IMPLEMENTATION OF A FAIR LOAD BALANCING ALGORITHM FOR STAR NETWORK CONFIGURED COMPUTER SYSTEMS

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

ABSTRACT

Networked computer systems are often used for executing computation intensive applications. The computers in these systems can have varying processing speeds and architectures. Due to this heterogeneity, the performance of applications can be impacted if the load is not properly balanced among the networked computers. In this paper, we model/implement a fair job-optimal load balancing algorithm for star network configured computer systems using M/M/1 queuing model. The objective of this algorithm is to provide fairness to all jobs in the system i.e. all jobs of approximately the same size will experience approximately the same average response time irrespective of the computers allocated for their execution.

Keywords: *Fairness, Load balancing, star networks, distributed computing.*

INTRODUCTION

The load on modern distributed computing systems such as the cloud computing systems has been increasing tremendously with the development of complex applications (*Rzadca et al., 2007*). Efficient job allocation to computing resources is essential for improving the performance of networked computing systems. In order to balance the load among the computers (nodes), load balancing algorithms consider several factors including the heterogeneity in the service/processing rate of computers, loads on computers, and the architecture of the computers. The objective of load balancing algorithms can be, for example, to minimize the overall average response time of the (jobs) in the system, or to minimize the average response time of a class of jobs, or to minimize the average response time of individual jobs. Load balancing algorithms may either take the average system statistics or current system information in making job allocation decisions.

In this paper, we implement a fair job-optimal load balancing algorithm for star network configured computer systems. We model the individual optimal policy for star network configured computer systems studied by (*Kameda, et al., 1997*) using M/M/1 queuing model. 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 objective of the implemented algorithm is to provide fairness to all jobs in the system *i.e.* all jobs of approximately the same size will experience approximately the same average response time irrespective of the computers allocated for their execution.

System optimal and job optimal load balancing policies for star networked systems have been studied in (*Kameda, et al., 1997*). An implementation of an optimal load balancing algorithm for star networked configurations based on M/M/1 queuing model for nodes and network links has been provided in (*Penmatsa, 2015*). Fair load balancing considering a single channel network with M/M/1 node and network models has been studied in (*Penmatsa, 2011*). Load balancing in distributed systems using cooperative game theory for providing fairness to all the jobs has been studied in (*Grosu et al., 2008*). Resource management in grid computing systems for providing fairness based on game theory has been studied in (*Rzadca et al., 2007*). A dynamic load balancing policy for distributed computer systems with star topology has been studied by (*Lim, et al., 1995*). A survey of some modern techniques for load balancing in distributed systems can be found in (*Aslam et al., 2015*).

FAIR LOAD BALANCING

We consider a star networked system consisting of n computers (nodes) connected to a central node (similar to *Penmatsa, 2015 and Kameda, et al., 1997)* as shown in Figure 1. The nodes and the network links are modeled as M/M/1 queuing systems (*Jain, 1991*). Based on the above assumption, the mean node delay for a job at computer i is given by *(Jain, 1991)*:

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

where μ_i is the service rate of node i and β_i is the job processing rate at node i.

The mean communication delay for a job at computer i is given by *(Jain, 1991)*:

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

where t is the mean communication time for sending or receiving a job from one computer to another and λ_i is the network traffic because of computer i.

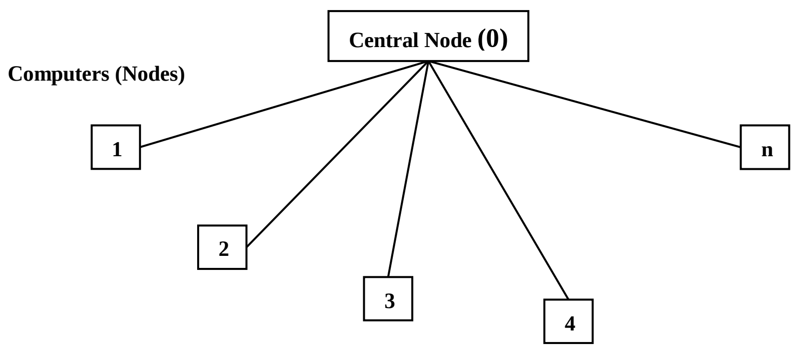

Figure 1. Star Networked Computer System

The nodes are classified as follows:

- *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 ($\beta_i = 0$) 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.

Let $$H_i(\beta_i) = F_i(\beta_i) - G_i(\lambda_i) \tag{3}$$

The following relations are equivalent to the equilibrium conditions *(Kameda, et al., 1997).* At the equilibrium solution (loads), a job cannot reduce its average (mean) response time even if it is transferred to another node for remote processing:

$$H_i(\beta_i) \geq \alpha, \qquad \beta_i = 0 \qquad for\ i \in R_d \tag{4}$$

$$H_i(\beta_i) = \alpha, \qquad 0 < \beta_i < \phi_i \qquad for\ i \in R_a \tag{5}$$

$$H_i(\beta_i) \leq \alpha, \quad \beta_i = \phi_i \quad for\ i \in N \tag{6}$$

$$\beta_0 = F_0^{-1}(\alpha) \tag{7}$$

subject to the flow constraint

$$F_0^{-1}(\alpha) + \sum_{i \in R_a} H_i^{-1}(\alpha) + \sum_{i \in N} \phi_i = \phi \tag{8}$$

where ϕ_i is the external job arrival rate at computer i, Φ is the total external job arrival rate into the system ($\sum_{i=1}^{n} \phi_i$), and α is the Lagrange multiplier. The following load balancing algorithm computes and allocates loads (job processing rates) to computers such that all jobs of approximately the same size will experience approximately the same average response time irrespective of the computers allocated for their execution.

FAIR-STAR ALGORITHM:

Input:

Node service rates: $\mu_1, \mu_2, \mu_3, \ldots \mu_n$
Node job arrival rates: $\phi_{1,}\phi_2, \phi_3, \ldots \phi_n$
Mean communication time: t

Output:

Node processing rates (load allocation to the nodes): $\beta_1, \beta_2, \beta_3, \ldots \beta_n$

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

The above algorithm takes the average system statistics such as the average arrival rates of jobs to the computers into account and hence is static in nature. The algorithm must be executed whenever the system loads (arrival rates) change over some threshold or periodically in order to compute a new load (job processing rates) allocation to the computers.

Preliminary results using a star networked system comprising of 32 computers showed that the mean response time of jobs achieved by FAIR-STAR algorithm is close to that of STAR-OPTIM (*Penmatsa, 2015*) and considerably lower than that of PROP (*Chow et al. 1979*). Also, the mean response times achieved by FAIR-STAR for all the computers are almost equal which shows that all jobs experience almost the same execution time independent of the allocated computer. The objective of STAR-OPTIM is to minimize the mean response time of jobs in the system in order to provide a system-optimal solution. PROP allocates jobs to computers in proportion to their service rates. The 32 node system considered has four types of computers with varying service rates to simulate a heterogeneous system.

CONCLUSION

In this paper, we presented a job-optimal load balancing algorithm for star network configured computer systems. The objective of this algorithm is to provide fairness to all jobs in the system *i.e.* all jobs of approximately the same size will experience approximately the same average response time irrespective of the computers allocated for their execution. In future work, we plan to evaluate the performance of FAIR-STAR algorithm by comparing it with a system-optimal algorithm using simulations with various system configurations.

REFERENCES

Aslam S. and Shah M. A., Load balancing algorithms in cloud computing: A survey of modern techniques, *IEEE National Software Engineering Conference (NSEC)*, Pages: 30 - 35, 2015.

Chow Y. C. and Kohler W. H. (1979), Models for dynamic load balancing in a heterogeneous multiple processor system, *IEEE Transactions on Computers*, C-28(5):354– 361, May 1979.

Grosu D., Chronopoulos A. T., and Leung M. Y. (2008), Cooperative load balancing in distributed systems, *Concurrency and Computation: Practice and Experience*, 20 (16) (2008) 1953–1976.

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.

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. (2015), Response Time Minimization in Star Networked Distributed Systems Using Optimal Load Balancing, *Journal of Global Information Technology*, Vol. 10, No. 1&2, pp. 1-8, ISSN: 1931-8162, 2015.

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.

Rzadca K., Trystram D., and Wierzbicki A., Fair game-theoretic resource management in dedicated grids, in: *Proc. of the 7th IEEE International Symposium on Cluster Computing and the Grid*, Brazil, May 2007, pp. 343–350.