Abstract

Heterogeneous computing systems may comprise of computing resources with varying service rates and work loads. The performance of these systems can be improved by proper load balancing of the jobs in the system among the computing resources. In this paper, we present a dynamic load balancing scheme whose objectives are to minimize the execution time of jobs in the system and to provide fairness to the users of jobs. The computing resources are modeled as central-server nodes with one processor and one or more input/output devices. Simulations are conducted with various heterogeneous system configurations to evaluate the performance of the presented dynamic load balancing scheme.

Keywords: Heterogeneous systems; fairness; dynamic load balancing.

1 Introduction

Distributed computing systems comprise of software components spread over multiple computers for improving efficiency and performance of applications. These distributed systems provide several benefits compared to centralized systems such as scalability and improved availability of services. Since the computers in these systems may consist of heterogeneous computing and communication resources with varying work loads, the performance of these systems can be impacted if proper load balancing of jobs is not performed. Given a large number of jobs, the load balancing schemes try to find an allocation of jobs to computers in the system optimizing an objective function (e.g. total execution time of jobs). Static load balancing schemes base their allocation decisions on collected statistical information about the system, where as, the dynamic (adaptive) load balancing schemes base their decision on the current state of the system [13].

Static optimal load balancing in distributed systems considering various network configurations (e.g. single channel and star network configurations) is studied in [6]. Dynamic load balancing policies considering single-channel network configuration with central-server node model have also been presented in [6]. All jobs in the system were regarded to belong to one group (e.g. one user/class). In central-server model, each node consists of one processor and one or more input/output (I/O) devices. Static and dynamic load balancing schemes with the objective of providing a system-optimal solution for multi-class (multi-user) jobs considering a central-server node model were studied in [11] and references there-in.

Fairness of allocation is an important factor in modern distributed systems and we can say that a load balancing scheme is fair if all the users in the system have approximately the same expected response time independent of the computers allocated for the execution of their jobs (of approximately the same size). In [10], we studied dynamic load balancing for minimizing execution costs of user jobs and for providing fairness to users in grid systems. The nodes were modeled as M/M/1 queuing systems [4] where the job inter-arrival times and service times are exponentially distributed and arrive in a single queue to a single computing resource. Static load balancing schemes for providing fairness considering a central-server node model have been presented in [8, 9]. Load balancing schemes based on game theory for providing fairness to users and their jobs have been presented in [1, 7, 2].

A classification of some most used load balancing algorithms in distributed systems (including cloud technology, cluster systems, and grid systems) is presented in [3]. Another survey of task allocation and load balancing in distributed systems with respect to aspects such as control models, resource optimization methods, and coordination mechanisms among heterogeneous nodes has been provided in [5]. In [12], performance analysis of greedy load balancing algorithms in heterogeneous distributed computing systems has been made.

Here, we present a dynamic load balancing scheme
whose objectives are to minimize the execution time of user jobs and to provide fairness to the users in a multi-user heterogeneous distributed system. The computing resources are modeled as central-server nodes with one processor and one or more I/O devices. Simulations are conducted with various heterogeneous system configurations to evaluate the performance of the presented dynamic load balancing scheme.

2 System Model

We consider a heterogeneous distributed system with \( n \) computers (nodes) connected by a communications network with jobs arriving from \( m \) different users to the nodes (similar to [11, 9]). We assume that \( \mu_i \) denotes the service rate of node \( i \) and \( \phi_j \) denotes the external job arrival rate into the system is denoted by \( \phi \). Each node is modeled as a central-server node as shown in Figure 1. \( p_0 \) denotes the probability that a job after departing from the processor finishes and \( p_1 \) denotes the probability that a job after departing from the processor requests I/O service. Therefore, \( \frac{p_1}{p_0} \) 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 \( x_{rs} \). The nodes and the communications network are assumed to have an exponential service-time distribution and that the job arrivals follow a Poisson distribution [4].

![Node Model](image)

Let \( \beta_i \) denote the job processing rate (load) of user \( j \) at node \( i \) and \( t_{IO} \) denote the service time of an I/O device. Based on the above assumptions, the expected response time (node delay) of a user \( j \) job processed at node \( i \) is given by [4]:

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

where \( \beta_i = [\beta_1^i, \beta_2^i, ..., \beta_m^i]^T \).

Let \( \lambda^j \) denote the job traffic through the network of user \( j \) and \( t \) denote the mean communication time for sending or receiving a job from one node to another for any user. The expected communication delay of a user \( j \) job is given by [4]:

\[
G^j(\lambda) = \frac{t}{(1 - t \sum_{k=1}^{m} \lambda^k k)} \sum_{k=1}^{m} \lambda^k < \frac{1}{t} \tag{2}
\]

In the above, it is assumed that \( G^j(\lambda) \) is independent of the source-destination pair \((r, s)\) but may depend on the total traffic through the network, \( \lambda \) where \( \lambda = \sum_{j=1}^{m} \lambda^j \). The overall average (expected) response time of user \( j \) is given by:

\[
T^j(\beta, \lambda) = \frac{1}{\phi^j} \sum_{i=1}^{n} \beta_i^j F_i^j(\beta_i) + \frac{\lambda^j}{\phi^j} G^j(\lambda) \tag{3}
\]

3 Dynamic Load Balancing

Dynamic load balancing schemes base their decision on the current state of the system. In this section, we present a dynamic load balancing scheme (named DNCOOPC-CS) whose objectives are to dynamically minimize the execution time (response time) of users (jobs) in the system and to provide fairness to all the users. We note that an allocation (of jobs) is said to be fair if all the users experience approximately the same expected response time for the execution of their jobs which are approximately of the same size independent of the computers allocated for their execution.

The DNCOOPC-CS scheme is based on the static job allocation scheme NCOOPC-CS presented in [9] whose objective is to improve the performance of E-commerce systems by minimizing the response time of users jobs (or transactions) and by providing fairness to all the users. NCOOPC-CS is based on non-cooperative economic game theory and it was shown that an (Nash) equilibrium solution provides an allocation which is fair to all the users. The following marginal node and marginal communication delay functions are defined in [9] for finding a solution implemented by NCOOPC-CS job allocation scheme.

\[
f_i^j(\beta_i) = \frac{\partial}{\partial \beta_i^j} [\beta_i^j F_i^j(\beta_i)] = \frac{\mu_i^j}{(\mu_i^j - \beta_i^j)^2} + \frac{p_1}{p_0} t_{IO} \tag{4}
\]
where $\mu_i = \mu_i - \sum_{k=1, k \neq j}^m \beta^k_i$.

$$g^j(\lambda) = \frac{\partial}{\partial \lambda} [\lambda^i G^j(\lambda)] = \frac{tg_{-j}}{(g_{-j} - t\lambda)^2}$$  \hspace{1cm} (5)

where $g_{-j} = (1 - t \sum_{k=1, k \neq j}^m \lambda^k)$.

$$\left(\mu_i^j - \sqrt{\frac{\mu_i^j}{\rho_i^j t_{1O}}}\right), \hspace{1cm} \text{if} \ x > \frac{1}{\mu_i^j} + \frac{\rho_i^j}{\rho_i^j t_{1O}}$$  \hspace{1cm} (6)

$$(f_i^j)^{-1}(\beta_i|_{\beta_i^j = x}) = \left\{ \begin{array}{ll}
0, \hspace{1cm} \text{if} \ x \leq \frac{1}{\mu_i^j} + \frac{\rho_i^j}{\rho_i^j t_{1O}}
\end{array} \right.$$

The state information that DNCOOPC-CS uses is the number of jobs waiting in queue to be processed (queue length) at the nodes. Each node $i$ ($i = 1, \ldots, n$) broadcasts the number of jobs of user $j$ ($j = 1, \ldots, m$) in its queue to all the other nodes. This state information exchange is done periodically, say every $P$ time units. Expressions for marginal node and communication delays in terms of current state information (instantaneous variables) are derived below (similar to [11, 10]).

Let $r_i$ ($i = 1, \ldots, n$) denote the mean service time of a job at node $i$, $N_i^j$ ($i = 1, \ldots, n; j = 1, \ldots, m$) denote the mean number of jobs of user $j$ at node $i$, $\rho$ denote the utilization of the communications network (where $\rho = t \sum_{k=1}^m \lambda^k$), and $\rho^j$ denote the utilization of the communications network by user $j$.

Using the relation $r_i = \frac{1}{\mu_i}$ [4] and Little’s law $(\sum_{k=1}^m N_i^k = \sum_{k=1}^m \beta^k_i F_i^j(\beta_i^j))$ [4], the marginal node delay of a user $j$ job at node $i$ (i.e. $f_i^j$ in eq. (4)) can be expressed in terms of $r_i$ and $N_i^j$ as:

$$f_i^j(\beta_i) = r_i^j(1 + \sum_{k=1}^m N_i^k)^2 + \frac{p_i}{p_0} t_{1O} \hspace{1cm} (7)$$

where $r_i^j = r_i - \sum_{k=1, k \neq j}^m \beta^k_i$.

Rewriting eq. (5) in terms of $\rho$, we have

$$g^j(\lambda) = \frac{t(1 - \rho + \rho^j)}{(1 - \rho)^2}, \hspace{1cm} \rho < 1, \rho^j < 1 \hspace{1cm} (8)$$

4 Experimental Results

We simulated a 32 node heterogeneous system with 20 users to evaluate the performance of DNCOOPC-CS load balancing scheme. The system has computers with 4 different service rates and the service rate of the fastest computers is 10 times that of the slowest computers (similar to [11, 10]). For comparison, the following load balancing schemes were also implemented: NCOOPC-CS [9] - a static scheme with the objective of minimizing the total execution time of users jobs and provide fairness to the users; DynamicGOS [11] - a dynamic load balancing scheme with the objective of providing a system optimal solution (but not fairness). 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. The overhead (OV) for job transfer we use in the following is defined as the percentage of service time that a computer has to spend to send or receive a job.

The average response times (expected response times) for executing the jobs of all users in the system for
system utilizations (system loads) ranging from 10% to 90% are presented in Figure 2. We assume that there is no overhead for job transfer in this case. It can be observed that the average response time achieved by all the schemes is close for low system utilizations. As the system load increases, the average response time increases, and the average response times achieved by the dynamic schemes, which use the instantaneous state information, are substantially lower (by at least about 40%) than that of the static NCOOPC-CS. Dynamic-GOS, whose objective is to provide a system optimal solution achieves a lower response time compared to DNCOOPC-CS, whose objective is not only to reduce the response time of jobs but also provide fairness to the users.

We use Fairness Index (FI) [4] as the metric to quantify the fairness of the load balancing schemes. If all the users have the same average response time, then \( FI = 1 \) and the system is 100% fair to all users and it is load-balanced. If FI decreases, then the load balancing scheme favors only some users. The Fairness Index (FI) of the load balancing schemes under consideration are presented in Figure 3 for various system utilizations. It can be observed that the FI of DynamicGOS falls from 1 at low system loads to about 0.93 at high system loads while the FI of DNCOOPC-CS is in the range \( \{0.97, 1\} \).

Figure 4 presents the average response times achieved by the load balancing schemes when the overhead for sending and receiving a job is set to 5% of the mean job service time at a node. It can be observed that the response times of DNCOOPC-CS are substantially lower than that of the static NCOOPC-CS for medium and high system loads. Figure 5 presents the Fairness Index (FI) achieved by the schemes when the overhead for sending and receiving a job is 5% of the mean job service time at a node. It can be observed that the Fairness Index of DNCOOPC-CS is in the range \( \{0.97, 1\} \) and the Fairness Index of DynamicGOS is in the range \( \{0.92, 0.99\} \).
Figure 5 presents the average response times achieved by the load balancing schemes when the overhead for job transfer is 10%. Although the response times increase considerably, it can be again observed that the response times of DNCOOPC-CS are lower than that of the static NCOOPC-CS for medium and high system loads. Figure 7 presents the Fairness Index (FI) when the overhead for job transfer is 10%. It can be observed that the FI of DNCOOPC-CS is in the range \( \{0.97, 1\} \) and the FI of DynamicGOS is in the range \( \{0.89, 0.99\} \).

From the above it can be observed that the performance of DNCOOPC-CS is not only close to that of DynamicGOS in terms of the average response time but also provides fairness to all the users (in terms of their experienced response times).

### 5 Conclusions

In this paper, a dynamic load balancing scheme (DNCOOPC-CS) for heterogeneous distributed systems is presented. The objectives of DNCOOPC-CS are to minimize the average response time of users jobs and to provide fairness to the users. The computers were modeled as central-server nodes. Based on experimental results with various system loads, it was observed that the Fairness Index achieved by DNCOOPC-CS is close to 1 and the average response times achieved by it were considerably lower than its static counterpart and comparable to a system-optimal dynamic scheme.

In future work, we plan to evaluate the performance of DNCOOPC-CS by varying the heterogeneity and the size of the distributed system.

### References


