

**AN ADAPTIVE QOS SCHEDULING ALGORITHM IN
SERVICE ORIENTED GRID AND CLOUD ENVIRONMENT**

ANG TAN FONG

**THESIS SUBMITTED IN FULFILMENT
OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY**

**FACULTY OF COMPUTER SCIENCE
&
INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR**

2011

ABSTRACT

The use of Grid and Cloud Computing for resource sharing has received tremendous attention in recent years. The merging of virtualization, utility computing and distributed computing with Service Oriented Architecture (SOA) has provided greater flexibility and ushered in a new paradigm of on-demand services. However, job scheduling remains a formidable challenge due to the dynamic and heterogeneous nature of Grid and Cloud Computing. Furthermore, the increasing and diverse end users requests raise new and greater urgencies to resolve the provisioning of Quality of Service (QoS).

This thesis proposes a Hybrid Scheduling Algorithm (HSA) with automatic deployment mechanism that maximizes resources utilization and minimizes total makespan. Subsequently, an Adaptive Scheduling Algorithm (ASA) that uses benchmarking is proposed to enhance the HSA. ASA is able to optimize the job scheduling performance over other approaches. Finally, Adaptive QoS Scheduling Algorithm (AQoSSA), an enhancement of ASA, is presented to meet the varied users QoS requirements. AQoSSA is able to maximize reliability and profit while guaranteeing the users' QoS requirements.

An experimental testbed is developed to evaluate the performances of all the proposed algorithms. The makespan results showed that the HSA and ASA outperformed the conventional MIN-MIN and MAX-MIN by between 1% - 10% and 5% - 17% respectively. Whereas, AQoSSA outperformed MIN-MIN QoS and MAX-MIN QoS by between 3% - 6% and 10% - 47% in terms of reliability and profit respectively while guaranteeing the users' QoS requirements.

ACKNOWLEDGEMENT

I would like to express my deep and sincere gratitude to my supervisor, Associate Professor Dr. Ling Teck Chaw for his guidance, support, continuous encouragement and valuable suggestions throughout the course of my research. Our weekly discussions in the research laboratory are very useful for the thesis.

A special thanks to University of Malaya and Ministry of Higher Education for the study leave and scholarship provided. In addition, special thanks to the staff of Faculty of Computer Science and Information Technology, University of Malaya for all the assistance rendered.

Last but not least, I would like to extend my heartfelt gratitude to my family. They have given me all the support and encouragement that I needed. I would never have done all this without their support.

This research is partially supported by E-science Fund under Ministry of Science, Technology & Innovation (MOSTI), Malaysia (01-01-03-SF0469).

TABLE OF CONTENTS

ABSTRACT	ii
ACKNOWLEDGEMENT	iii
TABLE OF CONTENTS	iv
LIST OF FIGURES	viii
LIST OF TABLES	x
LIST OF ABBREVIATIONS	xiii
CHAPTER 1 INTRODUCTION	1
1.1 Motivation	4
1.2 Objectives	5
1.3 Scope	5
1.4 Research Methodology	6
1.5 Thesis Organization	7
CHAPTER 2 PRELIMINARY	9
2.1 Grid Computing	9
2.1.1 Computational Grid	10
2.1.2 Data Grid	11
2.1.3 Grid Computing Architecture	11
2.1.4 Characteristics of Grid Computing	13
2.1.4.1 Heterogeneity	13
2.1.4.2 Dynamic Resource Performance	13
2.1.4.3 Sharing	14
2.1.4.4 Scalability	14
2.1.4.5 Adaptability	14
2.1.4.6 Transparency	14
2.1.4.7 Security	15
2.1.5 Problems of Grid Computing	15
2.1.6 Grid Evolution	16
2.1.6.1 Open Grid Services Architecture (OGSA)	17
2.1.6.2 Utility Computing	18
2.1.6.3 Software as Service (SaaS)	19

2.1.6.4	Virtualization.....	19
2.2	Cloud Computing.....	21
2.2.1	System Architecture.....	23
2.2.1.1	Infrastructure as Service (IaaS).....	24
2.2.1.2	Platform as Service (PaaS).....	26
2.2.1.3	Software as Service (SaaS).....	27
2.2.1.4	Benefits.....	28
2.2.2	Characteristics of Cloud Computing.....	29
2.2.2.1	Virtualization.....	30
2.2.2.2	Scalability.....	31
2.2.2.3	Accessibility.....	31
2.2.2.4	On-demand Service Provisioning.....	32
2.2.2.5	QoS Guaranteed.....	32
2.2.3	Web Services.....	33
2.2.3.1	XML.....	34
2.2.3.2	SOAP.....	35
2.2.3.3	WSDL.....	36
2.2.3.4	UDDI.....	37
2.2.3.5	REST.....	38
2.2.4	Service Oriented Architecture.....	39
2.2.4.1	Characteristics.....	40
2.2.4.2	Benefits.....	41
2.3	Research Opportunities.....	41
2.3.1	Job Scheduling.....	42
2.3.2	Quality of Service.....	42
2.3.3	Cost Management.....	43
2.4	Chapter Summary.....	43
CHAPTER 3 HYBRID SCHEDULING ALGORITHM (HSA) AND AUTOMATIC DEPLOYMENT.....		45
3.1	Grid and Cloud Scheduling Process.....	45
3.1.1	Resource Discovery.....	48

3.1.2	Resource Selection.....	50
3.1.3	Job Execution.....	50
3.1.4	Differences.....	53
3.2	Scheduling Algorithms.....	54
3.2.1	Structure of Scheduler	54
3.2.2	Job.....	56
3.2.2.1	Job Length	57
3.2.2.2	Communication Volume	57
3.2.2.3	Subtask Relationship.....	57
3.2.2.4	Hardware Requirement	59
3.2.2.5	Software Requirement.....	59
3.2.3	Time of Scheduling.....	59
3.2.3.1	Static Scheduling.....	59
3.2.3.2	Dynamic Scheduling	61
3.2.3.3	Hybrid Scheduling.....	62
3.2.4	Objective Function.....	63
3.2.5	Gap Analysis.....	65
3.3	Hybrid Scheduling Algorithm (HSA)	65
3.4	Automatic Deployment Mechanism.....	74
3.5	Experiment Setup	77
3.6	Results and Analysis	78
3.7	Chapter Summary.....	82
CHAPTER 4 ADAPTIVE SCHEDULING ALGORITHM (ASA)		85
4.1	Application Benchmarking	85
4.1.1	Related Works	87
4.1.2	Implementation of Application Benchmarking	89
4.2	Resource Benchmarking	94
4.2.1	Related Works	94
4.2.2	Implementation of Resource Benchmarking	96
4.3	Adaptive Scheduling Algorithm (ASA).....	100
4.4	Experiment Setup	103

4.5	Result and Analysis	105
4.6	Chapter Summary.....	109
CHAPTER 5 ADAPTIVE QoS SCHEDULING ALGORITHM (AQoSSA).....		111
5.1	QoS in Grid and Cloud Computing.....	111
5.1.1	Availability	112
5.1.2	Scalability	112
5.1.3	Deadline.....	113
5.1.4	Cost.....	113
5.1.5	Security.....	115
5.2	Service Level Agreement	116
5.3	Adaptive Quality of Service Scheduling Algorithm (AQoSSA).....	119
5.3.1	Profit Model.....	119
5.3.2	AQoSSA Components and Algorithms	123
5.4	Experiment Setup	125
5.5	Result and Analysis	126
5.6	Chapter Summary.....	130
CHAPTER 6 CONCLUSIONS AND FUTURE DIRECTIONS		132
6.1	Thesis Contributions	133
6.1.1	HSA with Automatic Deployment	133
6.1.2	ASA with Benchmarking.....	134
6.1.3	AQoSSA with Rescheduling	134
6.1.4	Grid and Cloud Testbed.....	135
6.1.5	Summary.....	136
6.2	Suggestions for Future Work	137
REFERENCE.....		139
APPENDIX A TEST CASES		148
A.1	Test Case 1: HSA and Automatic Deployment.....	148
A.2	Test Case 2: ASA with Benchmarking	154
A.3	Test Case 3: AQoSSA	162

LIST OF FIGURES

Figure 2.1: Layered Grid Computing architecture (adapted from Foster and Kesselman, 2004)	12
Figure 2.2: Principal elements of OGSA (adapted from Foster and Kesselman, 2004) ..	18
Figure 2.3: Types of hypervisor-based virtualizations (adapted from Sridhar, 2009)	20
Figure 2.4: Cloud Computing System Architecture (adapted from Vecchiola et al., 2009)	24
Figure 2.5: Cloud architecture related to Cloud services (adapted from Foster et al., 2008)	25
Figure 2.6: Relationship of Cloud Computing and other domains (adapted from Foster et al., 2008)	30
Figure 2.7: Communication between Web Services and UDDI (adapted from Diamadopoulou et al., 2008).....	38
Figure 3.1: A logical Grid scheduling architecture (adapted from Zhu, 2003).....	46
Figure 3.2: Grid scheduling process (adapted from Schopf, 2004).....	47
Figure 3.3: P2P network model (adapted from Ranjan et al., 2008).....	48
Figure 3.4: Centralized Scheduling (adapted from Hamscher et al., 2000)	55
Figure 3.5: Decentralized Scheduling (adapted from Hamscher et al., 2000).....	56
Figure 3.6: Directed Acyclic Graph (DAG)	58
Figure 3.7: System architecture of proposed algorithm	69
Figure 3.8: Makespan (seconds) of HSA for random job arrival and random parameter values using different z values	72
Figure 3.9: Proposed HSA.....	73
Figure 3.10: Main components of HSA and automatic deployment.....	74
Figure 3.11: Http.sys architecture	76
Figure 3.12: Makespan (seconds) of different algorithms for different experimental cases ..	81
Figure 4.1 Benchmark result: average execution time of service 1 running on resource A	92
Figure 4.2 Benchmark result: average execution time of service 1 running on resource B	93

Figure 4.3 Benchmark result: average execution time of service 1 running on resource C 93

Figure 4.4 Benchmark result: average execution time of service 1 running on different resources 98

Figure 4.5 Makespan (second) of HSA with and without benchmarking for different experimental cases 99

Figure 4.6 Main components of ASA 101

Figure 4.7 Proposed ASA..... 103

Figure 4.8 Makespan (seconds) of different algorithms with benchmarking for different experimental cases 107

Figure 5.1: Revenue vs. CPU Benchmark..... 121

Figure 5.2: Main components of AQoSSA 124

Figure 5.3: Proposed AQoSSA..... 125

Figure 5.4: Reliability (%) for different algorithms 127

Figure 5.5: Total profit of different algorithms 130

LIST OF TABLES

Table 3.1	Pricing of Amazon EC2 on demand instances (adapted from Amazon, 2010a)	64
Table 3.2	Makespan (seconds) of HSA for random job arrival and random parameter values using different z values	72
Table 3.3	A predefined set of resources and job parameters	78
Table 3.4	Makespan (seconds) of different algorithms for random job arrival and random parameter values	78
Table 3.5	Makespan (seconds) of different algorithms for random job arrival and big gap parameter values.....	79
Table 3.6	Makespan (seconds) of different algorithms for random job arrival and Poisson distribution parameter values	79
Table 3.7	Makespan (seconds) of different algorithms for different experimental cases and percentage of improvement.....	81
Table 4.1	Benchmark results: average execution time of service 1 running on resource A.....	91
Table 4.2	Benchmark results: average execution time of service 1 running on resource B.....	91
Table 4.3	Benchmark results: average execution time of service 1 running on resource C.....	91
Table 4.4	CPU mark for single CPU and multiple CPUs (PassMark, 2010).....	97
Table 4.5	Benchmark results: average execution time of service 1 running on different resources	98
Table 4.6	Makespan and percentage of improvement of HSA with benchmarking as compared to HSA with grouping for different test cases.....	100
Table 4.7	A predefined set of resources and job parameters	104
Table 4.8	Makespan (seconds) of different algorithms for random job arrival and random job length	105
Table 4.9	Makespan (seconds) of different algorithms for random job arrival and big gap job length.....	105

Table 4.10	Makespan (seconds) of different algorithms for random job arrival and job length in Poisson distribution.....	105
Table 4.11	Makespan (seconds) of different algorithms for Poisson arrival ($\lambda=50$) and random job length	106
Table 4.12	Makespan (seconds) of different algorithms for Poisson arrival ($\lambda=50$) and big gap job length.....	106
Table 4.13	Makespan (seconds) of different algorithms for Poisson arrival ($\lambda=50$) and job length in Poisson distribution ($\lambda_1=40000000, \lambda_2=4000, \lambda_3=50000$)	107
Table 4.14	Makespan (seconds) and percentage of improvement of ASA as compared to three different scheduling algorithms with benchmarking for different test cases	108
Table 5.1	Average Execution Time (second).....	119
Table 5.2	Revenue (unit).....	120
Table 5.3	Mean CPU Benchmark and relative value of CPU Benchmark from mean	121
Table 5.4	Modified Revenue (unit).....	122
Table 5.5	A predefined set of resources and job parameters	126
Table 5.6	Reliability (%) for different algorithms	127
Table 5.7	Reliability (%) for different algorithms and percentage of improvement of AQoSSA as compared to other algorithms	128
Table 5.8	Total profit (unit) for different algorithms	129
Table 5.9	Total profit of different algorithms and percentage of improvement of AQoSSA as compared to other algorithms	130
Table A.1	Resources Information	148
Table A.2	Tasks Information	149
Table A.3	Task scheduling with MIN-MIN (Parameter Value)	150
Table A.4	Task scheduling with MAX-MIN (Parameter Value).....	152
Table A.5	Task scheduling with HSA and Automatic Deployment	153
Table A.6	Resources Information	155
Table A.7	Tasks Information	155
Table A.8	Task scheduling with MIN-MIN (Benchmarking)	156
Table A.9	Task scheduling with MAX-MIN (Benchmarking).....	158

Table A.10	Task scheduling with HSA (Benchmarking)	159
Table A.11	Task scheduling with ASA (Benchmarking)	161
Table A.12	Resources Information	162
Table A.13	Tasks Information	163
Table A.14	Task scheduling with MIN-MIN (QoS).....	164
Table A.15	Task scheduling with MAX-MIN (QoS)	166
Table A.16	Task scheduling with AQoSSA (Rescheduling).....	168

LIST OF ABBREVIATIONS

Amazon EC2	Amazon Elastic Compute Cloud
API	Application Programming Interface
AQoSSA	Adaptive Quality of Service Scheduling Algorithm
ASA	Adaptive Scheduling Algorithm
CoS	Class of Service
DAG	Directed Acyclic Graph
DHT	Distributed Hash Table
GB	Gigabytes
GIS	Grid Information Service
HSA	Hybrid Scheduling Algorithm
HTTP	HyperText Transfer Protocol
IaaS	Infrastructure as a Service
IT	Information Technology
J2EE	Java 2 Platform Enterprise Edition
LAN	Local Area Network
MDS	Meta-computing Directory Service
MPI	Message Passing Interface
NGB	NAS Grid Benchmark
OGSA	Open Grid Services Architecture
OGSI	Open Grid Services Infrastructure
PaaS	Platform as a Service
QoS	Quality of Service

RAM	Random-Access Memory
REST	RE-presentational State Transfer
SaaS	Software as a Service
SDK	Software Development Kit
SIMD	Single Instruction Multiple Data
SLA	Service Level Agreement
SMP	Shared-memory Multi Processor
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SPEC	Standard Performance Evaluation Corporation
SSE	Streaming SIMD Extensions
SSH	Secure Shell
UDDI	Universal Description, Discovery and Integration
UI	User Interface
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name
VM	Virtual Machine
VO	Virtual Organization
WAN	Wide Area Network
WSDL	Web Services Description Language
WWW	World Wide Web
XML	Extensible Mark-up Language