Loughborough University

This item was submitted to Loughborough's Research Repository by the author.
Items in Figshare are protected by copyright, with all rights reserved, unless otherwise indicated.

Performance modelling and optimization for video-analytic algorithms in a cloud-like environment using machine learning

PLEASE CITE THE PUBLISHED VERSION

PUBLISHER

© Manal Nasser Khalfan Al-Rawahi

PUBLISHER STATEMENT

This work is made available according to the conditions of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0) licence. Full details of this licence are available at: https://creativecommons.org/licenses/by-nc-nd/4.0/

LICENCE

CC BY-NC-ND 4.0

REPOSITORY RECORD

Al-Rawahi, Manal N.K.. 2019. “Performance Modelling and Optimization for Video-analytic Algorithms in a Cloud-like Environment Using Machine Learning”. figshare. https://hdl.handle.net/2134/23359.

Performance Modelling and Optimization for Video Analytic Algorithms in a Cloud-like Environment using Machine Learning

by

Manal Nasser Khalfan AL-Rawahi

A Doctoral Thesis

Submitted in partial fulfilment
of the requirements for the award of

Doctor of Philosophy
of
Loughborough University

30th November 2016

Copyright 2016 Manal Nasser Khalfan AL-Rawahi

Abstract

CCTV cameras produce a large amount of video surveillance data per day, and analysing them require the use of significant computing resources that often need to be scalable. The emergence of the Hadoop distributed processing framework has had a significant impact on various data intensive applications as the distributed computed based processing enables an increase of the processing capability of applications it serves. Hadoop is an open source implementation of the MapReduce programming model. It automates the operation of creating tasks for each function, distribute data, parallelize executions and handles machine failures that reliefs users from the complexity of having to manage the underlying processing and only focus on building their application.

It is noted that in a practical deployment the challenge of Hadoop based architecture is that it requires several scalable machines for effective processing, which in turn adds hardware investment cost to the infrastructure. Although using a cloud infrastructure offers scalable and elastic utilization of resources where users can scale up or scale down the number of Virtual Machines (VM) upon requirements, a user such as a CCTV system operator intending to use a public cloud would aspire to know what cloud resources (i.e. number of VMs) need to be deployed so that the processing can be done in the fastest (or within a known time constraint) and the most cost effective manner. Often such resources will also have to satisfy practical, procedural and legal requirements. The capability to model a distributed processing architecture where the resource requirements can be effectively and optimally predicted will thus be a useful tool, if available. In literature there is no clear and comprehensive modelling framework that provides proactive resource allocation mechanisms to satisfy a user's target requirements, especially for a processing intensive application such as video analytic.

In this thesis, with the hope of closing the above research gap, novel research is first initiated by understanding the current legal practices and requirements of implementing video surveillance system within a distributed processing and data storage environment, since the legal validity of data gathered or processed within such a system is vital for a distributed system's applicability in such domains. Subsequently the thesis presents a comprehensive framework for the performance

modelling and optimization of resource allocation in deploying a scalable distributed video analytic application in a Hadoop based framework, running on virtualized cluster of machines.

The proposed modelling framework investigates the use of several machine learning algorithms such as, decision trees (M5P, RepTree), Linear Regression, Multi Layer Perceptron(MLP) and the Ensemble Classifier Bagging model, to model and predict the execution time of video analytic jobs, based on infrastructure level as well as job level parameters. Further in order to propose a novel framework for the allocate resources under constraints to obtain optimal performance in terms of job execution time, we propose a Genetic Algorithms (GAs) based optimization technique.

Experimental results are provided to demonstrate the proposed framework's capability to successfully predict the job execution time of a given video analytic task based on infrastructure and input data related parameters and its ability determine the minimum job execution time, given constraints of these parameters. Given the above, the thesis contributes to the state-of-art in distributed video analytics, design, implementation, performance analysis and optimisation.

Acknowledgments

My first great gratitude is for God, for giving me the opportunity to pursue a Phd and surrounded me with all blessing.

This thesis is the result of the support of many people and I would like express my deepest gratitude and sincere appreciation to all those who helped me complete my thesis.

I would like to thank my beloved parents Nasser Al-Rawahi and Farida Al-Hadhrami for their love, endless support and prayers. My mother came to UK many times to support me taking care of my children while I was doing my research work.

I would also like express my deep thanks to my supervisor Prof. Eran Edirisinghe for this continues guidance, support and patience from the start to the final stage of my Phd journy.

I would like to thank my dearest hasband Abdullah for his support and accepting my many years of research with open heart.

My special thanks to my children (Sama, Nasser & Ahmed) who give me hope and power in life.

My special thanks to my brother Hamood and my sisters Aziza, Maisa & Areen for their emotional support and prayers.

I would like to thank Dr Iain Philips my second supervisor for providing technical support and facilities to build Hadoop in a virtualized cluster.

I would also like to thank all the past and current school of science admin staff at Loughborough University.

Thanks to all my colleagues in the department of computer science who cooperate with comments and encouragement.

Finally, I would like to acknowledge the support of the Ministry of Manpower of the Sultanate of Oman for providing the funding for this research and Loughborough University UK for providing research facilities support.

List of Publications

The following papers have been published/accepted for publication:

Journal Paper:

(i) Manal, A.R. and Edirisinghe, E.A., 2015. Video Forensics in Cloud Computing: The Challenges & Recommendations. Journal of Information Sciences and Computing Technologies, 3(2), pp.201-216.

Conference Papers:

(ii) AL-Rawahi, M., Edirisinghe, E.A., and Jeyarajanz, T. "Machine Learning-Based Framework for Resource Management and Modelling For Video Analytic in Cloud-Based Hadoop Environment." IEEE Cloud and Big Data Computing. 2016. [accepted for publication]

(iii) AL-Rawahi, M., Edirisinghe, E.A. "Optimization Techniques to Enhance Allocation of Computing Resources for Cloud-based Video Analytic." In SAI Intelligent Systems Conference (IntelliSys), 2016. IEEE. [accepted for publication]

Contents

Abstract ii

Acknowledgments iv

List of Publications v

List of Acronyms xiv

1 Introduction 1

1.1 Aim & Objectives . . . 3
1.2 Research Questions & Contributions . . . 4
1.3 Thesis Structure . . . 6

2 Background & Literature Review 7

2.1 Video Analytic Systems . . . 7
2.1.1 Architecture . . . 7
2.2 Cloud Computing Paradigm . . . 10
2.2.1 Definition of Cloud Computing . . . 11
2.2.2 Cloud Service Models . . . 12
2.2.3 Cloud Deployment Models . . . 13
2.2.3.1 Amazon AWS . . . 14
2.2.4 Enabling Technologies . . . 15
2.2.4.1 Distributed Computing . . . 15
2.2.4.2 Virtualization . . . 15
2.2.4.3 Hypervisor . . . 16
2.2.5 Cloud Computing Architecture . . . 18
2.3 Hadoop1 Framework . . . 21
2.3.1 Hadoop Ecosystem . . . 23
2.3.2 MapReduce . . . 25
2.3.2.1 MapReduce Workflow . . . 25
2.3.3 HDFS . . . 27
2.4 Hadoop2 Framework (YARN) . . . 28

2.5 Cloud-based Hadoop . . . 30
2.5.0.1 Amazon EMR . . . 30
2.5.0.2 Sahara OpenStack . . . 31
2.6 Literature Review . . . 32
2.6.1 Hadoop Platform for Video Processing . . . 33
2.6.2 Hadoop Performance in a Virtualized Cluster . . . 35
2.6.3 Performance Modelling and Optimization . . . 37
2.7 Summary . . . 40

3 Video in Cloud Computing: The Challenges & Recommendations 41
3.1 Introduction . . . 41
3.2 Related Work . . . 43
3.3 Security and Privacy Requirements of a Video Surveillance System . 44
3.3.1 Review of the current legal framework that governs video surveillance systems installed in the UK . . . 45
3.3.2 Review of the legal framework governing video to be used as evidence . . . 48
3.3.3 Research Publications . . . 49
3.3.4 The Legal Aspects: Summary & Conclusions . . . 50
3.4 Cloud Computing Security Concerns . . . 50
3.4.1 The Cloud: Technical Issues . . . 51
3.4.2 The Cloud: Non-Technical Issues . . . 53
3.4.2.1 Data Protection . . . 53
3.5 Cloud Computing Performance Concerns . . . 57
3.6 Conclusion & Recommendation . . . 58

4 Video Analytics Applications Deployment on Hadoop 62
4.1 Introduction . . . 63
4.2 Methodology . . . 64
4.2.1 Video Dataset Description . . . 64
4.2.2 Video Applications Description . . . 65
4.2.2.1 Face Detection Algorithm . . . 65
4.2.2.2 Motion Detection Algorithm . . . 65
4.2.3 Hadoop System Design Overview . . . 66
4.2.3.1 Input Video . . . 67
4.2.3.2 HDFS . . . 68
4.2.3.3 MapReduce-based Video Analytic Application . . . 68
4.3 Experimental Testbed Set Up . . . 72

4.3.1 Virtual Cluster Configuration . . . 72
4.4 Experiments Results & Analysis . . . 73
4.5 Discussion . . . 78
4.6 Conclusion . . . 79

5 Performance Modelling for Hadoop-Based Video Analytics 81
5.1 Introduction . . . 81
5.2 Methodology . . . 83
5.2.1 Phase One: Analyse the characteristical behavioural of video analytic application in Cloud-Hadoop environment . . . 84
5.2.1.1 Experiment 1 . . . 85
5.2.1.2 Experiment 2 . . . 87
5.2.1.3 Experiment 3 . . . 88
5.2.1.4 Discussion . . . 88
5.2.2 Phase Two: Create Training Dataset . . . 89
5.2.2.1 Dataset Variables . . . 89
5.2.2.2 Data Collection . . . 90
5.2.2.3 Dataset Representation . . . 90
5.2.2.4 Data Preparation (Feature Selection) . . . 90
5.2.3 Phase Three: Modelling the Job Execution Time . . . 91
5.3 Experimental Results & Analysis . . . 92
5.3.1 Prediction Experiment Result 1 . . . 92
5.3.1.1 Training Datasets . . . 92
5.3.1.2 Prediction Models . . . 92
5.3.2 Prediction Experiment Result 2 . . . 95
5.3.2.1 Training Datasets . . . 95
5.3.2.2 Prediction Models . . . 95
5.4 Implementation Challenge & Discussion . . . 99
5.5 Conclusion . . . 100

6 Performance Optimisation for Hadoop-Based Video Analytics under Constraint Conditions 101
6.1 Introduction . . . 101
6.2 Problem Formulation . . . 103
6.3 Methodology . . . 105
6.3.1 Algorithms Description . . . 105
6.3.2 Fit GAs to Resource Allocation Problem . . . 107
6.4 Experimental Results and Analysis . . . 108
6.4.1 Face Detection Application . . . 108

6.4.1.1 Test1 . . . 108
6.4.1.2 Test2 . . . 110
6.4.2 Motion Detection Application . . . 111
6.4.2.1 Test1 . . . 111
6.4.2.2 Test2 . . . 112
6.4.3 Comparison of Different Optimization Results . . . 114
6.5 Conclusion . . . 115

7 Conclusion and Future Work **116**
7.1 Summary . . . 116
7.2 Conclusions . . . 117
7.3 Future Work . . . 117

References **120**

List of Figures

2.1 High-level block diagram of a video forensic system 8
2.2 Scientific analysis of academic disciplines on cloud computing research[86] 10
2.3 Cloud service models architecture[178] 12
2.4 Full-Virtualization 17
2.5 Para-Virtualization 17
2.6 Hardware-based Virtualization 17
2.7 Cloud reference architecture[110] 19
2.8 NIST conceptual reference model for cloud computing[50] 20
2.9 Cloud reference architecture[78] 21
2.10 Typical components of IaaS cloud infrastructure[67] 22
2.11 Google trends on Hadoop compared with other technologies. 22
2.12 Hadoop history 23
2.13 MapReduce data flow framework when executing a job 26
2.14 Hadoop Framework[127] 29
2.15 High level view of Amazon EMR [2] 31
2.16 Sahara architecture [33] 32

3.1 CSA Top Threats ranking in 2010 & 2013[42][64] 52
3.2 Roles of data controller & data processor 55

4.1 Image output from face detection Algorithm 66
4.2 Image output from motion detection Algorithm 66
4.3 Hadoop framework for video analytic application 67
4.4 Hadoop performing a video analytic job 68
4.5 Video (key,value) pairs generated from decoded frames 69
4.6 Hadoop virtual topology 72
4.7 Execution time against number of VMs for face detection algorithm 74
4.8 Execution time against number of VMs for motion detection algorithm 74
4.9 Speedup analysis of Hadoop-based face detection 75
4.10 Speedup analysis of Hadoop-based motion detection 76
4.11 Processing time when different video input size is used. 76

4.12 Processing time variation with different number of input video files when the number of VMs are held constant. 77

5.1 Impact of Reducer slots on CPU resource utilization 86
5.2 Impact of Reducer slots on the face detection job execution time . . 86
5.3 Impact of Reducer slots on the motion detection job execution time 86
5.4 Comparing the predicted vs actual execution time for different classifers 94
5.5 Comparing the predicted vs actual execution time for different classifiers 98

6.1 Evolution curves searching for best fitness 109
6.2 Evolution curves searching for best fitness 112
6.3 Evolution curves searching for best fitness 113

List of Tables

1.1 Research questions and thesis contributions. 5

2.1 Comparing Architecture of Hadoop1 & Hadoop2-YARN [101] 30

3.1 Summaries of the key legal requirements, the corresponding video surveillance system compliance and cloud computing challenges. . . 59
3.2 Summaries of performance requirements, the corresponding video surveillance system compliance and cloud computing challenges. . . 59

4.1 Video files details 65
4.2 Pseudo code for the implementation of a single-frame and overlapped-frame oriented applications based on Hadoop MapReduce. 71
4.3 Software configuration for the cluster of VMs. 73

5.1 Total execution time with different Reducer slots and tasks for face detection application. 87
5.2 Comparison of the Total Execution Time (TET) with two different VM resource types. 88
5.3 Attribute used for video analytic application performance modelling. 89
5.4 Training dataset for face detection application with videp type1. . . 90
5.5 Training dataset for motion detection application with videp type1. 90
5.6 Attributes for video analytic applications performance model. . . . 91
5.7 Results of the prediction models for face detection application with video type1. 93
5.8 Results of the prediction models for motion detection application with video type1. 93
5.9 Updated training dataset. 95
5.10 Results of the prediction models for face detection application with video type1. 96
5.11 Results of the prediction models for motion detection with video type1. 96

6.1 User input requirements and the system constraints 108

6.2 Results generated by the GAs operation 109
6.3 Analysis of results generated by the GAs operation 110
6.4 User input requirements and the system constraints 110
6.5 Results generated by the GAs operation 110
6.6 Analysis of results generated by the GAs operation 111
6.7 User input requirements and the system constraints 111
6.8 Results generated by the GAs operation for motion detection 111
6.9 Analysis of results generated by the GAs operation 112
6.10 User input requirements and the system constraints 112
6.11 Results generated by the GAs operation 113
6.12 Analysis of results generated by the GAs operation 114
6.13 Comparative optimization results for face detection application with two video types. 114
6.14 Comparative optimization results for motion detection application with two video types. 115

List of Acronyms

GA	Genetic Algorithm
HDFS	Hadoop Distributed File System
ML	Lagrange Multipliers
PS	Patern Search
VM	Virtual Machine
YARN	Yet Another Resource Negotiator

Chapter 1

Introduction

Large-scale distributed systems are required for video surveillance systems (VSS) in order to analyse large quantities of recorded video data which is a computing intensive activity. It is important to consider scalability as a factor for future video surveillance systems [60]. Existing solutions require demand in resources, which are unsuitable for future increased demands for video data. IBMs system, IBM Smart Surveillance System, (IBMSSS) [80], deploys a combination of database partitioning and web application server clustering that allows scalability. However, such solutions that attempt to resolve the scalability issue are expensive and increase the cost of hardware and overall investment expenses.

Video surveillance data processing is currently accomplished by techniques such as parallel computing and distributed computing to reduce costs. Such techniques provide performance enhancement and reduction in cost; yet suffer from limitations in resources, complex programming, scalable storage and limited support of fault tolerance. When considering these challenges within the current infrastructure, a data processing framework that is simple and automatically handles task scheduling, distribution and storage of data, load balancing, and machine failure is necessary in order to allow users to focus solely upon creating scalable applications.

One example of such a framework that has been widely adopted by major organisations, such as Amazon, Google, Yahoo and Facebook, as well as researchers and the community, is Hadoop. Hadoop [162] is an open source MapReduce implementation for data storage and intensive processing, designed to resolve a number of large data issues, such as searching, log analysis, indexing, multimedia analytics and machine learning. Hadoop has attracted researchers and other publicity indicating a move towards enhancing and developing Hadoop architecture, which has improved the overall stability of Hadoop[129]. These developments have inspired this research into using Hadoop as a platform to resolve large-scale video surveillance processing issues.

Forensic video analysis is a post-event processing, and only processed occasionally when needed. It is therefore both impractical and uneconomic to dedicate machines for the sole use of Hadoop within such an application. On the other hand video analytics are real-time alert systems that often require substantial amount of processing to be done, given the content of video that can change from time-to-time. Hence dedicating machines for the processing of video analytic data will be inefficient. Scalable computing and storage resources can be deployed on demand through cloud computing with minimum initial investment, providing full virtualisation and distributed computing technologies and can hence support applications such as video forensics and video analytics (collectively known as video surveillance).

Cloud computing services provide a range of resources and services to support video surveillance systems, which have been recently studied in literature[122][137][89]. Overall, this makes a good solution in the provision of scalable dynamic clustering of video analytic systems. In order to avoid potential risks of security and privacy breaches of the video data, there is a requirement to balance the benefits of cloud computing with an understanding of a variety of legal issues involved in deploying video surveillance. The process of risk-aware computing will assist in the creation of a more security comprehensive architecture as a protection against potential threats.

A literature review was undertaken in order to understand the current regulations and guidelines behind establishing a reliable, legal and trustworthy, cloud-based video surveillance system. The requirements of a legally acceptable video forensic system are discussed and current security and privacy challenges of cloud based computing systems are studied in order to recommend the design of a secure and reliable cloud-based video forensic system. The research focuses only on the performance of distributed video analytic applications using a cloud-based Hadoop platform after carefully considering the proposed recommendations and observations.

Preliminary experimental results conducted within the context of the research presented in this thesis indicates that a cloud-based Hadoop platform will be successful in speeding up video analytic processing and distribute computing of computer vision algorithms in a cluster of machines. It was also observed that the execution time of applications is determined mainly by both the size of the load, as well as cluster sizing. Hadoop MapReduce architecture is dependent upon the type of application, as well as upon hardware performance/configuration[162]. However there was no existing research in attempting to model Hadoop performance within a distributed computing framework serving data and processing intensive applications.

Video analytics/forensics is computationally intensive task, operating on a frame-by-frame basis to extract information from its content. Cloud workloads are characterised by their own resource and performance requirements, as well as constraints that are specified in service level agreements (SLA). Therefore, in order to meet performance goals, decisions relating to the correct resource to be deployed for a video analytic application workload requires careful analysis of its likely behaviour when applied to a cloud-based Hadoop environment. The ability to model and thus predict application performance and to subsequently optimise resource allocation will therefore be a useful contribution to the state-of-art in video surveillance research.

Identifying the above research gap, the behaviour and performance of a video analytic application running in a virtualised cluster is first studied in this thesis, which highlights the most significant factors that influence the execution time of an application. Based on these factors an experimental study was conducted in order to develop a prediction model for the application. This was undertaken by comparing different machine learning algorithms based on the prediction accuracies that reveal that decision-based models outperform linear regression models, whilst the Ensemble Bagging models outperform standard single-based classifiers. This research fills an existing gap in research relating to video analytic related comprehensive performance predictions. Current research maintains a focus upon different types of applications that are limited to using standard learning algorithms, such as Simple Linear Regression, SVM and Multilayer Perceptron (MLP).

To demonstrate the practical use of the prediction models obtained above, the thesis continues to study the use of Genetic Algorithms (GAs) as an optimisation method to search for the optimal resource allocations under given constraints to complete a job within a minimal period of time. The impact of both the infrastructure level parameters (e.g. number of VMs, number of slots etc.) and application level parameters (e.g. video input size, frame size etc.) to the above optimal resource allocations are studied in detail.

1.1 Aim & Objectives

The aim of the research presented within this thesis is to investigate the performance of a scalable video analytic application implemented on a cloud-based Hadoop environment. The results are used to propose a framework for modelling the performance and optimising resource allocations. In order to meet this aim, a number of objectives need to be met:

- Carry out a study and analysis of legal, ethical and security issues surround-

ing the deployment of video surveillance, within a cloud based environment and thus use the findings to recommendations for the design of such a system.

- Study the possible use of a selection of different distributed parallel processing techniques in the development of a cloud-based video analytics application and in particular, design, implement and anlyse a Hadoop based architecture, Determine the application and architecture specific parameters that has the most significant impact on performance.

- Based on machine learning algorithms develop models that are capable of accurately predicting the system's performance, i.e. the prediction of job execution time.

- Propose a method to optimise the performance of the proposed scalable video analytic system under given constraints of the system architecture and application related parameters.

1.2 Research Questions & Contributions

Table 1.1 provides a summary of the research questions, research method adopted to answer the research questions and resulting original contributions made by this thesis.

Table 1.1: Research questions and thesis contributions.

Research Question(RQ)	Research Method	Chapter/Published paper#	Contributions
RQ1: What are the general requirements for ensuring the legal compliance of a video surveillance system that is used to record surveillance data for legal prosecution and to what extent does a cloud infrastructure comply with this law?	Literature Review	3/(i)	1. Conducted a comprehensive analysis on video surveillance system legal requirements. 2. Provided literature on the reported cloud computing legal challenges in terms of security and privacy status. 3. Provided recommendations on the possible lawful architectural design and good practice. 4. Recommend enabling technologies; a private cloud platform to provide both storage and a facility computing platform and a Hadoop framework for distributed parallel processing platform.
RQ2: What is the performance gain obtainable when executing large-scale, highly scalable video analytic applications in a Hadoop based cloud like environment which supports distributed data processing and storage? What are the factors affecting performance in such the environment?	Experimental testbed on a small scale virtual cluster computing environment	4	1. Designed and constructed a Hadoop virtual cluster with nine virtual machines. 2. Illustrated the application performance gain obtainable by providing the speed up graph using Amdahl's Law. 3. Interpreted the results obtained from testing different factors that impacted the application perfomance. 4. Articulated the idea of modelling the video analytic application performance using infastracture level metrics as well as application-level metrics.
RQ3: Can a model be developed to predict the video service target performance in Hadoop based distributed processing environment based on system and video input related parameters? If a model can be generated, how accurate will such a model be?	Modelling using Machine Learning algorithms	5/(ii)	1. Built a prediction model for Hadoop video analytic application to predict the job execution time using different machine learning algorithms that fit our targeted model. 2. Compared the prediction accuracy between the machine learning algorithms. 3. Evaluated the constructed prediction model on new/unknown dataset to confirm the effeciency of the proposed method.
RQ4: Can the model be used to find the optimal performance of the video analytic application when deployed in the Hadoop based environment and executed under constraint conditions?	Optimization technique using genetic search algorithm	6/(iii)	1. Formulated optimization problem using the model derived from chapter5. 2. Applied genetic algorithms which produced a fast and an effective results in finding the minima job execution time under constraints condition.

1.3 Thesis Structure

The remainder of this thesis is organized as follows:

- Chapter 2: Provides a background knowledge and an overview of the related research topics, such as video surveillance systems, cloud computing and Hadoop/YARN.
- Chapter 3: Analyses the legal, ethical, security and performance requirements for carrying out video surveillance within a cloud based architecture.
- Chapter 4: Provides the design & the implementation of the virtualized Hadoop system and the performance analyis of the execution of the MapReduce-based video analytic application within this system.
- Chapter 5: Presents the development of machine learning based approaches to model and predict the total execution time of a video analytic application, deployed within a Hadoop virtual cluster.
- Chapter 6: Presents the use of a genetic algorithm based optimization approach to minimise the job execution time of a video analytic applications deployed on the Hadoop based architecture, subject to given constraints in resources.
- Chapter 7: Provides a conclusion and future work.

Chapter 2

Background & Literature Review

This chapter intends to present a discussion of the background knowledge required to undertake this study based on video analytic systems, cloud computing, and Hadoop MapReduce framework. These technologies are effectively utilised within the contributory chapters of this thesis, i.e. Chapters 3-6. In addition we describe related work in our research area.

2.1 Video Analytic Systems

Technological advances to improve security in society generally have included the installation of closed circuit television (CCTV), and although this technology enables operators to view or search recorded video data to investigate specific events, these processes of searching and monitoring are shown to be expensive in terms of labour, as well as being time consuming processes which can result in human errors. Typically the operator manually performs a visual search in recorded video footage for a given event to search for specific information, such as date, time periods, locations, colour of clothing, or gender of individuals from the database[69]. The advanced video surveillance systems use computer vision, machine learning and pattern recognition algorithms that can automatically track, classify and detect specific objects, and large quantities of visual data can be analysed with minimal interventions from operators, which is faster and less expensive in terms of labour. VSS performs either real-time alerts (video analytics) or post event on recorded and indexed video data stored in a database (video forensics).

2.1.1 Architecture

The review of literature of this topic suggests that the structure or architecture of these video surveillance systems is varied [80][114] depending on the observation/s to be carried out. However, in general, a typical video surveillance system consists

of a distributed set of video cameras covering an area/space that requires monitoring 24/7 for security purposes. We assume that these cameras are connected to a Video Database that stores the CCTV video footage for subsequent computer based processing.

In figure 2.1 a high-level block diagram of a typical video forensic system is presented. The input video footage is stored in a video database. In the case of manual inspection for forensic purposes a CCTV operator will play back the stored video, file-by-file in an attempt to locate the content/objects being searched for.

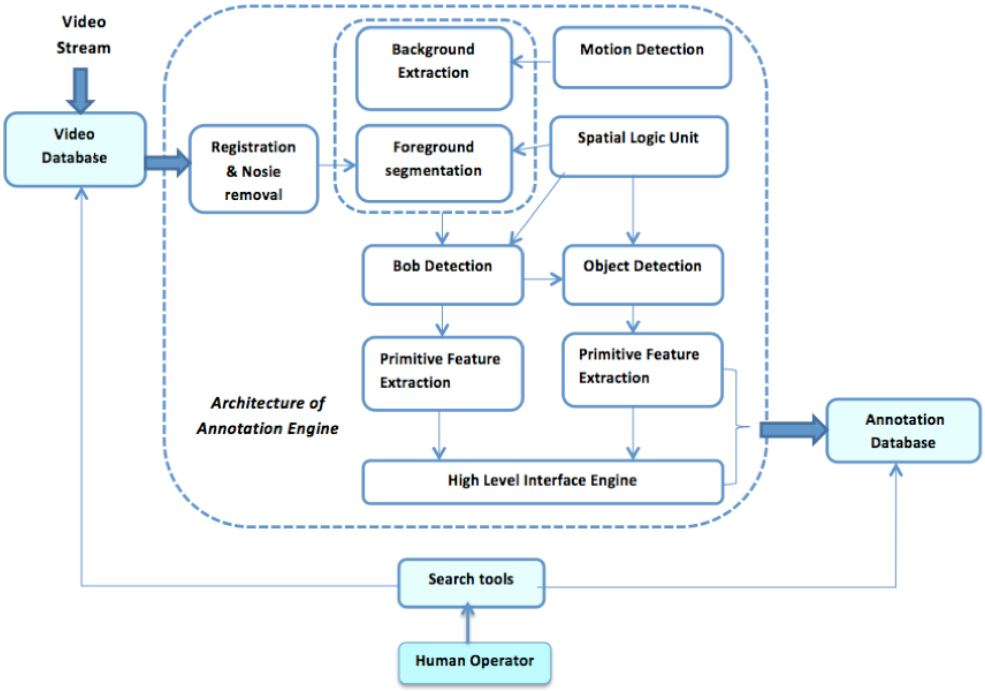


Figure 2.1: High-level block diagram of a video forensic system

In the case of computer based processing the videos are initially annotated at the time of storage with high level annotation information such as, the camera number/location, time of day etc. In addition to this high level annotations, depending on the content of the videos, lower level annotations are generated by an Annotation Engine (see figure 2.1). The video annotation engine is a collection of image processing, computer vision, pattern recognition, machine learning and optimization algorithms that work collectively to identify the presence of known objects (e.g. humans and vehicles) and are able to articulate their detail/appearance (e.g. shirt colour, vehicle type, number plate details, carrying a bag etc.). A typical architecture of the annotation engine is detailed in figure 2.1. It is noted that the annotation engine is the key computing/intelligent component of the forensic tool and is the location where most extensive computational tasks are carried out.

The annotated data, both the high-level (captured from camera input data directly at storage) and low-level (generated by the Annotation Engine) are stored in an annotation database. The availability of all annotation data real-time, would make the system efficient and hence highly desirable for video forensic analysis, post event applications. The challenge is the capture of low-level annotation data, real-time, given the complexity of the video processing algorithm. A typical computer with a single processor running at even the highest available typical clock speeds, would not enable real-time capture of low-level annotations (this is the basic research problem analysed in this thesis). Further the accuracy and trust of the data stored in the annotation database are key to conducting a forensic investigation that has any legal validity. The annotation information (i.e. metadata) will be used in the search process for the detection and recognition of people (e.g. man wearing a red jumper, carrying a bag), vehicles (e.g. a red van, speeding) and activities (e.g. man walking away from a blue car).

In making practical use of the system presented in figure 2.1 above in video forensics, a forensic search is initiated by a human operator (user) through an interface and using a search tool. The search tool searches through the Annotation Database created by the Annotation Engine detailed above. Once the objects/events with given descriptions are located, going through the Metadata, this will be used to fetch the data from the stored, original video footage.

A typical video surveillance system will comprise of many video cameras that are distributed over a public space being monitored. Often the cameras may have overlapping views. Even the same object that is visible via different views of cameras and at different times will look different due to changes of object size, angle of approach, clarity, partial occasions, varying camera specifications etc. In order to be able to process complicated scenes and still be able to do an accurate investigation, the computer vision algorithms have to be sophisticated. This results mostly in the need to use and execute complicated algorithms that will use a significant compute power. Having to process multiple videos captured by multiple cameras make the compute power requirements ever more. A solution to this problem exists in the use of distributing computing facilities where the tasks can be divided between multiple processors, often executing algorithms in parallel.

A rather modest number of research investigations have been carried out in the past to tackle the above problem. Depending on the type of solution adopted these have been reported in Chapters 3-6. One notable example is the system design by IBM that uses a collection of web application servers, clustering and database partitioning thus allowing scalability. The system designed is dedicated to serving a distributed video analytic task and the designed has been conducted with the

maximum computing resource needs in mind for a specified application. This solution attempts to solve scalability issue, but increases hardware investments cost due to the solution not being elastic, i.e. resources not being flexibility and effectively utilised dependent on real time usage. In order to address the resource allocation elasticity related issue, the current trend in video surveillance system design is to use cloud based architectures. However our literature review has revealed that this is an area that has not been investigated and reported in detail in literature[112]. Hence this has been one motivation factor behind the research presented in this thesis.

Following section provides background knowledge about the cloud computing paradigm.

2.2 Cloud Computing Paradigm

Cloud Computing is a new model that delivers Information Technology (IT) as a service to users. The services thus delivered can typically be classified into software, platform and infrastructure as a service, delivered on on-demand, with pay-per-use price model. The cloud offer elasticity and scalability in provisioning of resources, which can significantly reduce the cost of dedicated hardware/resource provisioning. This capability of cloud computing is the result of making effective use of existing technologies such as data centre automation, automatic computing, system management, utility computing, grids and clusters for distributed computing, Web 2.0, SOA, web services, and virtualisation[54].

The International Data Corporation (IDC) reports that worldwide spending on public IT cloud services reached 4 billion dollars in 2013, and is expected to be more than 107 billion doller in 2017[27]. Various studies report that cloud computing is a continuing trend for data storage and processing, and specifically supported in the field of computer science[35], as shown in figure 2.2, and with a peak of research studies between 2008 and 2009, based on cloud computing scientific analysis of Gartners Hype Cycle by Heilling and Vob [86][26][86].

Subject area	2008 (%)	2009 (%)	2010 (%)	2011 (%)	2012 (%)	2013 (%)	Avg. (%)
Computer Science	54.55	59.57	58.87	60.74	59.46	49.29	57.1
Engineering	20.66	11.65	11.24	13.82	14.90	23.48	16.0
Mathematics	4.13	11.55	14.85	13.82	10.43	11.03	11.0
Social Sciences	2.48	5.07	4.36	3.35	3.45	4.02	3.8
Business, Management and Accounting	7.44	3.04	2.08	2.52	2.22	1.61	3.2
Decision Sciences	1.65	2.03	2.11	1.33	1.65	1.33	1.7
Economics, Econometrics and Finance	4.13	-	0.85	0.89	0.34	0.22	1.1
Materials Science	0.83	0.91	1.19	1.14	0.78	1.11	1.0

Figure 2.2: Scientific analysis of academic disciplines on cloud computing research[86]

The following sections introduce cloud computing, its deployment and service models, and show how the development of cloud computing systems have contributed to commercial development, as well as open source development, which encourage more developer's to apply greater solutions and options.

2.2.1 Definition of Cloud Computing

Various academics and institutions have attempted to provide clear definitions of cloud computing, as this is a new and developing technology, and a recent attempt at defining cloud computing was suggested in 2011 that was based on an earlier attempt in 2009 by the National Institute of Standards and Technology (NIST)[118]. This suggests that the model of cloud computing involves service provider interaction and minimal management effort for releasing the required data in understandable formats accessed from services, applications, storage, servers and networks of computer resources that are configurable, which could be accessed from a network when required and when convenient to users.

The nature of cloud computing is complex and at present this technological advance lacks standardisation[37], so perceptions of the characteristics of cloud computing are varied and often challenged; for example, the management systems applied by Google do not use virtualisation as a factor, and cloud computing facilities can be accessed without accessing the Internet, as private clouds are available based at specific locations, so the Internet is not a characteristic of cloud computing [105][119][170]. However, the definition by NIST in terms of its key elements are widely noted in the literature on this topic and by many in the cloud computing scientific community, so that there are common factors that define cloud computing, which are described below.

- On-demand self-service: The cloud service provide has no dirct interaction with users when data is added and released from cloud computing resources, as this process is automatic.

- Resource pooling: Cloud users can select their requirements dynamically from pool of resources such as networks, computing and storage facilities. These resources can be shared by various tenants or consumer. With the exception of meeting legal requirements, users do not know the location of the resources that they are accessing[119].

- Broadband network access: Smartphones and laptop computers are part of a range of devices that can access various cloud services over the network via standardized interfaces.

- Rapid Elasticity: Resources of cloud computing assist users to avoid excessing computer power remaining unused when there is less demand, and reduced time and costs for procurement when adopting cloud computing capabilities that are automated. In addition to this elasticity, cloud computing offers faster speeds for users, so that when demand increases, cloud computing facilities increase rapidly, and when demand decreases, these facilities are dropped rapidly, so actual demand is matched quickly to cloud computing facilities that are available.

- Measured Service: Cloud computing provides a pay-per-use measurement model that enables users to pay when they use these services. For example in Amazon AWS customers are charged by the hour, and this model also gives users information about the efficiency of the resources they are using.

2.2.2 Cloud Service Models

Software (SaaS), Platform (PaaS) and Infrastructure (IaaS) services are defined by NIST as the cloud service models mostly used [118]. Access to cloud computing services is determined by model differences in terms of control and service types, and Figure 2.3 shows the layers of IaaS, PaaS and SaaS in models of cloud services.

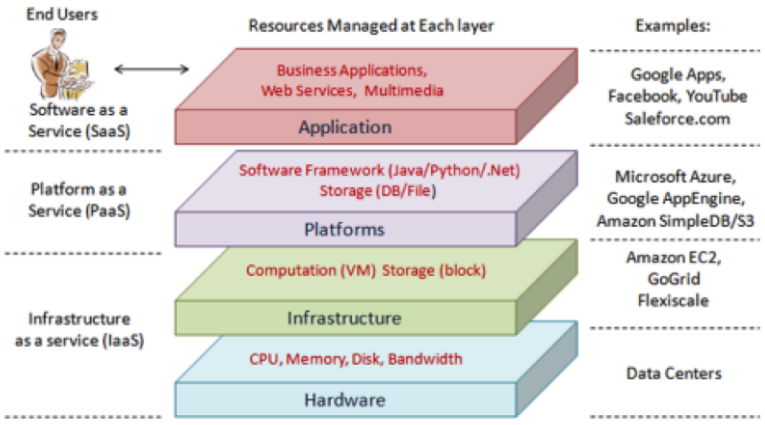


Figure 2.3: Cloud service models architecture[178]

- **Infrastructure-as-a-Service (IaaS):** This service is provided by Rackspace, GoGrid and Amazon AWS, to mention few, and although users cannot control this virtual resource hardware, they can release resource, update resources and create resources, and so change the environment with direct controls. This technology component applies KVM, VMware and Xen to enable virtualisation technology to be used with virtual resources, load balancing, networks, computing and data storage within this service[119][37].

- **Platform-as-a-Service (PaaS):** This service is provided by Salesforce Apex language, Microsoft Azure and Google App Engine, and interaction between the cloud environment and developers environment is enabled with API, so that developers use libraries, functions and programming tools to design their own application with this service.

- **Software-as-a-Service (SaaS):** This service is provided by Dropbox and Facebook, to mention few, and users are not required to deal with maintenance, updates and installations, and do not know about the software platform or infrastructure, so the cloud online software within this service includes CRM, email and storage.

  Some academics argue that a model for cloud computing should also include data, infrastructure, platform, hardware and software components, as an 'everything service' of (XaaS)[110] [134][134].

2.2.3 Cloud Deployment Models

Ownership of the provisioning location and infrastructure determines how cloud computing services are used, which disregards the model of delivery, and shown in figure 3.4.

- **Public clouds:** Upgrading and maintaining these services remains with the provider of these cloud computing services that maintains ownership of its resources of IT and the infrastructure, and although third parties would own public cloud computing services, the cloud environment has public accessibility.

- **Private clouds:** Organisations access their IT resources according to departments, locations or parts from a central database by the use of services and technologies provided by cloud computing facilities, which is owned by individual organisations[152]. Although some organisations use private cloud computing services located within their buildings, these are managed by third party companies, but most maintain their own cloud computing through various software options, such as Eucalyptus, OpenNepula, Cloudstack and Openstack. Private cloud computing services also mean that organisations can manage, process and service their data independently and avoid restrictions of legal requirements, exposure to security issues and limited bandwidth of networks of public cloud computing services [79]. Virtual private clouds are available from Amazon, so that organisations have a virtual environment that they control completely within a defined virtual

network, which is part of the Amazon AWS cloud services, as a section that is isolated from other parts[5].

- **Community clouds:** Management of community cloud computing services could be the responsibility of members or a third party, but members would have similar interests, such as policies, mission and requirements for security. Therefore, community cloud computing services are similar to public cloud computing services, but with limited access for community members[79].

- **Hybrid clouds:** The benefits of hybrid cloud computing services include portability of applications and data, as various models could be used, such as community, public and private. Advantages include scalability when using public cloud services, and control of security risks when using private cloud computing services, and consumers could use public cloud computing services for data that are not sensitive, and private cloud computing services for data that are highly sensitive, as these various models are combined by technology that is proprietary or standardised.

2.2.3.1 Amazon AWS

Cloud computer capacity that can be resized is available from Amazon Elastic Compute Cloud (Amazon EC2), and this service interface enables users to launch instances (i.e. virtual machines) with a variety of operating systems to fit different use case[1]. Each of these instances has different features and resources such as of CPU, memory, storage and network resources. Selecting any of them depends on customer's workload characteristics.

Amazon EC2 offers three purchasing models for renting instances The costs of these models are determined by the benefits offered by Amazon EC2, so that potential consumers need to ensure their organisation or company purchases one of the three models available that best meets their requirements [4] [153].

- On-Demand instances: Consumers make no advance payments or have any contract commitments in the long term, as cloud computing capacity is charged by the hour when used, or paid for on demand.

- Reserved instances: Consumers pay a reduced hourly rate by agreeing to a contract over a three year or one year period with an initial payment at the beginning of the contract, and no further payments until the contract ends.

- Spot instances: Consumers agree with the provider an hourly price rate that is determined as the maximum they would pay for these instances, so do not pay more than this rate, but prices vary according to the principle of supply

and demand. This could benefit consumers, but if prices fall too low, then providers can stop offering these services[153].

2.2.4 Enabling Technologies

Cloud computing is not a new concept[74], however, the technology already exists to control various capabilities of computing, such as distributed grid computing, distributed cluster computing, Web 2.0, SOA, Internet services and other Internet technologies, virtualisation and utility computing. The following gives an overview of some of these technologies that directly relates to our research:

2.2.4.1 Distributed Computing

- **Cluster**: is networks of computers that share computational workloads for computing and perform similar computing tasks by working together as commodity computers or parallel computers that are defined as a cluster, so that if one computer stops working, cluster distribution maintains availability and balance of service to consumers. Individual computers could have specifications that are different or similar to others, [140].

- **Grid Computing**: is a platform in which distributed resource are organised into logical pools and shared across multiple administrative domains connected by a network[71], which can consist of multiple clusters. The grid idea was initially developed to support scientific researchers who believed that computers should be developed to handle their complex data intensive experiments[73]. Open science Grid and EGEE are two examples of this. Cloud computing has same vision as Grid, yet the cloud is not limited to certain community users and provide services on-demand[133].

2.2.4.2 Virtualization

Virtualisation is considered to be a core technology within cloud computing that enables on-demand resources with elastic provision of resources[121]. Virtualisation is a process of abstracting physical IT resource such as server (CPU power), storage and network into software-based virtual resources to be used by multiple users. Each virtual resource is sharing underlying physical resources and is unaware of the virtualisation process as if it was running on a separate physical resource.

This technique optimises the use of resources and enables centralised management of pooled resources[147]. The term virtualisation emerged in the late 1960s

in different forms[130] and become a core enabling technology for cloud. Common examples of virtualized resources described below[152]:

- Server: This is a physical server transformed into virtual server called virtual machine (VM); examples include VMware, Xen and KVM[161].
- Storage: This is a physical storage devices used as a virtual storage machine or virtual disk, examples include NAS and SAN.
- Network: This is a physical network peripherals, such as firewall, router and switches are formed into a logical network fabric; examples include VPN and VLAN.

2.2.4.3 Hypervisor

Hypervisors or virtual machine monitors (VMM) are software solutions for server virtualisation, added between the hardware and operating systems responsible to launch multiple virtual machines from a single physical machine, sharing resources such as CPU, memory, storage and I/O devices [24]. This layer of virtualisation can be performed in three different techniques; full virtualisation, para-virtualisation and hardware-based virtualisation, as shown in figure 2.4, 2.5 & 2.6.

- full virtualization:Provides virtual abstraction that is completely de-coupled from the underlying hardware. The guest is not aware it is being virtualised and does not require modification. It provides isolation of virtual machines and simplifies migration and portability. Examples include VMware, KVM, Virtual box and Microsoft Virtual server.
- Para virtualization: Provides virtual abstraction, which is similar to underlying hardware. It requires change to kernel of guest operating systems, which makes it poor in compatibility and portability with unmodified operating systems; examples include Xen and Hyper-V.
- Hardware-based virtualization: Virtualises guest operating systems with a kernel that is the same as the host operating system. It creates isolation process contexts inside one OS kernel, which are only available for the Linux system.

Selecting a hypervisor is a critical task as it affects the system performance [121]. Analysis and study between hypervisors: KVM, Xen, VMware, Virtualbox has been intensively studied in the literature. One author[113] conducted a survey

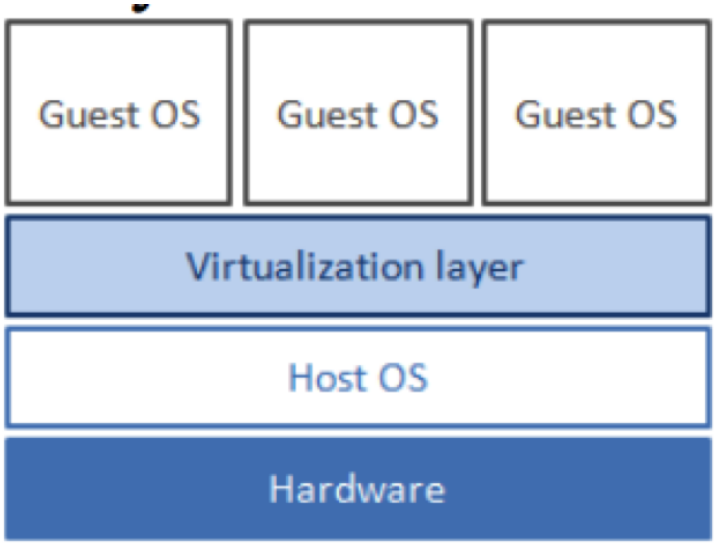


Figure 2.4: Full-Virtualization

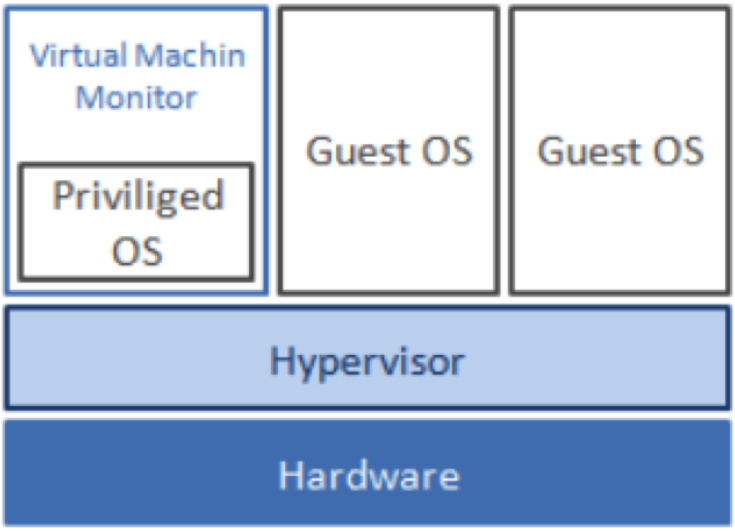


Figure 2.5: Para-Virtualization

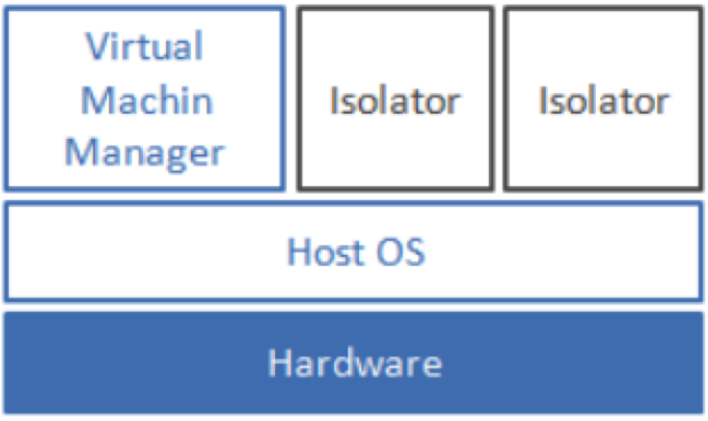


Figure 2.6: Hardware-based Virtualization

on these virtualisation technologies, which is a high level comparison related to type of techniques mentioned above. The author[173] states that KVM is the best choice for HPC cloud environment. The author[28] built KVM environment into a cloud system comparing its performance with a physical machine; KVM gives good result when computational is high. The performance for processing real-time data has been studied by[70] and the results show that KVM performs better with CPU intensive tasks. This author[109] analysed Xen and KVM performance and the results show best responsive time was achieved by KVM. The author[132] investigated the scalability of KVM with three parameters (overhead, linearity and isolation performance for three resources CPU, network and harddisk, the results show good scalability with CPU and network.

According to the findings above, KVM is used as a virtual environment in cloud systems for this research.

2.2.5 Cloud Computing Architecture

Cloud computing is a complex model, which involves different technologies and services that form that overall operation of the cloud. Alexander at al. [110] classifies cloud into different layers to ease explaining its process and interaction between components. This approach has been used and extended by many researchers to explain cloud architectures. Each focuses on describing certain components of cloud computing from the perspective of service deployment and delivery models [50][119][61]. In general, the architectures are guidance for vendors on how systems may be integrated to address issues of interoperability[133][110][177]and for researchers to ease analysis of cloud issues; for example, not limited to security, performance and management [174].

In 2009, Alexander Lenk et al. [110], proposed architectural categorisation of cloud technologies as a stack of service types, for instance, Iaas, PaaS and SaaS, each with their distinct features and current providers, see figure 2.7.

In September 2011, the government organisation NIST [50] published a Cloud Computing Reference Architecture model, which provides a high level architecture view of cloud computing. This architectural guidance is a starting point to understand the common standard terms and terminologies related to the major actors, their relationship, activities and functions in cloud computing, but not for design solution and implementation, see figure 2.8.

In 2011, Grobauer et al. [78] proposed cloud reference architecture based upon research funding from the University of California, Los Angeles, and IBM [174]. This architecture involves security-relevant cloud components that help analyse security issues relevant to each of the cloud services. As shown in figure 2.9 the

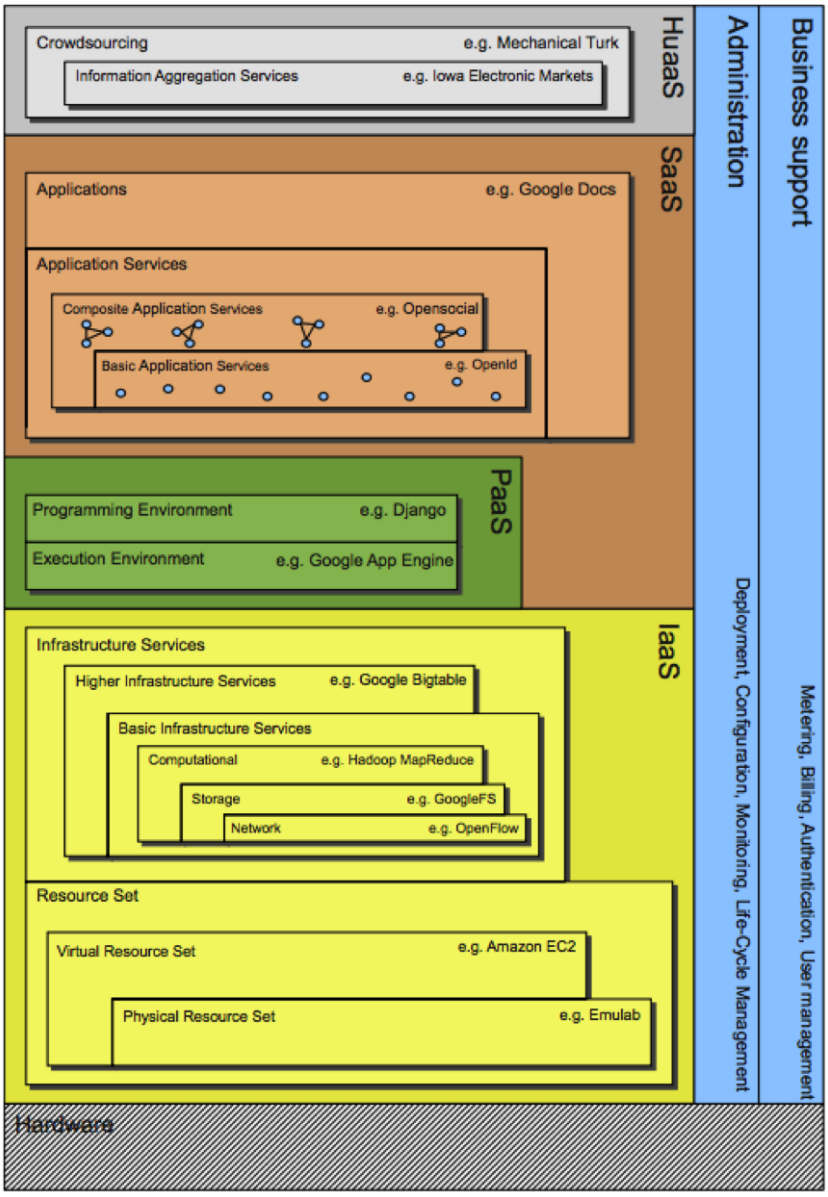


Figure 2.7: Cloud reference architecture[110]

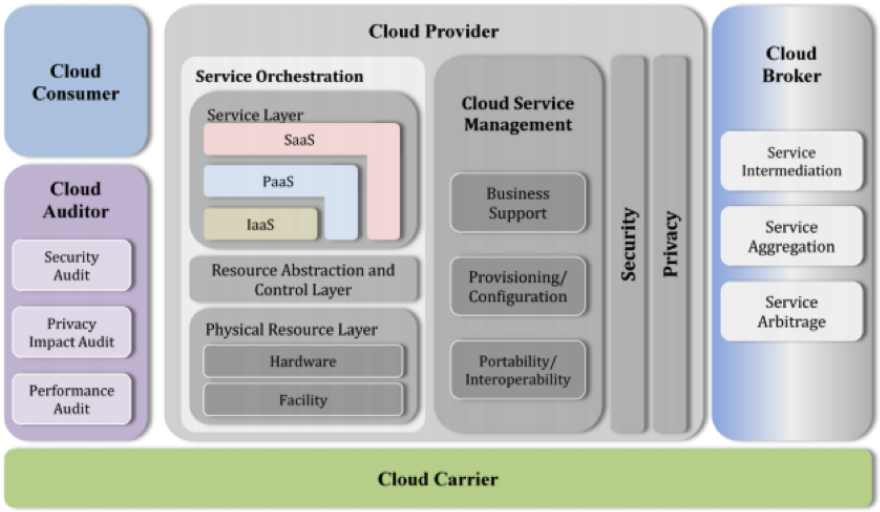


Figure 2.8: NIST conceptual reference model for cloud computing[50]

architecture shows the interaction between different layers divided into service customer, cloud specific infrastructure, supporting IT infrastructure and network carrier that connects cloud customers to cloud providers using standardised network protocols, such as SSH or HTTP. Cloud-specific infrastructure consists of three service models: IaaS, PaaS and SaaS, which are discussed in section 2.2.2. These services interact with customers through application programming interfaces (API) such as XML or REST, management access console and Identity, Authentication, Authorisation and Auditing mechanisms (IAAA) for user checks.

Cloud computing is composed of seven layers as described below [110][50][78]:

- Front-End: user, third party cloud, broker or auditor.
- Public network
- Application Layer
- Platform Layer
- Infrastructure layer
- Hypervisor layer
- Physical/hardware layer

In this research, the focus is upon IaaS and PaaS service model layers where video surveillance will be processed and stored. IaaS uses a virtualised infrastructure environment that consists of three main service components: computation, storage and communication. Frank Hans [67] presents a high level overview of typical technical infrastructure components of IaaS cloud, see figure 2.10. It is clear

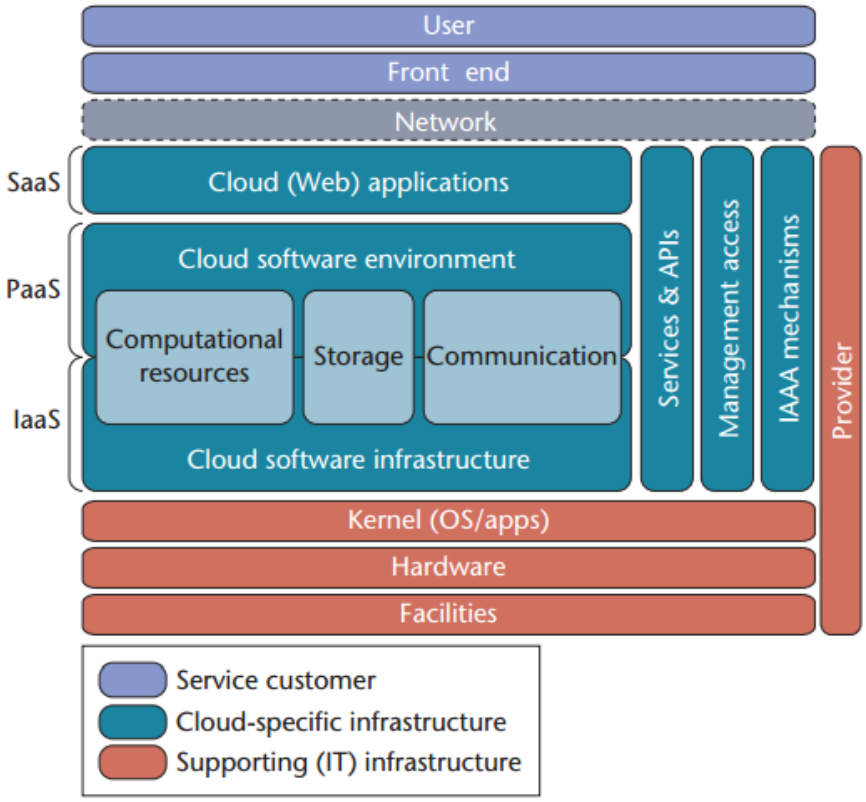


Figure 2.9: Cloud reference architecture[78]

that interaction between cloud customers and the virtualised services in cloud computing is through a cloud management system. Data can be hosted either in a shared multi-tenant or private single-tenant environment using virtual machine (VM) executed through Hypervisor software, such as VMware, Xen or KVM. The internal communication between VM's and storage, highlighted in blue, delivered through virtualised network components similar to the common IT data centre peripherals. PaaS is the application platform whereby video analytic application is installed on the top of IaaS. More information on PaaS is discussed in section (2.2.2).

2.3 Hadoop1 Framework

One open source distributed computing framework, capable of processing large-scale of data across cluster of computers whilst demonstrating a high degree of scalability and fault tolerance[162] is Hadoop, which is an implementation of the MapReduce model. Hadoop can be scaled up from a single machine to multiple machines, which together form a Hadoop cluster, with each machine performing

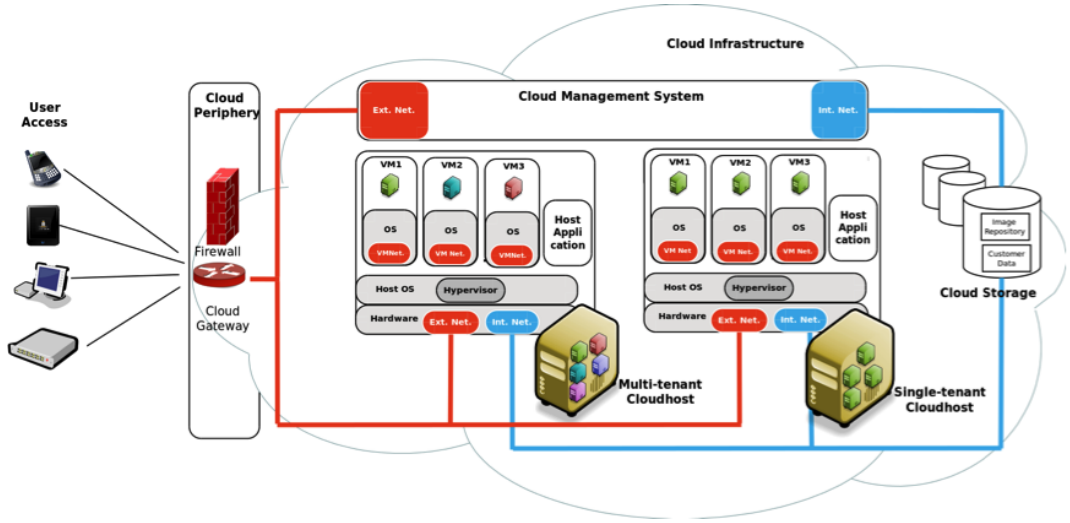


Figure 2.10: Typical components of IaaS cloud infrastructure[67]

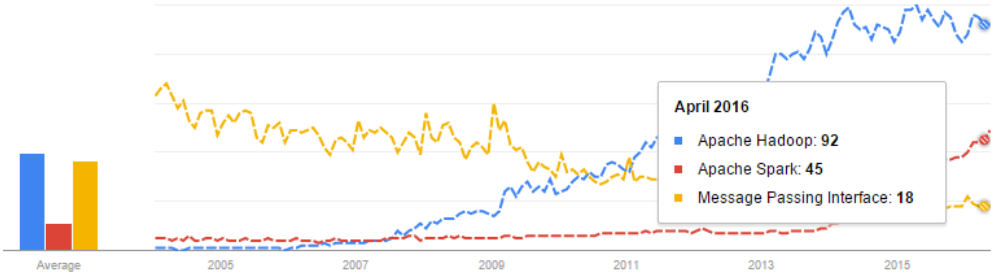


Figure 2.11: Google trends on Hadoop compared with other technologies.

local storage and computational data.

Developers and programmers focus upon the designing of parallel and distributed applications without concerns related to underlying details of the Hadoop framework, because it is an automated synchronisation and handling I/O process[141], which is capable of handling level failure applications. Hadoops distributed framework makes it sufficiently powerful and distinct when compared to existing frameworks, such Spark, MPI and other technologies, as shown in figure 2.11 using google trends tool [22].

Two Google published projects called Google e Systems (GFS) in 2004 and MapReduce programing model, which was invented in 2004 were the inspiration behind the implementation of Hadoop, which is currently licensed by Apache, initiated and led by Yahoo[23] in 2008, see figure 2.12.

The success of Hadoop is proven by one of Yahoo's Hadoop clusters, which processed one terabyte of data in 209 seconds, beating previous records of 297 seconds[8].

The platform that Hadoop platform has provided has resolved many large data problems in structured/non-structured data in disciplines, such as science

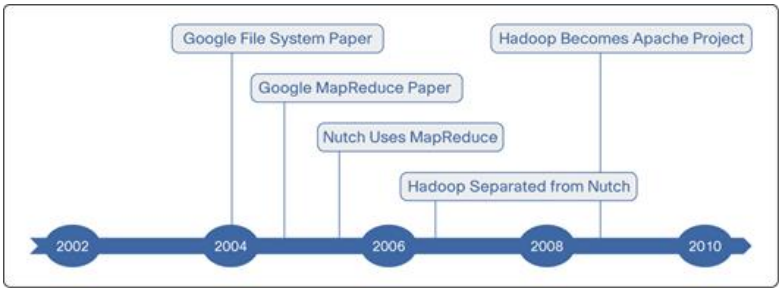


Figure 2.12: Hadoop developments.

and enterprise. Hadoop is currently deployed in large organisations, such as Yahoo, Facebook and eBay. Experimental issues with Hadoop may be categorised into computational problems that deal with massive data sets, requiring intensive computation per data element, computing-intensive applications, I/O intensive applications, CPU-intensive applications and processing video analytics. A number of enterprises adapt Hadoop in applications, such as web indexing, bioinformatics research, satellite data processing, computer visioning graphics, report generation, log analysis, data mining financial analysis, scientific simulation, medical imaging, weather forecasting and security analysis [162].

Hadoop is defined by different distribution channels ranging from open-source Apache Hadoop, pre-packaged commercial Hadoop providers such as Cloudera, IBM BigInsights, Horontworks and cloud-based platform such as Amazon Elastic, MapReduce (EMR) and Openstack Sahara enabling Hadoop as a service in the Cloud. There are two releases: Hadoop1 version 1.x series, which is a continuation of version 1.20.0 series used in production environments, and version 2.x, known as Hadoop2-YARN (Yet Another Resource Negotiator), which is a continuation of 0.23.x releases, discussed later in section 2.3.2.

Hadoop can work across commodity low-cost servers[162], since one piece of processed work can be distributed among many machines, which combines the total resource of each machine as one whole machine. Hadoop is able to distribute chunks of data into various nodes in advance where data locality is considered for computing to avoid storage and communication costs[162] [128], where according to[145] the moving of computation is cheaper than moving data, which makes Hadoop preferable to MPI.

2.3.1 Hadoop Ecosystem

There are two core components that form Hadoop: Hadoop Distributed File System (HDFS), which is a distributed storage saleable system, and a model that is responsible for distributed processing, called MapReduce. The Apache Software Foundation host other Hadoop based ecosystems, which have been de-

veloped and integrated within Hadoop in order to enhance the functionalities of the framework[162].

Some of the projects are briefly discussed in this thesis for their popularity, and information relating to flume, avero, parquet and crunch, we refer reader to [162]. The projects discussed below are: The systems are briefly introduced below:

- **Pig:** A scripting language and execution environment that is used to process large datasets.
- **Hive:** Is a distributed data warehouse built on top of Hadoop, which manages data stored in HDFS, providing a query language based on SQL called HiveQL. Facebook originally used Hive to manage large quantities of data produced daily on its social network and stored in HDFS.
- **HBase:** A column-oriented, distributive database built on top of HDFS for its underlying storage. HBase has the capacity to scale and work with large datasets, which supports batch-style computations using MapReduce, as well as point queries.
- **ZooKeeper:** A Hadoop distributed coordination service, which is a distributed, highly available coordination service. Distributed locks that can be used for building distributed applications through the use of ZooKeeper primitives.
- **Sqoop:** An open source tool designed to move data efficiently between relational databases, such as Sql Server, MySQL, Oracle, D2, Postange SQL and HDFS, as well as to the HBase system.
- **Spark:** A cluster computing framework designed for large-scale data processing [162], which is integrated with Hadoop. In can run in YARN, working on HDFS system and storage. Spark deploys its own distributed runtime for the execution of work in clusters to serve other types of applications, which need to use dataset across parallel operations. These include interactive applications and iterative jobs that have a limited use within the MapReduce framework[175] because of the integrated modules such as machine learning (ML-lib), stream processing (Spark Streaming), graphics processing (GraphX), and SQL (Spark SQL)[162]. Spark caches data in memory across iterations where in the datasets are loaded from disk in MapReduce, which allows Spark to run programs 100x faster[10].

2.3.2 MapReduce

MapReduce [162] is a programming model for distributed data processing, which can be used for writing in any language, it can be writing in any computer language, including Ruby, Java, Python and C. It works by distributing tasks across multiple machines through the use of a job scheduler, which is performed by the master machine. Each slave machine then processes the data stored on it. MapReduce consists of two components: Job Tracker, which resides in the master controls and monitors the distribution of task. Task Tracker, which resides in the slave and processes each assigned task, sending its status to Job Tracker. For the parallelisation process, MapReduce breaks the processing into 2 phases that can be executed in parallel on multiple machines[150]:

- Map function: This applies to every input record producing intermediate key-value pair, which are then stored on a local disk ready to be transferred to machines where a reducer is assigned to process the intermediate output.
- Reduce function: This merges the intermediate results from the Map phase and produces a final output result, which is stored in HDFS.

Each phase has key-value pairs as an input and output that a programmer specifies by map and reduce tasks. All jobs are executed on slave nodes as a map task or reduce task.

2.3.2.1 MapReduce Workflow

MapReduce applications that need to be run in Hadoop are called MapReduce jobs. In order to begin data processing, a client application submits a MapReduce job to Job Tracker(master) as a java code. Job Tracker communicates with Namenode (master) to find which Datanodes (slaves) contains blocks of input data.

The Job Tracker divides each MapReduce job into a set of tasks called map or reduce. Task Tracker running on those machines is then scheduled with the java code required to execute map function on local data. Several map and reduce tasks are running concurrently on each slave. The number of map slots and reduce slots are configured, which are dependent on the number of processors available in nodes to overlap computation and I/O. If all available slots are occupied, pending tasks must wait until some slots are freed up.

When the map task is complete, each machine stores the output result call intermediate data in its local temporary storage. It then sends the data over the network to a machine running reduce task for final computation. The communication between reducer and mapper happens through a TCP/IP protocol. There

are cases when data is not stored locally, such as when new nodes are added or when the node fails and the task is assigned to other node. In both cases, the new data node communicates with name node to be directed to nodes that have copies of the data; it then copies the data to local storage.

The following diagram explains the MapReduce programing model for the sequential phases (map and reduce) that the MapReduce data framework follows when executing a job, see figure 2.13 for illustration:

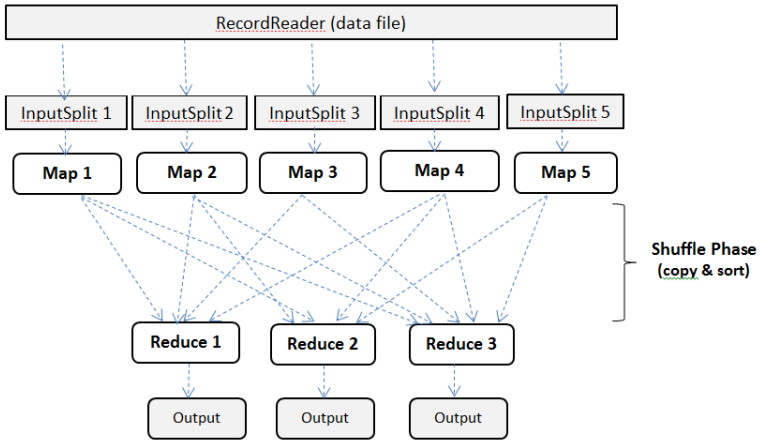


Figure 2.13: MapReduce data flow framework when executing a job

1. **Map Phase:**

   - **RecordReader:** This reads files from HDFS or any storage specified by the programmer; all data is then transformed into key-value pairs where the key is a unique id and the value is the corresponding data in bytes. It is then submitted to InputFormat in the form shown below:

     map(K1, V1) $\rightarrow$ list(K2,V2)

   - **One InputFormat:** Multiple types of key-value pairs provided by RecordReaders are accepted, all key-value pairs are combined and submitted to Mappers in Inputsplit form.

   - **Mapper:** Key-value pairs are generated through Inputsplit, with each node running one map task and run it in parallel. One map task takes a key-value pair, processes it and generates another key-value pair for reduced phase input. Mappers group key-value pairs according to requirements of algorithms and dispatch them to Reducers.

     Mappers group key-value pairs according to requirements of algorithms and dispatch them to Reducers.

2. **Reduce Phase:**

- **Shuffle phase:** When the nodes complete their map task they are ready for sort phase (copy), where nodes communicate with each other to pass key-value pairs to be sorted. This is the only phase where node communicates with each other.
- **Sort phase:** keys are sorted according to the key ID, presented as: Shuffle(list(K2,V2)) $\rightarrow$ (K2, list(V2))
- **Reducer phase:** This makes each reducer take all key-value pairs with the same key and merges them. It then performs computations on the values according to the instruction from java code. Reducer can take a subset of all the key-value pairs, but will always have all the values to one key. The result will be submitted to OutputFormat. Each reducer generates one output to storage (HDFS). This can be controlled through an implementation of Outputformat. Reducer phase takes the form shown below:

  Reduce(K2, list(V2)) $\rightarrow$ list(V3)
- **OutputFormat:** This deploys RecordWriter to write results back to HDFS ready for the client to read. The network is used when the blocks of the result have to be replicated by HDFS for redundancy.

2.3.3 HDFS

In a Hadoop distributed system, (HDFS) is one of Hadoops systems that provides scalable and shared storage network across cluster nodes. It is designed to work with the MapReduce framework written in Java that sits on top of a native local system. Files stored in HDFS are write only, but can be accessed and read many times. HDFS consists of two components:

- Namenode:
  - Resides on master machine and splitting data into blocks
  - Distribute blocks across cluster with replication for fault tolerance
  - Holds all metadata information about stored data blocks.
- Secondary Namenode:
  - Reside in master
  - Contains backup of all metadata stored in namenode.
- Datanode:

- Reside in each slave and stores blocks of data.
- Serves Read/write request from client with replication for fault tolerance.
- Propagate replication task as directed by NameNode.

Namenode is at the heart of HDFS where it stores all metadata information of the cluster and monitors the health of the datanodes. Datanodes stores the actual data of any stored file, which it sends back to the namenode every 3 seconds via the TCP protocol; every 10th heartbeat is block reported. The secondary namenode is a backup of namenode metadata, which connects to namenode after certain times to update its metadata information, which can be used to recover namenode is case of node failure.

Data is read and written to HDFS by going through different producers of communication between client and namenode through TCP. When a client sends a le to HDFS, namenode (master) split the le into blocks and replicates each block to a number of copies according to the Hadoop default setting, which is 3. It then distributes them to cluster nodes based on disk space availability to balance the load between the nodes.

In addition, Namenode uses rack awareness strategy (network topology) for replicas placement sorted in metadata les. This helps namenode to locate different copies of each block in away prevents the failure of one node from losing all copies of data. When the client is informed of the location of the blocks, it performs a pipeline[51] to sequentially copy data blocks between specified datanodes(slaves).

Figure 2.14, shows HDFS and MapReduce relationship within Hadoop framework.

2.4 Hadoop2 Framework (YARN)

Yet Another Resource Negotiator, YARN, is the next generation of Hadoop that is used for general computing platform that serves other large-scale programming models such as Spark, Dryad, Storm, and Graph processing [158], as well as MapReduce. These models provide different functionalities to data life cycle ranging from real time processing to interactive and batch processing that can be applied on the same data stored in single YARN cluster. Enterprises are therefore not required to retain separate clusters for different application types; they can work with data from the time that is generated from a single cluster [11]. Hadoop1 is designed for MapReduce implementation only. When a request from a client is submitted as a MR job, it will be the responsibility of job tracker to manage all

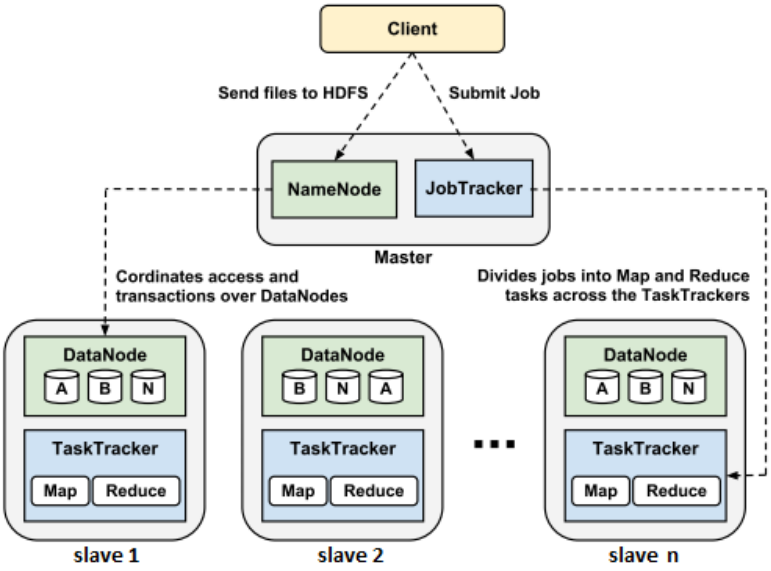


Figure 2.14: Hadoop Framework[127]

the execution of Mapreduce tasks in Hadoop cluster in terms of resource management and scheduling jobs across the cluster. However, in YARN architecture, this responsibility is separated into two functionalities:

- ResourceManager(previously JobTracker): Is a global distribution of available resources in a cluster among running applications (MapReduce, Spark etc.).
- NodeManager(previously TaskTracker): Provides per-node services within the cluster, which is responsible for launching application containers (resources), and monitors the resource usage of CPU, disk, memory, network bandwidth and reports back to the ResourceManager.

This separation of task makes managing multiple jobs running in YARN cluster easier. Figure 2.1 compares the architecture of Hadoop1 and Hadoop2, showing the role of ResourceManager in managing the jobs of different clients, each with separate NodeManagers. ApplicationMaster is a per-application component that works with NodeManager to manage the any job inside the cluster by negotiating resource containers with ResourceManager, tracking their status and progress. Containers are available in each node and they are allocated resources (CPU, memory, Network etc.) resulting from the negotiation between ResourceManager and Application manager.

The figure below shows that YARN continuing to use the HDFS layer, with its master NameNode for the storage of metadata services and DataNode for replicated storage services across a cluster[101]. However, in Hadoop1, it only supports one Namenode that manages the whole clustername space, which limits

system scalability. In YARN it supports multiple NameNodes in a single cluster for scalability and avoids single points of failure [11][9].

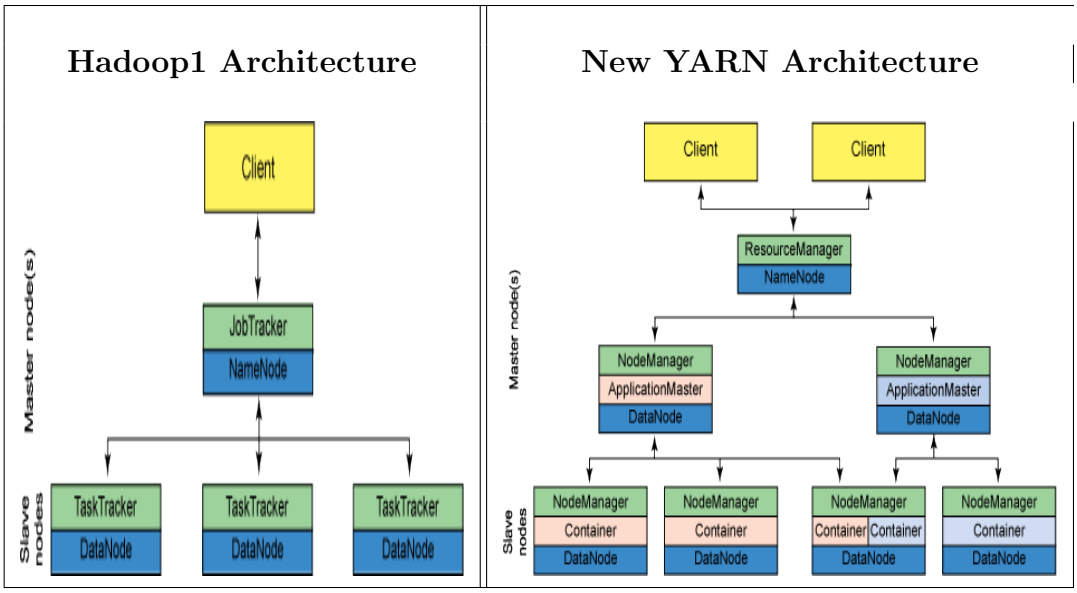


Table 2.1: Comparing Architecture of Hadoop1 & Hadoop2-YARN [101]

Benefits gained from this new Hadoop architecture include managing the life cycle of the application, improving the ability to scale Hadoop clusters to much larger configurations than previously possible, and allowing simultaneous execution of a variety of programming models[101].

2.5 Cloud-based Hadoop

2.5.0.1 Amazon EMR

Amazon Elastic MapReduce (EMR) [2] is a public cloud service for large-scale data analysis in a distributed environment. EMR uses the elastic infrastructure of Amazon taking advantage of EC2 computing and S3 storage to provide managed Hadoop framework releasing the customer from the expense of purchasing the underlying hardware and software, as well as its complexity. Figure 2.15 shows a high level view of EMR. Users only focus on analysing their data by loading their data to Amazon S2 storage, submitting their application and selecting the cluster size (number of machines). Amazon EMR provider takes care of Hadoop cluster deployment, management and security. EMR makes it easy and flexible for users to expand or shrink cluster size according to their analysis requirements. In addition, they provide virtual private cluster for users and organisation that want to be logically separated from other Amazon customers for enhanced security and

privacy. Other distributed processing are also powered by EMR, such as Apache, Spark, and different Hadoop jobs: Hive, Pig and Oozie.

Amazon EC2 (discussed in section 2.2.3.1), gives options for changing the type of an already provisioned instance, yet switching between instances types in Hadoop cluster can only be performed when instances are in their stopped state. Prior to running any job, a user should know what is the optimal number of instances (machines) and their types.

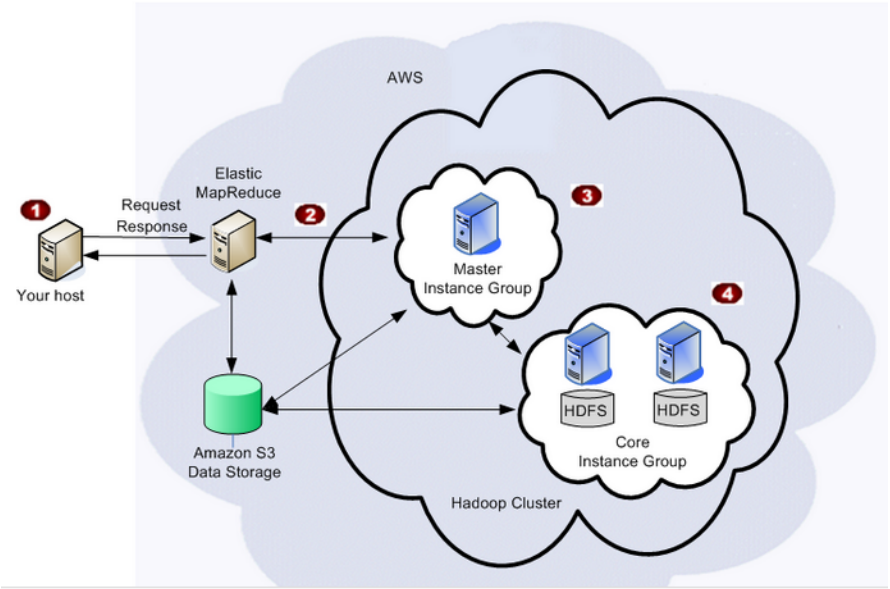


Figure 2.15: High level view of Amazon EMR [2]

2.5.0.2 Sahara OpenStack

2.3.2.3 Sahara OpenStack[33] is an open source data processing project that enables users to easily provision Hadoop cluster on top of Openstack infrastructure, which is a similar concept to Amazons EMR service. This project allows for collaboration between Horontworks, Redhart and Opetsack marinties. Sahara deploys clusters in few minutes and scales already provisioned clusters by adding/removing nodes on demand without the need to recreate the cluster. It supports different Hadoop distribution and vendor specific tools. Sahara use pre-designed templates for Hadoop configuration with the ability to modify parameter (e.g. heap size, map/reduce slot numbers). Figure 2.16 shows architecture of Sahara OpenStack [33], showing how Sahara interacts with Openstack components: Horizon, Keystone, Nova, Glance and Swift.

- Horizon: Is a graphical user interface (GUI) to be used by users to access all Saharas features.

- Keystone: Provides a security token used to work with the OpenStack, limiting authenticated user abilities in Sahara to OpenStack privileges.
- Nova: It is Hadoop cluster virtual machines provisioning unit.
- Glance: It is a pool of Hadoop VM images preconfigured with Hadoop and operating system.
- Swift: It is a data storage processed by Hadoop jobs.

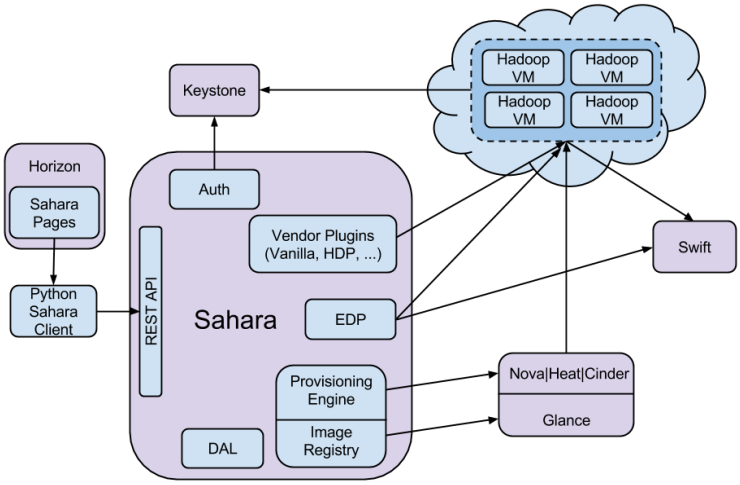


Figure 2.16: Sahara architecture [33]

According to Sahara, it is the users option to choose the cluster size, which means that a user should have knowledge about the running application prior to selecting a large cluster size that can cause under utilised virtual machines and a smaller cluster size can cause over-provision of resources in nodes, leading to performance degradation.

This drawback in also found in Amazon EMR service. The proposed research will investigate this issue and propose a novel automated technique to predict the number of VMs and corresponding resources required for video analytic applications.

2.6 Literature Review

A discussion of the existing literature is carried out in this section considering the solution for video processing running in cloud based Hadoop enviroenment. The categories of related work discussed based on the different proposed pipeline solution framework.

2.6.1 Hadoop Platform for Video Processing

Hadoop based platforms have been utilised in many application domains for distributed data processing. The focus of the following review of literature is limited to only video data processing, coding and transcoding.

Recently, with the popularisation of cloud based technologies, the use of a Hadoop based framework for processing video streams in a cloud like environment has become an active area of research, with the key focus being achieving time efficiency in processing large scale video data, due to the availability of distributed and abundant computer based processing and storage resources. The proposed work has been inspired by existing work on video/image processing and video coding/transcoding using a Hadoop based framework [126][148][141]. In the initial stages of research within this area, processing video using a Hadoop based framework was found to be challenging due to the fact that Hadoop was originally designed only to deal with text type of data. Therefore much of the early research focused on finding efficient ways to adapt video applications to a Hadoop framework and distribute the video stream in a manner that preserves its content[126][148][29].

The common approach used in literature for performing video analytics via a Hadoop cluster is by utilizing Hadoop related elements (i.e. projects) such as the Hadoop Distributed File System (HDFS), the MapReduce framework (see section 2.3) to carry out fast processing of large scale video data using open source tools such as OpenCV[30] and FFmpeg[18] that implements computer vision algorithms carrying out the required intelligent processing of data at a reduced cost[125]. In this research area one key focus has been in solving the two practical challenges faced by a Hadoop based architecture in processing video, namely, the modification of video processing libraries to work within a distributed computing cluster and transforming video analytic algorithms into map & reduce functions[149] compatible with the Hadoop-Mapreduce framework. Although all existing work follow similar approaches on solving the associated challenges, they differ in the methods used to read/write video files from the Hadoop distributed file system (HDFS). The following literature details some recent work in this area.

Ryu et al [59] proposed a framework for processing video analytic data based on a commodity physical Hadoop cluster and software tools such as OpenCV and FFmpeg. The author compared the performance of the system in single core and multi-core machines using a basic computer vision algorithm for face detection. Due to native library dependencies, FFmpeg was modified to be able to access data through the Hadoop distributed file system (hdfs), which the authors claim provide better performance than using the common mounting approach, fuse-hdfs

(see section 4.3.1.2). The video was split into Groups of Pictures (GOPs) [141], and was synchronized with the hdfs block size. The results showed a proportional increase of performance as the number of cores increase, a conclusion that was also confirmed by [85] for carrying out computer-intensive applications. However the paper did disclose the how the CPU was utilised nor discussed the factors that influence performance.

Heikkinen et al [85] also implemented video processing/analytics similar to that described in [59]. The key focus of the research was on effective data distribution that is based on video size and system performance parameters. The authors proposed the splitting of the input video file into 10MB size video clips as input, using an external tool, before inserting it into hdfs. However, the reason of selecting 10MB video input size is not well defined. The experimental results showed improvement in data distribution time compared to the traditional method of extracting frames first and then distributing them. For determining system performance, the authors measured the performance without considering the data transfer time, which is not accurate since data transfer causes overhead in i/o operations that could impact the overall execution time.

Another research by Hanlin [149], provided a technical implementation of video analysis on a physical Hadoop cluster. This work has enabled the research presented in this thesis to understand the design information of a video data workload in a cloud like environment. The author used open source tools such as fuse-dfs and traditional standalone software packages (FFmpeg, OpenCV and javacv). This is a approach similar to that used in [85] and [59]. However, the input video data written to hdfs is considered as one complete file. Which then internally goes through the common hdfs splitter. This fixed size video splits will then be decoded into a sequence of frames using FFmpeg tools during a MapReduce job, and each frame will be processed sequentially. The proposed system reduced processing time when compared to a typical local video analysis system. A similar approach was used in the research proposed in this thesis.

A recent study by Zhao et al [179] proposed a Hadoop video processing interface (HVPI) to help the user convert video analytic applications to a compatible Hadoop-Mapreduce framework.

In [16] Intel revealed a case study using a Hadoop based framework for implementing a distributed video monitoring syetem. However in this work the specific type of video application was not detailed and the technical implementation and system optimization was not discussed, making the contribution somewhat limited.

All of above works have focused on the implementation of video analytic algorithms using a Hadoop based framework. The experimental results demon-

strated the effectiveness of Hadoop in processing a video workload, considering video splitting techniques to optimize Hadoop performance. A similar architectural approach is used in the research presented in this thesis. However, the previous work are based on physical clusters and in the proposed research a virtualised execution environment is considered for the purposes of scalability and flexibility in assigning resources on demand. Virtualization provides a cost effective solution to the problem of building a Hadoop cluster with several physical machines. Given the above, the section 4.2.2 explores the existing research efforts that has experimented the visibility of using virtualization of Hadoop in processing large-scale data and have proposed solutions to improve performance.

2.6.2 Hadoop Performance in a Virtualized Cluster

Hadoop based implementation of Mapreduce applications using virtualization technology has been widely studied due its advantages, which include rapid provisioning, scalability, easy cluster management, cluster consolidation, optimal resource utilisation, live migration, network isolation, high availability and security [94][128][45][24].

Virtualization is one of cloud core technologies and recently attract attention in adding scalability and flexibility to big data issues (for details see section 2.2.4.2). Many existing industrial initiatives utilized the benefit of virtualization in a cloud-based Hadoop to run big data workload such as: IBM serengeti, VMware Hadoop, Openstack Sahara and Amazon Elastic Mapreduce(EMR). However, the benefit of virtualization may come with the price of reduction in performance due to overheads, resource competition and complex network communications. VMware reported in their virtualized Hadoop study that only 4% average performance improvement is possible as compared to utilising a physical Hadoop [45]. However this remains an insignificant observation when compared to the overall benefits of virtualization in Hadoop [45]. In literature several experiments have been conducted to test the performance of Hadoop on virtual machines(VM) using different approaches, [102] reported on experiments that presents a Hadoop performance analysis and diagnosis, [87] studied the cluster size variations, [63] presented the virtual machine configuration in detail and considered the use of different hypervisor types and Hadoop deployment strategies, [176] presented scheduling/load balancing algorithms, and [63] presented how to utilize cloud open source software as a platform. These previous research studies indicated a reduction on system performance due to overhead costs of a virtualization platform. Some research work proposed solutions to improve performance.

Hadoop and virtualization vendors, Hortonworks and VMware, have conducted

collaborative work to improve Hadoop support in a virtual cluster. For instance, adding extensions such as topology-aware plugins and providing elastic clusters by separating compute VM from local disk.

Ibrahim [94], evaluated the Hadoop framework in virtualized and non virtualized environments to address overheads caused by VM. He proposed a Cloudlet [93] as a new MapReduce framework by adding a local reducer for virtual machines in each physical environment to reduce data transfer during shuffle phase.

The author[84] designed a virtualized Hadoop cluster to study the scalability performance in two scenarios; first by adding extra node to a cluster (scale-out), and second by adding resources to the existing cluster node (scale-up). The experiment was conducted on a OpenNepula cloud platform [31], using Xen hypervisor. The result shows that different workloads require different types of scalability, for example the CPU-bound applications performed well with the scaling out method and vice versa for the I/O-bound application. In addition, they used a monitoring tool named Gangalia[21] to observe bottlenecks, while running Hadoop jobs and accordingly tuned Hadoop configuration parameters such as, map/reduce task slots, cpu, memory..etc, to reduce job execution time and improve resource utilization. The findings of this paper motivated the use of the scale-out mehtod for the investigations carried out within the context of the reseearch presented in this thesis. Morever, the proposed work also useds a similar tool to analyze video analytic resource consumption.

The author [127] proposed three types of topologies to test Hadoop performance: use of a fully virtualized cluster environment, use of separated data & computing nodes, and a topology that separates master and slave nodes. The work presented only investigated and experimented on a fully virtulized Hadoop using an openstack cloud. The results indicated a degradation of the performance when more VMs were added, due to increased overheads. Additionally, it was shown that the performance decreases due to the use of different HDFS block sizes and increase in the size of input data. These findings prompted the research conducted within the context of this thesis to consider adding a virtual machine to the network, only when the existing cluster machines are fully utilized, thus avoiding unnecessary overheads that degradate performance.

The paper [72] investigated the separation of data & compute operations conducted on both physical & virtual clusters, when conducting specific data operations. The implementation was not in a cloud environment. In the research proposed in this thesis, to avoid complexity, Hadoop's traditional and common architecture [162] containing all services from HDFS & MapReduce are operated on the same physical machine.

In [168] the authors have proposed a method to deploy Hadoop with Cloud-

stack [7] solving the cloned hostname issue that is caused when creating a virtual machine. The authors explained theoretically the drawbacks of a virtualization environment in running Hadoop and the fact that the efficiency of I/O scheduling are essential to reduce response time. There is no solution proposed to solve the performance issue.

In[92] the authors compared the Hadoop virtual cluster with a cloud-based Hadoop, using Openstack Sahara[33]. The authors used Hadoop benchmarks for performance analysis. The authors did not explain the variation in performance results of different MapReduce jobs running on both architecture scenarios. For example the case with the benchmark Hadoop application of calculation of mathematical Pi, which took less time to execute on a virtualized cluster.

In[63] the author analysed the impact of a Hadoop based deployment strategy on a cloud infrastructure in terms of performance, power consumption and resource utilization, by considering virtual machine placement of master and slaves within a virtual cluster for multi-tenant scenarios. The experiments were conducted on OpenNepula[31] cloud testbed with KVM hypervisor [28]. The results show that increasing the amount of virtual clusters within a cloud infrastructure has a direct impact on application performance and system behaviour. This conclusion is similar to the conclusion of [127].

In the absence of research investigations on how cloud-based Hadoop can effectively handle video analytic applications, it is important to identify the relevant issues when designing and implementing large scale video analytic applications in such environments. For this reason in Chapter-4 we aim to deploy a video analytic application in a Hadoop based virtual cluster with the objectives of investigating the system behaviour when the cluster size and the input data payload is varied. The results thus obtained will provide a solid foundation to the resource allocation modelling that is conducted in Chapter-5.

2.6.3 Performance Modelling and Optimization

Prediction-based performance modelling and optimisation of resources and provisioning based on characteristics of the workload are not new topics. They have been widely discussed in the literature using historical information to predict different performance metrics such as execution time for different field of applications in various computing environments such as datacentres [146], Cloud Computing[157][98][165] and Hadoop-MapReduce[102] report different approaches ranging from online/offline instrumental profiling, machine learning techniques to statistical modelling and control theory. Furthermore, within Hadoop-based applications, a number of optimisation methods were investigated with different ob-

jectives and constraints, such as cost, energy, quality and reducing job execution time. The management of resources for Hadoop includes issues with VM placement to physical machines [108][169], or placing applications to VM. Within this research, the focus is proactive prediction and optimisation of resources required by the application, and on the basis that cluster nodes deploy homogeneous VM types.

In this section we will review some relevant existing research on using machine learning for performance prediction in a given resource and using optimization approach for performanc optimization and resource provisioning.

Resource allocation and performance analysis for Cloud-based media applications have been studied in terms of considering their individual performance metrics. These exciting approaches focus on online resource adaptation (i.e. runtime estimation of resource usage) for media tasks being processed to meet QoS, job deadlines or performance goals. The authors [144] modelled resource allocation as a bin packing problem, considering only the CPU usage. They proposed dynamic resource allocation predictions based on Machine Learning algorithms SVM and KNN, to estimate tasks requirements and survival functions in order to estimate how much load a single VM can handle in parallel and shared environments so as not to violate QoS constraints. The authors [100] also aimed to predict the dynamic resource allocation, but specifically for real time video transcoding. This depends on the prediction of future user load demand using time series models. While the researchers have applied Machine Learning for resource prediction, no one has considered video analytic performance metrics in a Hadoop based distributed environment.

However, there are existing research conducted on a general Hadoop-based MapReduce using statistical models and Machine Learning techniques. The related work in this area was proposed by Kambatla et al.[102] who proposed the use of the online RSmaximizer tool that searches for optimum configuration parameters in terms of Hadoop applications by statistically matching its resource consumption with already known applications resource consumption signature and the optimal configuration stored in a database, using a brute force method. However, the optimisation techniques were built on fixed nodes and slots, which we proved in our study of Chapter-4 to have a great impact to overall performance. Verma et al. [160] developed online SLO-based resource provisioning, which can predict a reduced job completion time using the parameters from job profiles (Map, shuffle, reducer phases, completion times), input size and allocated resources. Herodotos et al. [87] proposed the use of an online Elastisizer tool which is an automated technique to optimise different configuration parameter settings and cluster resources (sizing) for a Hadoop job to meet performance needs. The authors used a

mix of a statistical blackbox model and a white-box model for estimation. They also used an instrumental application profiler which has high overheads. However, our application is processor-intensive, so this method will make it slower. Moreover, they used a fixed slot number when collecting job profiling data. This approach suffers from resource underutilisation or overutilisation. In addition, the method requires intensive job profiling. The author [107] proposed AROMA system that followed the same approach as [102] in classifying applications using clustering technqiues and SVM algorithm for hadoop job performance modelling. The author applied a pattern search technique that is based upon a SVM model to find the optimal resources to meet the required target at a reduced cost. The variation of slot and task numbers were ignored by AROMA when the model was built, which had a negative impact upon the accuracy of the model, and the optimisation goal. In addition did not describe how the problem was mathematically formulated.

The author in [104] improvd th work proposed by [160] and proposed a classical language multiplier to optimise resource provision, again based on the Hadoop performance model, which was generated by using locally weighted linear regression. The application specific characteristics were ignored and the aim was to focus on map and reduce task durations during modelling to estimate resources (i.e. map/reduce slot).

CRESP applies a search technique that is brute force in order to provision optimal resources within a Hadoop cluster[154] [58]. This is based upon a cost model deployed, which estimates job performance, as well as organising the jobs resources for the job through the use of a regression technique. The brute-force method generates a single solution for testing when there is a large search space with a number of representative and attributes that have a wide distribution of values, which takes longer to process when attempting to reach an optimal solution.

Our technique is different from all previous works in that we evaluate various Machine Learning algorithms using the WEKA tool [39] to find the best model with high prediction accuracy, modelled using feature vectors specifically related to video analytic application performance metrics such as resolution, file size, Hadoop configuration parameters and system performance (i.e. resource consumption) that affects video analytic services and uses them as input to the Machine Learning algorithms. In addition, for performance optimization and resource provisioning we have mathematically formulated the optimisation problem and presenting the genetic algorithm method for fast and effective results. We finally compared the result with other optimization technqiues Pattern Search and Language Multiplier technqiues that were proposed in literature.

2.7 Summary

This chapter introduces the background of intelligent video surveillance system, cloud computing paradigm and the Hadoop framework. In addition, the Hadoop2-YARN is discussed to show the improvements made in Hadoop2 over Hadoop1. In this thesis, research is limited to the Hadoop1 MapReduce processing engine in order to avoid version compatibility issues that are often found with the software and tools used to construct distributed video analytics. Moreover the related work were described to identify the gaps in the research field.

Having completed the above study of the research background and related work, in the following chapters the thesis aims to contribute with novel knowledge that will further extend the state-of-art in the area of video surveillance system deployment, analysis, modelling and optimisation.

Chapter 3

Video in Cloud Computing: The Challenges & Recommendations

In this chapter, we review the legal implications of deploying large scale video surveillance in a public cloud and determine the practicalities and challenges that need to be met to abide by the law. The research findings of this chapter provide recommendations for the design of a large-scale cloud-based video forensic system. The chapter brings together legal, policy related and technical requirements pertaining to the design, installation, commissiond and operation of large scale video surveillance in a public environment bridging an existing gap in academic and industry research.

3.1 Introduction

Present video surveillance systems that typically consist of a large number of distributed and networked CCTV cameras, collect significant quantities of digital evidence that can be used for crime forensics. The evolution of such systems have at present resulted in a significant proportion of the labour intensive video analytic and forensic tasks, usually carried out by trained CCTV operators, to be alternatively carried out by intelligent, automated, computer based analysis systems. Such systems use image processing, computer vision, pattern recognition and machine learning algorithms to detect and recognize objects of interest (e.g., people, vehicles etc.) and identify events of significance (e.g., person running, car speeding, people fighting etc.) enabling real-time alerts/warnings (i.e. video analytics) to be generated or objects/events to be indexed and stored in a database to allow on-line search to be carried out (e.g. search for a man wearing a red shirt who entered a specific named building between 1pm to 3pm during a given week) for video forensic investigations (i.e. post incident analysis). However conducting

efficient video forensics analysis on large datasets of video by distributed camera systems require high performance computing capabilities due to the complexities of computing algorithms to be utilized and the significant storage capacity required due to the sheer volume of data usually recorded. These two requirements increase the burden on the IT infrastructure to be used and introduce important challenges that need to be met to ensure practical viability of systems. In response to meeting the above challenges at present there are initiatives to move video analytics/forensics, typically carried out using dedicated storage and computing infrastructure to the cloud to best utilize its potential benefits in providing on-demand resource pooling (both compute power and storage). Although cloud computing and related infrastructure can support the above mentioned critical requirements of modern intelligent, automated video surveillance systems it also introduces other technical and non-technical challenges. Security and privacy risks are the most cited challenges in the area of cloud computing[75] due to the customers/users lack of physical control and the multi-tenancy nature of the cloud. Yet this is of fundamental importance in video evidence analytics and forensics, given the potential legal use of the evidence stored and/or created. Since video evidence gathering and use is regulated by law, it is crucial to review the legal implications of deploying video surveillance in the cloud and determine the practicalities and challenges that need to be met to abide by the law.

According to the research conducted within the remits of the research presented in this thesis there has not been any previous attempt in studying the legal requirements of a video forensic system and investigating the viability of developing a cloud based computing system for video forensics, given the known security and privacy threats of cloud computing.

While allocation and provisioning of virtual and physical sources in cloud are outside the control of cloud user, users need to specify the type and number of virtual machines that meets their application performance goal. This is a challenge since creating many virtual machines may lead to underutilized resources and may not also be cost-effective since in a public cloud the processing time is charged in an hourly basise[e.g Amazon EC2]. Furthermore, if less machines are created, it may affect performance expectation. This resource provisioning issue is an open research problem in cloud computing infrastructure management. This aim is to optimize the underlying resource utilization with a trade-off between resource cost and performance to meet a given customer's service level agreement(SLA) within a given budget.

This chapter attempts to bridge these research gaps and make relevant recommendations for the design of a large-scale, cloud-based video forensic systems.

3.2 Related Work

Some recent initiatives have focused on gathering video surveillance data from a system of distributed IP cameras and carring out basic video analytic tasks such as, motion detection, object identication, etc., in the cloud, overcoming storage capacity and processing power limitations of traditional video analytic applications. One example is the releases of the commercial cloud-based video surveillance systems, "Video Surveillance as a Service (VSaaS)", which is expected to grow in usage by 17% annually[38]. VSaaS is software-as-a service (SaaS) powered by Microsoft Windows Azure cloud platform. It provides High dimension (HD) video quality, real-time alerts performing motion detection, through heterogeneous connected devices. However, VSaaS is used for alert based video analytic tasks and do not support an extensive range of algorithms that can work together to support large-scale post incidence (i.e. video forensic tasks) video surveillance. Hence the basic dataset stored is nothing beyond the original video data captured and the usage of the service is so far not to support evidence in courts, but just as an alert system that can be used for monitoring security of a locality. A further drawback of VSaaS is that the infrastructure is beyond the user's control, which raises security and privacy concerns. In addition the compatibility issue of integrating cameras to VSaaS software adds extra hardware costs [122]. Some recent efforts from academic research addressed the challenges in the context of a cloud-based video surveillance system. The following sections introduce some of these research findings: Neal et al [122] investigated the capability of cloud services to support the requirements of hosting a high-resolution video surveillance management system and studied the cost in various cloud service models based on market pricing model. The author proposed cloud computing as a solution for VSM and highlighted issues to be considered such as the cost, legal requirements and compliance. These issues are considered and discussed in detail in this paper. Anwar Hussain has a number of contributions to video surveillance in the cloud. In 2012[90], he proposed a dynamic resource allocation scheme using a liner programming approach for composite video surveillance streams with cloud-based video surveillance system. A prototype of a system was implemented in Amazon AWS. In 2013[88], he analyzed the suitability of cloud solution by comparing video surveillance local infrastructure with his proposed cloud-based system in terms of performance, storage, scalability, reliability and collaborative sharing of media streams. The results demonstrated the capability of cloud computing to tackle the mentioned issues. In 2014 [89], a prototype design considering issues from his previous work was implemented and tested on Amazon EC2 platform. The author raised concerns in relation to the security and privacy factors and thus

suggested a hybrid-cloud solution as an alternative. Yong-Hua et al[167] proposed a prototype design for cloud-based video surveillance implemented in a private campus network. The design was focused on exploring the interaction between system components: the surveillance system, the browsing system and the storage system. Rodriguez and Gonzalez[137] proposed a cloud-based video surveillance system and focused on scalability and reliability issues in comparison to a traditional surveillance system. The proposed system was operated by optimizing the transmission of video streams between the client and cloud server, depending on network conditions, to avoid data loss in case of cloud failure or excessive network traffic. In this work video data was received and processed in the cloud, attending to security and privacy consideration. This was done by using security mechanisms such as, data encryption and secure transmission. The authors of [89],[167] and[137]utilized a cloud computing model to perform some basic image processing and computer vision algorithms. This work was limited with the design of fundamental video analytic tasks and no technical details were discussed.

As discussed above although some work has been presented in literature on cloud based video surveillance, this work has been limited to implementing simple video analytics tasks within a standard cloud based architecture. The key focus of such attempts have been to optimally use the available infrastructure and ensure security of video evidence gathered. However, the surveillance systems used were not of a scale that requires the storage of metadata about the stored videos thus requiring the safeguarding of such annotated data. Further the computing resource requirements were not sufficiently extensive to warrant considering the best use of a cloud based architecture. Further such work also did not discuss the legal requirements of a surveillance data gathering and investigatory system. Nevertheless such requirements warrant special features of both architectural and security requirements of a cloud based implementation. The key focus of the research presented in this paper is to bridge this research gap in making viable recommendations for a cloud based architecture for video forensics.

3.3 Security and Privacy Requirements of a Video Surveillance System

Intelligent CCTV surveillance systems used in public areas are installed by international, national and local governments to help prevent/detect crimes. Therefore they should be operated in such away to preserve confidentiality, integrity and personal privacy, by following appropriate laws and adopted codes of practice [41]. From country to country the legal requirements can differ in the details but the

essence of the requirements would be the same. In this section we focus our investigation on UK based legal and regulatory frameworks. It is noted that in the design, implementation and operation of a computer based, automated, CCTV video forensic system the legal and regulatory aspects would be taken into account. If not the practical use of such a system as a forensic evidence gathering and investigatory tool will be questionable. In this section we review and analyze the security and privacy requirements of video surveillance based on the following:

1. Legal frameworks: provides information on the Data Protection Act (DPA) that applies to video data processed in a cloud infrastructure[95] and also how it is accepted as evidence in court[111], and

2. Research publications: that address current problems, solutions, and future trends for research.

3.3.1 Review of the current legal framework that governs video surveillance systems installed in the UK

In the UK, the operation of CCTV is regulated by Data Protection Act of 1998 and Human Rights Act of 1998. In 2008, the UK Information Commissioners Office (ICO) issued guidance for the use of CCTV in the *"CCTV code of practice"* which was subsequently updated in 2014 titled, *"In the picture: A data protection code of practice for surveillance cameras and personal information"* to cover the inevitable widespread use of CCTV systems and thus the essential need to focus on data protection. The document provides practical guidance to those involved in operating surveillance camera systems and provide recommendations on how the legal requirements of Data Protection Act (DPA) can be met when monitoring individuals and disclosing images for the investigation of crimes. The guidelines highlight important criteria that should be considered in line with the requirements of designing a video surveillance architecture. The criteria can be summarised as follows:

- Ensuring effective administration - An individual/organization (i.e. the Data Controller) should be taking the ownership of the data gathered. The Data Controller is legally responsible for maintaining compliance with the DPA([96],page 10).

- Storing and viewing surveillance system information - Recorded material should be stored in a way that maintains the confidentiality and integrity of an image. In some cases when Cloud computing is used the controller has

to ensue that the cloud provider can ensure the security of the information following guidance from ICO ([96] , page 12).

- Disclosure - Video records must be secured and only accessed when there is a court order or information access right (freedom of information act 2012). This is to prevent the potential misuse of the system by operators who could spy on people, collect unauthorized copies, and manipulate data and marketing purpose which violate privacy and confidentiality of individuals. Disclosure of any image should be consistent with their purpose([96], page 14).

- Retention - The DPA does not prescribe specific minimum or maximum retention periods, which apply to all systems or footage. Rather retention should depend on an organizations own purposes for recording images ([96], page 19). Retention depends on the needs of a typical investigation that might be carried out by an organisation. After the retention period the data should be permanently deleted. However, recently UK government has introduced specific laws for dealing with data retention to protect public from criminals and terrorists [14].

A further guidance was published for the use of CCTV camera and Automatic Number Plate Recognition (ANPR) systems in the form of *"Surveillance Camera Code of Practice"* by Home Office and Lord Taylor of Holbeach CBE [46]. The guidelines include twelve principles that describe the best practices to be followed in using surveillance camera systems and processing images and footage in public places. This code of practice came into effect in England and Wales in 2013[15]. The guiding principles can be categorized into two groups as follows:

1. The development or use of surveillance camera systems, addressed in principles 1-4 (chapter 3-page 12) - These principles are related to the purpose of using the surveillance camera system, consideration of privacy and location of individual cameras, transparency/signage of cameras and clear responsibilities and accountability of surveillance systems.

2. The use or processing of images or other information obtained by virtue of such systems, addressed in principles 5-12 (chapter 4-page 16).

The eight principles under category (2) above are related to the way that the video feed is handled. These principles overlap with the requirements listed by the ICOs principles [96] listed above, including, video integrity and authorization access, retention and purpose of data disclosure. These principles are as follows:

- Principle 5: "Clear rules, policies and procedures must be in place before a surveillance camera system is used, and these must be communicated to all who need to comply with them."
- Principle 6: "No more images and information should be stored than that is strictly required for the stated purpose of a surveillance camera system. Such images and information should be deleted once their purposes have been discharged."
- Principle 7: "Access to retained images and information should be restricted and there must be clearly defined rules on who can gain access and for what purpose such access is granted; the disclosure of images and information should only take place when it is necessary for such a purpose or for law enforcement purposes."
- Principle 8: "Surveillance camera system operators should consider any approved operational, technical and competency standards relevant to a system and its purpose and work, in order to meet and maintain those standards"
- Principle 9: "Surveillance camera system images and information should be subjected to appropriate security measures to safeguard against unauthorized access and use."
- Principle 10: "There should be effective review and audit mechanisms to ensure that legal requirements, policies and standards are complied with in practice, and regular reports should be published."
- Principle 11: "When the use of a surveillance camera system is in pursuit of a legitimate aim, and there is a pressing need for its use, it should then be used in the most effective way to support public safety and law enforcement with the aim of processing images and information of evidential value."
- Principle 12: "Any information that is used to support a surveillance camera system, which compares against a reference database for matching purposes should be accurate and kept up to date."

A closer study of the above principles reveal that the annotation of stored video database (Principle 6) should be carried out only when there is a need for a forensic investigation (Principle 11). Therefore the data within the annotation database will only be created when it is necessary or for law enforcement purposes (Principle 7) and should be accurate and complete (Principle 12) at any given time. In other words, a need exist for carrying out on-demand, real time data processing of large datasets of captured video evidance.

3.3.2 Review of the legal framework governing video to be used as evidence

Video footage evidence, is defined as: "the presentation of visual facts about the crime or an individual that the prosecution presents to the court in support of their case"[62]. Once video evidence is collected from any type of storage media it must comply with legal requirements to ensure its admissibility in court procedures. In order for any digital evidence to be admissible in court, Nagel [120] listed a number of evidentiality rules required for any digital evidence to be relevant, authentic, original or an acceptable duplicate and hearsay.

Other evidentiality rules found in literature [151] such as those that relate to preservation, completeness and reliability is considered by Nagel as simply methods of authenticating digital evidence. The work presented in [139] explained how the court addresses legal issues when video is presented as evidence and emphasizes that video should be authenticated by testifying what is on the video is an exact representation of what should be on the video footage. If no witness is able to authenticate the surveillance video, then under the silent witness theory a judge can determine if the video can be authenticated if the following requirements are met [139]:

- There is evidence establishing the time and date of the video, which can be found in the metadata files of the captured videos.
- There was no tampering with the video.
- The video equipment used was sound.
- There is testimony identifying the participants depicted in the video.

This links to a reported court case in [120] which considered the use of hashing, metadata, and collection of data in its native format, as ways to authenticating evidence [111]. Even if evidence cleared the authentication process, additional evidential rules such as originality, preservation and hearsay will also apply [120]. An example of this is when a judge requests for a still-frame photo extracted from the video surveillance footage and compares it with the original video captured from the camera to ensure its originality and to avoid the possible misleading of the jury [123][43]. This confirms the importance of securing video surveillance data in-transit and at-rest, to preserve its integrity.

The process of investigating a crime via camera surveillance involves extracting the original video sequence and its associated meta-data files from recorded systems [32]. A given video files reliability to be used as evidence can be met by

technical authenticity methods such as using an audit trail, encryption or watermarking [17].

Modern video surveillance systems such as that presented in section 2.1.1, integrates various image processing, pattern recognition, machine vision and computer vision techniques for forensics video analysis. The operation of these algorithms affects the integrity of the resulting images but not their authenticity [62]. However, the use of processed images is not a problem in the law of England, Wales and Scotland as long as the user(investigator) is able to perform an audit trail to give evidence of the procedures used for generating, processing and storing digital images that proves the image is an accurate copy of the original[17].

3.3.3 Research Publications

In addition to the information presented above based on various laws and codes of practice, a number of research papers have been published in literature that relates to the use of video footage as evidence. Qasim and Christian[114] summaried the current state of the security and privacy requirements of modern distributed video surveillance with respect of integrity, confidentiality and access authorization mechanisms and underlined limitations of the existing approaches in large scale video surveillance systems. Real-time video encryption, key management, storage of video and its associated metadata, dynamic access controls are some research challenges identified by the authors. Another research effort by Winkler and Rinner [163] conducted a comprehensive survey of security and privacy protection related research work that have been published in the general area of visual sensor networks, also relates to video surveillance systems. In this paper[163], security requirements to ensure data integrity, authenticity and confidentiality are classified into four areas:

- Data-centric: include security of all data file cycles.
- Node-centric: include security of physical devices.
- Network-centric: include security of data transmission and communication.
- User-centric: related to awareness of how an individual's personal data is protected.

The solutions adopted to achieve these requirements range from trusted computing, encryption to access control. Authors highlighted the need for the protection of security and privacy within the application layer where more research were traditionally focused but also within the underlying infrastructure, a concern that this paper demonstrates to be genuine within cloud domain.

3.3.4 The Legal Aspects: Summary & Conclusions

The regulations and guidelines discussed above require appropriate technical and security safeguards to ensure the confidentiality, integrity, availability and authenticity of video, in order to be accepted as evidence in court and also to prevent breaches of an individual's privacy. The following is a summary of typical technical security practices adopted to ensure legal compliance with DPA:

- Encrypt data in transit and at rest, to maintain integrity and confidentiality.
- Implement a data backup plan to prevent data loss.
- Implement a mechanism to remove data from storage media after the retention period.
- Implement an audit mechanism to monitor that published polices and legal requirements are met.

There is one principle listed in DPA about international restrictions of data transfer. This principle is not mentioned in any of the legal frameworks discussed above. Data transfer is relevant to how cloud computing handles data for better performance and resource utilization; this will be discussed in (section). The implementation of the technical security practices mentioned above are based on common Information Technology (IT) practices presented in[114] and [163], However, there is no legislation that yet has specifically considered the use of cloud computing[55] and virtualization technology for CCTV video evidence gathering, processing and investigation. Therefore we consider security in cloud-based video surveillance as a research gap to be further explored.

The following sections present security concerns and the associated technical and non-technical issues relevant to using cloud computing as a environment for video surveillance.

3.4 Cloud Computing Security Concerns

Migrating a video surveillance system and its associated metadata outside the limits of an organization requires the cloud provider to provide a level of security protection similar to that could be provided if the system is operated within a local data centre [142], in a manner consistent with policies[99]. In fact, hosting data, whether in a local data centre or in a public cloud, makes data exposed to the same risks and breaches. Hence existing security measures can be implemented [56]. Nevertheless, cloud computing inherits risks from the core enabling technologies such as multi-tenancy, web services, utility computing and the internet[78][65].

This combination of cloud technologies makes the existing security controls not applicable, thus requiring further research and appropriate modification [136]. Besides, the concept of security and privacy are different depending on the law of a given country or business requirements. This leads to different requirements and protection mechanisms for data [57].

The centralized nature of resources and data in the cloud presents a more attractive target to attackers[55], where one successful attack can make way to follow up attacks against the whole system. This show how severe is the potential for security breaches in the cloud. A number of real world security incidents have been reported in literature that proves possibilities of cloud attacks [42][65][105][135][136]. The main causes of these security incidents are a customer's lack of physical control and the multi-tendency shared environment[55][61][135], which are vulnerabilities in cloud computing[166]. Surveys conducted by International Data Corporation (IDC)[75] in 2008 & 2009 shows that security is the top concern and barrier for cloud users, which reflects why the topic of security has been considered the primary research focus in the area of cloud computing [55][61]. The following sections refer to a review of literature that highlights the technical and non-technical issues that relates to the security and privacy of cloud computing.

3.4.1 The Cloud: Technical Issues

In literature several researchers have addressed cloud security and privacy from the perspective of industry, governmental and academia to determine research gaps, propose solutions and provide guidelines on best practices. Gartner Inc[53] was one of the first contributors to cloud computing. Their work titled "Assessing the Security Risks of Cloud Computing" published in 2008, warns organizations about the danger of migrating to the cloud, without performing a risk assessment in order to evaluate cloud specific risks, such as privileged user access, compliance, data location, data segregation, availability, recovery, investigative support and viability. Further the European Network and Information Security Agency (ENISA)[55] published a research article titled: Cloud Computing: Benefits, Risks and Recommendations for Information Security in November 2009. The document details a cloud computing risk assessment and provides guidelines on technical, organizational, and legal issues of cloud computing. It further introduced cloud vulnerabilities. Cloud Computing Security Alliance (CSA)[42] is another well-known organization that has conducted comprehensive research on cloud security, with a help of expert volunteers. They published their first report in December 2009 titled: Security Guidance for critical Areas of Focus in Cloud Computing [42]

and updated it in November 2011 as version 3.0. The report provides analysis of cloud risks identified in thirteen domain areas considering the architecture, legal and operational aspects of the cloud, with recommendation on technical security controls. In 2010, CSA released another set of guidelines titled: "Top Threats to Cloud Computing V1.0", which identified seven top threats related to cloud computing. In 2013 this work was extended and updated as "The Notorious Nine, Cloud Computing Top Threats in 2013", the threats ranked in order of severity [64], see figuee 3.1. As compared to the previous version of the guidelines, some shifts in ranking is noticeable, where data breaches have been moved from the 5th ranked in 2010 to the 1st ranked in 2013. This observation is not surprising due to the volume of data centralized in the cloud at present, which attract more attackers.

Ranking	2010	2013
1	Abuse and Nefarious Use of Cloud Computing	Data Breaches
2	Insecure Interfaces and APIs	Data Loss
3	Malicious Insiders	Account Hijacking
4	Shared Technology Issues	Insecure APIs
5	Data Loss or Leakage	Denial of Service
6	Account or Service	Malicious Insiders
7	Unknown Risk Profile	Abuse of Cloud Services
8		Insufficient Due Diligence
9		Shared Technology Issues

Figure 3.1: CSA Top Threats ranking in 2010 & 2013[42][64]

In [53][55][42] a number of organisations identified the security risks in cloud aiming to provide recommendations and guidelines when using cloud computing. However, no technical details have been provided as how to secure the infrastructure or data and how to achieve compliance to data protection law[68].

In publications, the paper [77] conducted a quantitative analysis on cloud security challenges and identified seven cloud-specific issues that have extensively received more attention in literature in terms of problems and solutions. The author classified them into a security model (considering network security, data security, interface, compliance, governance, legal issues, virtualization). The results showed that compliance, governance and legal issues received more solutions than problem citations, whereas the technical aspects such as virtualization, data leakage and isolation received less citation in terms of solutions. In [142][135][82], and the security and the protection of cloud infrastructure focused on trusted computing, cryptography and access control mechanisms. Similar mechanisms have been stated in video surveillance security[114]. Implementing any of the these mechanisms depends on the identified security metrics to quantify the improvement to system security and to compare security alternatives with similar functionalities

[116][114]. Given above, an attack surface metric can be used to identify the access entry points that attackers exploit to target data integrity, confidentiality or availability and hence decide on security measures. Frank [67] presented cloud specific security attacks in a technical infrastructure as a service(IaaS) cloud environment. The author considered these risks as attack surfaces in IaaS caused by malicious insiders (i.e a rogue cloud provider or malicious tenant). (C). Two scenarios of cloud infrastructure were illustrated and discussed, namely: multi-tenancy cloud host and single-tenant cloud host. In multi-tenancy scenario, multiple customers in a form of Virtual machines (VM) reside on the same physical machine and share resources. A single-tenant multiple virtual machine is only dedicated to a single customer, this concept is also called an off-private cloud. Both scenarios present security risks.

3.4.2 The Cloud: Non-Technical Issues

Legal issues and compliance have been recently addressed by researchers [48][111][103][66] analyzing the key issues outlined by ENISA. Within the context of this research we will focus on the legal issues related to data protection, data security and data location in the cloud, since they are considered main requirements for compliance with video surveillance laws. The following questions will be addressed in this section:

1. How data protection law applies and what are the responsibilities of the data controller (owner) and the data processor (provider) in a cloud environment?
2. How should data be stored and operated?
3. Where can data be stored?
4. Who can access data?

3.4.2.1 Data Protection

In common pubic cloud computing scenarios, personal data is processed and stored in a virtualized infrastructure, where multiple customers can share the same physical resources, and it can be transferred from one data center to another, without the knowledge of the next location of resources. This can violate data protection laws of an organization's asset if no prior risk assessment was performed [53]. Two documents providing guidelines have been published on the use of cloud computing by the European regulator [44] and UK Information Commissioners Office (ICO) [95], which approves the use of cloud computing. The documents provide guidelines to protect personal data in the cloud, explaining the procedures to be

considered prior to moving to cloud computing to protect personal data and lists the duties and obligations of data controller and data processors, in order to comply with the principles listed in EU Data Protection Directive 95/46EC and UK Data Protection Act 1998(DPA). Video data constitutes personal data thereby falls under DPA [40]. The following sections will discuss the main points in both ICO's and DPA's guidelines that are related to cloud computing.

1- Roles of the data controller & data processor:

How does the data protection law apply to the roles of the data controller and data processor in a cloud environment?

The guidelines emphasized the need to identify the data controller (owner) and the data processor (operator) and their interaction to identify who is responsible to be compliant with data protection laws. This helps the cloud customer to understand their obligation and what data protection risks that cloud computing presents and similarly, for the cloud provider to understand data protection requirements to make their service more efficient to customers that are subject to DPA laws [95].

The guidelines defined the controller as the one who determines the purpose of processing personal data and has the highest responsibility for complying with the DPA.The processor is the one who processes personal data on behalf of the controller [95]. Applying these roles to our proposed cloud-based video surveillance model gives the following assumption:

(The organization is the operator of the video surveillance system, for example a local government council. They use a third party application for forensic video analysis to run in a cloud computing environment. The organization will be a data controller for the video data processed by the application since they are the one who determine the purpose for which video data is processed. Cloud computing platform will be acting as the data processor.)

Now by identifying the organization as the data controller, we understand that all the duties and obligations imposed by the Data Protection Act 1998 are upon the controller (data owner). This relates to the collection, storage, retention, access, and ensuring that security measures are adequately placed by the processor.

2- Data Security:

How should data be stored and operated? The Seventh principle of the Data Protection Act states that: "Appropriate technical and organisational measures shall be taken against unauthorised or unlawful processing of personal data and

against the accidental loss or destruction of, or damage to, personal data" In accordance with this principle, the security requirement is only applied to the data processor by having to select the appropriate security measures taking into account the type of data being processed and the harm that might result from unauthorized access and misuse of the system. Putting this into the context of the cloud, the location of data in relation to the data controller is different for a public cloud. The data is stored remotely and the data control depends on the cloud service model. Compliance with the seventh principle requires that the cloud provider provides the basic security to data, and the customer (data controller) reviews the guaranteeing of availability, confidentiality and integrity of data through following an audit trail [95]. Figure 3.3, shows the relationship between the role of data controller and data processor.

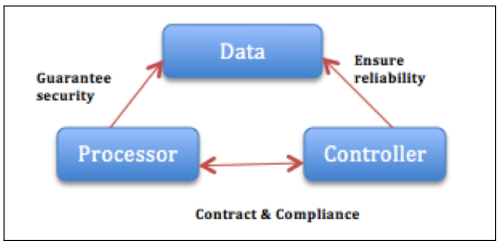


Figure 3.2: Roles of data controller & data processor

Given the above, The UK ICO guidance advices the data controller to assess and monitor the security measures by arranging an independent third party as a part of a standard certification to conduct a security audit of provider's services [117]. This will help a customer to monitor and check if the provider implements appropriate security and also to comply with its data protection obligation. It further reminds the customer to encrypt data in transit and at rest, to keep the encryption key at the customer premises, make sure all data copies made by the provider are completely deleted by the retention period. Data controller is not to be considered complying unless there is a written contract. Therefore, there should be a negotiation for SLA, including all requirements needed for data to be stored and processed in the cloud and to prevent the processor breaching the agreement.

3- Data Location:

Where is the data stored? The eighth principle of the Data Protection Act 1998 states that: "Personal data shall not be transferred to a country or territory outside the European Economic Area unless that country or territory ensures an adequate level of protection for the rights and freedoms of data subjects in relation to the processing of personal data", Cloud provider may have data centers

distributed across different geographical areas. This results in different laws and jurisdictions applying across countries. A consumer may specify the location of where data should be stored in their contract with the cloud provider (e.g. the Amazon cloud), However, determining which specific server or storage device will be used is difficult to verify due to the dynamic nature of cloud computing [155]. Even if they do, data may be subject to transfer without being informed[103]. This result in cross-jurisdiction by having to determine what law applies to which country and activity. Referring to the eighth principle, processing personal data is only restricted in EEA and to countries listed in the Safe Harbor Scheme[20] that can ensure an adequate level of protection to comply with all principles and the Act as a whole. There are some exceptional cases where data can be transferred to a non EEA country but this requires conducting a priori risk assessment. The cloud provider should guarantee lawfulness of cross-border data transfer and is included in a customer's contact agreement. Otherwise it could breach the eighth data protection principle.

4- Subpoena and E-discovery:

Who can access data? When there is a subpoena by law enforcement agencies for investigation, they may have the power to require the cloud provider to give them access to personal data. However due to the shared multi-tenancy architecture, this may cause other customers who may reside on the same physical servers to be at risk of the disclosure of their data to undesirable agents. One solution that can solve this problem is to encrypt data to ensure data protection in case provision for such disclosure [55]. However, malicious insider is another possible threat. It can be that the employers working for the cloud provider who have access to the system or an attackers virtual machine resides on the same physical machine where data is stored.

It has been shown above that many security issues are found in cloud computing, whether technical or non technical, due to a customer's lack of control and multi-tenancy nature of cloud computing. The security and privacy laws that regulate video data does not take the virtualization environment into account, which present challenges for a cloud provider to comply with [68] within a cloud based video forensic system. For example Amazon AWS [13] states that its virtual infrastructure has been designed to provide high security and ensure complete customer privacy to promote compliance with for e.g. healthcare and other governments needs [3][12]. However, a question of trust still remains as a challenge, whether cloud providers would comply with what they have promised,

and how transparent they are about security breaches. Therefore the potential use of clouds such as Amazon AWS for video forensics needs careful thought and trusted collaboration with the service provider.

3.5 Cloud Computing Performance Concerns

One of the guidlines presented in *"Surveillance Camera Code of Practice"* [46] highlights the need for a surveillance camera system to be capable of capturing, processing, analysing and storing images and information at a quality which is suitable for its intended purpose[46]. This principle is related to the performance issues around cloud computing, when deploying effective surveillance system in the cloud.

A cloud computing platform presents a unique apportunity for batch-processing video analytic tasks to deliver video analytics as a service by using multiple machines to analyse the significant scale of data at a reduced overall cost and less processing time, hiding the operational complexity of likely parallel execution from its user. The resources are provided from the cloud as virtual machines (VMs) which can reside on a single server or on different servers resulting in resource sharing for better system resource utilization. Many approaches exsit in processing big data in a cloud based platform to solve distributed computing problems and selecting of any particular one of them depends on the charactersitics of the data. For example, Hadoop, which is an open source implementation of the MapReduce model, has been widely adapted by the community for data storage and intensive processing. However, according to Ambrust et al.[47] the performance of a cloud computing based solution is unpredictable due to overheads sourced from virtulization and sharing of resources. A number of researchers have investigated this issue[172], but there has been no attempt to study the impact of virtualization on carrying out video analytics in a cloud based distributed system. In addition, cloud users will benefit from a deeper insight into the achievable performance improvement when a distributed computing approach is adopted for video analytics.

Cloud service providers such as the Amazon EC2 Cloud now support Hadoop user applications. However, a key challenge is that the cloud service provider's incapability to provide resource need estimate for user computing needss with specified requirements. For example a user requiring the real-time processing of 100 CCTV cameras simultaneuosly for video analytic/forensic purposes cannot obtain from the service provider an accurate estimate of the distributed computing resource, which will have to be allocated to the job. Currently, it is the user's responsibility to estimate the required amount of resources for their job running in a public cloud. While There are a number of proposed models for performance op-

timization of a Hadoop based system and for the associated resource provisioning for general data, the case of a performance model for hadoop-based video analytic system has not yet been studied. Nevertheless the bursty nature of video data that makes performance needs patchy and bursty makes video data different from general purpose data. Hence it is vital this research problem is investigated. In Chapters 4 & 5 a novel solution to this problem is presented.

3.6 Conclusion & Recommendation

In this chapter we have investigated the security and privacy related legal requirements and performance related concern in deploying cloud-based video surveillance systems. In particular the study was conducted in relation to a video forensic system that requires data storage both in its original and annotated formats, operating a number of video surveillance algorithms and conducting an effective search. Maintaining security at all levels of the video forensic system when deployed within a cloud is important. Table 3.1, summarises the key legal requirements that originate from the data protection act that governs the legal compliance of a video surveillance that can provide evidence that will be legally acceptable. The table further tabulates the challenges one must meet when using a cloud infrastructure to deploy a video forensic system.

Based on the information summarized in Table 3.1, the following recommendations can be made:

- Data within the annotation database of the video forensic tool should only be created when it is necessary for law enforcement purposes. This usually happens occasionally when there is an investigation request. When implemented within a cloud based environment to store the large amounts of annotated data produced when an investigation needs to be carried out, the on-demand resource pooling characteristics of a cloud should be effectively utilized. For effective processing of large-scale video data it may require several machines for parallel distributed processing. From the technical point of view, the use of cloud infrastructure brings many advantages to the video analytic architecture in terms of reducing investment cost on hardware that to be utilized occasionally and provide high scalability by easily increasing resources (server, processors, storage) to support a large number of cameras.

  Our research investigated the possible methods used to process video analytics in the cloud. Most of the previous work in cloud-based video surveillance system take advantage of the Amazon cloud [138][89] or build their own

Table 3.1: Summaries of the key legal requirements, the corresponding video surveillance system compliance and cloud computing challenges.

Legal Requirement (Data Protection Act)	**Video Surveillance System Compliance**	**Cloud Computing Challenge**
Fair & Lawful	Controller is responsible to ensure that the law is obeyed	Provider's Level of transparency is not clear
Purpose	Annotation of stored video database is carried out only when there is a need for a forensic investigation	Possibility exists for malicious insider attacks
Accuracy	Ensure authenticity & integrity of video data	Possible data loss /leakage/manipulation
Retention	Retention requirements can depends on organization using the system	For a complete removal of data a device need to be destroyed which is not possible in cloud environment. Also Attackers may be able to recover data due to resource sharing
Security	Protect annotation engine (i.e. processing algorithms), video database & annotation database	Protect annotation engine (i.e. processing algorithms), video database & annotation database
International data transfer	Transfer data only within EEA & countries having similar data protection laws	Specific data location is unknown

Table 3.2: Summaries of performance requirements, the corresponding video surveillance system compliance and cloud computing challenges.

Performance Concern	**Video Surveillance System Compliance**	**Cloud Computing Challenge**
Virtualization	Reliable quality	Perfromance unpredicted
Computing Resource	Scalability	Distributed computing efficiency

cloud-like environment[167], both approaches did not explain the processing workflow of video analysis algorithms used and how the video is distributed and stored. Distributed processing and distributed storage are the solution that our research focus on to solve storage and computational processing limits by choosing an effective data distribution scheme in terms of availability, security & performance.

- Security measures must be put in place to prevent video data from unauthorized access and to preserve accuracy, while in transit (network) and at rest (storage). Although various security measures are implemented by cloud providers, known real world examples exists of past, unpredicted breaches and outages. Although a definite solution does not exists as yet, on-going work by both academic and industry researchers should ensure improved levels of security in the future. The controller of a video surveillance system is responsible for ensuring that the system complies with security and privacy requirements. When implemented within a cloud based environment the cloud provider's level of transparency is not always clear to the controller. Therefore the controller can outsource to a reputable third party auditor to monitor security and levels of disclosure of data and if the cloud provider comply with SLA(Service Level Agreement).
  Nevertheless, video surveillance data and its associated metadata are very sensitive and not suitable to be stored in a public cloud. Using a private/hybrid cloud can be an alternative solutions at present to have wider control of the data. Processing video analytics in a private cloud will be the focus of this research.

This chapter identified issues that are important to consider when using cloud based technologies and the findings open new areas for significant research. The scope of this thesis is limited and related to the performance concern when using cloud, see Table 3.2. Considering the above requirements and challanges for deploying a scalable distributed video surveillance system, we found no literature that provide a significant research on the performance of cloud-based hadoop architures, specifically for video applications. However, the exsiting literature focused on the implementation of the system and the promising results motivated us to expand the research to provide a detailed analysis of the behaviour of video analytics application under different constraints and parameters applied to the cloud platform.

Given this, the chapters 4, 5 and 6 answer the research questions highlighted in Chapter-1 (i.e. RQ 2, 3 & 4) that are based on a selective video analytic application's performance analysis, modelling and performance optimization under

constraints computing resources, within a cloud-based Hadoop architecture.

Chapter 4

Video Analytics Applications Deployment on Hadoop

Deploying large scale video surveillance requires the use of significant computing resources that often need to be scalable. Such resources will also have to be of a distributed nature to satisfy practical design requirements and also sometimes procedural and legal reasons. The recent popularisation of cloud based technologies giving access to scalable and elastic computing resources make the cloud a viable option to support large scale video surveillance.

Chapter 3 described two key challenges behind the use of a cloud based approach for large scale video surveillance, namely, the security concerns around using the cloud for an application that has stringent data security requirements and the complexities behind allocating cloud based computing resources to an application that needs scalability and elasticity in resource allocation. Whilst addressing the first challenge is out of the scope of the research context of this thesis, the focus of this chapter is to initiate fundamental research that can eventually address the second challenge. However studying the resource allocation related issues within a real cloud is challenging due to the inability for a general user to control the allocation of resources. Building a private cloud for the purpose of research and development could be a costly and time-consuming task. As a result this is an area that has not been studied in detail in previous literature.

Considering the above observations in this chapter we propose the deployment of a selected video analysis task (i.e.face detection and motion detection) within a single Physical Machine (PM) virtualised to contain multiple Virtual Machines (VMs), supported by a Hadoop based architecture. In particular the focus is to identify the parameters that play a significant role in the distribution of computing resources and study their effect in the overall data processing speed. In Chapter-5 we demonstrate how these parameters can be used within a model that can then

be effectively used for the efficient allocation of resources.

4.1 Introduction

CCTV camera systems are installed in many public places to enhance security and surveillance. Often such data is gathered for manual processing by CCTV operators but more recently large-scale data collected by such syetems have been subjected to automated computer based processing, namely video analytics and forensics. Every CCTV camera produces large amount of video data per day. Therefore accumulating video streams often from a large amount of CCTV cameras present within a typical CCTV system produce a significant amount of data that conventional analysis platforms that are often supported by a single computer (or processor) may not be able to handle in a fast and efficient way. Thus there is a need for the use of a distributed data storage and processing platform such as a cloud (either public or private), to perform seamlessly scalable distributed video storage and processing. One important need in such a cloud based deployment is to fully understand the computing resource requirements so that such resources can be cost effectively allocated. For example a CCTV operator intending to use a public cloud would aspire to know what cloud resources need to be deployed so that the processing can be done in the fastest (or within a known time constraint) and the most cost effective manner. The same operator wanting to use a private, purpose built cloud will want to know the resource needs to estimate the cost of building and installing a private cloud that can serve the purpose for which it is to be used. Unfortunately such a resource modelling, estimation and prediction task is impossible to be carried out as a research exercise within a public cloud, due to limitations of knowledge of the operation of such a cloud to a general user. Further performing such an exercise on a purpose built private cloud will be both costly and time consuming. Review of literature conducted within the context of the research presented in this thesis has revealed that a number of industrial initiatives such as Intel[16],Pivotal[29] etc. and previous academic research have focussed efforts to deploy video analytic applications in a cloud-like, Hadoop environment to enhance performance and scalability. Such a design and deployment provides an environment that can be subjected to R&D in resource allocation in a flexible and unrestricted manner, thus making such an approach highly suited for the research being proposed. However it is noted that in a practical deployment the challenge of a Hadoop based architecture is that it requires several machines for effective processing, which then adds investment cost in the infrastructure. Therefore fully understanding the true resource requirements, given the knowledge of the CCTV task to be processed is important. In other words one should be able to model the

resource requirements in order to effectively predict and forecast resources to be utilised within the Hadoop based implementation. Unfortunately no work exists in literature addressing this issue in detail. Instead, the use of cloud computing infrastructure has been proposed in literature to solve both scalability and resource related cost [47], assuming unlimited scalability and ignoring cost-effective resource usage. It is noted here that cloud computing infrastructure is built on its core technology, virtualization, which provides on-demand elastic resource provisioning to meet scalable user's requirements, the same principle on which a Hadoop based architecture is built. Thus a Hadoop based architecture provides a cloud-like environment in which flexible, un-restricted research into resource allocation in a cloud based deployment can be effectively carried out. Given the above, in this chapter we investigate the resouce modelling and prediction of resource requirements in deploying a scalable video analytic application on a Hadoop based framework running on virtualized cluster. It is shown that this will enable one to model the resource needs when the same application is to be deployed in a cloud, hence providing answers to a number of open research and practical problems.

For clarity of presentation this chapter is organized as follows. In section 4.2 we introduce the design and implementation of a selected video processing application (face detection and motion detection algorithms) within a Hadoop MapReduce architecture. In section 4.3 we present the experiments that are conducted to characterise the performance of the implementation, enabling the modelling of resource requirements, in chapter-5. In section 4.4 we provide experimental results and a detailed analyses. In section 4.5 we provide discussion. We finally conclude in Section 4.6.

4.2 Methodology

In this section, the design and implementation of a simple video analytic system, i.e. a face detection and motion detection algorithems, in a Hadoop based virtual cluster environment is presented with the aim of investigating the research questions highlighted in section 1.2. Although the applications simple they are very much representatives of the type of most common video analytic tasks.

4.2.1 Video Dataset Description

The experiments were conducted on two different video datasets obtained from a benchmark website [34]. One video contains crowded scene with many images of different people (buddhist walking at a temple in queues), we refer to this video as type1. The other video contains less crowded scene of people walking in/out

a train station, we refer to this video as type2. These two terms will be used throughout the thesis to distinguish between the two videos. More details about each video file is given in Table 4.1.

Table 4.1: Video files details .

Video Type	Content	Resolution	Format	Frame Rate
type1	crowded	720x576 & 360x288	mp4	25
type2	less crowded	720x576 & 230x288	mp4	25

4.2.2 Video Applications Description

4.2.2.1 Face Detection Algorithm

The algorithm used in this thesis is based on Viola Jones face detection algorithm using Haar Feature-based Cascade Classifiers. The idea is to scan the detector many times through the same image each time with a new size. The face is detected and the feature is extracted using Haar feature where each feature is a single value obtained by subtracting sum of pixels under white rectangle from sum of pixels under black rectangle.

Viola jones algorithm uses a 24x24 windows as the base windows size to start evaluating these features in a given image. The algorithm uses Adaboost a machine learning algorithm to eliminate the large number of haar features for every single 24x24 sub window in any given image that can be redendent or not useful and select only the features that are very useful for the prupose of detection that needed to evaluate. After these features are found, a weight combination of all these features is used in evaluting and deciding any given window has a face or not. And to reduce the computational power needed to sum up all th pixel values under the black and white rectangles every time, the algorithm use the concept of integral image to find the sum of all pixels under a rectangle with just four corner values of the integral image. The algorithm uses a casdade classifer composed of stages each contaning a strong classifer. So all features are grouped into several stages where each stage has certain number of features to determin whether a given sub window is face or not. A given sub window is discarded as not a face. Figure 4.1 illustrate how the algorithm works in our video type1 dataset.

4.2.2.2 Motion Detection Algorithm

The algorithm used in this thesis is on background subtraction based on frame difference method. It detect moving of object from a sequence of frames, i.e. from

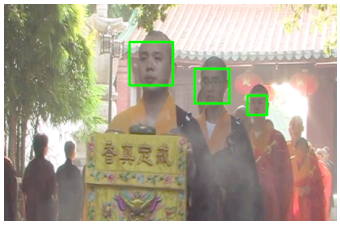


Figure 4.1: Image output from face detection Algorithm

the difference between consecutive frames. It adopts pixel-based difference to find the moving object. When there is no movement in the image sequence then the difference between the two images shows a black binary output image. When there is a movement the binary image of the difference between the two frames shows motion having white colour and where there is no change shows black colour. Figure 4.2 illustrates how the algorithm applies on our video type1 dataset.



Figure 4.2: Image output from motion detection Algorithm

4.2.3 Hadoop System Design Overview

The scalable Hadoop-based video analytic architecture used in this research is similar to that adopted in [149] and [179]. However, the proposed platform is virtualized. The system consists of an Apache Hadoop framework (discussed in section 2.3) and a collection of open source software applications such as the OpenCV library for video processing, FFmpeg for video splitting & frame extraction, javacv wrapper for integrating OpenCV and fuse-dfs, to build a distributed video analysis system. For each virtual node of the Hadoop cluster, a Hadoop distributed file system (HDFS) is used for storing video splits ready for processing. This approach is highly fault-tolerant and is suitable for large datasets[25] and using a MapReduce framework for distributed computation. Figure 4.3, shows the architecture of video processing using the Hadoop framework. The following

sections give a brief introduction about the functionality of the various system components.

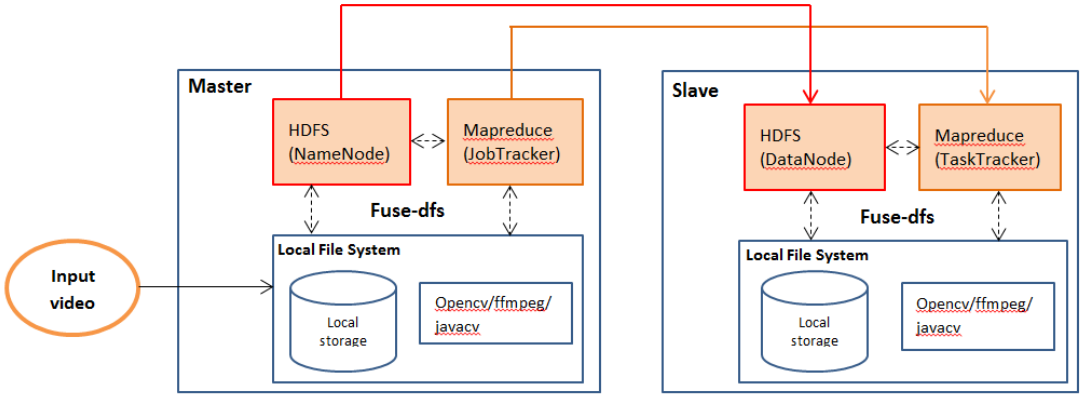


Figure 4.3: Hadoop framework for video analytic application

4.2.3.1 Input Video

The default way for HDFS to manage input data format is to split data into chunks or blocks to be spread across a cluster nodes for distributed parallel processing, each data chunk is an independent sequence of a data record. This method has been utilized in literature for text data formats in order to read line by line, records such as, logs, web documents etc. However, it is rather a challenge to adopt the same approach in handling video file formats, as video data comes in different format of containers, which consist of dependent frames that need an efficient mechanism to split video at particular boundaries that makes each video split meaningful[126]. Video files that comes from CCTV footage are stored as compressed files, therefore compression format should be taken into account when reading video chunks to decode into frames processed by mappers. Given this, the custom Inputformat & RecordReader classes of Hadoop are required to overwrite the default approach to read data.

In literature different approaches have been experimented on video input format to enhance overall performance. For example [85] suggested the use of a 10 MB input file size. However, our preliminarily results show an increase in execution time when the file split sizes are smaller than the block size. This performance degradation resulted from the overheads caused by starting and initiating many mappers to process each block individually. In addition, our result might also be in influenced by virtualization overheads. The authors of [59] used GoP techniques and the authors[149] read a video file as a single input file. In our work we followed[149] to avoid open GOP related issues.

4.2.3.2 HDFS

To analyse a video file, HDFS should be able to read/write the recorded video file and make it available to mpeg and OpenCV libraries for frame extraction and video processing. The challenge is that both libraries can not be directly accessed by HDFS since they are designed for a local system. Therefore, fuse-dfs module, based on the Filesystem in Userspace project (FUSE)[19], was selected as a method to mount HDFS on all nodes to a local file system.

4.2.3.3 MapReduce-based Video Analytic Application

For the proposed experiment a simple face detection algorithm and motion detection algorithms were implemented and tested as the custom MapReduce job. The system makes use of FFmpeg for video file decoding and encoding and OpenCV for the execution of the algorithms. Unfortunately these applications are C and C++ based native libraries, whereas Hadoop is a java based run time environment. Therefore the javacv wrapper was selected to provide a java API to Hadoop.

In the proposed research the Mapreduce-based algorithms were implemented by modifying the default java classes utilized in different phases of mapreduce data flow. Figure 4.4, illustrates mapreduce data flow showing the connection between system phases and detailed steps of processing one video file, named *InputSplit*).

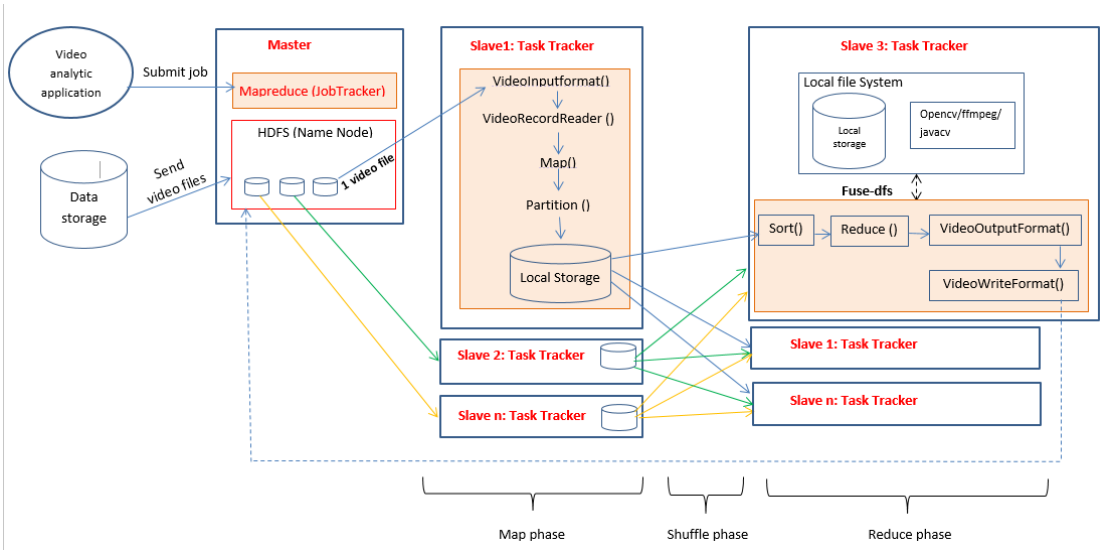


Figure 4.4: Hadoop performing a video analytic job

As illustrated in figure 4.4, initially when video file is stored in HDFS it is generally divided into logical separate files *InputSplits* of the same size and distributed them across the cluster of VM nodes (see Figure 4.4). The known storage locations of the *Inputsplits* are used by the Hadoop system (i.e. the master) to

schedule map tasks on the tasktracker (of VM nodes), where data splits resits. It is worth mentioning that a mapper takes the file as an input, so data locality becomes important.

In our case we consider the input video file as a complete file to be processed as one mapper by overwriting default Hadoop *isSplitable()* method in *FileInputFormat* class. We avoid splitting the input file for reasons detailed in section 4.3.1.1, i.e., a compressed video file consists of correlated frames and hence random splitting will cause dependent frames to be processed in different *Inputsplits* thus gives non-decodable files by FFmpeg.

When mapreduce face detection or motion detection task is executed, typically it should first calculate the splits for the job by calling *getSplits*. In the proposed configuration only one Inputsplit is considered as discussed above. The application will send this split to the master jobtracker to schedule a map task to be processed by the only tasktracker (a VM). The details of the map and reduce phases are as follows:

- **In map phase:**
  - **VideoRecordReader class:** Map task uses ReaderRecorder to decode and extract the sequence of frames out of the InputSplit by calling FFmpeg tool. Each decoded frame is then represented by a key-value pairs. The Key is a unique frame id corresponding to the frame number within the sequence and the value is the data of the corresponding frame. Subsequently Inputsplits in a form of key-value pair are sent to the map function to process. For instance a video file have the following sequence of frames & transformed into (key,value) pairs, see Figure 4.5:

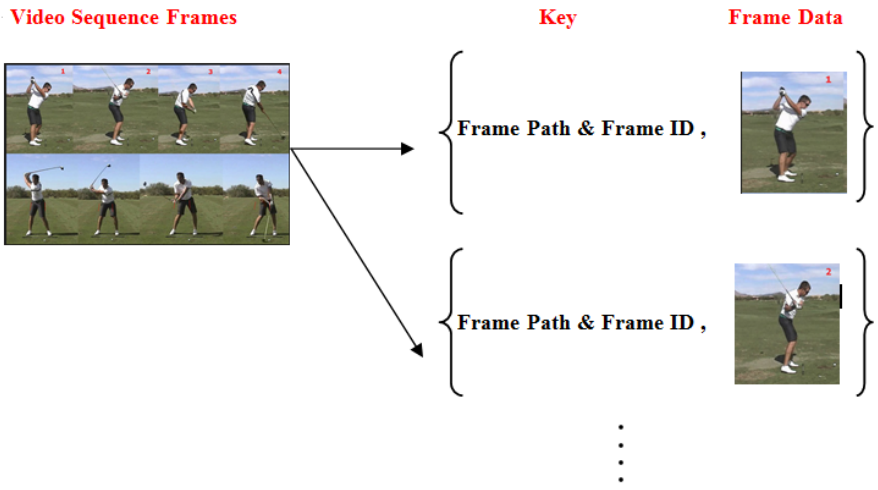


Figure 4.5: Video (key,value) pairs generated from decoded frames

  - **Mapper function:** Takes key-value pairs generated from previous phase and subsequently group them depending on the video analytic algorithms requirements as single frames like face detection algorithm or series of frames like motion detection algorithm. If there are more than one reduce mapper the output is partitioned by key and is sent to the buffer as input for the reduce phase. Map output is named as an intermediate output.

- **In Reducer phase:**

  - **Shuffle phase:** Transfers intermediate data from the mapper nodes to the reducer nodes scheduled by the jobtracker. Reducer deals with (key, value) as input, therefore any node (VM) can perform the reducer task and there is no need for concern about data locality.
  - **Sort phase :** It sorts intermediate inputs that comes from the different mappers, by key.
  - **Reducer function:** Each reducer takes all key-value pairs with the same key and merges them, and subsequently applies the face detection or motion detection algorithm on the frames (i.e., values) according to the instructions within the java code representing the computer vision algorithm. Finally the results are sent to the class OutputFormat.
  - **OutputFormat:** Generates output in a form of text including the frame number in which a face is detected and the locations of face/s on the frame. Finally the Record-Writer is used to write the results to the HDFS, ready for the application to read. .

The output for Mapreduce face detection application is written in a text file showing the coordinates (left, top, width, height) as location of faces in each video frame(images). The output for Mapreduce motion detection application is also written in a text file showing the number, time and duration of the detected motion. We checked the accuracy of these applications when running in hadoop environement with that running in stand-alone system, we found similar results in both scenarios. This is expected since in Hadoop distributed system each machine processes the same application code on every video file then merges the output.

Table 4.1 provides the pseudo code for the mapreduce functions of the face detection (i.e applied on frame by frame basis) and motion detection algorithm (i.e applied on overlaped frames).

Table 4.2: Pseudo code for the implementation of a single-frame and overlapped-frame oriented applications based on Hadoop MapReduce.

Map Phase:	
Inputs: <frameID,frame> **Outputs:** <groupID,EncodedFrame>	
if Single-Frame-App	// if the application is single frame oriented
groupID=frameID EncodedFrame=frame	
else	
groupID= get-episod(frameID)	// determine which group this frame belongs to
EncodedFrame= <frameID,frame>	//encapsulate each frame with its id
end	
Reduce Phase:	
Inputs: <groupID,encodedFrame-set > **Outputs:** <groupID,output-data>	
if Single-Frame-App **for** each frame **in** encodedFrame-set **do**	
output= proc-single-frame(frmae)	// a custom proedure processing a single frame
output-data.add(output)	
end	
else	
encodedFrame-array= sort (encodedFrame-set)	//restore the order of the frames in one group
output-data= proc-episode(encodedFrame-array)	// a procedure for processing an episode
end	

4.3 Experimental Testbed Set Up

4.3.1 Virtual Cluster Configuration

The Hadoop-based face detection/motion detection applications described in section 4.2.2 are implemented in a small scale virtual environment consists of one physical server machine deployed with KVM (Kernel-based Virtual Machine) that consolidated into multiple machines called virtual machines (VM). Each VM operates independently from the others. As described in Figure 4.6, master and slave nodes are built upon a virtual cluster sharing resources such as CPU, memory and network I/O. This type of deployment has advantage of saving power consumption and maximize resource utilization.

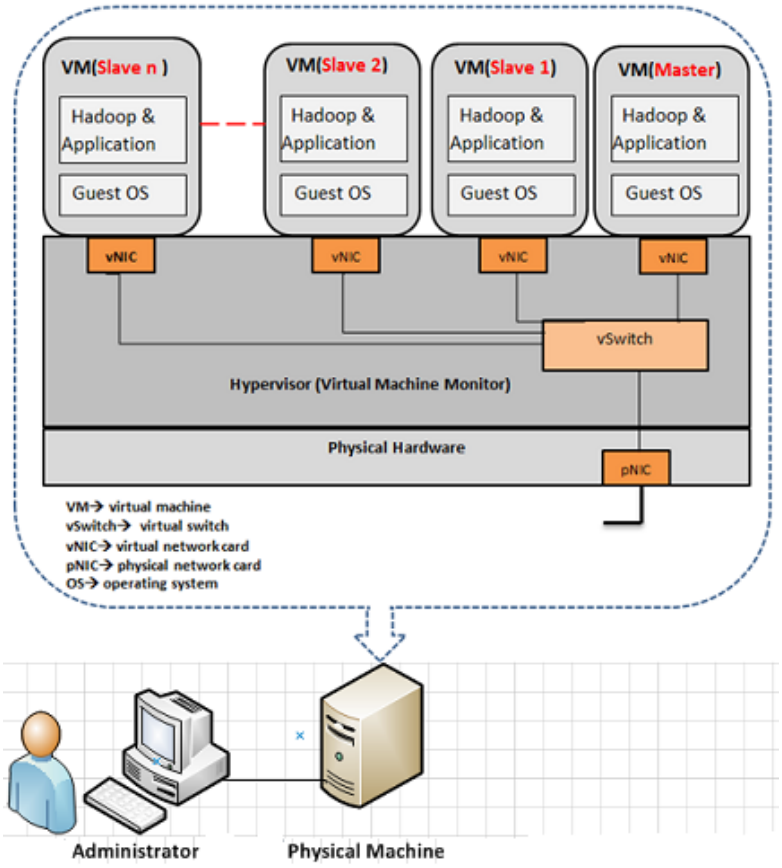


Figure 4.6: Hadoop virtual topology

The experiments were evaluated on a Hadoop virtual cluster consisting of nine virtual nodes, i.e., one master node and eight nodes dedicated as slaves. The hardware conguration is the same for all virtual nodes (4 cores & 8 GB memory). The software configuration across all node are given in Table 4.3.

The application run time is our performance metric to investigate the gained acceleration. We used cluster size, video format, video le size, resource capacity as

Table 4.3: Software configuration for the cluster of VMs.

Software	**Version**
Guest Operating System(OS)	CentOS 6.6
Java enviroement	jdk 1.7.0
Hadoop distribution	Apache Hadoop 1.0.4
Replication factor	2
OpenCV	v1.0.4
JavaCV	Compatible with OpenCV
Ffmpeg	v0.6.5

controlling variables to evaluate the behaviour of Hadoop cluster. Each experiment was conducted 3 times under exactly the same condition.

Hadoop comes with more than 100 default conguration parameters[162]. In our experiments we changed some of the signicant parameters to suit the underlying cluster resource capabilities in terms of handling the application in Hadoop. The replication factor was set to two, since it is a small scale cluster. The Java Heap size was set to 1024MB to avoid memory swap and reduce the limitations in terms of each running task. The number of map and reduce slot numbers which indicates how many parallel tasks to execute by one datanode(VM), vary depends on the experiment type and the available resources which is discussed in section 5.3.1.1.

4.4 Experiments Results & Analysis

In this section we present the performance of the face detection and motion detection algorithms implemented based on a virtualised Hadoop cluster, the design and implementation of which was described in the above sections. It is noted that the objectives of the research proposed in this chapter is twofold: first to investigate the performance gain achievable by the use of the Hadoop based virtual cluster when compared with running two video applications, face detection and motion detection algorithms in a standalone desktop computer that has been configured with setting similar to a virtual machine used by the Hadoop based cluster.Second, to evaluate the behaviour of the above systems when different video related parameters and computing resources are used.

It is beneficial to study how the Hadoop based virtualised cluster of machines can effect the algorithms performance, when different sizes of input video is processed. For this, we conducted a number of experiments to evaluate the performance gain considering execution time and speedup as performance metrics. Graphs plotted in Figure 4.7 shows how Hadoop based framework is able to accelerate the execution time using 1 to 8 virtual machines, when processing input

video footage from 4 min to 20 min duration in incremental steps of 4 min. It is observed that generally the execution time increases with the increase of input file size/duration. This is expected as the face detection algorithm operates on a frame-by-frame basis requiring ideally the same time to process each frame. The same observation found in motion detection algorithm, shown in figure 4.8.

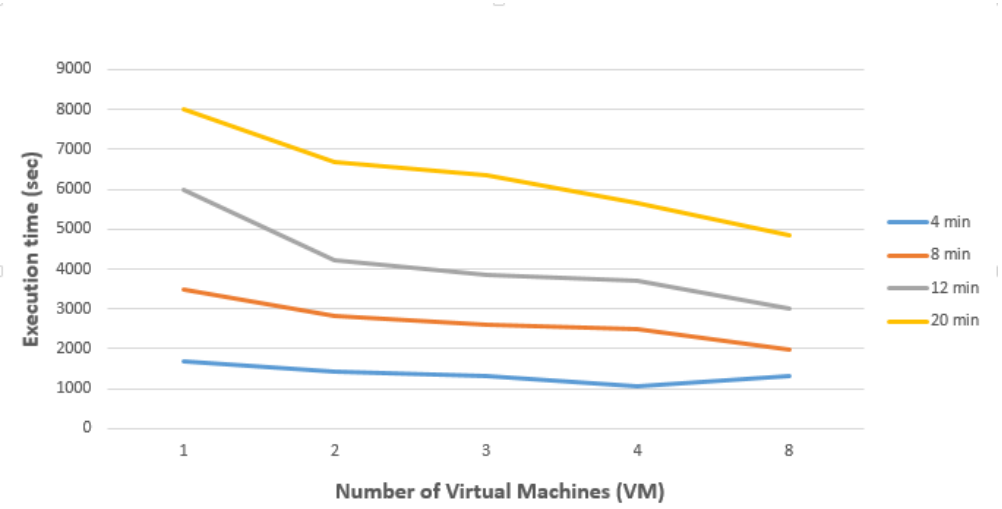


Figure 4.7: Execution time against number of VMs for face detection algorithm

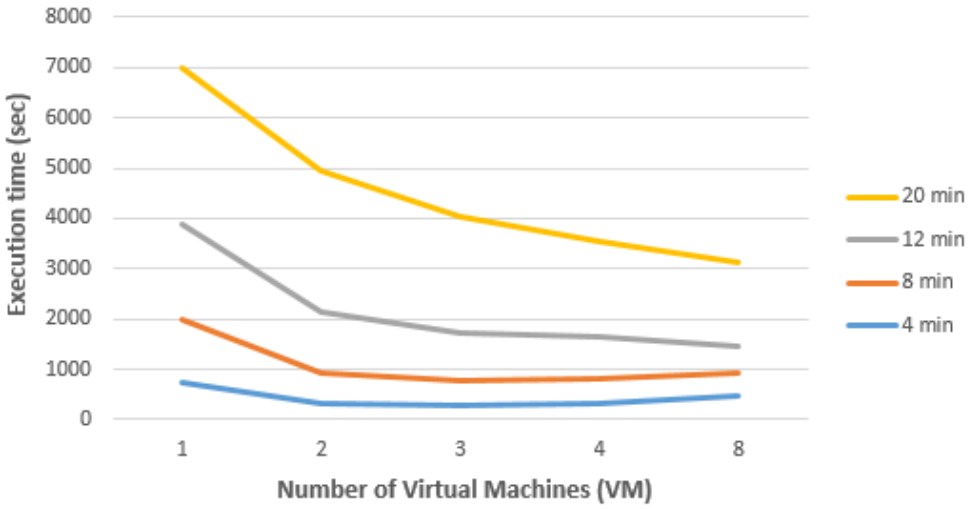


Figure 4.8: Execution time against number of VMs for motion detection algorithm

An interesting observation found in figures 4.7 and 4.8. In Figure 4.7 when using the 20 min input video the execution time falls by approximately 37.5% when the number of VM is increased from 1 to 8 and for the shorter input video of 4 min this reduction is only about 6%. The larger number of VMs is able to more effectively handle larger input video files due to the efficient handling of distributed processing of data within the virtualised Hadoop environment. A closer inspection

of figure 4.8 reveals that when the input video was 4 minutes long, upto 4 VMs the execution time gradually decreased but an addition of a further VM slightly increases the execution time. This is due to the fact that at a lower processing need of the 4 min input video the fifth VM will be largely under utilised but will need extra time with regards to overheads in the inter VM communication. It is noted that at higher input file sizes an increase of the VM number from 4 to 5 continues to decrease the execution time. This is due to the fact that under higher input file processing needs the fifth VM will be also be better utilised and will therefore outperform the cost of overhead communication.

To further analyse the results presented in the Figure 4.7 and 4.8, we calculate the speedup of Hadoop applications in terms of computation when processing the four different sized video files using Amdahl's Law[6]:

$$S = T_S / T_N \tag{4.1}$$

Where, T_S is the execution time of the face detection algorithm on a single VM and T_N represents the execution time of the Hadoop-based face detection on N number of VMs. The results of this calculation are displayed in figure 4.9 for face detection, and figure 4.10 for motion detection. It is clear that the Hadoop-based virtualised distributed architecture achieves the best speedup in computation when larger input file sizes are being handled.

These experiments demostrate the capability of hadoop to process large video files by different applications charactristics and different video types.

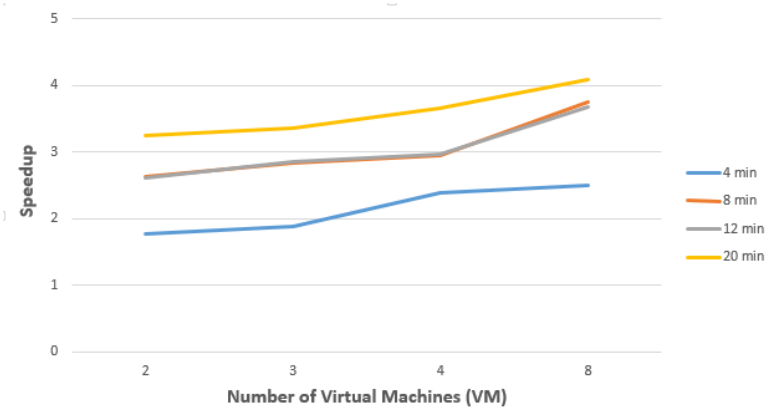


Figure 4.9: Speedup analysis of Hadoop-based face detection

After achieving the first research objective of the chapter as discussed above, a series of experiments were conducted on the implemented distributed video processing system to study the impact of various configuration parameters of the Hadoop architecture on system performance and performance effects of data scaling. The key observations are summarised as follows:

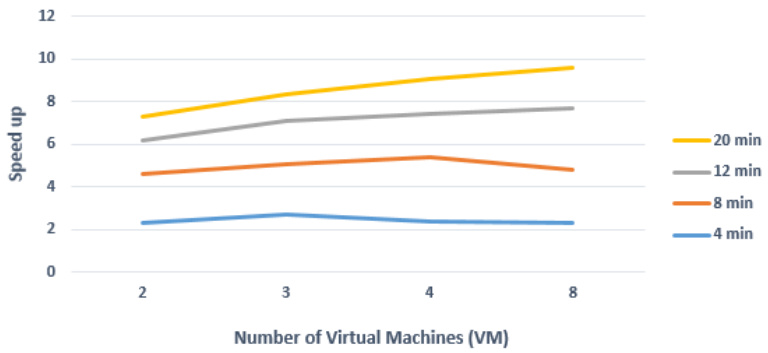


Figure 4.10: Speedup analysis of Hadoop-based motion detection

- When increasing the duration of the input video file, in other words when processing more frames, total processing time increased. The results tabulated in figure 4.11 shows that it is the reduce phase that takes the most amount of time for execution in all cases of different input video file size. We tested different video input file sizes, and in each time the reduce phase consumes most of the application job execution time. This is because it is in the reduce phase that the application algorithms are executed and applied on each single input frame. Initially for our experiments we used one slot within each reducer. This prevents any part of the video analytic application, run in a parallelised manner, hence needing the maximum time for the reduce phase.

Video file sizes (min)	Job Total Execution time(sec)	Completion time (sec)		
		Map phase	Shuffle phase	Reduce Phase
4	1601	280	306	1594
8	1608	273	305	1600
12	4721	806	833	4703
20	7353	1364	1393	7345

Figure 4.11: Processing time when different video input size is used.

- With the aim of improving the reducer performance we added more reducer slots to the VMs that contains the reducers, motivating the VM to attempt to parallel process the video content. Although we noticed a slight improvement in performance, this improvement was insignificant. The reasons are two fold. The video analytic application being investigated has not been implemented with parallelisation in mind (in fact it is serially written code) and

thus even in the presence of multiple reducer slots the reducer tasks cannot be effectively run in parallel. The slight improvement in performance is due to some presence of unintended parallelism that is being better exploited by the presence of multiple slots as compared to the case where each reducer has one slot. This experiment justifies the need for not only provisioning an appropriate virtual hardware configuration for parallelised implementation but also the need for software parallelism in the code implementing the video analytic application. It is noted that the latter is beyond the scope of the research context of this thesis and is suggested under further work in the thesis conclusion chapter, Chapter 7.

- Figure 4.12 shows an increase in processing time when number of video input files increase while the number of VMs or the cluster size is fixed. This is because the number of parallel tasks that can be run in the reducers gets subjected to an upper bound. Due to this reason if one increases the number of input video files each file requires separate mapping tasks (VM) to be processed in parallel. Moreover, the overhead caused by starting up and shutting down the required tasks increase processing time.

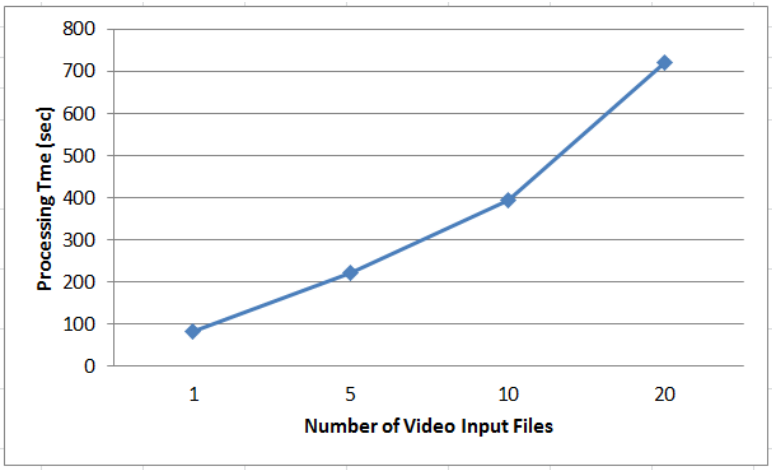


Figure 4.12: Processing time variation with different number of input video files when the number of VMs are held constant.

- Referring to Figures 4.7 & 4.8, they illustrate the change of execution time when the input video file size is held constant but the number of cluster nodes are increased. To further investigate the reasons behind these observations, we used the monitoring tool Ganglia[21], which is an open source tool, to monitor a virtual machines resource utilization during a job's run time. The tool allows the utilization analysis of individual nodes. A close look at the analysis results indicated that for the video analytic task at hand, upto 3 nodes, independent sub tasks from the video processing was automatically

identified by the operating system that could run in parallel and hence all three VMs were effectively utilised. However when more nodes are added, the actual advantage that three VMs provided was due to further attempted and unbalanced operations assigned to the fourth VM onwards. Whilst the excess VMs remain underutilised, the inter processor communication overhead increases the overall processing time needed. In chapter 5 we provide further experimental analysis that enables us to study this behaviour in more detail.

4.5 Discussion

The aim of this study was to investigate the performance gain of a typical video analytic application when implemented based on a Hadoop based virtual cluster of machines. The research conducted within the context of this chapter showed that a Hadoop based virtual system was easy to setup, provided the flexibility to easily manage and also proved to overperform local single processor based systems that are typically used to implement video analytic systems due to the advantage of used distributed processing.

The video analytic application used for the experiments conducted are a simple face detection algorithm that work based on a frame-by-frame basis to detect human faces and motion delection algorithm. Conceptually for face detection this means that the algorithm requires the same amount of processing to be carried out (i.e. same level of image processing) in each frame. For motion delection algorithm a group of overlapped frames are processed together. The software implementation used for the algorithms were not parallelised hence the amount of parallelism exploitable was limited to the highly likely possibility of divided processing of frames (as the face detection is done independently in each frame) but not the inherent and algorithmic parallelism. Given the application, algorithmic parallelism obtainable is anyway limited and hence deemed to be not important for the conducted experiments. It is noted that the following summary of performance is best applicable to only video analytic applications of the nature described above. The performance of the video analytic applications when implemented based on the Hadoop based virtual cluster were tested under different choice of distributed resources, obtainable easily by exploiting parameters of the Hadoop based configuration.

Keeping the resource availability fixed (i.e. fixed number of virtual machines, reducer slots and number of separate input files) when the input file size was increased (i.e. input data was scaled mimicking a typical application scenario of large scale video analytics) the required processing time increased. However for a

given input file size if the number of slots was increased, up to a certain limit, the processing time decreases due to multiple groups of (key-value) pairs processed in parallel. Beyond a certain number of reducer slots, the time taken gradually increases from reaching a minimum due to the under utilisation of clusters and the time needed for communication (i.e. hypervisor) overhead. Hadoop based parameters can be used to improve overall system performance [162][97]. In the case of changing the reducer slots the existing studies[59][149][179] have only used two reducer slots, the default value of the Hadoop configuration.

In addition to the above observations it was shown that increasing the number of input files (i.e. dividing and hence parallel feeding the input video) using a fixed cluster size (i.e. VMs), and fixed number of reducer slots, resulted in a almost linear increase of execution time as each files processing needs competes for computing resources, such as each file requires separate mapping tasks (VM) to be processed in parallel by the reducer slots. Moreover, the overhead caused by starting up and shutting down the required tasks increases processing time.

The experimental results summarised above shows that there should be a method to allocate the right resources to meet processing requirement of a video analytic algorithm in a Hadoop based architecture. As a number of configurable Hadoop parameters exists, it is important to determine which parameters play a more significant role in determining the total execution time. However this depends on the nature of the application and also the data being processed and will hence depend on parameters external to those configurable within Hadoop. Although previous studies have been conducted to investigate the impact of various Hadoop based parameters[93],[24],[171] [59],[149] no work exists that looks at the effect of the above mentioned wider set of parameters that impacts performance. The focus of the research presented in Chapter-5 is addressing this research gap in carrying out video analytics within a Hadoop based architecture.

Our study indicates that performance gain of the application in the Hadoop virtual cluster requires maximizing resource utilization in individual VM and provide appropriate number of VMs in cluster according to the input load. These findings have previously been under-presented in studies.

4.6 Conclusion

This chapter discussed the deployment of a scalable video analytic application on a cluster of virtual machines that are implemented on Hadoop based virtual framework. A face detection algorithm that works on a frame-by-frame operational basis on input video was used as the video analytic application as well as motion delection algorithm was used to confirm the effectivness of our approach

when using different type of video application. The results demonstrated the capability of the Hadoop based virtualised cluster to reduce the execution time if an appropriate Hadoop configuration was used, dependent on how the input video was fed. The experimental results and a detailed analysis revealed that a Hadoop based distributed and virtualised system provides a flexible, scalable and effective platform to carry out large scale video analytics. The careful design of the Hadoop based system plays a key role in the systems applicability in such domains.

Let us assume that there is a case for law enforcement and there is a need for a crime investigation analysis based on a collected set of CCTV video footage, within a specified (often limited) period of time. For such a task to be deployed having access to a computing platform with scalable computing resources is a must as one cannot pre-determine computational needs. Certainly a single processor, handling tasks sequentially, is only going to provide an ineffective solution. The solution lie in an environment that is reconfigurable and where the computing resources are elastic/scalable. In this chapter we have shown that a Hadoop based virtualised cluster of machines provides such a platform. In particular if practically implemented in a cloud or a cloud-like environment with scalable/elastic resource allocation capability, depending on the video processing needs the architecture can be reconfigured to provide an effective processing platform.

In this chapter we used a proof of concept approach and studied the Hadoop and video related parameters that play a significant role in such a deployment. In Chapter-5 and 6 we extend this work with the ultimate goal of proposing a framework that can be used for performance modelling and multi constraint based optimal allocation of resources when video analytics/forensics application are deployed in a cloud-like environment.

Chapter 5

Performance Modelling for Hadoop-Based Video Analytics

This chapter proposes a machine learning based approach to predict the total execution time of a video analytic application when deployed in a Hadoop virtual cluster based on the job execution profile, allocated Hadop configuration (i.e. Reducer slots and tasks), and the size of an input dataset. Experiments are conducted to determine which machine learning algorithms provide the most accurate prediction model for the system. The model is constructed by analysing the most influential input parameters, specifically related to video analytic applications, such as, video file characteristics (resolution, file size, frame rate etc.), cluster resource consumption (number of VMs) and Hadoop configurations values(Reducer slots and tasks).

5.1 Introduction

With the increasing demand on video forensics with regard to the investigation of criminal activities and terrorist attacks, responding to incidents require the analysis of large amounts of video data. From the perspective of law enforcement and investigation, this involves time and resource constraints. From the perspective of the computing environment, processing a large volume of video data requires a scalable solution. Putting the Hadoop-MapReduce framework approach into context, it enables scalable, fault-tolerant, automatically distributed and parallel processing framework across a cluster of machines [162].This reduces the execution time of an application, and hence speeds up the output in terms of the results associated with crime investigation. Hadoop is an open source framework that is popular within the research community alongside other large-scale data processing such as Phonesix[115] and Spark[175].

The Hadoop framework benefits from the virtual cloud infrastructure in scaling out to clusters of different VM numbers in order to meet the demand for video analytics. Cloud computing providers such as Amazon Elastic MapRedce (EMR) [2] offers Hadoop on the top of their infrastructure. Recently the private Cloud Openstack has integrated Hadoop as one of its components with the title Sahara[33]. This provision of Cloud-based Hadoop as a service is a cost-effective solution that relieves end users: companies and industries from having to establish dedicated Hadoop clusters, and being involved in maintenance and upgrading, which requires a great deal of capital expenditure.

The performance of Hadoop MapReduce depends on the type of application running and on the performance of the underlying hardware[162]. Video forensics is a computationally intensive application that operates on a frame-by-frame basis for a given video file to extract information from its content. The complexity of the operation and the amount of resources used depends on the algorithm type, video data, size, resolution, frame rate and intensity. In addition, video data takes many forms, such as video sequences from a single camera or views from multiple cameras. Both scenarios produce large video files that it will be impossible to process by a single machine with limited resources, or it may degrade performance in the Hadoop-based cluster when adding additional VM due to hypervisor overheads [171][45][94]and to VMs being under-utilised, explained in section 4.5,

Furthermore, Cloud workloads are characterised by their own performance profiles, resource requirements and constraints specified in service level agreements (SLA). Therefore, making decisions on the correct resource and job provisioning strategy for a video analytic application workload to meet performance goals, requires analysing its behaviour in a Cloud-based Hadoop environment in terms of resource usage patterns under different job configuration parameters. This performance model provides the ability to predict application performance, and hence can be used for resource management.

While existing research efforts in the multimedia domain have studied VM resource allocation in the cloud [91][144][100], their common focus has been on the dynamic requirements of different types of multimedia tasks in terms of a run time to meet QoS (i.e. delay-sensitive requirements) and cost goals. None have tackled the problem within the Hadoop domain, and the parameters used in our study to build the model have not been studied. Existing solutions with regard to the Hadoop domain have focused on resource allocation [107][160][159][87] and they vary from optimising Hadoop configuration parameters to optimising the number of VMs in cluster. All studies have been related to web server applications or Hadoop benchmarking for evaluation. The previous studies did not consider multimedia applications where resources depends on the media request type[144].

While there have been research efforts investigating the capability of the Hadoop framework in a Cloud environment to scale and speed up video analytic operations [85][59][149][179],the scalability of the application raises a new challenge with regard to resource utilisation and performance. Currently, Cloud users are assigned the task of selecting the amount of resources required for their application[2] without prior knowledge of its resulting performance. However, a wrong decision can cause overprovisioning or underprovisioning of virtual machine resources within the selected cluster (e.g. CPU, memory, network). This leads to application performance degradation affecting the user as well as the Cloud provider.

To address the above issue, we need a model that can predict how much time a video analytic application will take to process in a Hadoop Cloud-like environment with fully utilised resources to provide a QoS at a lower cost. In this chapter we present an experimental study to develop the model by comparing several machine-learning algorithms(ML) implemented with the Waikato Environment for Knowledge Analysis (WEKA) toolkit [39] with the aim of achieve the best predictive accuracy. The algorithms make different speed-accuracy-complexity tradeoffs[106], which will be used as metrics for decision making. The parameters used to build the model are proved to have a significant influence on processing time; such as video data settings (resolution and file size), the cluster CPU consumption and Hadoop configuration values(Reducer slots and tasks).The experimental results show that our model can successfully be applied to estimate the execution time for a face detection and motion detection tasks. While all tested ML classifiers give high prediction accuracy, the M5P and bagging algorithms proves to be the most accurate.we evalutated the accuracy of prediction models using various video file content.

The remainder of the chapter is organised as follows. Section 5.2 provides details of the experimental procedure followed. Section 5.3 provides experimental results and a detailed analysis. Section 5.4 discusses the implementation challenges. Finally Section 5.5 concludes the chapter.

5.2 Methodology

To meet the aim of this research, the proposed approach adopts Machine Learning (ML) techniques for the prediction using open source software WEKA which supports a large number of options for data pre-processing and modelling. The reason for selecting a Machine Learning approach was motivated by way the Hadoop-MapReduce functions. All distributed application tasks that belong to the same type of job apply the same computation in terms of data input which is controlled

by job configuration and cluster size. Therefore the resource usage pattern becomes recognisable for particular job application. As the result, the pattern tends to be fairly predictable. In this section we introduce the reader to the specific steps used for modelling that have been adopted within the research context of the proposed framework.

Our prediction process goes through three phases:

- **Phase one:** Study the behaviour of the applications performance in a Hadoop environment and identify features (i.e. attributes) that are related to the input data and the Hadoop configuration that affects performance. These features alongside the job execution times form the training dataset to construct the prediction model.
- **Phase two:** Apply feature selection techniques to minimise the above feature set, removing features that have insignificant impact, thus making the subsequent modelling process less complicated.
- **Phase three:** Train various ML algorithms with the dataset created in last phase to determine the Machine Learning Algorithm that results in the most accurate prediction.

5.2.1 Phase One: Analyse the characteristical behavioural of video analytic application in Cloud-Hadoop environment

The objective of this phase is to identify all features variables (attributes) that are needed for making the prediction decision. Previous research works[102][107] have shown that an optimal MapReduce configuration depends on the resource consumption profile of the job application. Therefore, we extract features from:

- The job Application level (see experiments 1 & 2): describes the Hadoop job configuration parameters (slots and tasks) and their impact on execution time.
- Cluster VM sizing level (see experiment 3): describes the resource allocation and consumption patterns of the application job.

We conducted an intensive analytical study of face detection/motion detection applications running on a virtualized Hadoop cluster to specifically investigate the features mentioned above. After each executing of the face detection and motion detection jobs within Hadoop system, we extract information about job execution

time from Hadoop jobtracker logs, and estimate its CPU resource usage observed from an online monitoring tool. The process is recorded after each execution for each individual virtual machine. Job counter logs gives application execution time during all phases: map, shuffle and reduce phases along with other information. In all experiments we considered various attribute values: video input file size, resolution, fps, Hadoop job configuration parameters such as(map/reducer slot number, map/reducer task number), job completion time, and CPU usage.

Previous work on MapReduce performance models were based on attributes that either focused on job execution time of the fine-granuality phases or on an applications resource consumption. Our proposed model is developed on video application specific parameters and resource consumption, since each has an impact on the total application run time.

The following experiments explain in detail the type of feature variables (attributes) we have been investigated and how they influence the job execution time.

5.2.1.1 Experiment 1

The main objective of this experiment is to observe the impact of reducer slots on CPU resource utilisation, and consequently on overall performance in a given cluster size. We run the face detection and motion detection application with video type1 and type 2 considering different variables: various input dataset sizes, video resolution, cluster size, reducer slots (the maximum number of parallel reducers per node), and reducer tasks. We focus on the resource allocation with regard to the reducer phase. This is because the mapper function in our face detection case(as described in chapter 4) is very simple, and most of the processing is being carried out by reducer.

Figure 5.1 shows the CPU consumption according to our hadoop virtual cluster configuration. We observe that 2 slots utilize 50% of the CPU, 3 slots utilize 75%, and 4 slots utilize 100%. Increasing the slot number to 5 results in same CPU usage as 4 slots. This observation was found when running both face detection and motion detection algorithms on two video types, which means that output is obtained depending on CPU resource usage regardless of input load and the application used. Moreover, number of slot number affects job execution time since increasing slots provide extra rooms for parallel processing. From figures 5.2 & 5.3 we observed a decrease in execution time when we increased the number of slots up to 4.

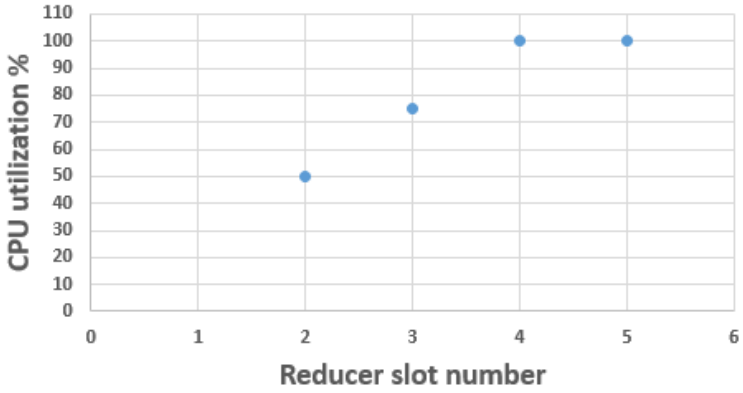


Figure 5.1: Impact of Reducer slots on CPU resource utilization

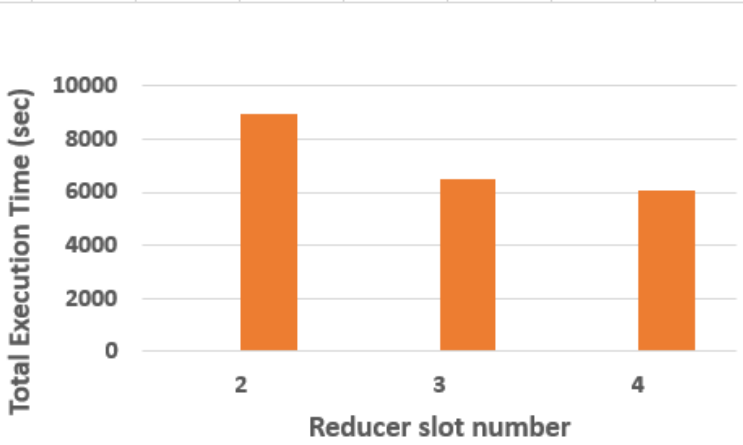


Figure 5.2: Impact of Reducer slots on the face detection job execution time

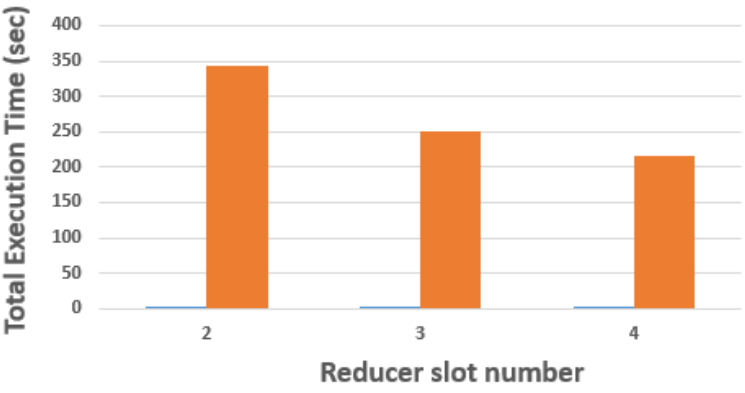


Figure 5.3: Impact of Reducer slots on the motion detection job execution time

5.2.1.2 Experiment 2

The objective of this experiment is to observe the impact of the Hadoop configuration parameter, the number of tasks on the job execution time. By running the experiments, we observed a job application that executes tasks in a single wave (round) is faster than if it is in multiple waves. For instance, if we configure three slots for a single VM, the total tasks to be processed in that VM should be equal to the VMs slot number(this is 3), which was found to fully utilise the CPU resource, and required less execution time. On the one hand, if less than three tasks are processed by a VM, this leads to CPU resource under-utilisation which wasting computing resources and energy[124]. On the other hand, when more than three tasks were assigned, this means that the extra tasks will be processed in another wave. This will need more time for processing since each wave will consume the same amount of time that was spent on the first wave. These observations were also seen in motion detection experiments.

$$N_T = N_{vm} * N_s \tag{5.1}$$

where, N_T is the number of tasks, N_{vm} is the number of nodes and N_s is the number of slots. Applying the eq.(5.1) to a cluster with certain VM size, the total tasks should be equal to the total number of individual VM slots in the cluster. This is, if cluster has 4 VMs each with 3 slots, then the number of tasks should be 12. Table 5.1 shows the job execution time for face detection algorithm using different slots and tasks. For instance, in the first row assigning 9 tasks leads to a better performance than with regard to tasks 12 and 18 using 3 nodes with 3 slots.

Table 5.1: Total execution time with different Reducer slots and tasks for face detection application.

File size (min)	**No. nodes**	**Reducer slots**	**Reducer tasks**	**Total execution time(sec)**
4	3	3	9	1278
4	3	3	12	1778
4	3	3	18	1440
4	3	4	6	1764
4	3	4	12	1328
4	3	4	18	1542

5.2.1.3 Experiment 3

The objective of this experiment is to observe the impact of different VM hardware configurations (resources) and the Hadoop parameter configurations (slots and tasks) on job execution time. In this section we demonstrate the result of Hadoop face detection application since motion detection application shows similer result. We conducted the experiments using two types of VM hardware configurations. Type1 consists of 2 CPU core and 4 GB RAM. Type2 consists of 4 CPU cores and 8 GB RAM. The application job was processed in both VM types using fixed video file characteristics (four minutes, 720x576 resolution, 25 fps), a fixed number of VMs and various slot numbers. We observed that the job execution time was reduced when the cpu usage reached its full utilisation using 3 slots in VM type 1, and when using 4 slots in VM type 2. As illustrated in Table 5.2, type 2 outperformed type1 due to the fact that face detection is a CPU-intensive application, where more CPU is needed for processing.

Table 5.2: Comparison of the Total Execution Time (TET) with two different VM resource types.

File size(min)	**No. nodes**	**Hadoop Configuration**		**TET for different VM Types**	
		Reducer Slot	**Reducer Tasks**	**Type1**	**Type2**
4	3	2	6	1901	1601
4	3	2	12	1946	1624
4	3	2	18	1953	1828
4	3	3	6	1920	1507
4	3	3	12	2400	1778
4	3	3	18	1920	1440
4	3	4	6	1896	1764
4	3	4	12	1916	1328
4	3	4	18	2324	1542

5.2.1.4 Discussion

The aim of the first phase of the study is to identify the features that contribute to providing optimal computing resources in order to reduce an application execution times, on which a performance prediction model can be constructed. The selected feastures are based on the observation found in both video applications(face detection and motion detection). We found that reducer slots control the resource utilisation with regard to application run time. Its optimal value

depends on the underlying VM hardware configuration. This leads to maximising resource usage and reducing execution time. Therefore, we select a slot number to be the parameter input for the model, and give users the options when it comes to selecting the level of CPU usage required in the cluster. Another parameter that is considered is the reducer task which causes different resource usages during the reducer processing phase. In some scenarios we found VMs to be underutilised or idle, due to queuing tasks that need to be processed in waves. Our approach to balance the task number with the total slots in the cluster resulted in fully utilized resources and less execution time. Therefore, we select the task number to be a parameter in the dataset that influences the models construction.

Other feature variables to be included in the dataset are: video input size, resolution and the number of VMs.

5.2.2 Phase Two: Create Training Dataset

5.2.2.1 Dataset Variables

The observation from last section provides an insight to the important features (parameters) that have a direct influence to the application performance as well as system performance. Table 5.3 shows the features (attributes) used to generate the training dataset. To build up the training dataset many experiments have been conducted with inputs such as: two resoutions[720x576 and 260x288], input file sizes [4, 6, 8 & 20 mins], each tested with reducer slots [2, 3, 4] and reducer tasks[6, 12, 18] and the frame rate fixed at 25 frames per second(fps).

Table 5.3: Attribute used for video analytic application performance modelling.

Attributes	**Values**
Input video size(min)	4,8,12,20
Resolution	360x288 and 720x576
Frame rate(fps)	25
Number of nodes (VMs)	2,3,6,8
mapred.reduce.tasks.maximum	2,3,4
mapred.reduce.tasks	6,12,18
Avg map time	
Avg shuffle time	
Avg reduce time	
Total job execution time	

5.2.2.2 Data Collection

After each job completion, the data collector extracts the information listed in Table 5.3 (execution time for map phase, shuffle phase, reduce phase and the total job completion time) from Hadoop jobtracker log files, and monitor the CPU utilization using the external monitoring tool named Uptime Cloud Monitor. It is worth noting that CPU utilization is bound by slot number as described in section (5.3.1.1).

5.2.2.3 Dataset Representation

Table 5.4 and 5.5 list sample datasets that are resulting from the above experiment.

Table 5.4: Training dataset for face detection application with videp type1.

Fps	**Resolution**	**File size (min)**	**No. nodes**	**Reducer slots**	**Reducer tasks**	**Total execution time(sec)**
25	720x576	8	2	2	6	2845
25	720x576	12	3	3	12	4907
25	720x576	20	4	4	18	7955
25	360x288	8	3	2	12	775
25	360x288	12	3	3	12	1111
25	360x288	20	8	4	18	1119

Table 5.5: Training dataset for motion detection application with videp type1.

Fps	**Resolution**	**File size (min)**	**No. nodes**	**Reducer slots**	**Reducer tasks**	**Total execution time(sec)**
25	720x576	8	2	2	6	772
25	720x576	12	3	3	12	382
25	720x576	20	4	4	18	972
25	360x288	8	3	2	12	300
25	360x288	12	3	3	12	475
25	360x288	20	8	4	18	633

5.2.2.4 Data Preparation (Feature Selection)

To increase the performance prediction accuracy we need to identify the most influential attributes and reduce attributes that do not significantly contribute to the

improved accuracy. According to [164], feature (attribute) selection has many benefits such as: improving the prediction performance of the predictors(classfiers), providing faster and more cost-effective predictors, and providing a better understanding of the underlying process that generated the data. WEKA has automated attribute selection option to determine which set of attributes are the best prediction for the application performance. For this study we selected among many options the wrapper method evaluator "cfsSubsetEval" with search method greedy step wise algorithm as it uses the prediction model to make selection assessment which provides higher accuracy for any tested algorithm. Table 5.5, shows the resulting attributes selected to predict the total execution time.

As shown in Table 5.6, we observe that the attribute frame rate(fps) was excluded, as expected, as in this selected case it is constant in all scenarios. In addition to this the attribute, shuffle phase execution time, was dropped as the cluster virtual machines are hosted in one single physical machine, and therefore, the process of coping data from the mapper to the reducer(data transfer activity) is internal, within same VM disk or between VMs which has no significat influance on the total execution time.

Table 5.6: Attributes for video analytic applications performance model.

Dataset Attributes	**Feature Selection**
Input video size(min)	✓
Resolution	✓
Frame rate(fps)	✗
Number of nodes (VMs)	✓
mapred.reduce.tasks.maximum	✓
mapred.reduce.tasks	✓
Avg map time	✓
Avg shuffle time	✗
Avg reduce time	✓
Total job execution time	✓

5.2.3 Phase Three: Modelling the Job Execution Time

This research adapted a WEKA tool (v3.6.13) for the implementation of Machine Learning classifiers; REPTree [164], Multi-Layer Perceptron(MLP) [164], M5P [131], Linear Regression(LR) [143] and the Ensemble Learning algorithm Bagging combined with M5P and LR [52] as the base classifier. The purpose is to find the best classiffer in order to predict the job execution time of the Hadoop-based video analytic application. Each learning algorithm was trained and tested using the same input dataset making use of ten-fold cross validation [81].

5.3 Experimental Results & Analysis

This section describes the analysis of the results obtained from the experiments to compare performance of various learning algorithms implemented within WEKA (using their default parameter settings) to model the system performance. We used the training datasets described in section 5.2.2.3 to train the eight algorithms in order to predict the total job execution time under different scenarios. In our experiment, the accuracy on the sample dataset has been obtained using 10-fold cross validation, which is helpful to prevent overfitting. The following sections present analysis of prediction models built by using different training datasets obtained from running both face detection and motion delection algorithms with two types of video files.

5.3.1 Prediction Experiment Result 1

5.3.1.1 Training Datasets

A total of 346 scenarios (instances) were recorded for each video type from various experiments presented in Microsoft excel .csv format. The dataset after feature selection technique consists of parameters which include, video resolution, input file size, number of nodes, number of slot slots, number of reducer tasks, map/reduce phases completion time and total job completion time.

5.3.1.2 Prediction Models

In this section we present various prediction models built by using the training datasets explained in section 5.3.11. For Predictive accuracy comparison, Table 5.7 and 5.8 tabulate the prediction accuracies obtained by each classifier with selected attributes, presented in terms of the time it takes to build the model, the correlation coefficient and the relative absolute error. From the table 5.7 we found interesting observations: First, the prediction accuracy from the classifiers RepTree, M5P, MLP and LR are high. Second, when we compare between the classfiers, we noted the ensemble classifier Bagging has a marginally increased accuracy when to the standard single classifiers, REPTRee, M5P , MLP and LR being used.

From table 5.8 we found that motion detection (overlapped frame application) producd less prediction accuracy comparing to face detection (a single frame oriented application). This is because number of overlapped frame that are required for procssing varies depending on the algorithm requirment. However, the results still show good prediction model with a correlation coefficient over 0.8.

We observed that with motion detection, the ensemble classifier Bagging has also increased accuracy when compared to the standard single classifiers, REPTRee, M5P , MLP and LR. The single classifier M5P model tree show better performance than LR, MLP, and REPTree.

Table 5.7: Results of the prediction models for face detection application with video type1.

Classifier	**Correlation Coefficient**	**Relative Absolute Error**	**Time Building Model(s)**
LR	0.991	9.6%	0.0
MLP	0.984	16.43%	0.29
M5P	0.993	7.9%	0.22
REPTree	0.972	15.4 %	0.04
Bagged LR	0.992	9.5%	0.02
Bagged MLP	0.993	7.9%	2.02
Bagged M5P	0.993	7.3%	0.39
Bagged REPTree	0.982	11.14%	0.03

Table 5.8: Results of the prediction models for motion detection application with video type1.

Classifier	**Correlation Coefficient**	**Relative Absolute Error**	**Time Building Model(s)**
LR	0.866	34%	0.03
MLP	0.831	53%	0.36
M5P	0.889	30%	0.1
REPTree	0.857	35%	0.0
Bagged LR	0.867	47%	0.0
Bagged MLP	0.833	45%	2.51
Bagged M5P	0.900	46%	0.67
Bagged REPTree	0.872	48%	0.0

To help visually compare the classification results, Figure 5.3 presents scatter plots of the predicted vs the actual execution times for each trained model. The figure illustrates the comparison of error spread between bagging and single classifiers. It shows an improved prediction capability of bagging with low error spread as compared to the others. While REPTree, LR and MLP have more spread of scatter points, indicating lower prediction accuracy when used as a single classifier. The single classifier with the best performance was M5P model.

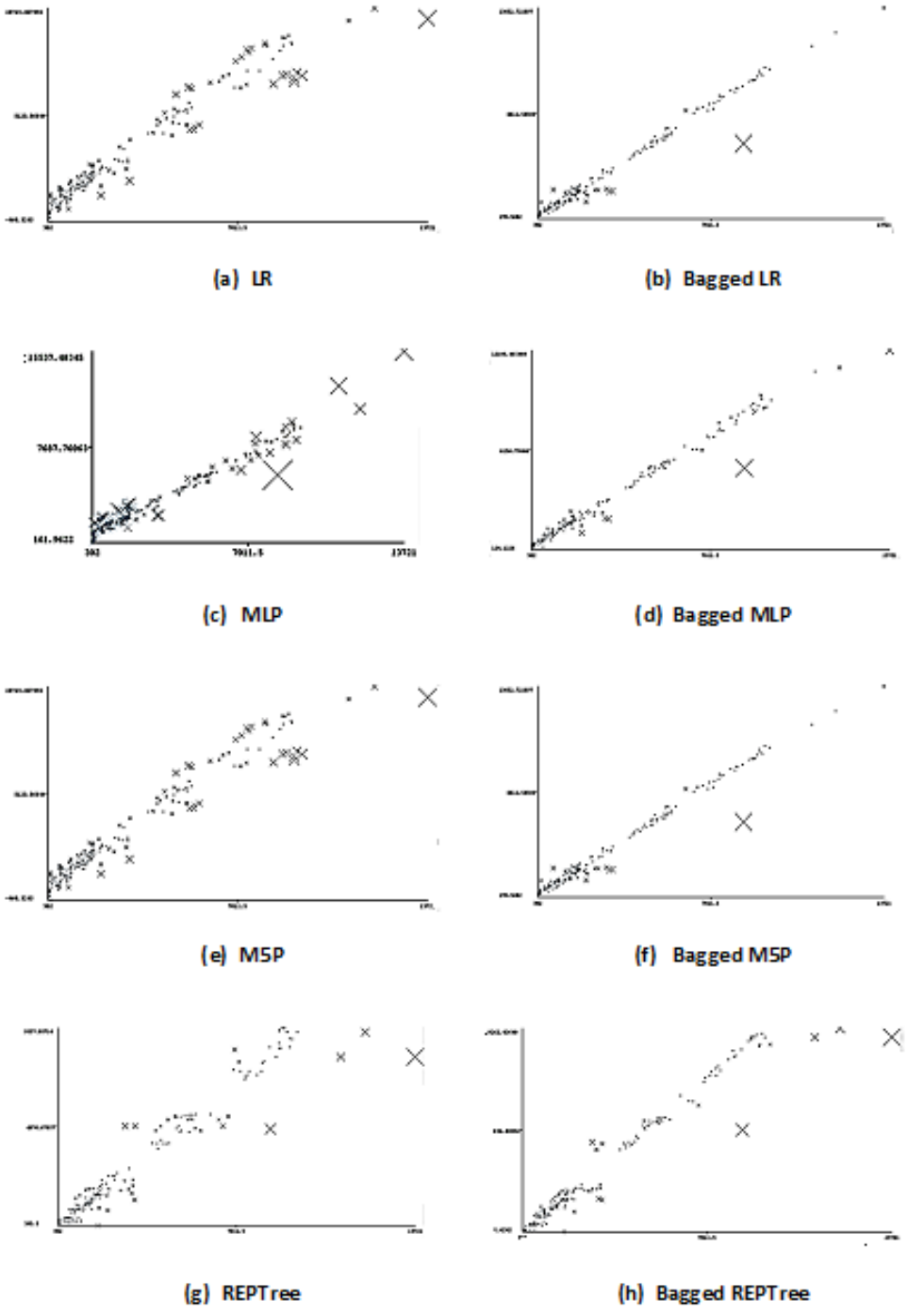


Figure 5.4: Comparing the predicted vs actual execution time for different classifers

5.3.2 Prediction Experiment Result 2

We discussed in the last section the process of predicting the total execution time of a video analytic applications by training ML classffers using dataset with predefined attributes. In view of the fact that some of the attributes for example, map, shuffle and reduce completion times, are not available before running an application, we found the models to be impractical. Thus the attribute training sets should be revised to be able to provide realistic inputs for prediction. As a result we removed all the phase completion times as attributes and kept only the system and video characteristic metrics as input parameters to the model. In this section we present prediction models created by four different training data sets that are obatined by running face detection and motion detection application using two different types of video file.

5.3.2.1 Training Datasets

As shown in Table 5.9, the new training datasets consist of six attributes: video resolution, input file size, number of nodes, number of reducer slots, number of reducer tasks and total job execution time.

Table 5.9: Updated training dataset.

Attributes
Input video size(min)
Resolution
Number of nodes (VMs)
mapred.reduce.tasks.maximum (slots)
mapred.reduce.tasks
Total job execution time

5.3.2.2 Prediction Models

Modelling was conducted and prediction results were obtained following a procedure similar to that described by section 5.3.1.2. Tables 5.10 and 5.11 tabulate the prediction accuracies obtained by each classiffer with the new selected attributes, presented in terms of the time it takes to build the model, the correlation coefficient and the relative absolute error.

Table 5.10 present predicition models for face detection application using video type1. An interesting observation found in face detection prediction model is that most of the classffiers are still performing at a high accuracy level as compared to the accuracies obtained in the previous experiment. We observe Linear Regression

Table 5.10: Results of the prediction models for face detection application with video type1.

Classifier	**Correlation Coefficient**	**Relative Absolute Error**	**Time Building Model(s)**
LR	0.854	52.6137%	0.0
MLP	0.9435	30.7926%	0.13
M5P	0.9638	21.8014%	0.07
REPTree	0.9628	19.9094%	0
Bagged LR	0.8537	52.6068%	0.01
Bagged MLP	0.9661	20.958%	1.11
Bagged M5P	0.9676	20.6712%	0.32
Bagged REPTree	0.9709	17.3204%	0.02

Table 5.11: Results of the prediction models for motion detection with video type1.

Classifier	**Correlation Coefficient**	**Relative Absolute Error**	**Time Building Model(s)**
LR	0.877	34.11%	0.02
MLP	0.845	49.81%	0.14
M5P	0.875	34.04%	0.05
REPTree	0.864	35.90%	0.0
Bagged LR	0.878	34.13%	0.02
Bagged MLP	0.884	35.91%	1.39
Bagged M5P	0.886	33.14%	0.28
Bagged REPTree	0.882	31.44%	0.02

model has slightly reduced performance, seeing that the correlation coeffcient has dropped to 0.85. To justify this, in the previous experiment the variable to be predicted (total execution time) depends linearly on the inputs: map/shuffle/reduce phases completion times, which explains why including these attributes presented more accurate modelling. For the new LR model, the dataset attributes include number of nodes, reducer slots and reducer tasks; which has nonlinear effects depending on the system behaviour, resource consumption and workload. For example, when increasing the number of node and keeping the reducer slot constant, the total execution time may not reduce, it depends on the processed load and accordingly on resource consumption. However the correlation coefficient of 0.854 is still considered to be sufficiently accurate for practical applications.

As illustrated in the Table 5.10, REPTree and M5P tree models are more efficient in predicting non-linear behaviour because they involve modelling based on tree structured algorithms. We observe the results using M5P is much better because prediction errors are consistently lower than those obtained by RepTree or Linear Regression.

Once again, the Ensemble method bagging outperforms the standard single classiffiers and proved its ability to improve prediction power of its base classifiers, as visually illustrated in figure 5.5. The reason for this is that it resamples the original training dataset and develops a prediction model from each sample using a single learning algorithm (e.g. LR, MLP,M5P or REPTree). Then it combines the output of these multiple prediction models by averaging the output or by voting, in order to increase performance over a single model.

Table 5.11 presents prediciton results for motion detection application using video type1 dataset. We found slight differences in accuracy comparing to the results illustrated in Table 5.8. However, the results in terms of best classifier shows that Ensemble bagging outperforms the standard single classiffiers which is similar to our conclusion on face detection.

To confirm the effectiveness of our prediction method we have trained the ML classifers with different video datasets for both face detection and motion detection applications. The results show exactly the same prediction accuracies as Tables 5.10 and 5.11. This is not superising since the new video datasets were obtained from running the same applications, which means the same procedure was applied to a frame or group of frames. However, because the job execution times have definitly changed in the dataset, this has ultimately resulted in some changes to the models coefficients to fit the new data. For illustration see the following regression model equations, it is clear that the coefficients differ in both equations.

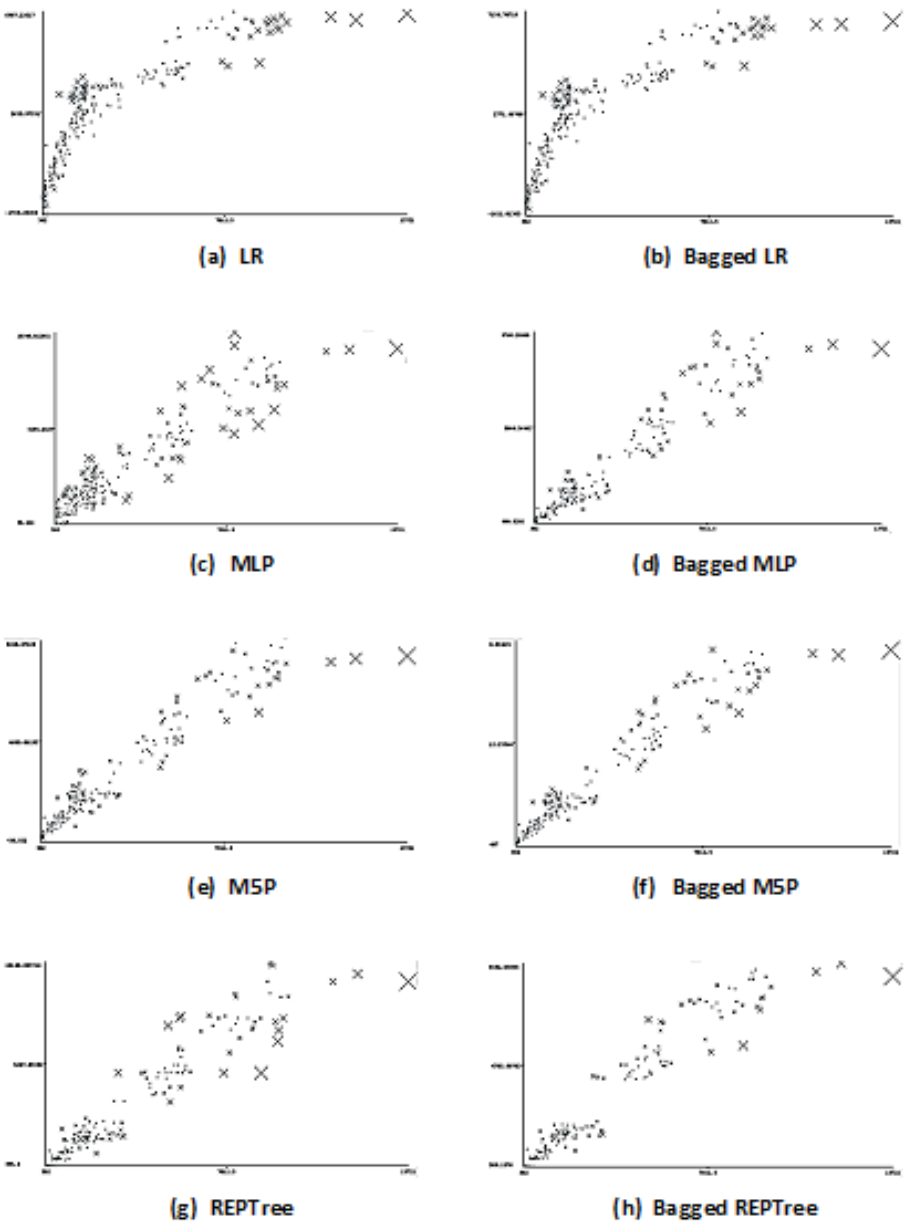


Figure 5.5: Comparing the predicted vs actual execution time for different classifiers

- Linear Regression model $f(x)$ for face detection using video type1 is given as:

$$f(x) = 4.2907 * x_1 + 5.3205 * x_2 + 260.58 * x_3 - 95.9374 * x_4 - 256.625 * x_5 - 27.0816 * x_6 - 3379.6942 \quad (5.2)$$

- Linear Regression model $f(x)$ for face detection using video type2 is given as:

$$f(x) = 2.9794 * x_1 + 3.7243 * x_2 + 182.406 * x_3 - 67.1562 * x_4 - 179.6375 * x_5 - 18.9571 * x_6 - 2365.7859 \quad (5.3)$$

5.4 Implementation Challenge & Discussion

Throughout the design and implementation phases of this project a number of practical challenges were met and successfully resolved. As these challenges may be important in a large scale deployment of the proposed system are presented as follows:

- In a real application to process larger video streams coming from different sources one will require a large-scale cluster environment that consists of many VMs that resides in different physical machines. In this case transferring data from the master to the slaves and from mappers to the reducers will have an impact to the network traffic and bandwidth. In our small scale cluster, all VMs reside on the same physical machine. The communication within VMs was performed inside that machine. This means that there is no network to monitor and analyse. Thus further tests are needed on network performance related aspects to confirm further application upscaling.

- Video input format was and is still an issue in processing Hadoop-based video analytic applications. For example: when processing one video file as a whole file it means only one mapper will process the file. This solution helps prevent splitting a video file when using Hadoop content-unaware splitting. Nevertheless, it has drawbacks in terms of resource limitation on one machine, since in some of our experiments we received error messages about java heap and memory limitation. In addition, reading a file as one

whole file does not mean that the file is read from one source but it actually had been split and distributed among the cluster VMs, then when the application job runs, all the data segments are fetched as a whole file to map the function to process. In terms of network related issues this causes network traffic and also increases processing time.

5.5 Conclusion

This chapter discussed performance modelling which can predict application execution times by given a resource allocation (e.g. a specified VM number) in a Hadoop based virtual environment. The prediction model was developed by comparing the performance of eight Machine Learning algorithms; M5P, REPTree, LR, MLP and Ensemble Learning algorithm Bagging with the same single classifiers, in terms of their predictive accuracy. We found out that all created models from our four training datasets, gave high prediction accuracies. The four training datasets represent data collected from running face deection as well as motion detection on two video types. In a first phase of experiments we included as predictor parameters the Hadoop phases completion times (map, shuffe and reducer). In a second phase, we removed the parameters and trained models with only video characteristics attributes (i.e. input size, fps and resolution), Hadoop configuration parameters (i.e. reducer slots, reducer tasks) and cluster size (i.e. num of nodes) that are available at the time of execution. Though the latter has lower accuracy than the former, it still provides accuracy with correlation coefficient above 0.80 .It is noted that the predictors for these models are available inputs that can be provided before any application runs. Whereas in the former scenario the Hadoop phases completion times are not known prior, therefore it is impractical to predict unless using online prediction methods similar to[107] [100].

Chapter 6

Performance Optimisation for Hadoop-Based Video Analytics under Constraint Conditions

In cloud-based Hadoop environments optimizing resource provisioning under contrainted conditions to minimise a job execution time has been a challenge. This chapter addresses this constrained problem by introducing genetic algorithm based optimization technique that makes use of one of the application performance model generated in Chapter-5 based on the Linear Regression approach. The optimization algorithm searches for the optimal resourc parameter settings(i.e VM number, slots number and task number) of the model to obtain minimum job execution time. This work closes an existing research gap in distributed processing of video analytic data and together with the content presented in Chapters 4 and 5, forms a framework that can be practically used in the performance optimisation of video analytic applications, when executed in a cloud-like environment.

6.1 Introduction

Cloud-based Hadoop environments are an active area of research focussing upon data processing and providing on-demand computing resources and storage that are appropriate for the needs of the user.

A cloud-computing platform offers mechanisms to automatically scale VM capacity, which makes the deployment of an on-demand Hadoop cluster a preferable choice for most users for scalability in building cluster sizes that are appropriate for a given task [58]. In particular the new Hadoop-YARN architecture includes resource management features, which manage resources across a cluster of machines, subject to the constraints of capacities of applications deployed. Features

such as this allows users to specify the cluster size, as well as the available resources for each machine in the cluster, together with assistance in redistributing the total resources available into containers that are appropriate for running a specific application. Further the allocation of resources can be automated by the cluster[36].

This dynamic allocation of resources enhances the utilisation of cluster resources, as well as providing enhanced performance, which removes users from being unduly concerned about capacity planning. However when a Hadoop application performance model is unknown prior to usage, it is a challenge to understand their resource capacity requirements in both Hadoop1 and Hadoop-YARN, which require careful consideration by new users due to implications on resource hiring costs.

A framework to model the performance of video analytic application, in terms of predicting the execution time considering the relationship between application specific characteristics, system performance parameters and available system resources, was discussed in a previous chapter. Making use of this model it will be useful to be able to determine the minimal execution time (i.e. the optimal value) of a video analytic application, when deployed within a Hadoop based virtual cluster under given multiple practical constraints (e.g. number of VMs, number of slots in a VM etc.). Although Hadoop can be adapted to the requirements of an application [162], the user is still required to specify the number of virtual machines for a cluster in order for the system to allocate the available resources. A user may be keen to know how the Hadoop based system could be most economically used to get the video analytic job completed within a time limit. Usually such a judgement is made based on users previous experience that could be subjective. Hence a sceintific and objective approach to optimal allocation of resources will be forthcoming. Thus the development of a multi-constraint optimisation framework to achieve this task will be a useful contribution to the present state-of-art. Several researchers have addressed the above problem[107][58][104] through the deployment of heuristic search techniques. However, these fail to guarantee optimum solutions, as well as being the only resource allocation considered that are based upon the Hadoop performance model.

In this chapter, resource constrained processing of video analytic data in a hadoop based distributed VM cluster is discussed and a method for resource allocation for optimised performance under multiple constraints using genetic algorithms (GAs) is investigated. In particular optimal CPU utilisation, video related characteristics/parameters and job execution time targets for a given video analytic application is discussed. Out of a number of possible alternative approaches a genetic algorithm based method was selected to support the research proposed due

to its multi-point search capability and robustness in global optimal value that guarantees the aim of minimising application execution time.

For clarity of presentation this chapter is organized as follows. In section 6.2, we define and formulate the minimization problem for optimization. In section 6.3 we provide a brief introduction to Genetic Algorithms, Pattern Search and Lagrange Multipliers technqiues. In section 6.4 we describe the details of the experimental procedure followed, provide experimental results and a detailed analysis and compare the performance of the algorithms. Finally section 6.5 concludes the chapter.

6.2 Problem Formulation

The decision problem considered in this chapter is a resource constrained problem, in which the objective is to minimize the total job execution time t of the video analytic application running in a defined Hadoop cluster. We aim to find the optimal execution time (e.g minimum execution time) under given multiple constraints. This problem is formulated as a single objective, multiple constraint, and optimization problem based on the generated Linear Regression model derived in Chapter 5. The aim is to minimize this time based on given constraints of computing resources and Hadoop based parameters.

Following the above mentioned general description of the problem formulation, the specific optimisation problem considered within the proposed framework can be described as follows:

The total execution time $f(x)$ is defined as

$$f(x) = \sum_{i=1}^{6} \theta_i x_i + \varepsilon \tag{6.1}$$

Where
$f(x)$= objective function (fitness)
θ_i = model parameters
x_i = model variables
ε = error term epsilon

For our Linear Regression model (based on face detection application tested with video type1)

$$f(x) = t = 4.2907 * x_1 + 5.3205 * x_2 + 260.58 * x_3 - 95.9374 * x_4 - 256.625 * x_5 - 27.0816 * x_6 - 3379.6942 \quad (6.2)$$

Therefore the problem formulation could be written as:

$$minimize f(x) \quad (6.3)$$

subject to

$$t \leq time_d \quad (6.4)$$

$$x_1 = framewidth, fixed, \quad (6.5)$$

$$x_2 = framehight, fixed. \quad (6.6)$$

$$x_3 = filesize, fixed, \quad (6.7)$$

$$lb_4 \leq x_4 \leq ub_4 \quad (6.8)$$

$$lb_5 \leq x_5 \leq ub_5 \quad (6.9)$$

$$lb_6 \leq x_6 \leq ub_6 \quad (6.10)$$

Where
t= total job execution time
$time_d$ = job execution time deadline
x_4 = number of virtual machines
x_5 = number of reducer slots
x_6 = number of reducer tasks
lb_4 & ub_4 = lower/upper bound of x_4
lb_5 & ub_5 = lower/upper bound of x_5
lb_6 & ub_6 = lower/upper bound of x_6

Similar procedures were applied to develop the following mathematical models to solve the optimization problem for face detection and motion detection with different video types:

- Linear Regression model for face detection with video type2.

$$f(x) = 2.9794 * x_1 + 3.7243 * x_2 + 182.406 * x_3 - 67.1562 * x_4 - 179.6375 * x_5 - 18.9571 * x_6 - 2365.7859 \quad (6.11)$$

- Linear Regression model for motion detection with video type1.

$$f(x) = 0.5959 * x_1 + 0.7449 * x_2 + 36.4812 * x_3 - -13.4312 * x_4 - 35.9275 * x_5 - 3.7914 * x_6 - 473.1572 \quad (6.12)$$

- Linear Regression model for motion detection with video type2.

$$f(x) = 0.4767 * x_1 + 0.5959 * x_2 + 29.185 * x_3 - 13.4312 * x_4 - 10.745 * x_5 - 3.0331 * x_6 - 378.5257 \quad (6.13)$$

The following section shows how a genetic algorithm is used to provide a solution for the above problem allocation and compare results with other optimization techniques.

6.3 Methodology

In this section we introduce Genetic Algorithm (GA) as a solution to our optimization problem. The result of GA is compared to other techniques used in literature: Pattern Search(PS) and Lagrange Multipliers(ML) optimisation techniques.

6.3.1 Algorithms Description

Pattern search is a numerical optimization method known as direct search. It begins with a point that satisfis the bounds throughout the search. It generates a sequence of iterations x_k. Given the current iterate x_k at each iteration k, the next point x_{k+1} is chosen from a finite number of candidats on a given mesh M_k (i.e . set of points). At each iteration the algorithm looks for a point in the mesh that minimise the objective function: $f(x_{k+1}) < f(x_k)$. This step is called search step. After that in poll step if the search step was unsuccessful, evaluate f at

points in the poll set P_k until an improved mesh point x_{k+1} is found. For further details on PS technique, the readers are referred to [156] .

Lagrange Multiplier is a technqiue to solve constrainted optimization problem to find maxima or minima of objective function $f(x)$ subject to a constraints by considering certain points in a surface. For further details on theory of ML technique, the readers are referred to [49] .

Genetic algorithms (GAs)[76][83] may be described as a global search optimisation technique based upon the principles of natural selection and evolution. John Holland and colleagues invented the technique during the 1960s-1970s[76]. From that time, GAs have demonstrated their usefulness in a number of different problems found in science, business and engineering applications.

The search process is carried out by GA in four stages: initialisation, selection, crossover and mutation. The initial population of chromosomes is defined by an algorithm, called individuals, with a variety of possible solutions with various genes structures, which are distributed randomly in the search space as the search starting position. The chromosomes are then calculated and evaluated by using a user-defined function, which is designed to numerically encode the performance of the chromosome.

GA is based on the idea of survival of the fittest, where the reproduction happens in such a way that only the highest performing chromosomes are selected from the initial population, and allowed to survive and breed their characteristics for coming generations, thus assisting in the search for the ideal solution; chromosomes that are poorly performing are discarded. At the crossover stage, two randomly selected chromosomes, exchange corresponding segments of a string representation of the parents, looking for a new solution in far-reaching directions. There are many different types of crossover: the one-point, the two-point, constrained and the uniform.

The mutation occurs when a member of the population(i.e chromosome) is randomly selected and one randomly selected bit in its string of bits is altered, which is a GA function. The reproduction and the crossover process produce many new strings, yet no new information is introduced at bit level into the population. If the mutant member is feasible, it replaces the member that mutated in the population. This mutation occurs with some probability, called the mutation rate running the algorithm for more generations. Mutation presence exists to ensure that the probability of reaching any point in the search space is never zero.

This process of natural selection occurs in all stages of the algorithm, which allows the population of chromosomes to evolve. The algorithm does not require cost function derivatives as with conventional analytic optimisation that deals with non-continuous cost functions and discrete variables. GAs are computation-

ally simple yet powerful in their search for improvement. In addition, genetic algorithms are evolutionary computing algorithms that distinguishes from other search and optimisation techniques, because they are processes that use the population of many individuals rather than a single individual to solve a problem.

6.3.2 Fit GAs to Resource Allocation Problem

The aim of the chapter is to solve the above minimization problem to determine optimal solutions thus making it possible for the optimal resource allocation in Hadoop cluster to the meet user's performance requirement in terms of minimizing the application execution time. Once the fitness function is defined, we began the process of fitting it to a GA by randomly generating the initial population by encoding the Linear Regression model data into set of chromosomes (possible solutions) named, individuals. Each gene in the chromosome represent a feature (variable) of the model that contributes to the prediction of the job execution time t . This is a multiple-dimensional optimization problem where each chromosome has six variables $(x_1, x_2, x_3, x_4, x_5, x_6)$ and it is written as an array of N_6 elements so that the:

$$chromosome = [x_1, x_2, x_3, x_4, x_5, x_6] \tag{6.14}$$

The set of chromosomes is called a population where each chromosome represnts a different solution containing optimal resources with a possibly minimized t.

In a population each chromosome's (individual) fitness value f is calculated by running all the data points (x_i) in the training dataset. The fitness fuction $f(x)$ is formulated as a single objective $f(x)$ as follows:

$$F(x) = f(chromosome) = f(x_1, x_2, x_3, x_4, x_5, x_6) \tag{6.15}$$

In our problem the fitness function equales to the total execution time, and the objective would be to minimise it subject to constraints of each variable, i.e. to optimise $f(x)$ under multiple constraints.

We limit the exploring a reasonable region of variables space by imposing a constraints and bounds using eq. (6.4 - 6.10). Deciding which chromosomes in the population are to survive and represent the offspring in the next generation, a fititness value for each chromosome is ranked from lower to highest cost, and the rest are discarded. The individuals with the highest-fitness (lower execution time) are selected to be parents for the next generation by applying the operators rank selection, crossover and mutation (with constrained option settings) to the current population. This process is repeated until a global optimization approaching

towards the ideal point (minimized t) is found, since none of the initial randomly selected variable values are particularly close to the global minimum.

6.4 Experimental Results and Analysis

This section, presents the results and an analysis of the experiment that was performed to illustrate the effectiveness of using GAs in finding the minimal execution time given multiple operational constraints. Two test cases are presented. The first test optimizes face detection and motion application with video type1. The second test optimizes the applications with video type2.

6.4.1 Face Detection Application

6.4.1.1 Test1

Suppose that the users requirements for running face detection application on a Hadoop based setup is listed in Table 6.1, and it is required to find the parameters that minimizes the execution time under constrained conditions. The experimental results obtained from applying GA to the eq.(6.2) are illustrated in Table 6.2. The optimum (i.e. the minimum) execution time is obtained as 2248 seconds (despite the given target of 2000 seconds) when the number of nodes, slots and tasks are set at 8, 4 and 32 respectively. It can be clearly seen that the algorithm can effectively assign the computing resources (i.e. number of nodes, slots & tasks) to obtain the optimal point of operation, i.e. minimising the execution time, whilst still satisfying the given constraints.

Table 6.1: User input requirements and the system constraints

Variables	**Parameter**
User input:	
Video Resolution (x_1),(x_2)	720x576
Video file size (x_3)	480 sec
Execution deadline ($time_d$)	2000 sec
Constraints:	
Number of nodes (x_4)	$2 \leq x_4 \leq 8$
Number of slots (x_5)	$2 \leq x_5 \leq 4$
Number of tasks (x_6)	$4 \leq x_6 \leq 32$

Figure 6.1 illustrates the evolution curves in searching for the optimal value of the execution time, i.e. minimal value of $f(x)$, that results from specific values for $x_4; x_5; x_6$, that were presented in Table 6.2. The function reaches its optimal value

Table 6.2: Results generated by the GAs operation

Variables	**GAs Solution**
Number of nodes (x_4)	8
Number of slots (x_5)	4
Number of tasks (x_6)	32
Fitness value $f(x)$	2248

with the increase of the generation number. The searching procedures is stopped when the function continuous to have the same minimal value with an increasing generation number. It is the global minimum that one should focus on achieving given that there are two other stable states of the graph.

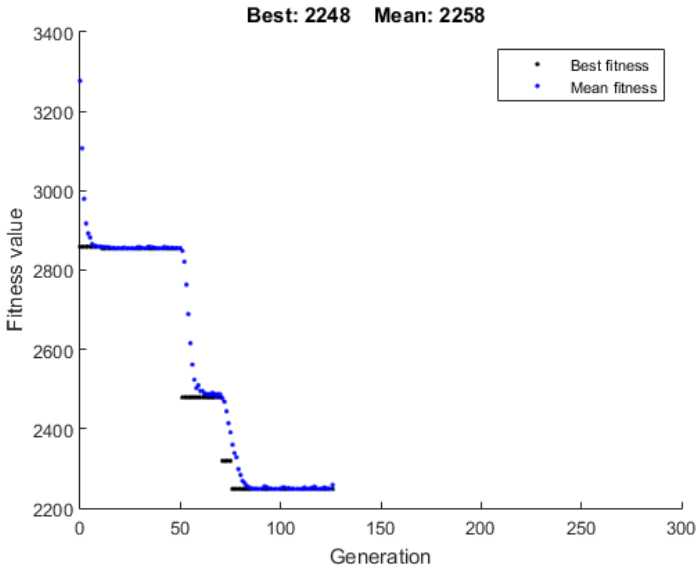


Figure 6.1: Evolution curves searching for best fitness

We evaluated the results generated from GAs operation further by substituting the values of $x_4; x_5; x_6$, that results in the optimal performance to the fitness function of our model to obtain the optimal reading for the execution time. Then the parameters were changed by slightly increasing and decreasing each parameter separately whilst keeping the other parameters fixed at the values that created the optimal execution time. The results are tabulated in Table 6.3 which demonstrates that what we have obtained is the optimal execution time under the given constraints. In Table 6.3 it is observed that some value of combinated parameters give close results, for example when we used 8 nodes/4slots with 32 tasks and 31 tasks, and also 7 nodes/4slots with 32 tasks. These marginal changes are critical when we consider the trade-off between cost and time where more resource results in more cost, but will minimise execution time.

Table 6.3: Analysis of results generated by the GAs operation

	x_4	x_5	x_6	$minf(x)$
GA optimal values	8	4	32	2248
Change x_6 value	8	4	31	2373
Change x_5 value	8	3	32	2603
Change $x_4 value$	7	4	32	2442

6.4.1.2 Test2

We used eq.(6.11) as objective function(fitness) for GA to optimize resources for face detection application running a different video file. We assume user input requirements and constraints are listed in Table 6.4. The experimental results gives the fitness value 260 seconds as a minimum job execution time when selecting the parameters nodes, slots and tasks as 3, 2, 6 respectively, shown in Table 6.5.

Table 6.6 shows that the obtained results are the optimal values when increasing and decreasing the parameters $x_4; x_5; x_6$ separately whilst keeping the other parameters fixed at the values that created the optimal execution time.

Table 6.4: User input requirements and the system constraints

Variables	**Parameter**
User input:	
Video Resolution (x_1),(x_2)	360x288
Video file size (x_3)	240 sec
Constraints:	
Number of nodes (x_4)	$3 \leq x_4 \leq 5$
Number of slots (x_5)	$2 \leq x_5 \leq 4$
Number of tasks (x_6)	$4 \leq x_6 \leq 20$

Table 6.5: Results generated by the GAs operation

Variables	**GAs Solution**
Number of nodes (x_4)	3
Number of slots (x_5)	2
Number of tasks (x_6)	6
Fitness value $f(x)$	260

Table 6.6: Analysis of results generated by the GAs operation

	x_4	x_5	x_6	$minf(x)$
GA optimal values	3	2	6	260
Change x_6 value	4	2	6	319
Change x_5 value	3	3	6	280
Change $x_4 value$	3	2	8	273

6.4.2 Motion Detection Application

6.4.2.1 Test1

We used eq.(6.12) as objective function and applied the data listed in Table 6.7 to find the parameter values of $x_4; x_5; x_6$. Table 6.8 shows the optimization results. The optimum execution time is obtained as 168 seconds when the number of nodes, slots and tasks are set at 4, 2, 12 respectively. GA minimised the execution time whilst still satisfying the given constraints. Table 6.9 analyses the results generated by GA.

Table 6.7: User input requirements and the system constraints

Variables	**Parameter**
User input:	
Video Resolution (x_1),(x_2)	720x576
Video file size (x_3)	480 sec
Execution deadline ($time_d$)	150 sec
Constraints:	
Number of nodes (x_4)	$2 \leq x_4 \leq 8$
Number of slots (x_5)	$2 \leq x_5 \leq 4$
Number of tasks (x_6)	$4 \leq x_6 \leq 32$

Table 6.8: Results generated by the GAs operation for motion detection

Variables	**GAs Solution**
Number of nodes (x_4)	4
Number of slots (x_5)	2
Number of tasks (x_6)	12
Fitness value $f(x)$	168

The simulation results when searching for the minimal value is shown in Figure 6.2. We notice the population remain stable when it reached the value 168, which is considered to be the global optimization point.

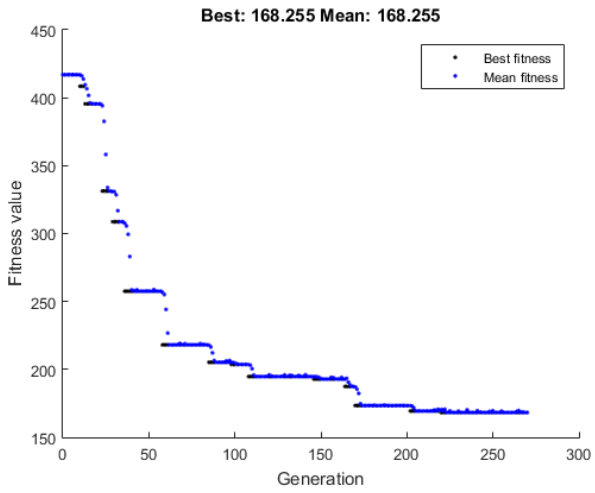


Figure 6.2: Evolution curves searching for best fitness

Table 6.9: Analysis of results generated by the GAs operation

	x_4	x_5	x_6	$minf(x)$
GA optimal values	4	2	12	168
Change x_4 value	3	2	12	227
Change x_5 value	4	3	12	194
Change $x_6 value$	4	2	6	270

6.4.2.2 Test2

We used eq.(6.13) as objective function and applied the data listed in Table 6.10 to find the optimal parameter values of $x_4; x_5; x_6$. The optimum execution time is obtained as 458 seconds when the number of nodes, slots and tasks are set at 4, 3, 8 respectively. Figure 6.3 illustrates the evolution curves when searching for the minimal value 458.

Table 6.10: User input requirements and the system constraints

Variables	**Parameter**
User input:	
Video Resolution (x_1),(x_2)	720x576
Video file size (x_3)	720 sec
Execution deadline $(time_d)$	400 sec
Constraints:	
Number of nodes (x_4)	$2 \leq x_4 \leq 8$
Number of slots (x_5)	$2 \leq x_5 \leq 4$
Number of tasks (x_6)	$4 \leq x_6 \leq 32$

We evaluated the results generated from GAs operation further by substituting

Table 6.11: Results generated by the GAs operation

Variables	**GAs Solution**
Number of nodes (x_4)	4
Number of slots (x_5)	3
Number of tasks (x_6)	8
Fitness value $f(x)$	458

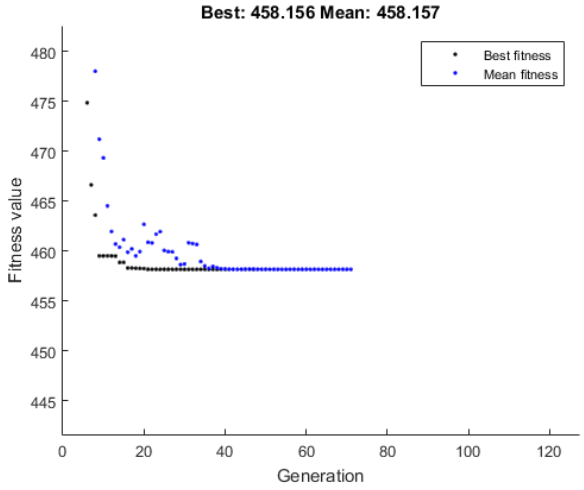


Figure 6.3: Evolution curves searching for best fitness

the values of $x_4; x_5; x_6$, that results in the optimal performance to the fitness function of our model to obtain the optimal reading for the execution time. Then the parameters were changed by slightly increasing and decreasing each parameter separately whilst keeping the other parameters fixed at the values that created the optimal execution time. The results are tabulated in Table 6.12 which demonstrates that what we have obtained is the optimal execution time under the given constraints

Table 6.12: Analysis of results generated by the GAs operation

	x_4	x_5	x_6	$minf(x)$
GA optimal values	4	3	8	458
Change x_4 value	3	2	12	528
Change x_5 value	4	4	12	468
Change $x_6 value$	4	3	6	614

6.4.3 Comparison of Different Optimization Results

We optimised the same fitness functions (objective functions for the execution time) using PS and LM optimisation techniques from Matlab optimization toolbox. We followed the same structure and scenarios applied to GA in (section 6.4) to compare the obtained results. Tables 6.13 $ 6.14 show the optimal values obtained by all the algorithms for both face detection and motion detection applications with video type1 and type2. We observed that PS and LM algorithms did not reached the optimal solution compared to GA. This is because GA starts with a population of points that are randomly distributed in the search space. This reduce the risk of falling in local minimum that is not global. Whereas PS and LM converged to a local optima due to the various distribution of variable values in our model that GAs is capable to avoid. Therefore, we conclude that GAs is a more effective approach to our problem optimisation as compared to the above algorithms.

Table 6.13: Comparative optimization results for face detection application with two video types.

	Video type1				**Video type2**			
Optimization Techniques	x_4	x_5	x_6	$minf(x)$	x_4	x_5	x_6	$minf(x)$
GAs optimal values	8	4	32	2248	3	2	6	260
PS optimal values	6	2	32	2858	3	2	7	326
LM optimal values	6	2	32	2858	4	2	5	260

Table 6.14: Comparative optimization results for motion detection application with two video types.

	Video type1				**Video type2**			
Optimization Techniques	x_4	x_5	x_6	$minf(x)$	x_4	x_5	x_6	$minf(x)$
GAs optimal values	4	2	12	168	4	3	8	458
PS optimal values	3	2	11	668	5	2	29	459
LM optimal values	3	3	8	672	6	4	8	458

6.5 Conclusion

In this chapter a Genetic Algorithm based method for the minimisation of the execution time based on the Linear Regression based performance prediction model obtained in Chapter-5 was proposed. The minimal execution time was effectively obtained subject to operation constraints under which the Hadoop based architecture executed the algorithm within the distributed cluster of VMs. It was shown that GAs can be used for this process with different video processing applications and video types. The use of two alternative algorithms i.e. PS and LM were also considered but were found to be less effective.

The work presented in this chapter concludes the final part of the novel performance modelling and optimisation framework proposed in this thesis, with Chapter 4 proposed a preliminary study to identify the significant parameters that affects the performance, Chapter-5 successfully modelling the performance based on these parameters and finally Chapter-6 proposing the optimisation approach. This successfully concludes the research agenda of this thesis.

Chapter 7

Conclusion and Future Work

This chapter summarises and concludes the major research findings of this thesis, and explains how these findings have contributed to achieving the research objectives. It also outlines potential opportunities to further improve the research presented in this thesis that could extend the knowledge in this subject and the applications that could benefit from this research.

7.1 Summary

The research presented in this thesis has investigated the feasibility of implementing video applications(face detection & motion detection) in a computing environment with distributed processing capabilities, so that the application could benefit from scalabilities of computing resources such an environment can provide. In particular due to the ultimate aim being on a comprehensive study of the use of a cloud based Hadoop infrastructure, a comprehensive review of the legal requirements and performance challenges in deploying video surveillance within a cloud infrastructure (both public and private) was initially investigated (see Chapter-3). The research findings concluded that cloud based architectures can be effectively used for distributed video surveillance provided some strict security guidelines and resource considerations were followed. Further the parameters of a Hadoop based cloud infrastructure that play a significant role in the distribution of computing resources and system performance have been studied in detail. The effect of these parameters on the overall data processing speed when the application is executed in the cloud-like environment has been studied in detail and a machine learning based approach has been adopted for modelling the performance based on the said parameters. A number of different learning algorithms have been adopted and their performance have been investigated and compared. The ability of these models to accurately model and subsequently predict computing resource

requirements of a distributed processing environment when running a selected video analytic application, has been demonstrated. Finally a Genetic Algorithm based single-object optimisation technique that is capable of using these models to obtain optimised performance under given multiple resource constraints, has been presented.

7.2 Conclusions

The following describes the contribution of this thesis:

- For legal rquirements of video surveillance, a private cloud can be used to harness the benefits of cloud based distributed scalable systems to support large scale video surveillance. This option can be more secured and controlled within an operator's premises.
- A Hadoop based framework was proposed in this research to perform the distributed parallel processing of video surveillance application within a private cloud environment.
- The performance of the video analytic application when implemented on the Hadoop based virtual cluster showed increasd performance from the aspect of scalability and efficiency in computation.
- ML methods helped in understanding the relationship between the different parameters (features) affecting the performance model. Bagging and decision tree models fit very well with our video application/data.
- Genetic Algorithms based approaches perform most effectively to our bound minimization problem.

7.3 Future Work

While the novel ideas presented in this thesis advances the current distributed processing framework in a number of areas related to resource allocation and performance in the application area of video surveillance system, the findings discussed in the last section highlighted a number of further opportunities and new directions that could be explored for future studies. These are presented below:

1. An important direction for improving the proposed performance modelling is to consider other video file attributes (features) when creating the model

such as different frame rates, input video with different levels of complexity, different video file formats that may need decoding using algorithms of different complexity before processing, etc.

2. The process of modelling the performance of the video application in a cloud-like environment can be made more accurate by considering other resource utilisations such as network data transmission between VMs, and available memory, alongside CPU resource which was considered in this thesis for better performance analysis of the underlying platform that can be used to improve the propose resource allocation optimization technique.

3. The use of virtualization makes our implementation much faster in terms of generating virtual machine and presents the ability to clone VMs for scalability when more nodes are needed. In the research presented in this thesis by using a simple virtual cluster the manual process adopted for adding or removing VM nodes was found to be time consuming since in each attempt one has to ensure the stored block data of the removed nodes are re-distributed to live nodes for availability. Thus automated configuration and rebalancing is needed. This can be obtained by using a cloud infrastructure integrated, Hadoop framework, for example such as Openstack Sahara[33].

4. The presented evaluation in this thesis was conducted on a small scale cluster of VMs running on a single physical machine. Therefore to generalize the results, the scale of the experiment set up should be extended to a large number of virtual machines, residing on different physical machines. Making use of existing public cloud services or a large dedicated private cloud is recommended

5. As mentioned in section 4.5, the face detection algorithm tested and evaluated runs on a frame by frame basis, with the same type of processing applied on each frame. As the Hadoop framework runs the application tasks in parallel, as the utilised code is of a sequential nature, the only parallelism it exploits is the division of input data into data segments (a collection of frames) that are independently processed by the distributed VMs. More complicated video processing algorithms contain a collection of basic algorithms (i.e. background/foreground extraction, colour correction, object detection, etc.) that could be run in parallel or in a more structured manner, within a distributed environment. It is recommended that such implementations are parallelised to take the best advantage of a distributed Hadoop based cloud like environment.

6. In this research we have used a multi-constraint, single objective optimization algorithm based on Genetic Algorithms, for the optimisation, i.e. the minimisation of the execution time. In other words we only have one objective, which has been optimised/minimised. Often in practice there will be a need to optimise together, two objectives, such as speed and the number of VMs. In such an attempt we can continue to use a Genetic Algorithms based approach for multi-objective optimisation.

7. We evaluated our model using the Hadoop MapReduce framework. We showed that the MapReduce approach has some practical limitations that are addressed by the more recent version of Hadoop YARN. It will be useful to repeat this research with the use of the Hadoop YARN framework to benefit from its resource management mechanism.

8. In the proposed research the Hadoop framework is used for online batch processing, it would be interesting to extend the work to involve online and interactive processing between different complex video analytics/forensics tasks. A possible future research direction is to use other integrated data processing frameworks, such as Spark[10] and Pig[162].

This research was motivated to respond to expectations of digital surveillance systems that apply advanced technology. We have resolved a number of fundamental research issues with regards to the performance of analysis, modelling and optimisation of video processing systems, when deployed in a cloud like environment by answering key questions related to efficiently of allocating computing resources to ensure an effective performance. In doing so a number of research gaps in existing research was closed. Despite the contributions of this thesis a substantial amount of further research can be carried out to further the advancement of this field.

References

[1] Amazon EC2 instance types. URL: http://aws.amazon.com/ec2/instance-types/.

[2] Amazon EMR. URL: https://aws.amazon.com/elasticmApreduce/.

[3] Amazon health care compliance. URL: http://media.amazonwebservices.com/AWS{_}HIPAA{_}Whitepaper{_}Final.pdf.

[4] Amazon Purchasing Options. URL: https://aws.amazon.com/ec2/purchasing-options/.

[5] Amazon Virtual Private Cloud (VPC). URL: http://aws.amazon.com/vpc/.

[6] Amdahl's law. URL: https://en.wikipedia.org/wiki/Talk:Amdahl{%}2527s{_}law.

[7] Apache CloudStack: Open Source Cloud Computing. URL: https://cloudstack.apache.org/.

[8] Apache Hadoop. URL: http://hadoop.apache.org/.

[9] Apache Hadoop 2.7.2 Apache Hadoop YARN. URL: http://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html.

[10] Apache Spark - Lightning-Fast Cluster Computing. URL: http://spark.apache.org/.

[11] Apache YARN. URL: http://hortonworks.com/apache/yarn/.

[12] AWS GovCloud (US) Region Overview Government Cloud Computing. URL: http://aws.amazon.com/govcloud-us/.

[13] AWS Security Center. URL: http://aws.amazon.com/security/.

[14] BBC News - Emergency phone and internet data laws to be passed. URL: http://www.bbc.co.uk/news/uk-politics-28237111.

[15] BBC News - Surveillance camera code of practice comes into force. URL: http://www.bbc.co.uk/news/uk-23636462.

[16] CASE STUDY Intel® Distribution for Apache Hadoop* Software Real-Time Video Surveillance Across Locations Shanghai. URL: http://www.intel.com/content/dam/www/public/us/en/documents/case-studies/big-data-apache-hadoop-china-telecom-case-study.pdf.

[17] Digital Images as Evidence - CCTV Information. URL: http://www.cctv-information.co.uk/i/Digital{_}Images{_}as{_}Evidence.

[18] FFmpeg. URL: https://www.ffmpeg.org/.

[19] Filesystem in Userspace - Wikipedia. URL: https://en.wikipedia.org/wiki/Talk:Filesystem{_}in{_}Userspace.

[20] Frequently asked questions related to transfers of personal data from the EU/EEA. URL: FREQUENTLYASKEDQUESTIONSRELATINGTOTRANSFERSOFPERSONALDATAFROMTHEEU/EEA.

[21] Ganglia Monitoring System. URL: http://ganglia.sourceforge.net/.

[22] Google Trends. URL: https://www.google.co.uk/trends/http://whatis.techtarget.com/definition/Google-Trends.

[23] Hadoop - YDN. URL: https://developer.yahoo.com/hadoop/.

[24] Hadoop on Virtual Machines. URL: http://www.slideshare.net/rjmcdougall/hadoop-on-virtual-machines.

[25] HDFS Architecture Guide. URL: http://hadoop.apache.org/docs/r1.2.1/hdfs{_}design.html.

[26] Hype Cycle for Cloud Computing, 2011. URL: https://www.gartner.com/doc/1753115/hype-cycle-cloud-computing-.

[27] IDC Forecasts Worldwide Public IT Cloud Services Spending to Reach Nearly $108 Billion by 2017 as Focus Shifts from Savings to InNovation - prUS24298013. URL: http://www.idc.com/getdoc.jsp?containerId=prUS24298013.

[28] KVM. URL: http://www.linux-kvm.org/page/Main{_}Page.

[29] Large-Scale Video Analytics on Hadoop — Pivotal P.O.V. URL: https://blog.pivotal.io/data-science-pivotal/features/large-scale-video-analytics-on-hadoop.

[30] OpenCV. URL: http://opencv.org/.

[31] OpenNebula — Flexible Enterprise Cloud Made Simple. URL: http://opennebula.org/.

[32] Retrieval of Video Evidence and Production of Working Copies from Digital CCTV Systems v2.0. URL: https://www.gov.uk/government/uploads/system/uploads/attachment{_}data/file/378448/66-08{_}Retrieval{_}of{_}Video{_}Ev12835.pdf.

[33] Sahara - OpenStack. URL: https://wiki.openstack.org/wiki/Sahara.

[34] Test Media. URL: https://media.xiph.org/derf/.

[35] Trends in Computer Science Research. URL: http://cacm.acm.org/magazines/2013/10/168170-trends-in-computer-science-research/fulltext.

[36] Tuning YARN. URL: http://www.cloudera.com/documentation/enterprise/5-3-x/topics/cdh{_}ig{_}yarn{_}tuning.html.

[37] Twenty-One Experts Define Cloud Computing — Cloud Computing Journal. URL: http://cloudcomputing.sys-con.com/node/612375.

[38] VSaaS - Video Surveillance as a Service. URL: http://www.vsaas.com/.

[39] Weka 3 - Data Mining with Open Source Machine Learning Software in Java. URL: http://www.cs.waikato.ac.nz/ml/weka/.

[40] Determining what is personal data, 1998. URL: https://ico.org.uk/media/for-organisations/documents/1554/determining-what-is-personal-data.pdf.

[41] CCTV code of practice, 2008. URL: http://www.belb.org.uk/downloads/foi{_}cctv{_}code{_}of{_}practice.pdf.

[42] Top Threats to Cloud Computing V1.0. Technical Report March, 2010.

[43] ACPO Good Practice Guide for Digital Evidence. 2012.

[44] Article 29 Data Protection Working Party, 2012. URL: http://ec.europa.eu/justice/data-protection/article-29/documentation/opinion-recommendation/files/2012/wp196{_}en.pdf.

[45] Virtualized Hadoop Performance with VMware vSphere 5.1, 2012. URL: http://www.vmware.com/files/pdf/vmware-virtualizing-apache-hadoop.pdf.

[46] Surveillance Camera Code of Practice, 2013. URL: https://www.gov.uk/government/publications/surveillance-camera-code-of-practice.

[47] Michael Armbrust, Ion Stoica, Matei Zaharia, Armando Fox, Rean Griffith, Anthony D. JoSeph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, and Ariel Rabkin. A view of cloud computing. *Commun. ACM*, 53(4):50, Apr 2010. URL: http://dl.acm.org/ft{_}gateway.cfm?id=1721672{&}type=html, doi:10.1145/1721654.1721672.

[48] Paolo Balboni, Via Mascheroni, Avv Paolo, and Balboni Law. Data Protection and Data Security Issues Related to Cloud Computing in the EU. *Soc. Sci. Res.*, 022(022):1–12, 2010. URL: http://ssrn.com/abstract=1661437, doi:10.2139/ssrn.1661437.

[49] Dimitri P Bertsekas. *Constrained optimization and Lagrange multiplier methods*. Academic press, 2014.

[50] Robert B. Bohn, John Messina, Fang Liu, Jin Tong, and Jian Mao. NIST cloud computing reference architecture. In *Proc. - 2011 IEEE World Congr. Serv. Serv. 2011*, pages 594–596, 2011. doi:10.1109/SERVICES.2011.105.

[51] Hedlund Brad. Understanding Hadoop Clusters and the Network, 2011. URL: http://bradhedlund.com/2011/09/10/understanding-hadoop-clusters-and-the-network/.

[52] Leo Breiman. Bagging predictors. *Mach. Learn.*, 24(2):123–140, Aug 1996. URL: http://link.springer.com/10.1007/BF00058655, doi:10.1007/BF00058655.

[53] By Jon Brodkin, Network World Cloud, Security Risks, Cloud Computing, and Google App Engine. Gartner : Seven cloud-computing security risks. pages 2–3, 2008.

[54] RajkuMar Buyya, James Broberg, and Andrzej M. Goscinski. Cloud Computing Principles and Paradigms. Mar 2011. URL: http://dl.acm.org/citation.cfm?id=1971955.

[55] Daniele Catteddu and Giles Hogben. Cloud Computing: Benefits, risks and recommendation for information security. Technical report, 2009.

[56] Deyan Chen and Hong Zhao. Data Security and Privacy Protection Issues in Cloud Computing. *2012 Int. Conf. Comput. Sci. Electron. Eng.*, (973):647–651, 2012. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6187862, doi:10.1109/ICCSEE.2012.193.

[57] Deyan Chen and Hong Zhao. Data Security and Privacy Protection Issues in Cloud Computing. In *2012 Int. Conf. Comput. Sci. Electron. Eng.*, volume 1, pages 647–651. IEEE, 2012. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6187862, doi:10.1109/ICCSEE.2012.193.

[58] Keke Chen, James Powers, Shumin Guo, and Fengguang Tian. CRESP: Towards Optimal Resource Provisioning for MApreduce Computing in Public Clouds. *IEEE Trans. Parallel Distrib. Syst.*, 25(6):1403–1412, 2014. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6678508, doi:10.1109/TPDS.2013.297.

[59] Ryu Chungmo, Lee Daecheol, Jang Minwook, Kim Cheolgi, and Seo Euiseong. Extensible Video Processing Framework in Apache Hadoop. In *2013 IEEE 5th Int. Conf. Cloud Comput. Technol. Sci.*, volume 2, pages 305–310. IEEE, Dec 2013. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6735441, doi:10.1109/CloudCom.2013.153.

[60] Christopher Chute and David Reinsel. Network Video Surveillance : Addressing Storage Challenges, 2012. URL: http://www.emc.com/collateral/analyst-reports/ar-idc-nw-video-surv.pdf.

[61] Cloud Security Alliance. Security Guidance for Critical Areas of Cloud Security in Cloud Computing. URL: https://cloudsecurityalliance.org/research/security-guidance/.

[62] Neil Cohen and Ken Maclennan-brown. Digital Imaging Procedure, 2007.

[63] Javier Conejero, Blanca Caminero, and Carmen Carrion. Analysing Hadoop performance in a multi-user IaaS Cloud. In *2014 Int. Conf. High Perform. Comput. Simul.*, pages 399–406. IEEE, Jul 2014. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6903713, doi:10.1109/HPCSim.2014.6903713.

[64] CSA. The Notorious Nine Cloud Computing Top Threats in 2013, 2013.

[65] Kamal Dahbur, Bassil Mohammad, and Ahmad Bisher Tarakji. A survey of risks, threats and vulnerabilities in cloud computing. In *Proc. 2011 Int. Conf. Intell. Semant. Web-Services Appl. - ISWSA '11*, pages 1–6, New York, New York, USA, Apr 2011. ACM Press. URL: http://dl.acm.org/citation.cfm?id=1980822.1980834, doi:10.1145/1980822.1980834.

[66] Sam De Silva. Key Legal Issues with Cloud Computing: A UK Law Perspective. In Al Bento and A. K Aggarwal, editors, *Cloud Comput. Serv. Deploy. Model.*, chapter 13. IGI Global, Oct 2012. URL: http://www.igi-global.com/chapter/content/70144, doi:10.4018/978-1-4666-2187-9.

[67] Frank Doelitzscher. *Security Audit Compliance For Cloud Computing*. Thesis, Plymouth University, 2014.

[68] Frank Doelitzscher, Christoph Reich, and Anthony Sulistio. Designing Cloud Services Adhering to Government Privacy Laws. In *2010 10th IEEE Int. Conf. Comput. Inf. Technol.*, pages 930–935. IEEE, Jun 2010. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5578475, doi:10.1109/CIT.2010.172.

[69] eds. Ma, Yunqian, and Gang Qian. *Intelligent video surveillance: systems and technology*. CRC Press, 2009. URL: http://www.crcpress.com/product/isbn/9781439813287.

[70] Kathleen Ericson and Shrideep Pallickara. On the Performance of Virtualized Infrastructures for Processing Realtime Streaming Data. In *2012 IEEE Fifth Int. Conf. Util. Cloud Comput.*, pages 176–183. IEEE, Nov 2012. URL: http://dl.acm.org/citation.cfm?id=2415689.2415725, doi:10.1109/UCC.2012.15.

[71] Thomas Erl, Ricardo Puttini, and Zaigham Mahmood. Cloud Computing: Concepts, Technology & Architecture. May 2013. URL: http://dl.acm.org/citation.cfm?id=2500934.

[72] Eugen Feller, Lavanya Ramakrishnan, and Christine Morin. On the performance and energy efficiency of Hadoop deployment models. In *2013 IEEE Int. Conf. Big Data*, pages 131–136. IEEE, Oct 2013. URL: http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=6691564, doi:10.1109/BigData.2013.6691564.

[73] Ian Foster and Carl Kesselman. The Grid 2: Blueprint for a New Computing Infrastructure. Nov 2003. URL: http://dl.acm.org/citation.cfm?id=996313.

[74] Ian Foster, Yong Zhao, Ioan Raicu, and Shiyong Lu. Cloud Computing and Grid Computing 360-Degree Compared. In *2008 Grid Comput. Environ. Work.*, pages 1–10. IEEE, Nov 2008. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4738445, doi:10.1109/GCE.2008.4738445.

[75] Frank Gens. IDC eXchange Blog Archive New IDC IT Cloud Services Survey: Top Benefits and Challenges, 2009. URL: http://blogs.idc.com/ie/?p=730.

[76] David E. Goldberg and John H. Holland. Genetic Algorithms and Machine Learning. *Mach. Learn.*, 3(2-3):95–99. URL: http://link.springer.com/article/10.1023/A{%}253A1022602019183, doi:10.1023/A:1022602019183.

[77] Nelson Gonzalez, Charles Miers, Fernando Redígolo, Marcos Simplício, Tereza Carvalho, Mats Näslund, and Makan Pourzandi. A quantitative analysis of current security concerns and solutions for cloud computing. *J. Cloud Comput. Adv. Syst. Appl.*, 1(1):11, 2012. URL: http://www.journalofcloudcomputing.com/content/1/1/11, doi:10.1186/2192-113X-1-11.

[78] Bernd Grobauer, Tobias Walloschek, and ElMar Stocker. Understanding Cloud Computing Vulnerabilities. *IEEE Secur. Priv. Mag.*, 9(2):50–57, Mar 2011. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5487489, doi:10.1109/MSP.2010.115.

[79] Cloud Computing Use Case Discussion Group. Cloud Computing Use Cases. pages 1–57, 2010. URL: papers2://publication/livfe/id/21564.

[80] A. Hampapur, L. Brown, J. Connell, and A. Senior. IBM sMart surveillance system (S3): a open and extensible framework for event based surveillance. In *Proceedings. IEEE Conf. Adv. Video Signal Based Surveillance, 2005.*, pages 318–323. IEEE, 2005. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1577288, doi:10.1109/AVSS.2005.1577288.

[81] Jiawei Han, Micheline Kamber, and Jian Pei. *Data mining: concepts and techniques*. 2012. URL: http://medcontent.metApress.

com/index/A65RM03P4874243N.pdfhttp://books.google.com/books?hl=en{&}lr={&}id=AfL0t-YzOrEC{&}oi=fnd{&}pg=PP2{&}dq=Data+Mining+-+Concepts+and+Techniques{&}ots=Uv-{_}qO7qF9{&}sig=uOlQMuZWqvolT8EjxUbOuq{_}oe84, doi:10.1002/1521-3773(20010316)40:6<9823::AID-ANIE9823>3.3.CO;2-C.

[82] Keiko Hashizume, David G Rosado, Eduardo Fernández-Medina, and Eduardo B Fernandez. An analysis of security issues for cloud computing. *J. Internet Serv. Appl.*, 4(1):5, 2013. URL: http://www.jisajournal.com/content/4/1/5, doi:10.1186/1869-0238-4-5.

[83] Randy L. Haupt and Sue Ellen Haupt. *Practical Genetic Algorithms*. John Wiley & Sons, 2004. URL: https://books.google.com/books?hl=en{&}lr={&}id=k0jFfsmbtZIC{&}pgis=1.

[84] Yanzhang He, Xiaohong Jiang, Zhaohui Wu, Kejiang Ye, and Zhongzhong Chen. Scalability Analysis and Improvement of Hadoop Virtual Cluster with Cost Consideration. In *2014 IEEE 7th Int. Conf. Cloud Comput.*, pages 594–601. IEEE, Jun 2014. URL: http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=6973791, doi:10.1109/CLOUD.2014.85.

[85] Arto Heikkinen, Jouni Sarvanko, Mika Rautiainen, and Mika Ylianttila. Distributed multimedia content analysis with MApreduce. In *2013 IEEE 24th Annu. Int. Symp. Pers. Indoor, Mob. Radio Commun.*, pages 3497–3501. IEEE, Sep 2013. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6666755, doi:10.1109/PIMRC.2013.6666755.

[86] Leonard Heilig and Stefan Voss. A Scientometric Analysis of Cloud Computing Literature. *IEEE Trans. Cloud Comput.*, PP(99):1–1, 2014. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6808484, doi:10.1109/TCC.2014.2321168.

[87] Herodotos Herodotou, Fei Dong, and Shivnath Babu. No one (cluster) size fits all: automatic cluster sizing for data-intensive analytics. In *Proc. 2nd ACM Symp. Cloud Comput. - SOCC '11*, pages 1–14, New York, New York, USA, Oct 2011. ACM Press. URL: http://dl.acm.org/citation.cfm?id=2038916.2038934, doi:10.1145/2038916.2038934.

[88] M. Anwar Hossain. Analyzing the Suitability of Cloud-Based Multimedia Surveillance Systems. In *2013 IEEE 10th Int. Conf. High Perform. Comput. Commun. 2013 IEEE Int. Conf. Embed. Ubiquitous Comput.*, pages 644–650. IEEE, Nov 2013. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6831978, doi:10.1109/HPCC.and.EUC.2013.96.

[89] M Anwar Hossain. Framework for a Cloud-Based Multimedia Surveillance System. 2014, 2014.

[90] M. Shamim Hossain, M. Mehedi Hassan, M. Al Qurishi, and Abdullah Alghamdi. Resource Allocation for Service Composition in Cloud-based Video Surveillance Platform. In *2012 IEEE Int. Conf. Multimed. Expo Work.*, pages 408–412. IEEE, Jul 2012. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6266418, doi:10.1109/ICMEW.2012.77.

[91] M. Shamim Hossain, M. Mehedi Hassan, M. Al Qurishi, and Abdullah Alghamdi. Resource Allocation for Service Composition in Cloud-based Video Surveillance Platform. In *2012 IEEE Int. Conf. Multimed. Expo Work.*, pages 408–412. IEEE, Jul 2012. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6266418, doi:10.1109/ICMEW.2012.77.

[92] Muhammad Faraz Hyder, Muhammad Ali Ismail, and Hameeza Ahmed. Performance comparison of Hadoop Clusters configured on virtual machines and as a cloud service. In *2014 Int. Conf. Emerg. Technol.*, pages 60–64. IEEE, Dec 2014. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7021017, doi:10.1109/ICET.2014.7021017.

[93] Shadi Ibrahim, Hai Jin, Bin Cheng, HaiJun Cao, Song Wu, and Li Qi. CLOUDLET. In *Proc. 18th ACM Int. Symp. High Perform. Distrib. Comput. - HPDC '09*, page 65, New York, New York, USA, Jun 2009. ACM Press. URL: http://dl.acm.org/citation.cfm?id=1551609.1551624, doi:10.1145/1551609.1551624.

[94] Shadi Ibrahim, Hai Jin, Lu Lu, Li Qi, Song Wu, and Xuanhua Shi. Evaluating MApreduce on virtual machines: The Hadoop case. In *Cloud Comput.*, volume 5931 LNCS, pages 519–528. Springer Berlin Heidelberg, 2009. URL: http://dx.doi.org/10.1007/978-3-642-10665-1{_}47, doi:10.1007/978-3-642-10665-1_47.

[95] ICO. Guidance on the use of cloud computing, 2012. URL: https://ico.org.uk/media/for-organisations/documents/1540/cloud{_}computing{_}guidance{_}for{_}organisations.pdf.

[96] ICO. In the picture : A data protection code of practice for surveillance cameras and personal information. Technical report, 2015. URL: https://ico.org.uk/media/for-organisations/documents/1542/cctv-code-of-practice.pdf.

[97] Impetus. Hadoop performance tuning. (October 2009):1–13, 2009.

[98] Sadeka Islam, Jacky Keung, Kevin Lee, and Anna Liu. Empirical prediction models for adaptive resource provisioning in the cloud. *Futur. Gener. Comput. Syst.*, 28(1):155–162, Jan 2012. URL: http://www.sciencedirect.com/science/article/pii/S0167739X11001129, doi:10.1016/j.future.2011.05.027.

[99] W A Jansen. Cloud Hooks: Security and Privacy Issues in Cloud Computing. In *2011 44th Hawaii Int. Conf. Syst. Sci.*, pages 1–10. IEEE, Jan 2011. URL: http://dl.acm.org/citation.cfm?id=1955602.1956037, doi:10.1109/HICSS.2011.103.

[100] F. Jokhio, A. Ashraf, S. Lafond, I. Porres, and J. Lilius. Prediction-Based Dynamic Resource Allocation for Video Transcoding in Cloud Computing. In *2013 21st Euromicro Int. Conf. Parallel, Distrib. Network-Based Process.*, pages 254–261. IEEE, Feb 2013. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6498561, doi:10.1109/PDP.2013.44.

[101] Tim Jones and Micah Nelson. Moving ahead with Hadoop YARN, 2013. URL: http://www.ibm.com/developerworks/library/bd-hadoopyarn/.

[102] Karthik Kambatla, Abhinav Pathak, and Himabindu Pucha. Towards Optimizing Hadoop Provisioning in the Cloud. *Proc. First Work. Hot Top. Cloud Comput.*, page 118, 2009. URL: http://www.usenix.org/events/hotcloud09/tech/full{_}papers/kambatla.pdf, doi:10.1.1.148.9460.

[103] Michael L. Kemp, Shannon Robb, and P. Candace Deans. The Legal Implications of Cloud Computing. In Al Bento and A. K Aggarwal, editors, *Cloud Comput. Serv. Deploy. Model.*, chapter Chapter 14. IGI Global, Oct 2012. URL: http://www.igi-global.com/chapter/legal-implications-cloud-computing/70145/, doi:10.4018/978-1-4666-2187-9.

[104] Mukhtaj Khan, Yong Jin, Maozhen Li, Yang Xiang, and ChangJun Jiang. Hadoop Performance Modeling for Job Estimation and Resource Provisioning. *IEEE Trans. Parallel Distrib. Syst.*, 27(2):441–454, Feb 2016. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7045505, doi:10.1109/TPDS.2015.2405552.

[105] Md. Tanzim Khorshed, A.B.M. Shawkat Ali, and Saleh A. Wasimi. A survey on gaps, threat remediation challenges and some thoughts for proactive at-

tack detection in cloud computing. *Futur. Gener. Comput. Syst.*, 28(6):833–851, Jun 2012. URL: http://www.sciencedirect.com/science/article/pii/S0167739X12000180, doi:10.1016/j.future.2012.01.006.

[106] Y Kodratoff. *Introduction to Machine Learning*. 2014. URL: https://books.google.co.il/books?id=AQyjBQAAQBAJ.

[107] Palden Lama and Xiaobo Zhou. AROMA: automated resource allocation and configuration of mApreduce environment in the cloud. In *Proc. 9th Int. Conf. Auton. Comput. - ICAC '12*, page 63, New York, New York, USA, Sep 2012. ACM Press. URL: http://dl.acm.org/citation.cfm?id=2371536.2371547, doi:10.1145/2371536.2371547.

[108] Gunho Lee, Niraj Tolia, Parthasarathy Ranganathan, and Randy H. Katz. Topology-aware resource allocation for data-intensive workloads. In *Proc. first ACM asia-pacific Work. Work. Syst. - APSys '10*, page 1, New York, New York, USA, Aug 2010. ACM Press. URL: http://dl.acm.org/citation.cfm?id=1851276.1851278, doi:10.1145/1851276.1851278.

[109] Dionisio Leite, Maycon Peixoto, Marcos Santana, and Regina Santana. Performance Evaluation of Virtual Machine Monitors for Cloud Computing. In *2012 13th Symp. Comput. Syst.*, pages 65–71. IEEE, Oct 2012. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6391765, doi:10.1109/WSCAD-SSC.2012.22.

[110] Alexander Lenk, Markus Klems, Jens Nimis, Stefan Tai, and Thomas Sandholm. What's inside the Cloud? An architectural map of the Cloud landscape. In *2009 ICSE Work. Softw. Eng. Challenges Cloud Comput.*, pages 23–31. IEEE, May 2009. URL: http://dl.acm.org/citation.cfm?id=1564595.1564625, doi:10.1109/CLOUD.2009.5071529.

[111] Terrence V. Lillard. *Digital Forensics for Network, Internet, and Cloud Computing: A Forensic Evidence Guide for Moving Targets and Data*. Syngress Publishing, Jun 2010. URL: http://dl.acm.org/citation.cfm?id=1875294.

[112] Thanathip Limna and Pichaya Tandayya. A flexible and scalable component-based system architecture for video surveillance as a service, running on infrastructure as a service. *Multimed. Tools Appl.*, Dec 2014. URL: http://link.springer.com/10.1007/s11042-014-2373-8, doi:10.1007/s11042-014-2373-8.

[113] Meriam Mahjoub, Afef Mdhaffar, Riadh Ben Halima, and Mohamed Jmaiel. A Comparative Study of the Current Cloud Computing Technologies and Offers. In *2011 First Int. Symp. Netw. Cloud Comput. Appl.*, pages 131–134. IEEE, Nov 2011. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6123450, doi:10.1109/NCCA.2011.28.

[114] Qasim Mahmood Rajpoot and Christian D. Jensen. Security and Privacy in Video Surveillance: Requirements and Challenges. In Nora Cuppens-Boulahia, Frédéric Cuppens, Sushil Jajodia, Anas Abou El Kalam, and Thierry Sans, editors, *ICT Syst. Secur. Priv. Prot.*, volume 428 of *IFIP Advances in Information and Communication Technology*, pages p. 169–184. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014. URL: http://link.springer.com/10.1007/978-3-642-55415-5, doi:10.1007/978-3-642-55415-5.

[115] C. P. Malloth, P. Felber, A. Schiper, and U. Wilhelm. Phoenix: A Toolkit for Building Fault-Tolerant Distributed Applications in Large Scale. In *Work. Parallel Distrib. Platforms Ind. Prod.*, 1995. URL: http://infoscience.epfl.ch/record/50153.

[116] Pratyusa K. Manadhata and Jeannette M. Wing. An Attack Surface Metric. *IEEE Trans. Softw. Eng.*, 37(3):371–386, May 2011. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5482589, doi:10.1109/TSE.2010.60.

[117] Renzo Marchini. *Cloud Computing: A Practical Introduction to the Legal Issues*. BSI, 2010. URL: http://books.google.co.uk/books/about/Cloud{_}Computing.html?id=7RBKewAACAAJ{&}pgis=1.

[118] Peter Mell and Timothy Grance. The NIST Definition of Cloud Computing (Draft) Recommendations of the National Institute of Standards and Technology. *Nist Spec. Publ.*, 145.

[119] Peter Mell and Timothy Grance. The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology.

[120] Jeffrey L Nagel, G P C Ibbons, and L Jeffrey. Getting ESI Evidence Admitted : Lorraine v . Markel American Insurance Co ., 2007. URL: http://www.metrocorpcounsel.com/articles/9210/getting-esi-evidence-admitted-lorraine-v-Markel-american-insurance-co.

[121] Robayet Nasim and Andreas J. Kassler. Deploying OpenStack: Virtual Infrastructure or Dedicated Hardware. In *2014 IEEE 38th*

Int. Comput. Softw. Appl. Conf. Work., pages 84–89. IEEE, Jul 2014. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6903110, doi:10.1109/COMPSACW.2014.18.

[122] David Neal and Syed M. Rahman. Video surveillance in the cloud-computing? In *2012 7th Int. Conf. Electr. Comput. Eng.*, pages 58–61. IEEE, Dec 2012. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6471484, doi:10.1109/ICECE.2012.6471484.

[123] Dj Neal. Video Surveillance in the Cloud? *Int. J. Cryptogr. Inf. Secur.*, 2(3):1–19, Sep 2012. URL: http://www.airccse.org/journal/ijcis/papers/2312ijcis01.pdf, doi:10.5121/ijcis.2012.2301.

[124] Peter P. Nghiem and Silvia M. Figueira. Towards efficient resource provisioning in MApreduce. *J. Parallel Distrib. Comput.*, Apr 2016. URL: http://www.sciencedirect.com/science/article/pii/S0743731516300077, doi:10.1016/j.jpdc.2016.04.001.

[125] I. G. B. B. Nugraha. Video analysis tools for cloud-based motion detection. In *2012 Int. Conf. Cloud Comput. Soc. Netw.*, pages 1–4. IEEE, Apr 2012. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6215716, doi:10.1109/ICCCSN.2012.6215716.

[126] Rafael Pereira, Marcello Azambuja, Karin Breitman, and Markus Endler. An Architecture for Distributed High Performance Video Processing in the Cloud. In *2010 IEEE 3rd Int. Conf. Cloud Comput.*, pages 482–489. IEEE, Jul 2010. URL: http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=5557958, doi:10.1109/CLOUD.2010.73.

[127] Rosangela De Fatima Pereira, Walter Akio Goya, Jan-Erik Mangs, and Azimeh Sefidcon. Exploiting Hadoop Topology in Virtualized Environments. *2014 IEEE World Congr. Serv.*, (scenario 1):301–308, Jun 2014. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6903282, doi:10.1109/SERVICES.2014.60.

[128] Rosangela de Fatima Pereira, Walter Akio Goya, Jan-Erik Mangs, and Azimeh Sefidcon. Exploiting Hadoop Topology in Virtualized Environments. In *2014 IEEE World Congr. Serv.*, pages 301–308. IEEE, Jun 2014. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6903282, doi:10.1109/SERVICES.2014.60.

[129] Ivanilton Polato, Reginaldo Ré, Alfredo Goldman, and Fabio Kon. A comprehensive view of Hadoop researchA systematic literature review. *J. Netw.*

Comput. Appl., 46:1–25, Nov 2014. URL: http://www.sciencedirect.com/science/article/pii/S1084804514001635, doi:10.1016/j.jnca.2014.07.022.

[130] Michael Price. The Paradox of Security. (Vmm), 2008.

[131] J R Quinlan. Learning with continuous classes. In *Mach. Learn.*, volume 92, pages 343–348, 1992. URL: http://sci2s.ugr.es/keel/pdf/algorithm/congreso/1992-Quinlan-AI.pdf{%}5Cnhttp://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.34.885{&}rep=rep1{&}type=pdf, doi:10.1.1.34.885.

[132] Fayruz Rahma and Teguh Bharata Adji. Scalability Analysis of KVM-Based Private Cloud For IaaS. 2(4):288–295, 2013.

[133] Hassan Rajaei and Jeffrey Wappelhorst. Clouds & grids: a network and simulation perspective. pages 143–150, Apr 2011. URL: http://dl.acm.org/citation.cfm?id=2048416.2048435.

[134] Bhaskar Prasad Rimal, Eunmi Choi, and Ian Lumb. A Taxonomy and Survey of Cloud Computing Systems. In *2009 Fifth Int. Jt. Conf. INC, IMS IDC*, pages 44–51. IEEE, 2009. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5331755, doi:10.1109/NCM.2009.218.

[135] Thomas Ristenpart, Eran Tromer, Hovav Shacham, and Stefan Savage. Hey, you, get off of my cloud. In *Proc. 16th ACM Conf. Comput. Commun. Secur. - CCS '09*, page 199, New York, New York, USA, Nov 2009. ACM Press. URL: http://dl.acm.org/citation.cfm?id=1653662.1653687, doi:10.1145/1653662.1653687.

[136] Francisco Rocha and Miguel Correia. Lucy in the sky without diamonds: Stealing confidential data in the cloud. In *2011 IEEE/IFIP 41st Int. Conf. Dependable Syst. Networks Work.*, pages 129–134. IEEE, Jun 2011. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5958798, doi:10.1109/DSNW.2011.5958798.

[137] D.A. Rodriguez-Silva, L. Adkinson-Orellana, F.J. Gonz'lez-Castano, I. Armino-Franco, and D. Gonz'lez-Martinez. Video Surveillance Based on Cloud Storage. In *2012 IEEE Fifth Int. Conf. Cloud Comput.*, pages 991–992. IEEE, Jun 2012. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6253615, doi:10.1109/CLOUD.2012.44.

[138] D.A. Rodriguez-Silva, L. Adkinson-Orellana, F.J. Gonz'lez-Castano, I. Armino-Franco, and D. Gonz'lez-Martinez. Video Surveillance Based on Cloud Storage. In *2012 IEEE Fifth Int. Conf. Cloud Comput.*, pages 991–992. IEEE, Jun 2012. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6253615, doi:10.1109/CLOUD.2012.44.

[139] Robert I Rubin and Mark J Stempler. Video Surveillance in Personal Injury Cases. 2010.

[140] Naidila Sadashiv and S. M. Dilip KuMar. Cluster, grid and cloud computing: A detailed comparison. In *2011 6th Int. Conf. Comput. Sci. Educ.*, pages 477–482. IEEE, Aug 2011. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6028683, doi:10.1109/ICCSE.2011.6028683.

[141] Rainer Schmidt and Matthias Rella. An Approach for Processing Large and Non-uniform Media Objects on MApreduce-Based Cluster. In Chunxiao Xing, Fabio Crestani, and Andreas Rauber, editors, *Digit. Libr. Cult. Heritage, Knowl. Dissemination, Futur. Creat.*, volume 7008 of *Lecture Notes in Computer Science*, pages 172–181. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. URL: http://www.springerlink.com/index/10.1007/978-3-642-24826-9, doi:10.1007/978-3-642-24826-9_23.

[142] Eugene J Schweitzer. Reconciliation of the cloud computing model with US federal electronic health record regulations. *J. Am. Med. Inform. Assoc.*, 19(2):161–5, 2011. URL: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3277612{&}tool=pmcentrez{&}rendertype=abstract, doi:10.1136/amiajnl-2011-000162.

[143] George A. F. Seber and Alan J. Lee. *Linear Regression Analysis*. John Wiley & Sons, 2012. URL: https://books.google.com/books?hl=en{&}lr={&}id=X2Y6OkXl8ysC{&}pgis=1.

[144] Krisantus Sembiring and Andreas Beyer. Dynamic resource allocation for cloud-based media processing. In *Proceeding 23rd ACM Work. Netw. Oper. Syst. Support Digit. Audio Video - NOSSDAV '13*, pages 49–54, New York, New York, USA, Feb 2013. ACM Press. URL: http://dl.acm.org/citation.cfm?id=2460782.2460791, doi:10.1145/2460782.2460791.

[145] Sangwon Seo, Ingook Jang, Kyungchang Woo, Inkyo Kim, Jin-Soo Kim, and Seungryoul Maeng. HPMR: Prefetching and pre-shuffling in shared

MApreduce computation environment. In *2009 IEEE Int. Conf. Clust. Comput. Work.*, pages 1–8. IEEE, 2009. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5289171, doi:10.1109/CLUSTR.2009.5289171.

[146] Warren Smith, Ian Foster, and Valerie Taylor. Predicting application run times with historical information. *J. Parallel Distrib. Comput.*, 64(9):1007–1016, Sep 2004. URL: http://www.sciencedirect.com/science/article/pii/S0743731504000991, doi:10.1016/j.jpdc.2004.06.008.

[147] G Somasundaram. *Information storage and management storing, managing, and protecting digital information*. Wiley Pub.,, Indianapolis, Ind. :, 2009.

[148] Chris Sweeney and Sean Arietta. HIPI : A Hadoop Image Processing Interface for Image-based MApreduce Tasks, 2010. URL: https://cs.ucsb.edu/{~}cmsweeney/papers/undergrad{_}thesis.pdf.

[149] Hanlin Tan and Lidong Chen. An approach for fast and parallel video processing on Apache Hadoop clusters. In *2014 IEEE Int. Conf. Multimed. Expo*, pages 1–6. IEEE, Jul 2014. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6890135, doi:10.1109/ICME.2014.6890135.

[150] Khaled Tannir. *Optimizing Hadoop for MApreduce*. Packt Publishing Ltd, 2014. URL: https://books.google.com/books?hl=en{&}lr={&}id=GTnoAgAAQBAJ{&}pgis=1.

[151] Mark Taylor, John Haggerty, David Gresty, and David Lamb. Forensic investigation of cloud computing systems. *Netw. Secur.*, 2011(3):4–10, Mar 2011. URL: http://www.sciencedirect.com/science/article/pii/S1353485811700241, doi:10.1016/S1353-4858(11)70024-1.

[152] Zaigham Mahmood Thomas Erl and Ricardo Puttini. *Cloud computing concepts, technology and architecture by Thomas Erl, Zaigham Mahmood and Ricardo Puttini*, volume 39. ACM, Aug 2014. URL: http://dl.acm.org/citation.cfm?id=2632434.2632462, doi:10.1145/2632434.2632462.

[153] Cheng Tian, Ying Wang, Feng Qi, and Bo Yin. Decision model for provisioning virtual resources in Amazon EC2. pages 159–163, Oct 2012. URL: http://dl.acm.org/citation.cfm?id=2499406.2499427.

[154] Fengguang Tian and Keke Chen. Towards optimal resource provisioning for Running MApreduce programs in public clouds. *Proc. - 2011 IEEE 4th Int. Conf. Cloud Comput. CLOUD 2011*, pages 155–162, 2011. doi:10.1109/CLOUD.2011.14.

[155] Shahed Lafif Tim Mather, Subra KuMaraswamy. *Cloud Security and Privacy*.

[156] Virginia Torczon. On the Convergence of Pattern Search Algorithms. *SIAM J. Optim.*, 7(1):1–25, Feb 1997. URL: http://epubs.siam.org/doi/abs/10.1137/S1052623493250780, doi:10.1137/S1052623493250780.

[157] Nedeljko Vasić, DeJan Novaković, Svetozar Miučin, DeJan Kostić, and Ricardo Bianchini. DejaVu: accelerating resource allocation in virtualized environments. *ACM SIGPLAN Not.*, 47(4):423–436, Jun 2012. URL: http://dl.acm.org/citation.cfm?id=2248487.2151021, doi:10.1145/2248487.2151021.

[158] Vinod KuMar Vavilapalli, Siddharth Seth, Bikas Saha, Carlo Curino, Owen O'Malley, Sanjay Radia, Benjamin Reed, Eric Baldeschwieler, Arun C. Murthy, Chris Douglas, Sharad Agarwal, Mahadev Konar, Robert Evans, Thomas Graves, Jason Lowe, and Hitesh Shah. Apache Hadoop YARN. In *Proc. 4th Annu. Symp. Cloud Comput. - SOCC '13*, pages 1–16, New York, New York, USA, Oct 2013. ACM Press. URL: http://dl.acm.org/citation.cfm?id=2523616.2523633, doi:10.1145/2523616.2523633.

[159] Abhishek Verma, Ludmila Cherkasova, and Roy H. Campbell. ARIA: automatic resource inference and allocation for mApreduce environments. In *Proc. 8th ACM Int. Conf. Auton. Comput. - ICAC '11*, page 235, New York, New York, USA, Jun 2011. ACM Press. URL: http://dl.acm.org/citation.cfm?id=1998582.1998637, doi:10.1145/1998582.1998637.

[160] Abhishek Verma, Ludmila Cherkasova, and Roy H. Campbell. *Resource provisioning framework for MApreduce jobs with performance goals*, volume 7049 LNCS. Springer Berlin Heidelberg, 2011. URL: http://dx.doi.org/10.1007/978-3-642-25821-3{_}9, doi:10.1007/978-3-642-25821-3_9.

[161] Xiaolong Wen, Genqiang Gu, Qingchun Li, Yun Gao, and Xuejie Zhang. Comparison of open-source cloud management platforms: OpenStack and OpenNebula. In *2012 9th Int. Conf. Fuzzy Syst. Knowl. Discov.*, pages 2457–2461. IEEE, May 2012. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6234218, doi:10.1109/FSKD.2012.6234218.

[162] Tom White. *Hadoop: The Definitive Guide*. O'Reilly Media, Inc., fourth edition, May 2012. URL: http://dl.acm.org/citation.cfm?id=2285539.

[163] Thomas Winkler and Bernhard Rinner. Security and Privacy Protection in Visual Sensor Networks. *ACM Comput. Surv.*, 47(1):1–42, Jul 2014. URL: http://dl.acm.org/citation.cfm?id=2620784.2545883, doi:10.1145/2545883.

[164] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition*. Morgan Kaufmann Publishers Inc., 2005.

[165] Zhen Xiao, Weijia Song, and Qi Chen. Dynamic Resource Allocation Using Virtual Machines for Cloud Computing Environment. *IEEE Trans. Parallel Distrib. Syst.*, 24(6):1107–1117, Jun 2013. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6311403, doi:10.1109/TPDS.2012.283.

[166] Zhifeng Xiao and Yang Xiao. Security and Privacy in Cloud Computing. *IEEE Commun. Surv. Tutorials*, 15(2):843–859, 2013. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6238281, doi:10.1109/SURV.2012.060912.00182.

[167] Yong Hua Xiong, Shao Yun Wan, Yong He, and Dan Su. Design and implementation of a prototype cloud video surveillance system. *J. Adv. Comput. Intell. Intell. Informatics*, 18(1):40–47, 2014.

[168] Guanghui Xu, Feng Xu, and Hongxu Ma. Deploying and researching Hadoop in virtual machines. In *2012 IEEE Int. Conf. Autom. Logist.*, pages 395–399. IEEE, Aug 2012. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6308241, doi:10.1109/ICAL.2012.6308241.

[169] Cairong Yan, Ming Zhu, Xin Yang, Ze Yu, Min Li, Youqun Shi, and Xiaolin Li. Affinity-aware Virtual Cluster Optimization for MApreduce Applications. In *2012 IEEE Int. Conf. Clust. Comput.*, pages 63–71. IEEE, Sep 2012. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6337857, doi:10.1109/CLUSTER.2012.13.

[170] Haibo Yang and Mary Tate. A Descriptive Literature Review and Classification of Cloud Computing Research. 31, 2012.

[171] Kejiang Ye, Xiaohong Jiang, Yanzhang He, Xiang Li, Haiming Yan, and Peng Huang. vHadoop: A Scalable Hadoop Virtual Cluster Platform for

MApreduce-Based Parallel Machine Learning with Performance Consideration. In *2012 IEEE Int. Conf. Clust. Comput. Work.*, pages 152–160. IEEE, Sep 2012. URL: http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=6355859, doi:10.1109/ClusterW.2012.32.

[172] Nezih Yigitbasi, Alexandru Iosup, Dick Epema, and Simon Ostermann. C-Meter: A Framework for Performance Analysis of Computing Clouds. In *2009 9th IEEE/ACM Int. Symp. Clust. Comput. Grid*, pages 472–477. IEEE, May 2009. URL: http://dl.acm.org/citation.cfm?id=1577849.1577939, doi:10.1109/CCGRID.2009.40.

[173] Andrew J. Younge, Robert Henschel, James T. Brown, Gregor von Laszewski, Judy Qiu, and Geoffrey C. Fox. Analysis of Virtualization Technologies for High Performance Computing Environments. In *2011 IEEE 4th Int. Conf. Cloud Comput.*, pages 9–16. IEEE, Jul 2011. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6008687, doi:10.1109/CLOUD.2011.29.

[174] Lamia Youseff, Maria Butrico, and Dilma Da Silva. Toward a Unified Ontology of Cloud Computing. In *2008 Grid Comput. Environ. Work.*, pages 1–10. IEEE, Nov 2008. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4738443, doi:10.1109/GCE.2008.4738443.

[175] Matei Zaharia, Mosharaf Chowdhury, Michael J Franklin, Scott Shenker, and Ion Stoica. Spark : Cluster Computing with Working Sets. In *HotCloud'10 Proc. 2nd USENIX Conf. Hot Top. cloud Comput.*, page 10, 2010. doi:10.1007/s00256-009-0861-0.

[176] Matei Zaharia, Andy Konwinski, Anthony D. JoSeph, Randy Katz, and Ion Stoica. Improving MApreduce Performance in Heterogeneous Environments. In *Proc. 8th USENIX Conf. Oper. Syst. Des. Implement.*, pages 29–42. USENIX Association, 2008. URL: http://static.usenix.org/legacy/events/osdi08/tech/full{_}papers/zaharia/zaharia{_}html/.

[177] Liang-Jie Zhang and Qun Zhou. CCOA: Cloud Computing Open Architecture. In *2009 IEEE Int. Conf. Web Serv.*, pages 607–616. IEEE, Jul 2009. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5175875, doi:10.1109/ICWS.2009.144.

[178] Qi Zhang, Lu Cheng, and Raouf Boutaba. Cl[1] Q. Zhang, L. Cheng, and R. Boutaba, Cloud computing: state-of-the-art and research chal-

lenges, J. Internet Serv. Appl., vol. 1, no. 1, pp. 718, Apr. 2010.oud computing: state-of-the-art and research challenges. *J. Internet Serv. Appl.*, 1(1):7–18, Apr 2010. URL: http://www.springerlink.com/index/10.1007/s13174-010-0007-6, doi:10.1007/s13174-010-0007-6.

[179] Xiaomeng Zhao, Huadong Ma, Haitao Zhang, Yi Tang, and Yue Kou. HVPI: Extending Hadoop to Support Video Analytic Applications. In *2015 IEEE 8th Int. Conf. Cloud Comput.*, pages 789–796. IEEE, Jun 2015. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7214119, doi:10.1109/CLOUD.2015.109.