IMPLEMENTATION OF AN OPTIMAL LOAD BALANCING ALGORITHM FOR STAR NETWORK CONFIGURATIONS

Satish Penmatsa, Framingham State University, USA
(spenmatsa@framingham.edu)

ABSTRACT

Star network configured computer systems consist of a central node to which all other computing nodes are connected. The central node can be a switch, hub or a computer, and acts as a conduit to transmit messages. The computing nodes in these networks may be heterogeneous and have varying workloads. Without proper load balancing, the performance of these systems would be degraded. In this paper, we model/implement an optimal load balancing algorithm for star network configured computer systems using M/M/1 queuing model for nodes and network links. In M/M/1 queuing model, the inter-arrival times of jobs and the service times of computers are exponentially distributed. The objective of the optimal load balancing algorithm is to minimize the average response time of jobs in the system.

Keywords: *Load balancing, distributed systems, star networks, optimal solution.*

INTRODUCTION

In distributed computer systems, the computing nodes (computers/nodes) can be arranged in various ways depending on how they need to communicate with each other. A network topology determines how different nodes in a network are connected to each other and how they communicate. Examples of network topologies include point-to-point, bus, star, ring, mesh, and tree. The point-to-point is the simplest topology with a permanent link between two nodes. In systems where bus topology is used, each node is connected to a single cable (bus). Star network configured computer systems consist of a central node to which all other computing nodes are connected. The central node can be a switch, hub or a computer, and acts as a conduit to transmit messages.

The computing nodes in these distributed systems may be heterogeneous and have varying workloads. Without proper load balancing, the performance of these systems would be degraded. In this paper, we model/implement an optimal load balancing algorithm for star network configured computer systems using M/M/1 queuing model for nodes and network links. In M/M/1 queuing model, the inter-arrival times of jobs and the service times of computers are exponentially distributed. The objective of the optimal load balancing algorithm is to minimize the average response time of jobs in the system.

A dynamic load balancing policy for distributed computer systems with star topology has been studied by (*Lim, et al., 1995*). In this study, a sender-initiated dynamic load balancing policy is considered. The problem of minimizing the average job response time in a star configured distributed computing system with N peripheral processors and one central processor is studied by (*Dattatreya, et al., 1996*). Optimal static load balancing in star network configurations with two-way traffic has been studied by (*Li, et al., 1994*). A general model for optimal static load balancing in star network configurations has been provided by (*Tantawi, et al., 1984*). A more efficient load balancing algorithm than the one proposed by (*Tantawi, et al., 1984*) for a distributed computer system with star network configurations is derived by (*Kameda, et al., 1997*). In this paper, we model the optimal load balancing algorithm for star network configured computer systems studied by (*Kameda, et al., 1997*) using M/M/1 queuing model.

The rest of the paper is organized as follows. In the next section, the star network configuration system, terminology, and assumptions are presented. Then, the load balancing problem for obtaining an optimal solution and an algorithm for computing the optimal solution are presented. Finally, conclusions and future work are presented.

THE STAR NETWORK CONFIGURATION SYSTEM

In this section, the system model is presented. The star network configuration system consists of *n* heterogeneous nodes (computers) connected to a central node as shown in Figure 1.

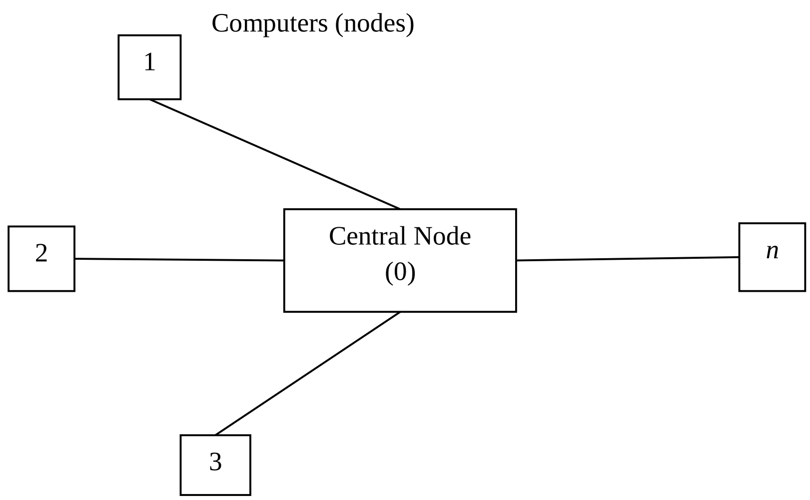

Figure 1. System Model

The terminology and assumptions used (similar to (*Kameda, et al., 1997*) and (*Penmatsa, et al., 2011*)) are as follows: The nodes and the network links are modeled as M/M/1 queuing systems (*Jain, 1991*). In these queuing systems, the inter-arrival times and the service (processing) times are exponentially distributed and jobs arrive in a single queue (which is assumed to have infinite capacity) to a single computing resource with a First Come, First Served service discipline.

The service rate of computer i is denoted by μ_i (*i.e.* the number of jobs that can be processed by the processor at computer i per unit time). The external job arrival rate at computer i is denoted by ϕ_i (*i.e.* the number of external jobs arriving at computer i per unit time). Hence, the total external job arrival rate to the system is $\sum_{i=1}^{n} \phi_i$ which is denoted by Φ. The job processing rate at computer i is denoted by β_i (this is the number of jobs per second that will be processed at computer i per unit time as allocated by the load balancing scheme). The mean communication time for sending or receiving a job from one computer to another is denoted by t. The network traffic because of computer i is denoted by λ_i.

Based on the above assumptions, the mean node delay (mean response time) for a job at computer i is given by:

$$F_i(\beta_i) = \frac{1}{(\mu_i - \beta_i)} \quad (1)$$

The mean communication delay for a job at computer i is given by:

$$G_i(\lambda_i) = \frac{t}{(1 - t\lambda_i)} \qquad \lambda < \frac{1}{t} \quad (2)$$

The nodes are classified as follows (similar to (*Tantawi, et al., 1984*)):

- *Sink nodes (S)*: Only receive jobs from other nodes but do not send out any jobs.
- *Idle source nodes (R_d)*: Do not process any jobs (β_i = 0) and send all the jobs to other nodes. Do not receive any jobs from other nodes.
- *Active source nodes (R_a)*: Processes part of the jobs that arrive and send the remaining jobs to other nodes. But, they do not receive any jobs.
- *Neutral nodes (N)*: Process jobs locally without sending or receiving jobs.

OPTIMAL LOAD BALANCING

The problem of minimizing the mean response time of a job can be expressed as (*Kameda, et al., 1997*)):

$$minimize\ D(\beta) = \frac{1}{\Phi}\left[\sum_{i=0}^{n} \beta_i F_i(\beta_i) + \sum_{i=1}^{n} (\phi_i - \beta_i) G_i(\phi_i - \beta_i)\right] \quad (3)$$

subject to the following *conservation*, *stability*, and *positivity* constraints.

Conservation constraint: $\sum_{i=0}^{n} \beta_i = \Phi$ (4)

Stability constraint: $\beta_i \leq \phi_i, \quad i = 1, 2, \dots, n$ (5)

Positivity constraint: $\beta_i \geq 0, \quad i = 0, 1, 2, \dots, n$ (6)

The conservation constraint states that the total external job arrival rate into the system should be equal to the total job processing rate (of all the computers). The stability constraint states that the job processing rate at a computer should not be greater than the external job arrival rate to that computer. The positivity constraint states that the job processing rate at a computer should not be negative.

The marginal node delay is defined as:

$$f_i(\beta_i) = \frac{\delta}{\delta\beta_i}(\beta_i F_i(\beta_i)) \quad (7)$$

Substituting Equation (1) in Equation (7), we have:

$$f_i(\beta_i) = \frac{\delta}{\delta\beta_i}\left(\frac{\beta_i}{\mu_i - \beta_i}\right) = \frac{\mu_i}{(\mu_i - \beta_i)^2} \quad (8)$$

The marginal communication delay is defined as:

$$g_i(\lambda_i) = \frac{\delta}{\delta\lambda_i}(\lambda_i G_i(\lambda_i)) \quad (9)$$

Substituting Equation (2) in Equation (9), we have:

$$g_i(\lambda_i) = \frac{\delta}{\delta\lambda_i}\left(\frac{t\lambda_i}{1 - t\lambda_i}\right) \quad (10)$$

Hence,

$$g_i(\phi_i - \beta_i) = \frac{\delta}{\delta\beta_i}\ (\phi_i - \beta_i) G_i(\phi_i - \beta_i) = \frac{1}{\left(1 - t(\phi_i - \beta_i)\right)^2} \quad (11)$$

The difference between the marginal node delay and the marginal communication delay is denoted by $h_i(\beta_i)$. So,

$$h_i(\beta_i) = f_i(\beta_i) - g_i(\phi_i - \beta_i), \quad i = 1, 2, \dots, n \quad (12)$$

Substituting Equation's (8) and (11) in Equation (12), we have,

$$h_i(\beta_i) = \frac{\mu_i}{(\mu_i - \beta_i)^2} - \frac{1}{\left(1 - t(\phi_i - \beta_i)\right)^2}, \quad i = 1, 2, \dots, n \quad (13)$$

Kameda, et al., 1997 derived three properties (Property 1.6, 1.7, and 1.8) based on the Tantawi-Towsley Theorem for star network configurations (*Tantawi, et al., 1984*). The following load balancing algorithm provides an optimal solution for star network configurations using M/M/1 queuing model for nodes and network links by substituting the expressions derived above in Equations (8), (11), and (13) for $f_i, g_i,$ and h_i respectively in Property's 1.6, 1.7, and 1.8 in (*Kameda, et al., 1997*).

ALGORITHM:

1. Order computers (nodes) in decreasing order of their $h_i's$.
2. Find α (Lagrange multiplier) such that $\lambda_0(\alpha) = \lambda_R(\alpha)$, by using, for example, a binary search.
3. Determine the optimal loads of computers:
   - $\beta_i = 0\ for\ i\ in\ R_d(\alpha)$
   - $\beta_i = h_i^{-1}(\alpha) for\ i\ in\ R_a(\alpha)$
   - $\beta_i = \phi_i\ for\ i\ in\ N(\alpha)$

CONCLUSION

In this paper, we presented an optimal load balancing algorithm for star network configured computer systems using M/M/1 queuing model as node and network link models. The objective of the optimal load balancing algorithm is to minimize the average response time of jobs in the system. In future, we plan to evaluate the performance of the optimal algorithm using simulations with various system configurations.

REFERENCES

Dattatreya G. R. and Venkatesh R. (1996), Static and decentralized-adaptive load balancing in a star configured distributed computing system, *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 26(1): 91 – 104, 1996.

Jain R. (1991), The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling, *Wiley-Interscience*, 1991.

Kameda H., Li J., Kim C. and Zhang Y. (1997), Optimal load balancing in distributed computer systems, *Springer Verlag*, London, 1997.

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

Lim K. S. and Kim C. G. (1995), Dynamic load balancing in distributed computer systems with star topology, *Proceedings of the 5th IEEE Computer Society Workshop on Future Trends of Distributed Computing Systems*, Pages: 514 – 517, 1995.

Penmatsa S. and Chronopoulos A. T. (2011), Game-Theoretic Static Load Balancing for Distributed Systems, *Journal of Parallel and Distributed Computing*, 71(4): 537-555, April 2011.

Tantawi A. N. and Towsley D. F. (1984), A general model for optimal static load balancing in star network configurations, *Proceedings of the 10th International Symposium on Computer Performance Modelling, Measurement and Evaluation (Performance '84)*, Pages 277-291, 1984.