EPiC Series in Computing
Volume 91, 2023, Pages 108–115
Proceedings of 38th International Conference on Computers and Their Applications

Adaptive Fair and Cost-Effective Load Balancing in Distributed Computing Systems

Satish Penmatsa[1] and Penmatsa Krishna Kanth Varma[2]
[1] Framingham State University, MA, USA
[2] SRKR Engineering College, AP, India
spenmatsa@framingham.edu, pkkvarma@srkrec.ac.in

Abstract

Load balancing is one of the main challenges in distributed computing systems such as the cloud computing systems. It helps improve throughput while reducing the response time and cost for data and computation intensive applications. In this paper, we present an adaptive load balancing scheme for heterogeneous distributed computing systems whose objective is to provide a fair allocation of jobs to computing resources that reduces the cost of executing the jobs in the system. Using simulations, we compare the performance of the presented scheme with that of existing load balancing schemes.

1 Introduction

Emerging technologies and applications related to artificial intelligence, data science, and big data analytics rely on high-performance systems for computational needs. These systems often comprise of computing nodes and software components spread over wide area networks and offer benefits such as scalability, efficiency, and cost-effectiveness. Load balancing is essential for these heterogeneous distributed systems to avoid some nodes becoming overloaded and some becoming underloaded. Effective load balancing can help improve resource utilization, response time, and cost for data and computation intensive applications.

A survey of load balancing algorithms for distributed systems such as the cloud computing systems has been made in [2] [6] [10]. Static load balancing mechanisms require prior knowledge of the system state in making job allocations to the computing resources. However, these mechanisms cannot adapt to changing system parameters and application load. Adaptive load balancing mechanisms monitor the system and application load at run-time and hence can recompute the allocation of jobs to resources with any changes in the system and application load [7].

With the advent of artificial intelligence-based applications in several fields, load balancing algorithms for heterogeneous networking systems based on machine learning are also being devised for data intensive applications [5]. Static load balancing for providing fairness in distributed utility computing systems has been studied in [8]. An adaptive scheme with the objective of minimizing the response time of jobs and providing fairness has been presented in [9]. In this paper, we present an adaptive load balancing scheme for heterogeneous distributed computing systems whose objective is to reduce the cost for executing the jobs in the system by considering fairness. A client-server node model is considered. Simulations are conducted to study the performance of the proposed scheme.

2 Distributed Computing System Model

We consider a heterogeneous distributed system with *n* computing nodes. The service rate of node *i* is denoted by μ_i (*i.e.* the number of jobs that can be processed by the processor at node *i* per unit time). The job arrival rate at node *i* is denoted by ϕ_i (*i.e.* the number of jobs arriving at node *i* per unit time). Hence, the total job arrival rate to the system is $\sum_{i=1}^{n} \phi_i$ which is denoted by ϕ. The job processing rate at node *i* is denoted by β_i (this is the number of jobs that will be processed at node *i* per unit time). Each node is modeled as a central-server model [9] [11]. We assume that P_0 denotes the probability that a job after departing from the processor (at a node) finishes and P_1 denotes the probability that a job after departing from the processor requests an I/O service. Hence, $\frac{P_1}{P_0}$ denotes the average number of I/O requests per job. The service time of an I/O device is denoted by t_{IO}.

Based on the above assumptions, the mean response time of a job at node *i* is given by:

$$T_i(\beta_i) = \frac{1}{(\mu_i - \beta_i)} + \frac{P_1}{P_0} t_{IO} \tag{1}$$

The computing nodes in these distributed systems may belong to various agencies/institutions and have different pricing policies. In order to obtain the prices charged by the resource owners, we use a pricing model based on bargaining game theory proposed in [3]. An incomplete information alternating-offer non-cooperative bargaining game is played (simulated) between the owners of the jobs and the computing resources to obtain an agreed price-per-unit-resource. Let k_i be a constant that maps the response time at node *i* to the amount of resources consumed at node *i* and k_{IO} be a constant that maps the I/O delay to the amount of resources consumed from the I/O device. Also, let p_i denote the price per unit resource at node *i* and p_{IO} denote the price per unit resource of an I/O device.

Based on the above assumptions, the mean cost for processing a job at node *i* is given by:

$$C_i(\beta_i) = \frac{k_i p_i}{(\mu_i - \beta_i)} + k_{IO} p_{IO} \left(\frac{P_1}{P_0}\right) t_{IO} \tag{2}$$

3 Load Balancing

Static load balancing can be achieved by using the average system statistics. Consider the average system parameters μ_i (service rate of node i) and ϕ_i (job arrival rate at node i). Based on economic game theory, the load balancing problem is formulated as a cooperative game [4] between the computing nodes. The nodes cooperate in making a job allocation such that each operates at its optimum (the objective of each node is to minimize its mean cost given by Equation 2). The cooperative game assumes that each node plays (performs) at least at a minimum by servicing jobs given by its service rate and that the job processing rate at a node is less than that of its service rate, the job processing rate is non-negative, and the total job arrival rate of the system is equal to the total job processing rate of the nodes in the system ($\sum_{i=1}^{n} \beta_i = \sum_{i=1}^{n} \phi_i = \phi$).

The Nash Bargaining Solution [4] provides a Pareto optimal and fair solution (COOPS (Cooperative Solution)) that reduces the cost for executing all the jobs in the system and also provides fairness to all the jobs in the system *i.e.*, all the jobs of approximately the same size will incur approximately the same expected cost independent of the nodes allocated for their execution.

Adaptive load balancing considers the current system state information. Let r_i denote the mean service time of a job at node i, N_i denote the mean number of jobs at node i, and n_i denote the number of jobs at node i at a given instant. So, $\mu_i = \frac{1}{r_i}$.

Based on the above assumptions, Equation 2 can be written as:

$$C_i(\beta_i) = \frac{k_i p_i r_i}{(1 - r_i\beta_i)} + k_{IO} p_{IO} \left(\frac{P_1}{P_0}\right) t_{IO} \tag{3}$$

Using Little's Law, we have $N_i = \beta_i C_i(\beta_i)$. So, the above equation can be written as:

$$C_i(\beta_i) = r_i(k_i p_i + N_i) + k_{IO} p_{IO} \left(\frac{P_1}{P_0}\right) t_{IO} \tag{4}$$

The virtual node cost based on the current number of jobs at a node can be expressed (using Equation 4) as:

$$C_i(\beta_i) = r_i(k_i p_i + n_i) + k_{IO} p_{IO} \left(\frac{P_1}{P_0}\right) t_{IO} \tag{5}$$

Here, we use the number of jobs at a node at a given instant (job queue length) as the state information. The job queue lengths will be exchanged between the nodes periodically. When a new job arrives at a node, the scheduler (DCOOPS (Dynamic Cooperative Solution)) will determine whether the job has to be processed locally or should be sent to another node for remote processing. For a job arriving at node i, C_i will be compared with $C_j, j = 1, \ldots, n, j \neq i$. If $C_i > C_j \; \forall \, j = 1, \ldots, n, j \neq i$, then it is costlier for the job to be executed at node i than at a remote node. The remote node to which the job should be transferred will be computed based on the difference in the costs between the nodes. If $C_i < C_j \; \forall \, j = 1, \ldots, n, j \neq i$, the arriving job will be processed locally.

4 Performance Evaluation

A distributed computing system with 16 nodes is simulated to evaluate the performance of DCOOPS. To make the system heterogeneous, the nodes are classified into four groups with each group having a different service rate for the nodes. Six nodes have a service rate of 10 jobs/sec, five nodes have a service rate of 20 jobs/sec, three nodes have a service of 50 jobs/sec, and two nodes have a service of 100 jobs/sec. Static load balancing schemes COOPS (which provides a Pareto optimal and fair solution) and Proportional (PROP) (allocates jobs in proportion to the service rate of the nodes) [1] are also implemented for comparison purposes.

Figure 1 presents the expected cost (in some monetary unit) achieved because of the job allocation performed by COOPS, PROP, and DCOOPS for various system loads ranging from 10% to 90%. The cost in the case of PROP is high because its allocation overloads the slower nodes resulting in higher expected response time and higher cost. The cost achieved by the adaptive DCOOPS is close to that of static COOPS for low to medium system loads and is significantly lower for higher system loads.

Figure’s 2, 3, and 4 present the expected cost for executing the jobs at nodes 1 through 16 when the system load is 70%. It can be observed that the expected cost varies significantly in the case of PROP. The cost is approximately the same at all the nodes in the case of COOPS and DCOOPS which means that all the jobs incur approximately the same expected cost independent of the nodes to which they are allocated (fair allocation). The cost achieved in the case of DCOOPS is lower than that of COOPS due to its consideration of the current state information (the job queue lengths at the nodes) in making allocation decisions.

Figure’s 5, 6, and 7 present the expected cost for executing the jobs at nodes 1 through 16 when the system load is 80%. It can again be observed that the expected cost varies significantly in the case of PROP and the cost is approximately the same at all the nodes in the case of COOPS and DCOOPS. The cost achieved in the case of DCOOPS is lower than that of COOPS.

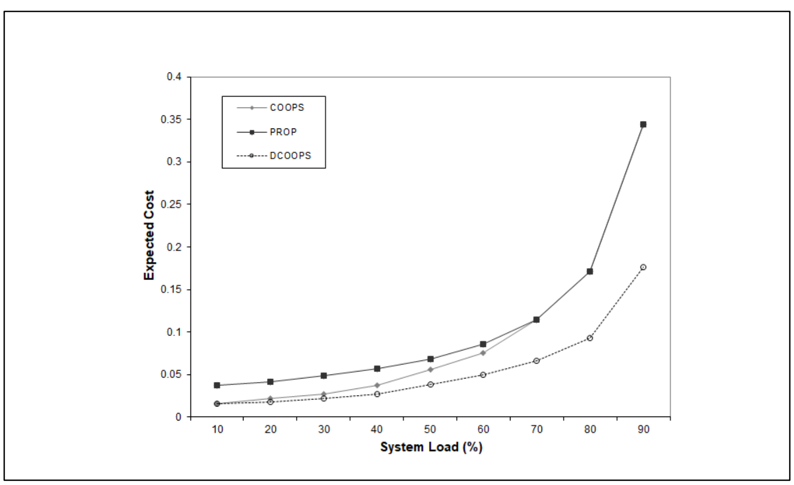

Figure 1: System Load vs Expected Cost

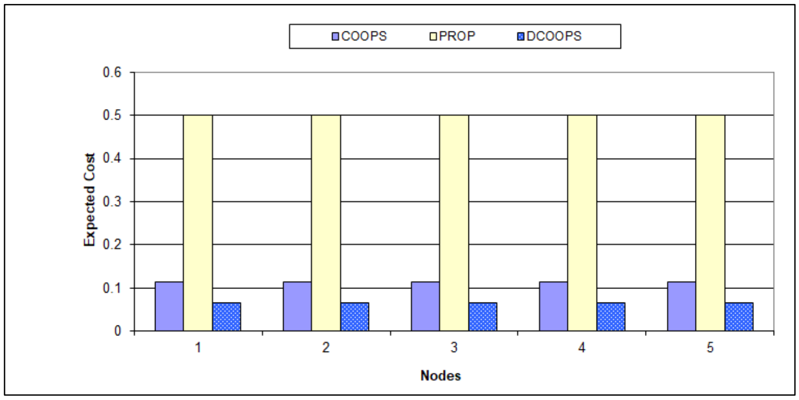

Figure 2: Expected Cost at Nodes 1 – 5 at 70% System Load

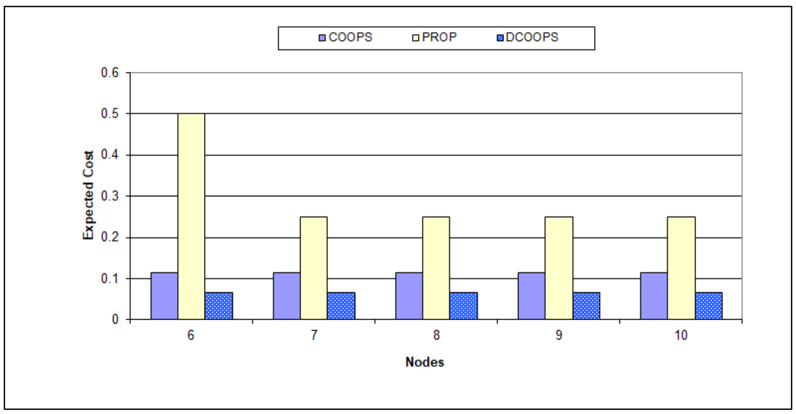

Figure 3: Expected Cost at Nodes 6 - 10 at 70% System Load

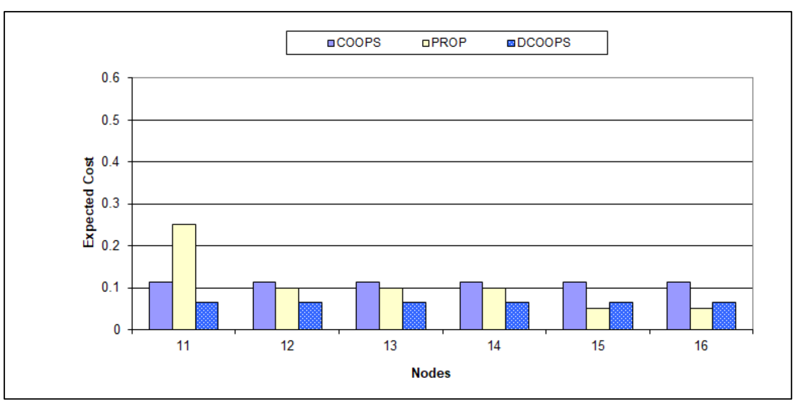

Figure 4: Expected Cost at Nodes 11 - 16 at 70% System Load

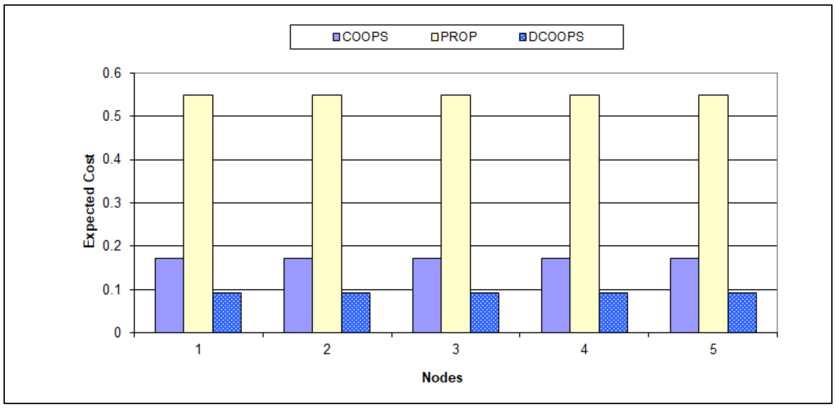

Figure 5: Expected Cost at Nodes 1 – 5 at 80% System Load

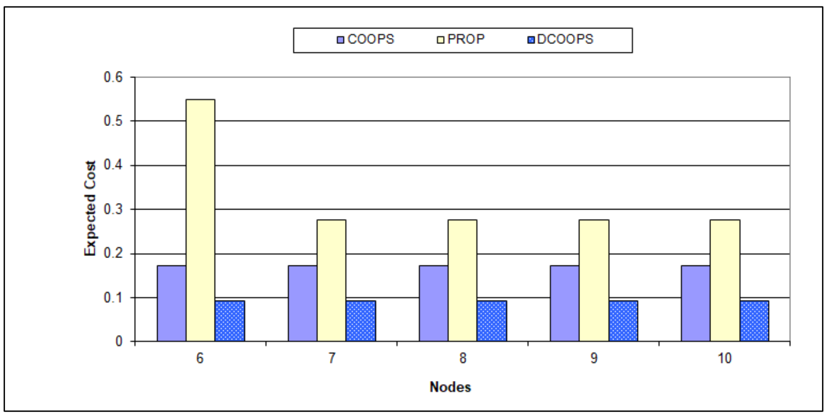

Figure 6: Expected Cost at Nodes 6 - 10 at 80% System Load

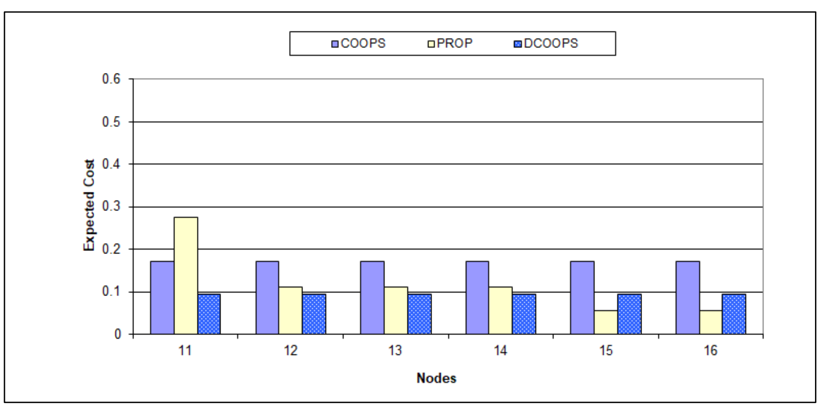

Figure 7: Expected Cost at Nodes 11 - 16 at 80% System Load

5 Conclusions

In this paper we presented an adaptive load balancing scheme, DCOOPS, which allocates the jobs to the computing nodes so as to minimize the cost for executing the jobs in the system and also to provide fairness. The current job queue lengths at the nodes is considered by DCOOPS in making allocation decisions. Based on simulations, it was observed that the cost achieved by the allocation of DCOOPS is close to that of static schemes PROP and COOPS for low to medium system loads and is significantly lower from medium to high system loads. In future work, we plan to evaluate the performance of DCOOPS by varying the system heterogeneity and the system size and compare with other adaptive load balancing schemes that have a similar objective.

References

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

[2] E. J. Ghomi, A. M. Rahmani and N. N. Qader, "*Load-balancing algorithms in cloud computing: A survey*", Journal of Network and Computer Applications, Vol. 88, pp. 50-71, Jun. 2017.

[3] P. Ghosh, N. Roy, S. K. Das, and K. Basu, "*A pricing strategy for job allocation in mobile grids using a non-cooperative bargaining theory framework*", Journal of Parallel and Distributed Computing, 65(11):1366–1383, 2005.

[4] 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 Symposium, Fort Lauderdale, Florida, USA, pp 501-510, April 14-15, 2002.

[5] E. Gures, I. Shayea, M. Ergen, M. H. Azmi and A. A. El-Saleh, "*Machine Learning-Based Load Balancing Algorithms in Future Heterogeneous Networks: A Survey*", IEEE Access, Vol. 10, pp. 37689-37717, 2022.

[6] I. N. Ivanisenko and T. A. Radivilova, "*Survey of major load balancing algorithms in distributed system*", Information Technologies in Innovation Business Conference (ITIB), Kharkiv, Ukraine, pp. 89-92, 2015.

[7] A. S. Milani, N. J, Navimipour, "*Load balancing mechanisms and techniques in the cloud environments: Systematic literature review and future trends*", Journal of Network and Computer Applications, Vol. 71, pp. 86-98, 2016.

[8] S. Penmatsa, "*Load Balancing for Providing Fairness in Utility Computing Systems with Central-Server Model*", The 30th International Conference on Computers and their Applications, Honolulu, Hawaii, USA, March 9 -11, 2015.

[9] S. Penmatsa and G. S. Hura, "*Response Time Minimization and Fairness in Distributed Systems with Central-Server Node Model Using Dynamic Load Balancing*", International Journal of Computers and their Applications, Vol. 26, No. 2, June 2019.

[10] D. A. Shafiq, N. Z. Jhanjhi, A. Abdullah, "*Load balancing techniques in cloud computing environment: A review*", Journal of King Saud University - Computer and Information Sciences, Vol. 34, Issue 7, pp. 3910-3933, 2022.

[11] 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, pp. 332–340, Phoenix, AZ, 1995.