

**GUARANTEEING SERVICE LEVEL AGREEMENTS
FOR TRIANGLE COUNTING VIA OBSERVATION-
BASED ADMISSION CONTROL ALGORITHM**

Weerakkody Arachchige Chinthaka Rukshan Weerakkody

(189356P)

Thesis submitted in partial fulfillment of the requirements for the degree
Master of Science in Computer Science

Department of Computer Science and Engineering
Faculty of Engineering

University of Moratuwa
Sri Lanka

October 2022

DECLARATION

I declare that this is my own work and this dissertation does not incorporate without acknowledgement any material previously submitted for degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature:

Date:.....

Name: Chinthaka Weerakkody

The above candidate has carried out research for the Masters dissertation under my supervision.

Signature of the supervisor:

Date:

Name: Prof. Sanath Jayasena

Signature of the supervisor:

Date:

Name: Dr. Miyuru Dayarathna

ACKNOWLEDGEMENT

My sincere appreciation goes to my family for the continuous support and motivation given to manage my MSc research work. I also express my heartfelt gratitude to Prof. Sanath Jayasena and Dr. Miyuru Dayarathna, my supervisors, for the supervision and advice given throughout this research period. I'm eternally grateful for Prof. Toyotaro Suzumura for helping us by providing necessary remote servers for the project. I am also thankful to WSO2. Telco (Pvt) Ltd, Virtusa (Pvt) Ltd, Macrometa and WSO2 Lanka (Pvt) Ltd for providing support to complete this project.

ABSTRACT

Increasingly large graph processing applications adopt the approach of partitioning and then distributed processing. However, maintaining guaranteed Service Level Agreement (SLA) on distributed graph processing for concurrent query execution is challenging because graph processing by nature is an unbalanced problem. We investigate on maintaining predefined service level agreements for commonly found graph processing workload mixtures. We develop a Graph Query Scheduler Mechanism (GQSM) which maintains a guaranteed service level agreement in terms of overall latency.

The proposed GQSM model is implemented using the queueing theory. Main component of GQSM is a job scheduler which is responsible for listening to an incoming job queue and scheduling the jobs received. The proposed model has a calibration phase where the Service Level Agreement data, load average curve data, and maximum load average which can be handled by the hosts participating in the cluster without violating SLA is captured for the graphs in the system. After completing the calibration phase the job scheduler is capable of predicting the load average curve for the incoming job requests. The scheduler checks whether the maximum load average extracted from the predicted load average curve exceeds the load average threshold values captured in the calibration phase. Based on the result the job scheduler accepts or rejects the job requests received.

Results show that SLA is successfully maintained when the total number of users is less than 6 in a JasmineGraph cluster deployed in a single host. For distributed clusters the number of users can go up to 10 without violating SLA. The proposed model is scalable and it can be applied to a distributed environment as well.

As future work, the proposed model can be extended to work with less initial calibration steps and the scheduling algorithm can be improved with intelligent workload management among hosts for more efficient resource consumption.

TABLE OF CONTENTS

DECLARATION	i
ACKNOWLEDGEMENT	ii
ABSTRACT	iii
TABLE OF CONTENT	iv
LIST OF FIGURES	vi
LIST OF TABLES	viii
LIST OF ABBREVIATIONS	ix
LIST OF APPENDICES	x
CHAPTER 1: INTRODUCTION	1
1.1 Introduction	1
1.2 Research Problem	2
1.3 Motivation	2
1.4 Aim and Objectives	3
CHAPTER 2: LITERATURE REVIEW	4
2.1 Distributed Graph Database Servers	4
2.2 Elastic Scaling	12
2.3 VM Load Balancing	15
CHAPTER 3: SOLUTION VIA GRAPH QUERY SCHEDULING MECHANISM	17
3.1 Methodology	17
3.1.1 Job Classification	17
3.1.2 Calibration	18
3.1.3 GQSM Model	21
3.2 Data Collection Model	28
CHAPTER 4: IMPLEMENTATION OF GQSM	31
4.1 Job Scheduler Implementation	31
4.2 Performance Statistic Collection	33
CHAPTER 5: EVALUATION	37
5.1 Experimental Setup	37
5.1.1 Computing Infrastructure Setup	37
5.1.2 Resource and Workload Combinations	38
5.1.3 Data Collection	42

5.2 Model Evaluation	47
5.2.1 GQSM Parameter Evaluation	47
5.2.2 GQSM Evaluation in Single Server Cluster	50
5.2.3 GQSM Evaluation in Distributed Cluster	54
5.2.4 Load Average Curve Data Prediction	61
5.2.5 GQSM Auto Calibration	62
5.3 Results Discussion	64
CHAPTER 6: CONCLUSION	66
6.1 Summary	66
6.2 Limitations	67
6.3 Future Work	67
References	69
Appendix - A: Load Average Curves for Epinion, Google and Astro-Physics Graphs for Cluster 1	73
Appendix - B: Job Scheduler Implementation	74
Appendix - C: Performance Statistics Collection	75

LIST OF FIGURES

Figure	Description	Page
Figure 2.1	Traditional vs Traffic and network aware vertex cut algorithms	5
Figure 2.2	H-load approach overview	6
Figure 2.3	Correspondence between bisections and the partition sketch for the process of partitioning the graph	7
Figure 2.4	Acacia System Architecture	8
Figure 2.5	Overview of Acacia	8
Figure 2.6	Components of Acacia	9
Figure 2.7	System Architecture of JasmineGraph	10
Figure 2.8	ESM operation	14
Figure 3.1	Flow chart for auto calibration process	20
Figure 3.2	Graph Query Scheduling Mechanism (GQSM)	22
Figure 3.3	Job Request Pickup and Scheduling by Job Scheduler	23
Figure 3.4	Job Scheduler High Level Architecture	24
Figure 3.5	Flow Chart of the Scheduler	25
Figure 3.6	Deriving aggregate curve for two high priority jobs	26
Figure 4.1	New JasmineGraph architecture	34
Figure 4.2	Flow chart for job calibration	35
Figure 5.1	JasmineGraph dockerized environment	38
Figure 5.2	Master worker distribution of cluster 1	39
Figure 5.3	Master worker distribution of cluster 2	40
Figure 5.4	Master worker distribution of cluster 3	40
Figure 5.5	Master worker distribution of cluster 4	41
Figure 5.6	Load average curve for Youtube in cluster 1	43
Figure 5.7	Expected load average curve	45
Figure 5.8	Expected SLA vs actual response time	46
Figure 5.9	Load average curves for multiple iterations-cluster 1	48
Figure 5.10	Load average curves for multiple iterations-cluster 2	49

Figure	Description	Page
Figure 5.11	Predicted and actual work load curves for Epinions 2 users and 4 users	51
Figure 5.12	Predicted and actual work load curves for Epinions 8 users and 16 users	52
Figure 5.13	Average response time and guaranteed SLA information for Epinion	53
Figure 5.14	Calibrated load average curve for Web Google in cluster 2	54
Figure 5.15	Predicted and actual workload curves for Web Google in cluster 2 server 1 for 4 users and 8 users	55
Figure 5.16	Predicted and actual workload curves for Web Google in cluster 2 server 1 for 12 users and 16 users	56
Figure 5.17	Predicted and actual workload curves for Web Google in cluster 2 server 2 for 4 users and 8 users	57
Figure 5.18	Predicted and actual workload curves for Web Google in cluster 2 server 2 for 8 users and 16 users	58
Figure 5.19	Average response time and guaranteed SLA information for Web Google in cluster 2 server 1	59
Figure 5.20	Average response time and guaranteed SLA information for Web Google in cluster 2 server 2	60
Figure 5.21	Base amplitude adjustment curve for epinion graph in cluster 1	61
Figure 5.22	Comparison of derived curve, derived amplitude adjusted curve and expected curve for epinions graph for 6 users in cluster 1	62
Figure 5.23	Obtained graphs during auto calibration process of Epinion in cluster 1	63
Figure 5.24	Obtained graphs during the auto calibration process of Epinion in cluster 1 with two youtube calibrated graphs	64

LIST OF TABLES

Table	Description	Page
Table 2.1	Comparison Between GraphH and Acacia	12
Table 3.1	Deriving Aggregated Load Average Curve	26
Table 3.2	Experiment Results Capturing Format	30
Table 5.1	Combination of hardware	39
Table 5.2	JasmineGraph cluster setup	39
Table 5.3	List of graph data sets for experiments	42
Table 5.4	SLA values for graphs	42
Table 5.5	SLA upper bounds (Guaranteed SLAs) for graphs	44
Table 5.6	Collected maximum load average, guaranteed SLA and actual response time data	45
Table 5.7	Load average thresholds for cluster1	46
Table 5.8	Response times and maximum load average data for Epinion	53
Table 5.9	Load average thresholds for Web Google in cluster 2	54
Table 5.10	Average response times and maximum load averages for Web Google in cluster 2	59

LIST OF ABBREVIATIONS

Abbreviation	Description
SLA	Service Level Agreement
CPU	Central Processing Unit
VM	Virtual Machine
HM	Host Machine
GPU	Graphics Processing Unit
GQSM	Graph Query Scheduling Mechanism
ESM	Elastic Switching Mechanism

LIST OF APPENDICES

Appendix	Description	Page
Appendix - A	Load Average Curves for Epinion, Google and Astro-Physics Graphs for Cluster 1	73
Appendix - B	Job Scheduler Implementation	74
Appendix - C	Performance Statistics Collection	75