Performance Evaluation of System Optimal Load Balancing Schemes for Multi-User Job Distributed Systems

Satish Penmatsa[1] **and Gurdeep S. Hura**[2]
[1]Dept. of Computer Science, Framingham State University, Framingham, MA, USA
[2]Dept. of Math. & Computer Science, University of Maryland Eastern Shore, Princess Anne, MD, USA

Abstract – *Load balancing of computing resources in distributed computing systems is essential for improving the performance of computation-intensive applications. In this paper we study static and dynamic load balancing schemes whose objective is to provide a system optimal solution for all the users in the system. We consider a central-server node model with communication costs when transferring jobs for remote processing. The performance of the static and dynamic schemes is evaluated using simulations with various distributed system configurations.*

Keywords: Load balancing, system optimal solution, distributed computing.

1 Introduction

Modern distributed systems such as cloud computing systems often comprise of several computing resources (nodes) with varying processor speeds distributed over a wide geographic area. Jobs or applications being executed by these systems may be submitted by several users with varying submission rates. Due to the system heterogeneity and differences in the job arrival rates, the performance or the executing time of these applications may be hugely impacted. Balancing the load of the computing resources by taking their heterogeneity and the job arrival rates into account can significantly improve the performance of distributed systems and their applications.

In this paper, we study and compare two load balancing schemes (*StaticGOS* and *DynamicGOS*) whose objective is to minimize the average response time of all jobs in the system to provide a global (system) optimal solution. The StaticGOS tries to load balance by taking the average system behavior into account whereas the DynamicGOS tries to load balance by taking the instantaneous system states into account. Related work on static load balancing for providing a system optimal solution has been studied in [6]. Here, we consider a central-server node model and take the communication costs into account when transferring jobs to other nodes for remote processing. Static and dynamic load balancing schemes for single-class (single-user) jobs by considering a central-server node model have been studied in [15]. Here, we consider a multi-class (multi-user) job distributed system.

Several other studies have been made for load balancing in distributed systems considering various system models and objectives. For example, the impact of various factors such as scalability and resource utilization on load balancing schemes has been studied in [16]. An adaptive load balancing algorithm for wireless distributed computing networks based on the channel conditions and available energy among the cooperating nodes has been studied in [1]. In [14], the behavior of load balancing strategies with regard to the network structure in distributed computing systems has been studied. Performance analysis of greedy load balancing algorithms in heterogeneous distributed computing systems has been made in [13].

Game theoretic approaches for load balancing with the objective of providing fairness have been proposed in [4, 8]. A non-cooperative game theoretic framework was employed to achieve load balancing among selfish users/consumers. Static load balancing for providing a job-optimal solution based on cooperative game theory has been proposed in [17]. In [5], a load balancing policy for fair workload allocation in heterogeneous systems has been proposed. Static resource allocation for heterogeneous computing environments with tasks having dependencies, priorities, deadlines, and multiple versions has been studied in [2]. System cost optimization in multi-class utility computing systems has been studied in [11]. In [10], a static job allocation scheme for distributed systems with selfish agents using a central-server node model has been proposed. Adaptive cost optimization for fair resource allocation in computational grid systems has been studied in [12]. In [9], dynamic load balancing of multi-user jobs considering an M/M/1 node model has been studied.

The rest of the paper is organized as follows. In Section 2, we present the system model. In Section 3, the StaticGOS and DynamicGOS load balancing procedures are presented.

Simulation results comparing the load balancing schemes are presented in Section 4 and conclusions are drawn in Section 5.

2 Distributed System Model

A distributed system with n computing nodes connected by a communications network is considered as shown in Figure 1. In the following, we present the notations and assumptions used (similar to [6, 15]).

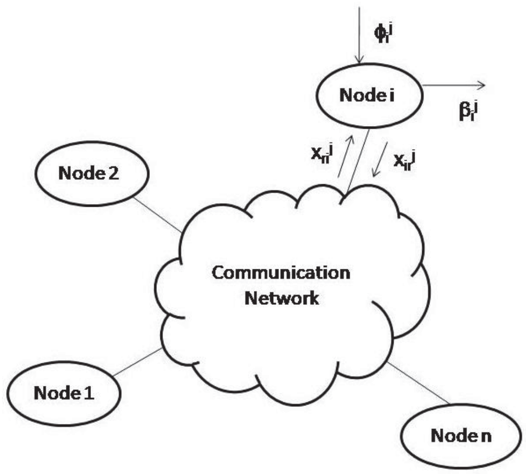

Figure 1: System Model

The service rate of each node i is denoted by μ_i . Jobs arriving to each node may belong to m different classes (users). The job arrival rate of user j to node i is denoted by ϕ_i^j . The total job arrival rate of user j is denoted by ϕ^j and the total job arrival rate into the system is denoted by ϕ. Each node is modeled as a central-server model as shown in Figure 2.

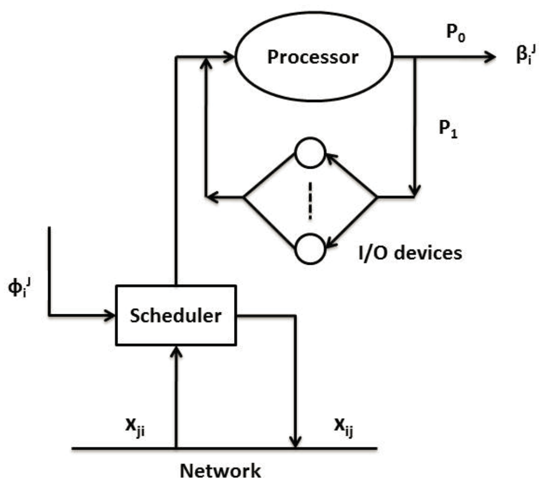

Figure 2: Node Model

p_0 denotes the probability that a job after departing from the processor finishes. p_1 denotes the probability that a job after departing from the processor requests I/O service. Therefore, p_1/p_0 denotes the average number of I/O requests per job. The job flow rate of user j from node r to node s is denoted by x_{rs}^j .

We assume that the nodes and the communications network have an exponential service-time distribution and that the arrivals of jobs follow a Poisson distribution [7]. For each user j, nodes are classified into the following: *Idle source nodes* (R^j_d), *Active source nodes* (R^j_a), *Neutral nodes* (N^j), and *Sink nodes* (S^j). Idle source nodes do not process any jobs and send all the jobs to other nodes. Active source nodes processes part of the jobs arriving to them and send the rest to the other nodes. Neutral nodes processes local jobs and do not receive or send any jobs from or to other nodes. Sink nodes receive jobs from other nodes and processes all local and remote jobs (received from source nodes). They do not send jobs to other nodes.

Based on the above assumptions, the average response time of a user j job processed at node i is given by [7]:

$$F_i^j(\beta_i) = \frac{1}{(\mu_i - \sum_{k=1}^{m} \beta_i^k)} + \frac{p_1}{p_0} t_{IO} \qquad (1)$$

where β_i^j is the job processing rate (load) of user j at node i, t_{IO} is the service time of an I/O device, and $\beta_i = [\beta_i^1, \beta_i^2, \ldots, \beta_i^m]^T$.

The average communication delay of a user j job is given by:

$$G^j(\lambda) = \frac{t}{(1 - t\sum_{k=1}^{m} \lambda^k)}, \qquad \sum_{k=1}^{m} \lambda^k < \frac{1}{t} \qquad (2)$$

where t is the mean communication time for sending or receiving a job form one node to the other for any user, λ^j is the job traffic through the network of user j, and λ is the total traffic over the communications network.

3 Optimal Load Balancing

In this section, we present the static and dynamic load balancing schemes whose objective is to minimize the overall response time of all user jobs in the system in order to provide a system optimal solution. The *StaticGOS* takes the average system behavior into account in balancing the load whereas the *DynamicGOS* takes the instantaneous system information into account for balancing the load among the computing nodes.

3.1 Static Optimal Load Balancing (StaticGOS):

The problem of minimizing the overall average response time of the system can be expressed as:

$$\min D(\beta) = \frac{1}{\phi}\sum_{j=1}^{m}[\sum_{i=1}^{n}\beta_i^j F_i^j(\beta_i) + \lambda^j G^j(\lambda)] \quad (3)$$

subject to the constraints:

$$\beta_i^j \geq 0 \quad (4)$$

$$\sum_{i=1}^{n}\beta_i^j = \phi^j \quad (5)$$

$$\sum_{j=1}^{m}\beta_i^j < \mu_i \quad (6)$$

The user j *marginal node delay* $f_i^j(\beta_i)$ and *marginal communication delay* $g^j(\lambda)$ are defined as:

$$f_i^j(\beta_i) = \frac{\partial}{\partial\beta_i^j}\sum_{k=1}^{m}\beta_i^k F_i^k(\beta_i)$$

$$= \frac{\mu_i}{(\mu_i - \sum_{k=1}^{m}\beta_i^k)^2} + \frac{p_1}{p_0}t_{IO} \quad (7)$$

$$g^j(\lambda) = \frac{\partial}{\partial\lambda^j}\sum_{k=1}^{m}\lambda^k G^k(\lambda) = \frac{t}{(1 - t\sum_{k=1}^{m}\lambda^k)^2} \quad (8)$$

where $\beta_i^k F_i^k(\beta_i)$ denotes the average number of user k jobs at node i and $\lambda^k G^k(\lambda)$ denotes the average number of user k jobs in the communications network.

Following the methodology proposed in [6], the optimal solution to the problem in Eq. (3) satisfies the following relations:

$$\begin{array}{lll}
f_i^j(\beta_i) \geq \alpha^j + g^j(\lambda), & \beta_i^j = 0 & (i \in R_d^j), \\
f_i^j(\beta_i) = \alpha^j + g^j(\lambda), & 0 < \beta_i^j < \phi_i^j & (i \in R_a^j), \\
\alpha^j + g^j(\lambda) \geq f_i^j(\beta_i) \geq \alpha^j, & \beta_i^j = \phi_i^j & (i \in N^j), \\
f_i^j(\beta_i) = \alpha^j, & \beta_i^j > \phi_i^j & (i \in S^j)
\end{array}$$

subject to the total flow constraint,

$$\sum_{i\in S^j}(f_i^j)^{-1}(\beta_i|_{\beta_i^j=\alpha^j}) + \sum_{i\in N^j}\phi_i^j + \sum_{i\in R_a^j}(f_i^j)^{-1}(\beta_i|_{\beta_i^j=\alpha^j+g^j(\lambda)}) = \phi^j$$

where α^j is the Lagrange multiplier.

In order to find the optimal solution (*i.e.* optimal β_i^j's), we initially assume that the loads (job processing rates) at node *i* (*i = 1 … n*) for user *j* (*j = 1 … m*) jobs to be ϕ_i^j. Then, in an iterative manner, the loads of each user to the nodes are updated by fixing the other user loads and by choosing an optimal α^j that makes the total traffic out of the source nodes equal (to some tolerance) to the total traffic into the sinks (for that user). This procedure continues until the desired norm is reached. This procedure balances the marginal node delays of all the users statically for providing a system optimal solution.

3.2 Dynamic Optimal Load Balancing (DynamicGOS):

In the following, we discuss a dynamic load balancing procedure for providing a near system optimal solution.

Let r_i denote the average service time of a job at node *i* and N_i^j denote the average number of jobs of user *j* at node *i*. The *marginal node delay* of a user *j* job at node *i* from Eq. (7) can be expressed in terms of r_i and N_i^j (using Little's law [7]) as:

$$f_i^j(\beta_i) = r_i(1 + \sum_{k=1}^{m}N_i^k)^2 + \frac{p_1}{p_0}t_{IO} \quad (9)$$

Let ρ denote the average utilization of the communications network where $\rho = t\sum_{k=1}^{m}\lambda^k$. Rewriting Eq. (8) in terms of ρ, we have:

$$g^j(\lambda) = \frac{t}{(1-\rho)^2}, \qquad \rho < 1 \quad (10)$$

Let n_i^j denote the number of jobs of user *j* at node *i* at a given instant and ρ' denote the utilization of the communications network at a given instant. Rewriting Eq.'s (9) and (10) in terms of these instantaneous variables, we have:

$$f_i^j = r_i(1 + \sum_{k=1}^{m}n_i^k)^2 + \frac{p_1}{p_0}t_{IO} \quad (11)$$

$$g^j = \frac{t}{(1-\rho')^2}, \qquad \rho' < 1 \quad (12)$$

f_i^j and g^j are used as the estimates of user *j marginal virtual node delay* at node *i* and user *j marginal virtual communication delay*. The dynamic load balancing procedure tries to balance the marginal virtual node delays of each user at all the nodes dynamically.

Each node *i* broadcasts the number of jobs of user *j* in its queue (n_i^j's) and the arrival rates (ϕ_i^j's) to all the other nodes periodically. Using the arrival rate information, each node executes the StaticGOS procedure to determine the optimal loads, which are then converted (using Little's law) to *thresholds* (T_i^j) *– the optimal number of jobs of user j that can be present at node i*.

An arriving job of user *j* to node *i* will be transferred to another node for remote processing if $n_i^j > T_i^j$. Else, the job will be processed locally. If $f_i^u > f_j^u + g^u$, then node *i* is said to be more heavily loaded than node *j* for a user *u* job. Using this expression, each other node will be compared with the local node to determine the lightest node for remote processing a user *j* job.

4 Experimental Results

In this section, we present the experimental results and compare the performance of StaticGOS and DynamicGOS. We simulated a distributed system with 32 nodes and 20 users. To make the system heterogeneous, we considered nodes with different service rates with the service rate of the fastest node 10 times that of the slowest. The total job arrival rate of the system is determined by the system utilization (load) and the total service rate of the system. The total job arrival rate of the system is divided among the 20 users unevenly to simulate heterogeneous user job arrival rates. The mean communication time is assumed to 1 millisecond. We also implemented a Proportional load balancing scheme (PROP_M) [3] for comparison purposes. With PROP_M each user allocates her/his jobs to computing nodes in proportion to their service rates.

In Figure 3, we present the effect of system utilization (system load) on the performance of the load balancing schemes when the overhead for job transfer is 0. The overhead for job transfer is defined as the percentage of service time that a node has to spend to send or receive a job. It can be observed that at low system loads (10% through 30%), the average response time for a job achieved by StaticGOS and DynamicGOS is close. As the system load increases, the average response time achieved by DynamicGOS is substantially less than that of StaticGOS. For example, at 80% system utilization, the average response time of DynamicGOS is about 50% less than that of StaticGOS. The average response time achieved by PROP_M is much higher than the other two schemes because it overloads the slower nodes.

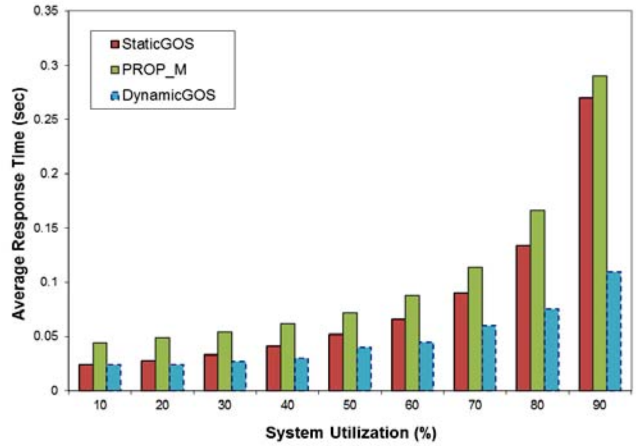

Figure 3: Average Response Time vs System Utilization (Overhead = 0)

Figure 4 presents the effect of system utilization on the performance of the load balancing schemes when the overhead for job transfer (OV) is 5%. It can be observed that at low system loads the performance of StaticGOS is close to that of DynamicGOS. As the system utilization increases, the average response time for a job achieved by DynamicGOS is substantially lower than that of StaticGOS and PROP_M. However, the response time of the load balancing schemes at high system utilization (90%) is much higher than in Figure 3 due to the involved overhead costs.

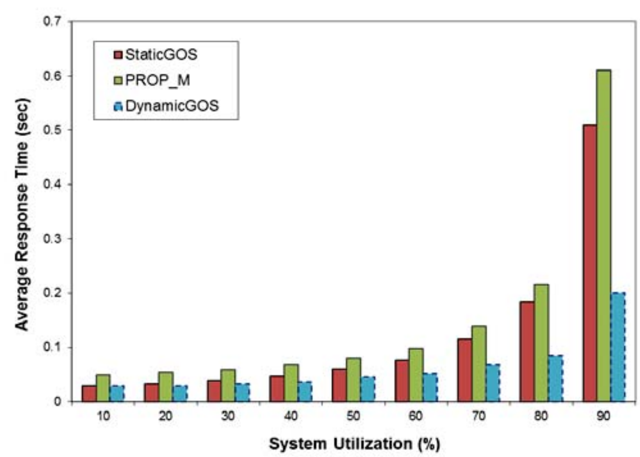

Figure 4: Average Response Time vs System Utilization (Overhead = 5%)

Figure 5 presents the effect of system utilization on the performance of the load balancing schemes when the overhead for job transfer (OV) is 10%. It can again be observed that the average response time achieved by StaticGOS is close to that of DynamicGOS at low system loads. At medium system loads (50% through 70%), the average response time achieved by DynamicGOS is about 50% less than that of StaticGOS. However, for high system utilizations, the performance of DynamicGOS approaches to that of the other two static schemes due to the involved high overhead costs for job and state information exchange by the nodes.

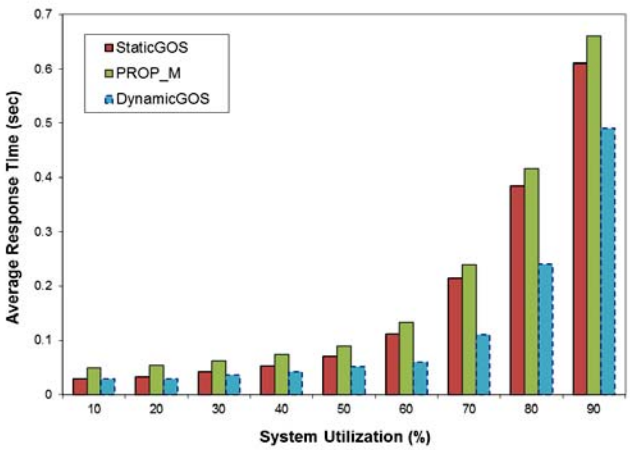

Figure 5: Average Response Time vs System Utilization (Overhead = 10%)

5 Conclusions

In this paper, static and dynamic load balancing schemes for providing a system optimal solution have been studied and their performance has been compared using a heterogeneous system model. It was observed that at low system loads, the performance of the StaticGOS is close to that of DynamicGOS. For medium and high system loads, the average response time achieved by the DynamicGOS is substantially lower than that of StaticGOS. However, as the communication overhead costs increase, the performance of DynamicGOS tends to approach to that of StaticGOS.

In future work, we plan to study the impact of the variation of state information exchange period on DynamicGOS, and to implement efficient mechanisms for exchanging state information among the nodes to reduce the impact of the communication overhead costs on the performance of DynamicGOS.

6 References

[1] M. I. M. Alfaqawi, M. H. Habaebi, M. U. Siddiqi, Md. R. Islam, S. Khan, and D. Datla, “Adaptive load balancing algorithm for wireless distributed computing networks”, International Conference on Intelligent Systems Engineering (ICISE), Year: 2016, Pages: 256 – 261.

[2] T.D. Braun, H.J. Siegel, A.A. Maciejewski, and Y. Hong, “Static resource allocation for heterogeneous computing environments with tasks having dependencies, priorities, deadlines, and multiple versions”, Journal of Parallel and Distributed Computing, 68 (11) (2008) 1504–1516.

[3] Y.C. Chow, W.H. Kohler, “Models for dynamic load balancing in a heterogeneous multiple processor system”, IEEE Transactions on Computers, C-28 (5) (1979) 354–361.

[4] D. Grosu and A.T. Chronopoulos, “Noncooperative load balancing in distributed systems”, Journal of Parallel and Distributed Computing, 65 (9) (2005):1022–1034.

[5] L. Georgiadis, C. Nikolaou, and A. Thomasian, “A fair workload allocation policy for heterogeneous systems”, Journal of Parallel and Distributed Computing, 64 (4) (2004) 507–519.

[6] H. Kameda, J. Li, C. Kim, and Y. Zhang, “Optimal load balancing in distributed computer systems”, Springer Verlag, London, 1997.

[7] R. Jain, “The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling”, Wiley-Interscience, 1991.

[8] S. Nouri and S. Parsa, “A non-cooperative approach for load balancing in heterogeneous distributed computing platform”, 4th International Conference on Computer Sciences and Convergence Information Technology, Year: 2009, Pages: 756 – 761.

[9] S. Penmatsa and A. T. Chronopoulos, “Dynamic multi-user load balancing in distributed systems”, Proceedings of the 21st IEEE International Parallel and Distributed Processing Symposium (IPDPS 2007), Long Beach, California, USA, March 26-30, 2007.

[10] S. Penmatsa and G. S. Hura, “Job allocation in e-commerce systems involving self-interested agents”, Journal of Global Information Technology, Vol. 5, No. 1, pp. 1-11, 2010.

[11] S. Penmatsa, "System cost optimization in multi-class utility computing systems", Proceedings of the 2013 MTMI International Conference, Review of Business and Technology Re- search, Virginia Beach, VA, USA, Sept. 20-21, 2013.

[12] S. Penmatsa and G. S. Hura, "Adaptive cost optimization and fair resource allocation in computational grid systems", Proc. of the 29th International Conference on Computer Applications in Industry and Engineering, Sept. 26-28, Denver, CO, 2016.

[13] B. Sahoo, D. Kumar, and S. K. Jena, "Performance analysis of greedy load balancing algorithms in heterogeneous distributed computing system", International Conference on High Performance Computing and Applications (ICHPCA), Year: 2014, Pages: 1 – 7.

[14] M. A. Salman, C. Bertelle, and E. Sanlaville, "The behavior of load balancing strategies with regard to the network structure in distributed computing systems", 10th International Conference on Signal-Image Technology and Internet-Based Systems, Year: 2014, Pages: 432 – 439.

[15] Y. Zhang, K. Hakozaki, H. Kameda, and K. Shimizu, "A performance comparison of adaptive and static load balancing in heterogeneous distributed systems", In Proc. IEEE 28th Annual Simulation Symposium, pages 332–340, Phoenix, AZ, 1995.

[16] A. Vig, R. S. Kushwah; and S. S. Kushwah, "An efficient distributed approach for load balancing in cloud computing", International Conference on Computational Intelligence and Communication Networks (CICN), Year: 2015, Pages: 751 – 755.

[17] D. Grosu, A.T. Chronopoulos, and M.Y. Leung, "Load balancing in distributed systems: an approach using cooperative games", Proc. of the 16th IEEE Intl. Parallel and Distributed Processing Symp., Ft Lauderdale, Florida, USA, April 2002.