COOPERATIVE AND NON-COOPERATIVE APPROACHES FOR LOAD BALANCING DISTRIBUTED COMPUTING SYSTEMS

Satish Penmatsa, Framingham State University, USA, spenmatsa@framingham.edu
Gurdeep S. Hura, University of Maryland Eastern Shore, USA, gshura@umes.edu

ABSTRACT

Load balancing in distributed computing systems is essential for improving their performance. With load balancing, the execution time of jobs can be reduced and fairness to the jobs and users can be provided. In this paper, we review load balancing schemes that provide fairness based on cooperative and non-cooperative game theory concepts. Cooperative game theory was used for providing fairness to all the jobs in the system and non-cooperative game theory was used for providing fairness to all the users in the system. Experimental results that compare the performance of the cooperative and non-cooperative load balancing schemes with other existing schemes are presented.

1. INTRODUCTION

Distributed computing systems comprise of multiple computing resources connected by communication networks. The computing resources may span a local area network or may spread across the globe using wide area networks. The goal of distributed computing is to make such a network of resources work as a single computer (*http://publib.boulder.ibm.com*). The advantages of distributed systems over centralized systems include scalability and availability.

Load balancing is essential for improving the performance of these distributed computing systems. With load balancing, the execution time of jobs can be reduced and fairness to the jobs and users can be provided. In this paper, we review load balancing schemes that provide fairness based on cooperative and non-cooperative game theory concepts. Cooperative game theory was used for providing fairness to all the jobs in the system and non-cooperative game theory was used for providing fairness to all the users in the system. The following load balancing schemes are reviewed: Cooperative load balancing scheme (COOP), Nash non-cooperative load balancing scheme (NASH), Cooperative load balancing scheme with communication costs (CCOOP), and Non-cooperative load balancing scheme with communication costs (NCOOPC).

The cooperative schemes provide fairness to all the jobs in the system *i.e.* they find an allocation of jobs to computers that yields an equal or approximately equal expected response time for all the jobs (of approximately the same size). The non-cooperative schemes provide fairness to all the users in the distributed system *i.e.* they find an allocation of users' jobs to computers that yields an equal or approximately equal expected response time for all the users (with jobs of approximately the same size).

2. COOPERATIVE LOAD BALANCING SCHEME (COOP)

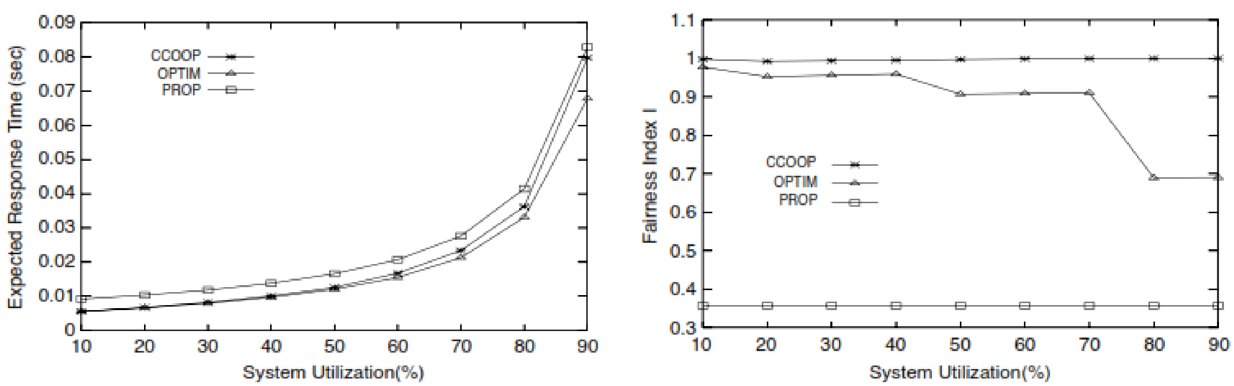
The COOP scheme is based on the cooperative game theory framework. Using this framework, the load balancing problem in single-class job distributed systems is formulated as a cooperative game among the computers in Grosu et al. (2008). The cooperative load balancing game consists of (i) computers (in the distributed system) as players; (ii) a set of strategies for each computer; (iii) an objective function for each computer; and (iv) an initial performance guarantee by each computer. The set of strategies of each computer are defined by the stability, conservation, and positivity constraints. The objective (function) of each computer is to minimize the average (expected or mean) response (execution) time of jobs executing on it.

A solution (load balancing solution) to the above cooperative game was obtained using the concept of Nash Bargaining Solution which provides a Pareto-optimal operating point for the distributed system. Here, Pareto optimality means that it is impossible to find another job allocation that leads to superior performance for one computer, without strictly decreasing the performance of another computer. Based on the Nash Bargaining Solution, the COOP load balancing algorithm is derived.

3. COOPERATIVE LOAD BALANCING SCHEME WITH COMMUNICATION COSTS (CCOOP)

The CCOOP scheme (Penmatsa et al. (2011)), is also based on the cooperative game theory framework. The load balancing problem in single-class job distributed systems is formulated as a cooperative game among the computers and the communication subsystem. The players (computers and the communication subsystem) cooperate in making decisions such that each of them will operate at its optimum. The players have complete freedom of pre-play communication to make joint agreements about their operating points. Based on the Nash Bargaining Solution (NBS) which provides a Pareto optimal and fair solution, an algorithm (CCOOP) is provided for computing the NBS for the cooperative load balancing game.

The performance of CCOOP is compared with other existing schemes (OPTIM (Kim et al. (1992)) and PROP (Chow et al. (1979))) using simulations. Figure 1 presents the results using a distributed system comprising of 32 computers ((Penmatsa et al. (2011)). From Figure 1a, it can be observed that the performance of CCOOP (in terms of expected response time) is close to OPTIM for low and medium system utilizations and approaches PROP for high system utilizations. From Figure 1b, it can be observed that the fairness index (used to quantify the fairness of load balancing schemes) of CCOOP is close to 1 for any system utilization.



(a) Expected response time. (b) Fairness index.

Figure 1. Performance Evaluation of CCOOP

4. NASH NON-COOPERATIVE LOAD BALANCING SCHEME (NASH)

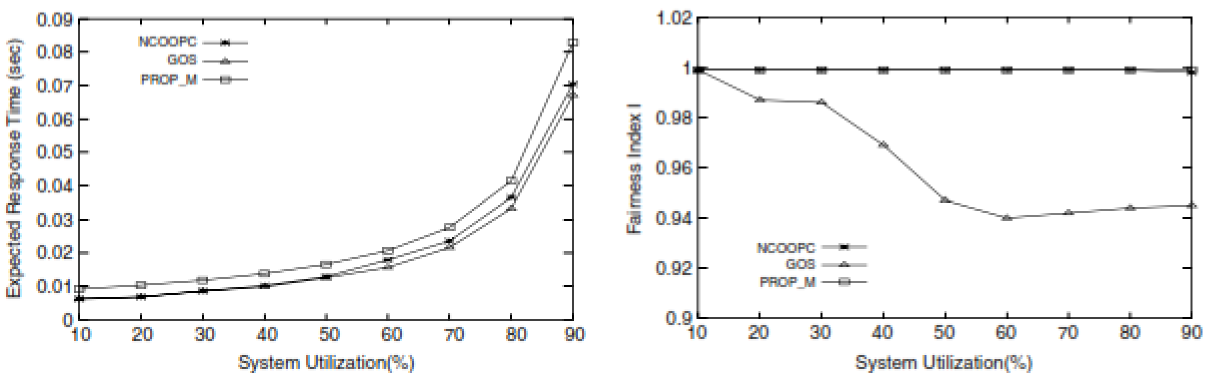
The Nash scheme is based on non-cooperative game theory. A game theoretic framework for obtaining a user-optimal solution in multi-class job distributed systems is presented in Grosu et al. (2005). The load balancing problem is formulated as a non-cooperative game among the users in the distributed system. The non-cooperative load balancing game consists of a set of players, a set of strategies, and preferences over the set of strategy profiles. Here, the players are the users in the system, the strategies are defined by the stability, conservation, and positivity constraints, and the preferences are represented by the player's expected response times. Each user prefers to have a lower expected response time than the other users.

For the above non-cooperative load balancing game, the structure of the Nash equilibrium is presented. Nash equilibrium provides a user-optimal operating point for the distributed system. Nash equilibrium for the load balancing game is a strategy profile with the property that no user can decrease her/his expected response time by choosing a different load balancing strategy given the other users' load balancing strategies. Based on the structure of the Nash equilibrium, the Nash load balancing algorithm is derived.

5. NON-COOPERATIVE LOAD BALANCING SCHEME WITH COMMUNICATION COSTS (NCOOPC)

The NCOOPC scheme (Penmatsa et al. (2011)) is also based on non-cooperative game theory framework. The load balancing problem in multi-class job distributed systems is formulated, taking into account the users' mean node delays and the mean communication delays, as a non-cooperative game among the users. Each user minimizes her/his own response time independently of the others and they all eventually reach an equilibrium. The concept of Nash equilibrium is used as the solution of the non-cooperative game and a distributed algorithm (NCOOPC) is derived for computing it.

The performance of NCOOPC is compared with other existing schemes (GOS (Kim et al. (1990)) and PROP_M (Chow et al. (1979))) using simulations. Figure 2 presents the results using a distributed system comprising of 32 computers and 20 users (Penmatsa et al. (2011)). From Figure 2a, it can be observed that the performance of NCOOPC (in terms of expected response time) is close to GOS and substantially better than PROP_M for almost any system utilization. From Figure 1b, it can be observed that the fairness index of NCOOPC is close to 1 for any system utilization.



(a) Expected response time. (b) Fairness index.

Figure 2. Performance Evaluation of NCOOPC

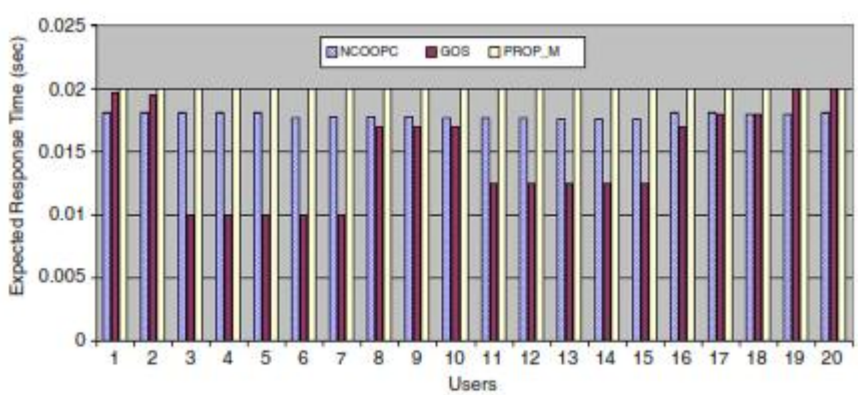


Figure 3. Expected Response Time of each User

Figure 3 presents the expected response time of each user at a system utilization of 60%. It can be observed that the response time of all users is almost the same in the case of NCOOPC. Although the response times in the case of PROP_M are also the same for all users, they are substantially higher than that provided by NCOOPC. There are large differences in the response times of the users computed by GOS.

6. CONCLUSION

In this paper, we reviewed load balancing schemes that provide fairness based on cooperative and non-cooperative game theory concepts. COOP and CCOOP which are based on cooperative game theory provide fairness to all the jobs in the system and NASH and NCOOPC which are based on non-cooperative game theory provide fairness to all the users in the system. Experimental results that compare the performance of the cooperative and non-cooperative load balancing schemes with other existing schemes are presented.

REFERENCES

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, 1979.

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

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

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

Kim C. and Kameda H (1990), Optimal static load balancing of multi-class jobs in a distributed computer system, *Proceedings of the 10th International Conference on Distributed Computing Systems*, pp. 562–569, May 1990.

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.

Penmatsa S (2011), Distributed Algorithms for providing Fairness in Heterogeneous Computer Systems, *International Journal of Science and Informatics*, 1(1): 28-33, 2011.