Anticipatory Models of Load Balancing in Cloud Computing

A Thesis Submitted for the Degree of

Doctor of Philosophy

By

Shahrzad Aslanzadeh

University of Technology Sydney

New South Wales, Australia

CERTIFICATE OF ORIGINAL AUTHORSHIP

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Signature of Student: *Shahrzad Aslanzadeh*

Date: 04/14/2016

Acknowledgment

I would like to express my sincere gratitude to my supervisor Dr. Zenon Chaczko for his support, continuous guidance, meticulous suggestions and his inexhaustible patience especially during preparation of this dissertation. At many stages in the course of this research, I benefited from his advice, particularly so when exploring new ideas.

Also I would like to thank Dr. Christopher Chiu, who as a good friend was always willing to help and give his best suggestions. Many thanks to Christopher Mcdermid for helping me through my case study design.

I will be thankful to my close friends for their friendly support and caring.

I express my personal love and appreciation to my mum, dad and my sister for constant motivation and support.

Finally, I would like to thank my husband Alireza, who was always there cheering me up and stood by me through the good times and the bad.

Abstract

Cloud Computing is a recent arrival to the world of IT infrastructure. The concept allows companies to maximise utilisation of their potentials and consequently boost their performance. One of the main benefits of Cloud Computing is the significant increase in efficiency of executing business plans. Additionally, Cloud Computing provides large-scale applications with powerful computing power across global locations. Yet Cloud users are able to share their data easily by using replication methodologies.

Cloud Computing structure has been developed based on a multi-tenancy concept. Therefore, availability and efficiency of the resources are important factors in the Cloud architecture. However, as the numbers of users are increasing rapidly, the load will have a significant impact on performance and operation of the Cloud systems. Accordingly, optimised load balancing algorithms that can manage the Cloud load in a time- and cost-efficient manner are required.

Much research in recent years has been dedicated to optimising load balancing in Cloud Computing. This optimisation is demonstrated through a balanced network of interacting resources. The goal of this network is to minimise the wait time and maximise utilisation of the throughput.

This thesis provides a set of solutions which mitigate the problem of load balancing in the Cloud. The dissertation investigates a novel class of heuristic scheduling algorithms that improves load balancing in workflow scheduling applications.

Furthermore, it proposes a new anticipatory replication methodology with the objective of improving data availability to enhance the load balancing between the Cloud sites.

In summary, this research innovation implicates the design of optimised load balancing algorithms that consider the magnitude and direction of the load in workflow applications. Furthermore, by architecting the anticipatory replication algorithm, it minimises the numbers of the replicas and enhances the effective network usage in Cloud-based systems.

Contents

I. Principal Theory and Concepts ... 2

Background ... 3

1.1 Introduction ... 3

1.2 Rational ... 4

1.3 Research Contribution ... 6

1.4 Research Questions ... 6

1.5 Formulation of the Hypothesis ... 8

1.5.1 Approach and Hypothesis Validation ... 9

1.5.1.1 Research Action Study ... 9

1.5.1.2 Experiments ... 9

1.5.2 Expected Outcomes ... 10

1.6 Outline of the Thesis ... 11

1.7 Related Publications ... 17

Literature Review ... 19

2.1 Cloud Computing ... 19

2.1.1 Cloud Characteristics ... 24

2.1.2 Cloud Taxonomy ... 25

2.1.3 Cloud Architecture ... 26

2.1.4 Cloud Stack ... 28

2.1.4.1 Infrastructure as a Service (IaaS) ... 28

2.1.4.1.1 Characteristics of IaaS ... 29

2.1.4.2 Platform as a Service (PaaS) ... 29

2.1.4.2.1 Characteristics of PaaS ... 30

2.1.4.3 Software as a Service (SaaS) ... 30

2.1.4.3.1 Characteristics of SaaS ... 30

2.2 Cloud Computing: Challenges and Benefits ... 31

2.3 Load Balancing in Cloud Computing ... 34

2.3.1 Load Balancing: Concept, Definition, Benefits, Challenges ... 34

2.3.2 Load Balancing Measurement Criteria ... 35

2.3.3 Load Balancing Methods 36

2.4 Focus of This Research 38

2.4.1 Load Balancing Algorithms: Independent vs Dependent 38

2.4.1.1 Dependent Load Balancing Algorithms 39

2.4.1.2 Independent Load Balancing Algorithms 45

2.5 The Quest of This Research 49

2.5.1 The Anticipatory Approach 49

2.5.2 Workflow Scheduling 51

2.5.3 Replication Strategy 53

2.5.4 Research Issues 55

2.5.4.1 Workflow Scheduling 55

2.5.4.2 Replication Methodology 55

2.6 Summary 58

Research Methodology 59

3.1 Research Design 59

3.2 Research Method 60

3.2.1 Research Rules 62

3.3 Modelling Approaches 62

3.3.1 Black-box Modelling Approaches 62

3.3.2 White-Box Modelling Approaches 63

3.3.3 Gray-Box Modelling Approach 63

3.4 Experimental Methodologies Overview 65

3.4.1 Anticipatory Approach 65

3.4.2 Spring Tensor Model 65

3.4.2.1 Mathematical Apparatus 66

3.4.3 Anticipatory Replication Methodology 69

3.5 Summary of the Chapter 70

Research Action Study 71

4.1 Introduction 71

4.2 Load Balancing 72

4.3 Presence 75

4.4 Cloud Metrics 77

4.5 The Case Study: HDM Load Monitoring 79

4.5.1 Load Monitoring Tool 81

4.5.2 HDM Load Tolerance 83

4.5.2.1 Testing Tool 83

4.5.2.2 Page Analyser 84

4.5.2.3 Load Test-10 Virtual Users 86

4.5.2.3.1 10 Users-Load time vs Clients Active 86

4.5.2.3.2 10 Users - Number of the Active Requests vs Clients Active 86

4.5.2.3.3 10 Users - HDM Content Type Distribution 87

4.5.2.3.4 10 Users HDM Content Load Distribution 87

4.5.2.4 Load Test - 20 Virtual Users 88

4.5.2.4.1 20 Users-Load Time Vs Clients Active 88

4.5.2.4.2 20 Users- Number of the Active Requests Vs Clients Active 88

4.5.2.4.3 20 Users - HDM Content Type Distribution 89

4.5.2.4.4 20 Users HDM Content Load Distribution 89

4.5.2.5 Load Test - 50 Virtual Users 90

4.5.2.5.1 50 Users-Load Time Vs Clients Active 90

4.5.2.5.2 50 Users -Number of the Active Requests vs Clients Active 90

4.5.2.5.3 50 Users- HDM Content Type Distribution 91

4.5.2.5.4 50 Users HDM Content Load Distribution 91

4.5.2.6 Load Test - 100 Virtual Users 92

4.5.2.6.1 100 Users-Load Time Vs Clients Active 92

4.5.2.6.2 100 Users- Number of the Active Requests Vs Clients Active 92

4.5.2.6.3 100 Users - HDM Content Type Distribution 93

4.5.2.6.4 100 Users HDM Content Load Distribution 93

4.5.2.7 Results Analysis 94

4.5.2.7.1 Average Load Time Results 94

4.5.2.7.2 HDM Content Type Distribution Results 94

4.5.2.7.3 HDM Content Load Distribution Results 96

4.6 Overall Analysis and Proposed Solution 97

4.7 Conclusion 98

II. Contribution to Research 99

STEM-PSO Based Task Scheduling Algorithm 100

5.1 Introduction 101

5.2 Load Balancing Problem Formulation 102

5.3 STEM-PSO Scheduling Algorithm 104

5.3.1 Particle Swarm Optimisation..................... 104

5.3.2 STEM Algorithm..................... 105

5.3.3 The Fundamentals of STEM-PSO..................... 111

5.4 Experiment Results 112

5.4.1 Total Execution Time..................... 113

5.4.2 CPU Load and CPU Time 114

5.4.3 Memory Rate..................... 115

5.4.3.1 Workflow Soft Error Rates..................... 116

5.5.4 Comparison of STEM-PSO Results..................... 117

5.5.5 Experiment Analysis 118

5.5 Conclusion..................... 119

Load balancing & Data Replication Strategy..................... 120

6.1 Introduction 120

6.2 Problem formulation 122

6.3 Proposed Methods..................... 123

6.3.1 SDDRC Architecture Design 123

6.3.2 SDDRC Algorithm..................... 125

6.4 Results and Analysis 131

6.4.1 Mean Job Execution Time..................... 133

6.4.2 Effective Network Usage 134

6.4.3 Total Number of the Replicas 135

6.4.3.1 Applying Soft Error Rates on Memory 136

6.4.4. Analysis of SDDRC Results..................... 137

6.4.5 Experiment Analysis 138

6.5 Conclusion..................... 139

Conclusion..................... 140

7.1 Review..................... 140

7.2 Thesis contribution 141

7.2.1 Discussion 141

7.2.2 Limitations 143

7.3 Future work 144

7.4 Final Remarks 146

III. Bibliography and publications 147

Bibliography 148

Appendix 175

9.1 Experiment Specifications 175

9.1.1 HDM System 175

9.1.1.1 Development specifications 176

9.1.2 STEM-PSO Implementation Details 179

9.1.3 Cloudsim Specification Details 183

9.1.3.1 Adding a new file in Tier structure 188

List of Figures

Figure 1.1- Thesis outline 14
Figure 2.1-Litriture review high-level mind map 20
Figure 2.2- IT evolution (KPMG 2011) 21
Figure 2.3- Comparing Cloud Computing, Cluster Computing and Grid Computing (Modified from Google trend) 22
Figure 2.4 -Cloud taxonomy 26
Figure 2.5 -Reference model of Cloud architecture, adopted from, (Buya et al. 2008) 27
Figure 2.6- Cloud stack categorisation, adopted from (Barkat et al. 2014) 28
Figure 2.7 -Load balancing categories 36
Figure 3.1 - Cloudsim architecture (Adopted from Calheiros 2011) 61
Figure 3.2 - Thesis modelling approaches (Adopted from DTU compute website) 64
Figure 3.3- Elastic sub-network model 67
Figure 4.1-Server availability model (Chaczko et al. 2014) 73
Figure 4.2- Hybrid load balancing model (Chaczko et al. 2014) 74
Figure 4.3- HDM System 79
Figure 4.4- Conceptual Architecture, Data Centric Model (Chaczko et al. 2011) 80
Figure 4.5- High-level HDM database design 81
Figure 4.6- HDM load balancing model (Chaczko et al. 2011) 82
Figure 4.7- System health monitoring UI 83
Figure 4.8- Load time vs clients active - 10 virtual users 86
Figure 4.9- Requests per second vs clients active – 10 virtual users 86
Figure 4.10- HDM content type distribution - 10 virtual users 87
Figure 4.11-HDM content load distribution - 10 virtual users 87
Figure 4.12- Load time vs clients active - 20 virtual users 88
Figure 4.13- Requests per second vs clients active – 20 virtual users 88
Figure 4.14- HDM content type distribution - 20 virtual users 89
Figure 4.15- HDM content load distribution -20 virtual users 89
Figure 4.16- Load time vs clients active - 50 virtual users 90
Figure 4.17- Requests per second vs clients active – 50 virtual users 90
Figure 4.18-HDM content type distribution - 50 virtual users 91
Figure 4.19- HDM content load distribution -50 virtual users 91
Figure 4.20-Load time vs clients active - 100 virtual users 92
Figure 4.21- Requests per second vs clients active – 100 virtual users 92
Figure 4.22- HDM content type distribution - 100 virtual users 93
Figure 4.23 - HDM content load distribution - 100 virtual users 93
Figure 5.1- Example of workflow modelling 103
Figure 5.2- STEM workflow parameters (Bond, angle, dihedral, non-local interaction) 107
Figure 5.3- Sample tasks allocations on (PC1-PC4) 112
Figure 5.4- Comparison of total execution time between STEM-PSO and HEFT 113
Figure 5.5- Comparison of total CPU Time between STeM-PSO and HEFT 114
Figure 5.6-Comparison of total Memory utilisation rate between STEM-PSO and HEFT 115
Figure 6.1– High level SDDRC system architecture 124

Figure 6.2- GRMS high-level architecture 125
Figure 6.3- GRMS Tier insertion 126
Figure 6.4-Creating the VM data inputs in Cloudsim 132
Figure 6.5 -Initiaing the VM creating at runtime 132
Figure 6.6- Mean job execution time 133
Figure 6.7- Effective network usage 134
Figure 6.8 -Total number of replications 135
Figure 9.1-HDM system high-level design 175
Figure 9.2- STEM-PSO high-level logical design 179
Figure 9.3-Pegasus workflow data generator 180
Figure 9.4- STEM-PSO front end 180
Figure 9.5- Hessian Evaluation Rate 181
Figure 9.6- PSO algorithm input -front end 181
Figure 9.7- STEM-PSO main class code 182
Figure 9.8- Cloudsim high-level design 183
Figure 9.9- Cloudsim class architectures 185
Figure 9.10-Cloudsim submission time frontend 187
Figure 9.11- Inserting a node in Tier structure 188
Figure 9.12- Updating the pointer place in Tier structure 189

List of Tables

Table 1.1- Thesis part I and part II structure 12
Table 2.1- Benefits of the Cloud Computing 32
Table 2.2 -Challenges in the Cloud Computing 33
Table 2.3- Summary of the reviewed dependent scheduling algorithms 44
Table 2.4- Summary of the reviewed independent algorithm 48
Table 2.5- Summary of the reviewed workflow scheduling algorithms 56
Table 2.6- Summary of the reviewed replications methodologies 57
Table 4.1- VM properties 85
Table 4.2- Summary of No. of clients vs average load time 94
Table 4.3-HDM content type distribution results – 10, 20,50, 100 virtual users 95
Table 4.4-HDM average content type load time distribution results 10,20,50,100 virtual users96
Table 4.5- Average user load time vs average html load time 97
Table 5.1- Interpretation of bond, angle, dihedral and non-local connections between tasks on workflow model (Aslanzadeh & Chaczko 2015). 108
Table 5.2 -CPU load utilisation rate using STeM-PSO 114
Table 5.3-Analysing the impact of soft error rates on Memory rate 116
Table 5.4- Total workflow execution time applications using STEM-PSO and HEFT 117
Table 5.5- CPU utilisation rate of workflow applications using STEM-PSO and HEFT 117
Table 5.6- Memory utilisation rate of workflow applications using STEM-PSO and HEFT .. 117
Table 6.1 -Analysing the impact of soft error rates on Memory rate 136
Table 6.2 - Mean job execution time using SDDRC (in milliseconds) 137
Table 6.3 - Effective network usage using SDDRC (in percentage) 137
Table 6.4 - Total number of replication using SDDRC 137

Glossary

AMAZON EC2	Amazon Elastic Cloud Computing
ANM	Anisotropic Network Model
DAG	Directed Acyclic Graph
DPSO	Discrete Version Of PSO
ENM	Elastic Network Models
FCFS	First Come First Served
FMOC	Finite Multi-Order Context
GA	Genetic Algorithm
GBEST	Global Best Position
GNM	Gaussian Network Model
GRMS	Global Replica Management System
HDM	Hospital Data Management
HEFT	Heterogeneous Earliest Finish Time
HTML	Hyper Text Mark-up Language
I/O	Input/Output
IAAS	Infrastructure As A Service
ICT	Information & Communication Technology
IOT	Internet Of Things
IT	Information Technology
LFU	Least Frequently Used
LRU	Least Recently Used
MAKESPAN	Total Length Of Schedule
MBIT/S	Megabit Per Second
NIST	National Institute Of Standards And Technology
NP PROBLEM	Non-Deterministic Polynomial Time
PAAS	Platform As A Service
PBEST	Best Local Position
PSO	Particle Swarm Optimisation
QOS	Quality Of Service
RC	Relative Cost

SAAS	Software As A Service
SDDRC	Smart Dynamic Data Replication In Cloud Computing
SHEFT	Scalable Heterogeneous Earliest Finish Time
SLA	Service Level Agreement
STEM	Generalised Spring Tensor Model
STEM-PSO	Generalised Spring Tensor Model- Particle Swarm Optimisation
URL	Uniform Resource Locator
VM	Virtual Machine
XML	Extensible Mark-up Language
XMPP	Extensible Messaging And Presence Protocol