Telecommunication Networks and Computer Systems

Series Editors

Mario Gerla
Aurel Lazar
Paul Kühn
Hideaki Takagi

Springer
London
Berlin
Heidelberg
New York
Barcelona
Budapest
Hong Kong
Milan
Paris
Santa Clara
Singapore
Tokyo

Other titles published in this Series:

Teletraffic: Theory and Applications
Haruo Akimaru and Konosuke Kawashima

Scrambling Techniques for Digital Transmission
Byeong Gi Lee and Seok Chang Kim

Multiservice Loss Models for Broadband Telecommunication Networks
Keith W. Ross

Metropolitan Area Networks
Marco Conti, Enrico Gregori and Luciano Lenzini

Hisao Kameda, Jie Li, Chonggun Kim and
Yongbing Zhang

Optimal Load Balancing in Distributed Computer Systems

With 49 Figures

Springer

Hisao Kameda
Jie Li
Institute of Information Sciences
and Electronics,
University of Tsukuba,
1-1-1 Tennoudai, Tsukuba-shi,
Ibaraki 305, Japan

Chonggun Kim
Department of Computer Engineering,
Yeungnam University,
Gyongsan, Kyongbuk, 712-749, Korea

Yongbing Zhang
Institute of Policy and Planning Sciences,
University of Tsukuba,
1-1-1 Tennoudai, Tsukuba-shi,
Ibaraki 305, Japan

Series Editors

Mario Gerla
Department of Computer Science,
University of California,
Los Angeles,
CA 90024, USA

Paul Kühn
Institute of Communications
Switching and Data Technics,
University of Stuttgart,
D-70174 Stuttgart, Germany

Aurel Lazar
Department of Electrical Engineering and
Center for Telecommunications Research,
Columbia University,
New York, NY 10027, USA

Hideaki Takagi
Institute of Policy and Planning Sciences,
University of Tsukuba,
1-1-1 Tennoudai, Tsukuba-shi,
Ibaraki 305, Japan

British Library Cataloguing in Publication Data
Optimal load balancing in distributed computer systems.
(Telecommunication networks and computer systems)
1.Electronic data processing - Distributed processing
I.Kameda, Hisao
004.3'6

ISBN-13: 978-1-4471-1246-4 e-ISBN-13: 978-1-4471-0969-3
DOI: 10.1007/978-1-4471-0969-3

Library of Congress Cataloging-in-Publication Data
A catalog record for this book is available from the Library of Congress

Softcover reprint of the hardcover 1st edition 1997

Typesetting: Camera ready by authors

69/3830-543210 Printed on acid-free paper

Preface

A recent trend in computer systems is to distribute computation among several physical processors. There are basically two schemes for building such systems. One is a *tightly coupled* system and the other is a *loosely coupled* system. In the tightly coupled system, the processors share memory and a clock. In these *multiprocessor* systems, communication usually takes place through the shared memory.

In the loosely coupled system, the processors do not share memory or a clock. Instead, each processor has its own local memory. The processors communicate with each other through various communication lines, such as high-speed buses or telephone lines. These systems are usually referred to as *distributed* computer systems.

The processors in distributed computer systems may vary in size and function. They may include small microprocessors, workstations, minicomputers, and large general-purpose computer systems. These processors are referred to by a number of different names such as *sites*, *nodes*, *computers*, and so on, depending on the context in which they are mentioned. The term *node* is mainly used through this book.

Distributed computer system research includes many areas, including: communications networks, distributed operating systems, distributed databases, concurrent and distributed programming languages, theory of parallel and distributed algorithms, interconnection structures, fault tolerant and ultrareliable systems, distributed real-time systems, distributed debugging, and distributed applications [Sta84]. Therefore, a distributed computer system consists of the physical networks, nodes, and all the controlling software.

There are five major reasons for building distributed computer systems: *resource sharing*, *performance improvement*, *reliability*, *communication*, and *extensibility*. We will briefly elaborate on each of them.

(1) Resource Sharing: If a number of different nodes are connected to each other, then a user at one node may be able to use the resources available at another. For example, a user at node A may be using a laser printer that is provided by node B. Meanwhile, a user in B may access a file that resides at A. In general, resource sharing in a distributed computer system provides mechanisms for sharing files at remote nodes, processing information in a distributed database, printing files at remote nodes, using

remote specialized hardware devices, and other operations.

(2) Performance Improvement: If a particular computation can be partitioned into a number of subcomputations that can run concurrently, then the availability of a distributed computer system may allow us to distribute the computation among the various nodes, to run it concurrently. In addition, if a particular node is currently overloaded with jobs, some of them may be moved to other, lightly loaded nodes. This movement of jobs is called *load balancing* .

(3) Reliability: If one node fails in a distributed computer system, the remaining nodes can potentially continue operating. If the system is composed of a number of large general-purpose computers, the failure of one of them should not affect the rest. In general, if enough redundancy exists in the system in both hardware and software, the system can continue with its operation, even if some of its nodes have failed.

(4) Communication: When a number of nodes are connected to each other via a communications network, the users at different nodes have the opportunity to exchange information. In a distributed system we refer to such activity as *electronic mail.* The similarity between an electronic mail system, which is an application visible to the users, and a message system, which is intended for communication between processes, should be apparent.

(5) Extensibility: This is the ability to easily adapt to both short and long term changes without significant disruption of the system. Short term changes include varying workloads and subnet traffic, and host or subnet failures or additions. Long term changes are associated with major modifications to the requirements of the system.

One of the most interesting problems in distributed computer systems is improving the performance of the system by balancing the load among nodes. In this book, we study load balancing problems.

Load balancing may be either static or dynamic. Static load balancing strategies are generally based on the information about the average behavior of system; transfer decisions are independent of the actual current system state. Dynamic strategies, on the other hand, react to the actual current system state in making transfer decisions. It seems that dynamic strategies are more effective than static strategies whereas the former may have more overhead than the latter. Furthermore, it seems that there currently exists no optimal dynamic strategy that is sophisticated enough to be applicable generally to various environments and analytically tractable.

This book investigates mainly the optimal static load balancing in distributed computer systems. The results of optimal static load balancing might be helpful in designing a distributed computer system or making parametric adjustments to improve the system performance.

In this book, the optimal load balancing in five important network configurations are considered. These network configurations are single channel network configurations, star network configurations with one-way traffic, tree hierarchy network configurations, star network configurations with two-way traffic, and tree network configurations with two-way traffic.

The single channel network configuration is common in distributed computer systems. In the network configurations, host computers share the same communication media (e.g., bus) for communication.

In the star network configuration with one-way traffic, there are many satellite nodes (stations) and a central node (hub). All satellite nodes are connected to the central node by communication lines. A job externally arriving at a satellite node can be processed locally or be forwarded to the central node for processing. A job arriving at a central node, however, can only be processed locally. That is, there is only one way traffic from satellite nodes to the central node in this network configuration.

The tree hierarchy network configuration is an extension of the star network with one-way traffic. Notice that many of distributed computer communication networks naturally have the tree hierarchy network configurations. That is, we often have tree organizations of computing machines. For example, a super–power center machine is connected with remote station mainframe computers each of which is, in turn, connected with server-type workstations each of which is connected with client-type workstations and personal computers, etc. In a tree hierarchy network, all nodes are connected in a hierarchy structure. We assume that a job arriving at a node can only be processed locally or be forwarded to a node in a higher layer for processing in a tree hierarchy network. This assumption is reasonable if the processing capacities of the nodes in the higher layer of a node i are larger than that of node i, and the processing capacities of the nodes in the lower layer are smaller than that of node i.

In the same as the star network configuration with one-way traffic, there are many satellite nodes (stations) and a central node in a star network with two-way traffic. A job externally arriving at the central node, however, can be forwarded to a satellite node for processing. In this network configuration, the central node can be taken as a switch so that a job arriving at a node (a satellite node or the central node) can be sent to any nodes in the network for processing.

In tree network configurations with two-way traffic, there are no hierarchical structures. Every node is connected to one of its neighbor nodes by one and only one communication line. Since there is two-way traffic between any two nodes (host computers) in this configuration, a job arriving at a node can be processed locally or be forwarded to any other nodes in a network for processing.

Furthermore, two types of load balancing policies each of which has distinct performance objective are considered. One is the *overall* optimal policy and the other is the *individually* optimal policy. The overall optimal policy whose goal is to minimize the system-wide mean job response time. The solution is referred as the *optimum*. The individually optimal load balancing, on the other hand, is the policy whereby each job is scheduled so as to minimize its own expected job response time, given the expected node and communication delays. The solution of the individually optimal load balancing is referred as the *equilibrium*.

Chapter 1 studies the optimal load balancing in single channel and one-way traffic star network configurations. At first, two effective load balancing algorithms are presented for the single channel and one-way traffic star network configurations one for each. For the multi-class job environment of the single channel network configurations, the model of the distributed system is also presented. On the basis of the model, the load balancing problem is formulated as a nonlinear optimization problem. An effective load balancing for the multi-class job environment is also presented.

Chapter 2 provides comparative study on the overall optimal load balancing and individually optimal load balancing policies in single channel and one-way traffic star network configurations by parameter analysis in detail.

Three important parameters in distributed computer systems can be considered: the communication time of the network, the processing capacity of each node, and the job arrival rate of each node. The effects of the three parameters on the behavior of the systems in the overall optimal policy and in the individually optimal policy is studied for the multi-class job environment through numerical experimentation.

Chapter 3 presents optimal load balancing in tree hierarchy network configurations. We derive theorems which give the necessary and sufficient conditions for the optimal load balancing to the tree hierarchy problem. It is proven that the tree hierarchy optimization problem can be solved by solving much simpler star sub–optimization problem iteratively. A decomposition algorithm to solve the optimal static load balancing problem in tree hierarchy network configurations is presented. Furthermore, the effects of link communication time and node processing time on optimal load balancing in tree hierarchy network configurations are studied. Clear and simple analytical results are presented.

In chapter 4, optimal load balancing in star network configurations with two-way traffic is presented. It is proven that in the optimal solution the satellite nodes of the star network are classified into the following categories: the idle source nodes, the active source nodes, the neutral nodes, and the sink nodes. The necessary and sufficient conditions for optimal solution are studied, and an effective $O(n)$ algorithm is proposed to solve the optimization problem for an n-satellite system. Furthermore, the effects of link communication time on optimal load balancing in a star network are studied analytically.

As an important extension the work on load balancing in star network configurations with two-way traffic, chapter 5 contains the study of optimal load balancing in tree networks with two-way traffic. It is demonstrated that the optimization problem of the particular structure can be solved effectively by a decomposition technique. An effective algorithm is developed for obtaining the optimal solution.

In finding the optimal load balancing policies, the study of the uniqueness of the solution is important. Chapter 6 provides the necessary and sufficient conditions of the uniqueness for a certain class of the static optimal load balancing problems.

As we have noted above, there are two kinds of load balancing: static and dynamic. The dynamic load balancing is also an active research field. Although this book mainly studies the static load balancing problems in distributed computer systems, a survey of studies on dynamic load balancing is provided in chapter 7.

Chapter 8 provides a comparative study of static and dynamic load balancing. Dynamic load balancing policies offer the possibility of improving load distribution at the expense of additional communication and processing overhead. The overhead of dynamic load balancing may be large, in particular for a large heterogeneous distributed system. Comparing the performance provided by static load balancing policies and dynamic load balancing policies, it shows that the static load balancing policies are preferable when the system loads are light and moderate or when the overhead is not negligible high.

Acknowledgements

We have been fortunate in having the help of many people during our work in The University of Electro-Communications and University of Tsukuba. Without their help, it would be impossible to finish this work. We would like to express our special gratitude to Professor Hideaki Takagi for his kindly introducing us the opportunity to publish our work in the Springer-Verlag London Limited series, Telecommunications Networks and Computer Systems (TNCS). We are greatly indebted to the support of our publisher, Spring-Verlag London Limited.

Credits

We are grateful to the copyright holders of the following articles for permitting us to use material contained within them.

C. Kim and H. Kameda. Optimal static load balancing of multi-class jobs in a distributed computer system. In *10th IEEE International Conference on Distributed Computing Systems*, pages 562–569, Paris, France, June 1990. Reprinted with permission of IEEE.

C. Kim and H. Kameda. Optimal static load balancing of multi-class jobs in a distributed computer system. *The Transactions of the IEICE*, E73(7):1207–1214, July 1990. Reprinted with permission of IEICE.

C. Kim and H. Kameda. An algorithm for optimal static load balancing in distributed computer systems. *IEEE Trans. on Computers*, 41(3):381–384, March 1992. permission of IEEE.

C. Kim and H. Kameda. Parametric analysis of static load balancing of multi-class jobs in a distributed computer system. *The Transactions of the IEICE*, E75-D(4):527–534, July 1992. Reprinted with permission of IEICE.

H. Kameda and Y. Zhang. Uniqueness of the solution for optimal static routing in open bcmp queueing networks. *Mathl. Comput. Modelling*, 22:119–130, 1995. Reprinted with permission of Elsevier Science Ltd.

J. Li and H. Kameda. Optimal static load balancing in tree hierarchy network configurations. In *Proc. of the 10th IASTED International Conference, Applied Informatics*, pages 13–16, Innsbruck, Austria, February 10 - 12 1992. Acta Press. Reprinted with permission of IASTED.

J. Li and H. Kameda. Effects of node processing time on optimal load balancing in tree hierarchy network configurations. *Computer Communications*, 16(12):781–793, 1993. Reprinted with permission of Elesevier Science B.V.

J. Li and H. Kameda. Optimal load balancing in tree networks with two-way traffic. *Computer Networks and ISDN Systems*, 25(12):1335–1348, 1993. Reprinted with permission of Elsevier Science Publishers B.V.

J. Li and H. Kameda. A decomposition algorithm for optimal static load balancing in tree hierarchy network configurations. *IEEE Trans. on Parallel and Distributed Systems*, 5(5):540–548, 12 1994. Reprinted with permission of IEEE.

J. Li and H. Kameda. Optimal load balancing in star network configurations with two-way traffic. *Journal of Parallel and Distributed Computing*, 23(3):364–375, 12 1994. Reprinted with permission of Academic Press, Inc.

J. Li, H. Kameda, and K. Shimizu. Effects of link communication time on optimal load balancing in tree hierarchy network configurations. *IEICE Trans. on Information and Systems*, E76-D(2):199–209, 1993. Reprinted with permission of IEICE.

Y. Zhang, K. Hakozaki, H. Kameda, and K. Shimizu. A performance comparison of adaptive and static load balancing in heterogeneous distributed systems. In *Proc. 28th Annual Simulation Symposium, IEEE Computer Society Press*, pages 332–340, Phoenix, A.Z., 1995. Reprinted with permission of IEEE.

Y. Zhang, H. Kameda, and K. Shimizu. Parametric analysis of static load balancing in star network configurations. *Transactions IEICE*, J74-D-I(9):644–655, 1991. Reprinted

with permission of IEICE.

Y. Zhang, H. Kameda, and K. Shimizu. A comparison of adaptive and static load balancing strategies by using simulation methods. In *Proc. 1992 Singapore Int. Conf. Intelligent Control and Instrumentation, IEEE Singapore Section*, pages 1162–1167, Singapore, February 1992. Reprinted with permission of IEEE.

Y. Zhang, H. Kameda, and K. Shimizu. Parametric analysis of optimal static load balancing in distributed computer systems. *Journal of Information Processing*, 14(4):433–441, 1992. Reprinted with permission of the Informatin Processing Society of Japan.

Y. Zhang, H. Kameda, and K. Shimizu. Adaptive bidding job scheduling in heterogeneous distributed systems. In *Proc. 1994 Int. Workshop Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, IEEE Computer Society Press*, pages 250–254, Durham, N.C., 1994. Reprinted with permission of IEEE.

Contents

List of Tables

List of Figures

Chapter 1

Load Balancing in Single Channel and Star Network Configurations

1.1 Introduction

In this chapter, the static load balancing in single channel and star network configurations is studied. Tantawi and Towsley [TT84, TT85] studied load balancing of single class jobs in a distributed computer system that consists of a set of heterogeneous host computers connected by single channel and star communications networks. They considered an optimal static load balancing strategy which determines the optimal load at each host so as to minimize the mean job response time, and derived two algorithms (called single-point algorithms) that determines the optimal load at each host for given system parameters. Static load balancing may be useful for system sizing (e.g., allocation of resources, identification of bottleneck, sensitivity studies, etc.). The solution of optimal static load balancing may help us design the system.

First, the same model for sigle channel network configurations as Tantawi and Towsley is considered. Some additional properties that the optimal solution satisfies are derived. On the basis of these properties, another single-point algorithm that seems more easily understandable and more straightforward than that of Tantawi and Towsley is derived. The performance of the derived algorithm is compared with that of Tantawi and Towsley (T & T). The number of program steps for implementing the new algorithm is about a third of that of the T & T algorithm. During the course of numerical experiment, the new algorithm required about two thirds of the computation time that is required by the T & T algorithm [KK92a]. For star network configuration system, an efficient load balancing algorithm than that of Tantawi and Towsley [TT84] is also derived [KK92a].

Second, the single job class model is extended to a multiple job class environment of single channel network configurations [KK90a] [KK90b]. An efficient algorithm for optimal load balancing of multi-class jobs is proposed. The performance of the proposed algorithm

H. Kameda et al., *Optimal Load Balancing in Distributed Computer Systems*

for multi-class jobs is favorably compared with those of other well-known algorithms: the Flow Deviation (FD) algorithm [FGK73] and the Dafermos algorithm [DS69]. Both the proposed algorithm and the FD algorithm require a comparable amount of storage that is far less than that required by the Dafermos algorithm. During the course of numerical experimentation, the proposed algorithm and the Dafermos algorithm required comparable computation times for obtaining the optimal solution that are far less than that of the FD algorithm [KK90a] [KK90b].

1.2 Load Balancing in the Single Job Class Environment

1.2.1 Introduction

Tantawi and Towsley [TT85] have considered an optimal static load balancing strategy which determines the optimal load at each host so as to minimize the mean job response time. They have proposed a model of a distributed computer system that consists of a set of heterogeneous host computers connected by a single channel communications network. A key assumption of theirs was that the communication delay does not depend on the source-destination pair. This assumption may apply to single channel networks such as satellite networks and some LAN's. Given this assumption, they determined the requirement that the optimal load at each host satisfies, and derived an algorithm that determines the optimal load at each host for given system parameters. It is this algorithm that they call a single-point algorithm.

The Tantawi and Towsley single-point algorithm [TT85] is surprising in the sense that it does not calculate the load at each node iteratively. Note that previous algorithms on related models such as flow-deviation type algorithms (see, e.g., Fratta, Gerla and Kleinrock [FGK73]) and Gauss-Seidel type algorithms (see, e.g., Dafermos and Sparrow [DS69] and Magnanti [Mag84]) require iterative calculation of loads. However, the algorithm appears to be complicated and difficult to understand.

In this section, the same model as Tantawi and Towsley [TT85] under the same assumptions concerning the communication delay is considered. Additionally, some properties that the optimal solution satisfies are derived. On the basis of these properties, another single-point algorithm that is more easily understandable and more straightforward than that of Tantawi and Towsley is offered [TT85]. Furthermore, several properties relating to the convergence of our algorithm are identified and its performance is demonstrated. A more efficient load balancing algorithm than that of Tantawi and Towsley [TT84] for a distributed computer system with star network configurations is also de-

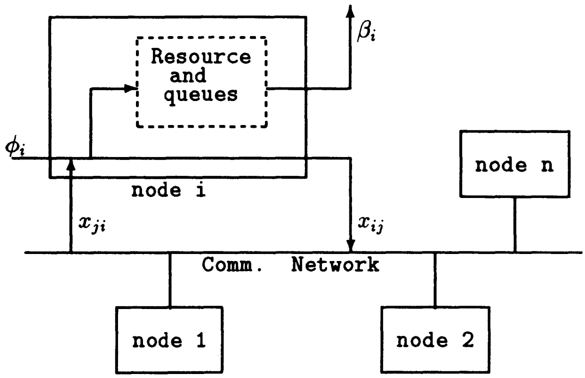

Figure 1.1: A distributed computer system.

rived.

1.2.2 Model description

The same model as that of Tantawi and Towsley is assumed [TT85]. That is, the system consists of n nodes (hosts) connected by a single channel communications network (Figure 1.1).

Jobs arrive at node i, $i = 1, 2, ..., n$, according to time-invariant Poisson process. A job arriving at node i may be either processed at node i or transferred through the communications network to another node j. We assume that the decision of transferring a job does not depend on the state of the system, and hence is *static* in nature. Also, we assume that a transferred job from node i to node j receives its service at node j and is not transferred to other nodes. It is assumed that the expected communication delay from node i to node j is independent of the source-destination pair (i, j).

For reference, a portion of the notation and assumptions contained in [TT85] are repeated here (see Appendix C of [TT85]).

n Number of nodes.

β_i Job processing rate (load) at node i.

$\boldsymbol{\beta}$ $[\beta_1, \beta_2, .., \beta_n]$.

x_{ij} Job flow rate from node i to node j.