

Design, Analysis, and Simulation of Multiple Divisible Load Scheduling Strategies in Distributed Computing Networks

Project ID: RP3982732 / R-263-000- 073-112

Principle Investigator: Bharadwaj Veeravalli¹

Collaborator: ²Professor Ko Chi Chung

¹ *Open Source Software Laboratory* (<http://opensource.nus.edu.sg>)

² *Digital Systems and Applications Laboratory*

Department of Electrical and Computer Engineering

The National University of Singapore,

10 Kent Ridge Crescent, Singapore 119260

Email : {elebv,elekoc}@nus.edu.sg

¹Corresponding author

Acknowledgements

Dr. Bharadwaj Veeravalli and Prof. Ko Chi Chung deeply appreciate and thank the funding support for this project from the Department of ECE, NUS. *Special thanks* goes to Prof. Daniel S H Chan (HoD in the year 1998) and his research committee members for approving this project in the first place. The success of the project is also shared by the following skilful researchers who showed tremendous sustained interest in rendering a continuous support in all possible ways. The principal investigator (PI) would like to thank Prof. T. G. Robertazzi, SUNY Stony Brook, USA, and Dr. Jeeho Sohn, AT&T Bell Labs, USA, for rendering their test data and results presented in [32] and in Section 12.3 of [9], for conducting comparative studies with his strategy in Chapter 3 of this report. PI would also like to thank Dr. Gerassimos Barlas, American University of Sharjah, for his simulation results for Chapter 3 and Prof. Debasish Ghose, Indian Institute of Science, Bangalore, India, for his support in Chapter 4. Both the investigators would like to thank all the students contributed to this project via final year bachelor's projects (Mr. Chan Sze Kong, 1999, Lui Khai Kok, 2000), and PhD thesis (Dr. Xiaolin Li). Lastly, the principal investigator appreciates all the help obtained from international referees during the process of publishing most of the contributions in this report in high-quality journals.

Dr. Bharadwaj renders a special and a BIG THANK YOU to his little daughter *Tanya Veeravalli*, for turning 180 degrees upside a sheet of paper with a timing diagram kept on a table before a mirror in his living room, during her peak-prank hours! The reflection of the timing diagram in the mirror now figures in as a **first ever work** in the domain of long-duration movie retrieval on distributed networks employing multiple servers strategy, which is subsequently published in *Multimedia Tools and Applications* journal, in a special issue, in the year 2000. In fact, this is the first attempt in the domain of multimedia to use the results of divisible load theory (DLT) to significantly improve the performance.

Finally, the PI renders his sincere thanks to Mr. Lin Wujian, his PhD student, in helping him to fix the numbering of figures and references, in order, and for proof-reading on the whole, while preparing this report.

Abstract

In this project, our objective is to minimize the total processing time of several divisible jobs arriving at any computer site in a distributed computing system (DCS). Divisible loads are loads characterized by a large data volume and demands very high intensive CPU times for computations. Typical examples include, satellite image processing, low-level pixel processing, large volume data fusion and data processing for military surveillance applications, long-duration multimedia data retrieval over delay-sensitive networks, etc, to quote a few. We consider a Ethernet (bus) network that has a single scheduler/bus control unit (BCU) that schedules and disseminates the load to the processors in the system. We first propose a revised affine model, for computation and communication events, to the theory of divisible load scheduling and demonstrate its influence in scheduling divisible loads. We revisit the existing model by considering the impact of overheads involved in computation and communication. These overheads do exist in real-life situations. Although some studies considered these overheads, there does not exist any supporting analysis to quantify the influence of these overheads on the performance rigorously. We extend the analysis and propose integer approximation algorithms that can be directly used for real-life applications. For Ethernet-like networks, we perform an exact analysis of the concept of sequencing amidst the above-mentioned overheads. We then design, analyze and conduct rigorous simulation experiments of multiple loads scheduling using multi-installment strategies and also propose certain heuristics whenever optimal solution ceases to exist. Finally, we attempt rigorous theoretical and experimental verifications on several real-life problems - large scale matrix-vector products on PC clusters, Edge-detection application for image processing, and long-duration multimedia movie retrieval. We provide several illustrative examples throughout this report for the ease of understanding.

Contents

List of Figures	6
1 Introduction	1
1.0.1 Research contributions	2
1.0.2 Related work	3
2 Load Distribution Strategy in Bus Networks	5
2.0.3 Definitions and some remarks	5
2.0.4 Closed-form Solutions for the Processing Time	7
2.0.5 Load sharing conditions	8
2.0.6 Concept of equivalent processor	9
2.0.7 Homogeneous systems	11
2.1 Processing Time Minimization Using the Concept of Sequencing	12
2.1.1 Motivating example	12
2.1.2 Optimal sequencing	14
2.1.3 Discussions on simulation results	16
2.2 Integer approximation algorithms	19
2.2.1 Performance bounds	21
2.2.2 Constrained Partitioning	23
2.2.3 Sub-Optimal Processing Time Solutions	27
2.2.4 Algorithm IIA	30
3 Processing Multiple Loads on Bus Networks	38
3.0.5 Definitions and notations	38
3.0.6 Optimal Processing Strategies	39
3.0.7 Heuristic strategies	44
3.0.8 Simulation Results and Discussions	46

4	Experimental Verification	57
4.0.9	Large size matrix-vector product computations	57
4.0.10	Divisible data and the test-bed	57
4.0.11	Measurement of speed parameters	58
4.0.12	Effect of solution back-propagation	60
4.0.13	Software development system support	61
4.0.14	Discussion of Results	63
4.1	Edge-Detection application in image processing: Implementation on PVM Platform	66
4.1.1	Extensions to the case of finite-size buffers	69
4.2	Long-duration Movie Retrieval: Multiple-server retrieval strategy . . .	70
5	Conclusions	86