Probabilistic Approach to Scheduling Divisible Load on Network of Processors

Manar Arafat¹, Sameer Bataineh²* and Issa Khalil³

¹Department of Computer Science, An-Najah National University, Nablus, Palestine
²Faculty of Computer and Information Technology, Jordan University of Science and Technology, Jordan
³Qatar Foundation, Qatar

Abstract

Divisible Load Theory (DLT) is a very efficient tool to schedule arbitrarily divisible load on a set of network processors. Most of previous work using DLT assumes that the processors' speeds and links' speeds are time-invariant. Closed form solution was derived for the system under the assumption that the processors' speed s and the links' speeds stay the same during the task execution time. This assumption is not practical as most of distributed systems used today have an autonomous control. In this paper we consider a distributed system (Grid) where the availability of the processors varies and follows a certain distribution function. A closed form solution for the finish time is derived. The solution considers all system parameters such as links' speed, number of processors, number of resources (sites), and availability of the processors and how much of power they can contribute. The result is shown and it measures the variation of execution time against the availability of processors.

Keywords: Divisible load theory; Scheduling; Availability; Distributed systems

Definitions and Notations

N: The number of available data sources (sites)
M: The number of available nodes to process the grid tasks
Sᵢ: Data source (site). i=1…N
Pᵢ: Available nodes to process the grid tasks. i=1…M
L: Total load at the originating load node
Cᵢ: Computational speed of node i in the network
wᵢ: A constant that is inversely proportional to the computational speed of node i, wᵢ = 1/Cᵢ
zᵢ: A constant that is inversely proportional to the link speed between site i and node j.
fᵢ: The fraction of the load L assigned to data source i=1 … N
Lᵢ: The load assigned to data source i=1, … N
αᵢ,j: The fraction of load assigned from site i to node j.
Tᵢ: The time it takes the i th node to process the entire load when wᵢ=1. The entire load can be processed on the i th processor in time wᵢTᵢ.
Tᵢ: The time it takes to transmit the entire load over a link when zᵢ=1. The entire load can be transmitted over the i th link in time zᵢTᵢ.
Tᵢ: Time to finish processing αᵢ,j.
γᵢ,j: The probability of finding processor j is willing to contribute its whole computing power at time t during αᵢ,j wᵢTᵢ, where (i=1…N) and (j=1…M).
Tᵢ: The time it takes to send αᵢ,j.
Tᵢ: The time it takes to process αᵢ,j.
Tᵢ: The minimum expected finish time for the total load.

Cᵢ: The equivalent communication power for site i, (i=1,2, … N).
Cᵢ: The total communication power in the system.

Introduction

Geographically distributed heterogeneous systems become very powerful and popular in the past decade. Good examples of such systems are Clusters, Grids and Clouds. Grid can be viewed as a distributed large-scale cluster computing. From another perspective, it constitutes the major part of Cloud Computing Systems in addition to thin clients and utility computing [1-4]. Hence, Grid computing has attracted many researchers [5]. The interest in Grid computing has gone beyond the paradigm of traditional Grid computing to a Wireless Grid computing [6].

There are many attempts to find an analytical solution for scheduling load on the nodes (processors) on those systems. Queuing theory is a very famous tool participated to find analytical solution on such system [7,8]. Divisible Load Theory (DLT), deterministic in nature, was also used and proved that it is very much the same as Markov Chain Modeling [9]. However, DLT has shown that it is an excellent tool to schedule independent jobs on a Grid originated from multiple resources [7,10-14].

As Scheduling plays an important role in determining the performance of Grids, there are many algorithms in literature that discuss the scheduling on Grids [15-23]. It was shown that finding an...
analytical solution for general scheduling problem in a Grid is a very
difficult task [24].

There are several attempts to use the DLT to model scheduling
arbitrarily divisible load on the Grid [25-27]. However, communication
time is rarely considered [11]. In [7], communication time is studied
but not in dividing the load, so the transfer input time of the load
was not part of the model. Though all parameters of the system were
considered in [10], the paper did not provide a closed form analytical
solution for the finish time.

In our previous work in [28], we managed to come up with
closed form solution for the minimum finish time of executing an
arbitrarily divisible application on the Grid taking into consideration
the communication time and the computation time simultaneously.
In [28], we assumed that the nodes are always available to execute the
grid task. In distributed environments, this assumption is unrealistic.
A node may not be willing to contribute its whole computing power
during the time span of a grid task execution.

The objective of this paper is to develop an analytical model to
distribute the grid load from multiple sites to all nodes in the Grid such
that the load is executed in a minimum time. The model deals with
the dynamic availability of each node to serve the Grid tasks. We will
derive closed form solutions for the load fraction of each node, and the
minimum expected finish time of the total load. The solution considers
all system parameters such as the links’ speed, number of processors,
number of resources (sites), and availability of the processors and how
much of their power they can contribute.

The rest of the paper is organized as follows: In section 5 we present
the model of the system, in section 3 we present the notations used
throughout the paper. The system equations are in section 6. The
results and discussion are in section 7. Finally the concluding remarks
are in section 8.

System Model

The system has three types of nodes: originating load node, N data
sources (or sites) Si, i=1,..., N, and M available nodes Pi, i=1,...,M to
process the grid tasks. Nodes have different speeds wi, i=1,2,...,M. The
links that connect the data sources with the nodes have also different
speeds zi, i=1,2,...,N, j=1,2,...,M.

The system works as shown in Figure 1. The originating node
receives the total load L and distributes it to N available sites. The
fraction of the load to be assigned to each site is fi, i=1 ... N. It follows
that the share of the load Li to be assigned to data source i is given by

\[ L_i = \frac{1}{C_i} \sum_{j=1}^{N} \frac{F_i}{Z_{ij}} \]

where \( C_i \) is the total communication power in the system and \( Z_{ij} \) is
the communication time between site i and node j.

In section 6, first we obtain the fraction of load that has to be
assigned to each site from the load originating node. Each site will
be assigned a fraction that depends on the equivalent communication
power Ci of the site. Consequently, the fraction of the load to be
assigned to site i is \( f_i = \frac{C_i}{C_T} \) for i=1 ... N, such that \( \sum_{i=1}^{N} f_i = 1 \). It follows
that the share of the load \( L_i \) to be assigned to each site i=1...N is given by

\[ L_i = f_i L \]
Next, we obtain the fraction of load that has to be assigned from each site to each of the available nodes in the network. The assumption that computational resources at nodes are dedicated to Grid tasks is impractical. Each of the M nodes may not be willing to contribute its whole computing power during the time span of a grid task execution. Each node $P_j (j=1...M)$ will be assigned a load fraction that depends on the speed of the links, the speed of the nodes and the dynamic availability of each node in the grid. So it is not possible to take advantage of the equivalent power concept at this stage.

In Figure 3, each data source has the illusion as if it is the only data source that is distributing its load to the M available nodes. In other words, we can view the system as N multiples of a single data source as shown in Figure 4.

In Figure 4, each site will distribute its load to M available nodes such that the total load is executed in a minimum time. The solution is based on the optimality principle [30]. Optimality solution assumes that all processors finish at the same time. It is analytically proved that a minimal solution time is achieved when the computation by each node finishes at the same time [31]. Intuitively, this is because otherwise, the processing time could be reduced by transferring some fractions of load from busy nodes to idle nodes [32-36].

We now derive closed form solution for the load fraction of each node, and the minimum expected finish time of the total load. The solution is based on the optimality principle [37]. Optimality solution assumes that all processors finish at the same time.

In general, the time to complete execution of $\alpha_{ij}$ on node $j$ consists of the communication time $c_{ij}T(t)$ and processing time $p_{ij}T(t)$ divided by the probability of finding node $j$ available during $T(t)$

$$T_0 = T_0 + \frac{T(t)}{P_j(t)} \quad \text{Where } i=1,2,\ldots, N ; j=1,2,\ldots M$$

(4)

$$T_0 = \alpha_iZ_iT_{on} + \frac{w_jT(t)}{P_j(t)}$$

(5)

Let $i=1$, Then the system equations are:

$$T_0 = \alpha_1(Z_1T_{on} + \frac{w_jT(t)}{P_j(t)})$$

(6)

Applying the optimality criterion that all processors should stop computing at the same time [31,32]:

$$\alpha_{ij} = \alpha_{1i} = \frac{P(t)Z_iT_{on}P(t) + w_jT(t)}{P(t)Z_iT_{on}P(t) + w_jT(t)}$$

(7)

From equation (7), we obtain equations (8), (9), and (10) respectively.

$$\alpha_{1j} = \alpha_{1i} = \frac{P(t)Z_iT_{on}P(t) + w_jT(t)}{P(t)Z_iT_{on}P(t) + w_jT(t)}$$

(8)

$$\alpha_{2j} = \alpha_{1i} = \frac{P(t)Z_iT_{on}P(t) + w_jT(t)}{P(t)Z_iT_{on}P(t) + w_jT(t)}$$

(9)

$$\alpha_{ij} = \alpha_{1i} = \frac{P(t)Z_iT_{on}P(t) + w_jT(t)}{P(t)Z_iT_{on}P(t) + w_jT(t)}$$

(10)

Now we can write all the above equations as a function of $\alpha_{ij}$ and the parameters of the Grid as follows:

$$\alpha_{ij} = \alpha_{1i} = \frac{P(t)Z_iT_{on}P(t) + w_jT(t)}{P(t)Z_iT_{on}P(t) + w_jT(t)}$$

(11)

$$\alpha_{ij} = \alpha_{1i} = \frac{P(t)Z_iT_{on}P(t) + w_jT(t)}{P(t)Z_iT_{on}P(t) + w_jT(t)}$$

(12)

In general,

$$\alpha_{ij} = \alpha_{1i} = \frac{P(t)Z_iT_{on}P(t) + w_jT(t)}{P(t)Z_iT_{on}P(t) + w_jT(t)} \quad \text{Where } j=1,2,\ldots, M$$

(13)

Obviously, the summation of all fractions of the load must equal 1.
\[
\alpha_1 + \alpha_2 + \alpha_3 + \ldots + \alpha_{M} = 1
\]  
(14)

Using equations (13) and (14), we can find the exact value of \( \alpha_1 \) as a function of grid parameter

\[
\alpha_1 = \frac{1}{1 + \sum_{i=1}^{M} \rho_i (T + w_i / T_p) + \sum_{j=1}^{N} \rho_j (T + w_j / T_p)}
\]  
(15)

Since \( \alpha_1 \) is found, \( \alpha_2, \alpha_3, \ldots, \alpha_{M} \) can be calculated using equation (13) and the minimum expected finish time can also be determined using equations (6) and (15):

\[
T_{\text{finish}} = T_i \frac{1}{\sum_{j=1}^{M} \rho_j (T + w_j / T_p)}
\]  
(16)

Results and Discussion

In this section we demonstrate that the analytical model obtained is correct in that it produces results that are in agreement with intuitively expected results. The results are compared with the deterministic systems discussed in all previous work [7,10-14]. The total load in the system is found in \( N \) load sources. Each load source will distribute its load fraction \( f_j (k=1\ldots N) \) to the \( M \) available nodes in the network. The availability of each node \( i \) \((i=1\ldots M)\) varies with time and is based on certain probability \( P_i(t) \). If \( P_i(t)=1 \), the total computing power of node \( j \) is available and if \( P_i(t)=0.5 \), half of the total computing power of node \( j \) can be lent to the newly arriving job. In general \( P_i(t) \) of the computing power of node \( j \) can be devoted to a newly arriving job. The probability function can be derived from a certain realistic distribution. An estimate of \( P_i(t) \) can be measured over a reasonable period of time. As an example let us think of each processor as an M/M/1 queue where the customers arrive according to independent Poisson processes with rate \( \lambda \) and the service times of all customers are exponentially distributed with mean \( 1/\mu \). We require that \( \rho = \lambda/\mu < 1 \), since, otherwise, the queue length will explode. The quantity \( \rho \) is the fraction of time the server is working. From the equilibrium probabilities we can derive expressions for the mean number of customers in the system \( E(l) \), which is:

\[
E(l) = \frac{\rho}{1 - \rho}
\]

If the power of a processor \( j \) is uniformly distributed among all jobs in the queue of processor \( j \), then the speed of the processor \( j \) at an instant of time devoted to each job served by processors \( j \) is given by the following equation

\[
p_j(t) = \frac{1}{E_i(t)} = \frac{1 - \rho}{\rho}
\]

One can further consider an M/M/1 system serving different types of customers. As a simple example assume that there are two types of jobs only, type 1 and 2, but the analysis can easily be extended the situation with more types of customers. Type 1 and type 2 customers arrive according to independent Poisson processes with rate \( \lambda_1 \) and \( \lambda_2 \), respectively. The service times of all customers are exponentially distributed with the same mean \( 1/\mu \). One can easily find a \( p_j(t) \) even if a preemptive resume priority rule is applied.

In general an explicit solution for the probabilities \( p_j(t) \) is

\[
p_j(t) = \sum_{k=1}^{n} k \times p_k(t)
\]

Where \( p_k(t) \) is the probability that at time \( t \) there are \( k \) jobs in the system which can handle at most \( n \) jobs. \( p_j(t) \) equation can be found in [38].

In the following we use the equations derived in the paper to study the effect of different parameters on the system performance. The results are compared with the deterministic case. Deterministic results are those obtained with \( t=1 \).

In Figures 5 and 6, we assume that the available nodes in the network will have the same probability of availability to serve the grid loads. Comparing the results with the deterministic case, when the nodes devoted all power to one job, in other words when the probability estimate = 1, we notice that the expected finish time could be 5 times slower when the system has a large number of processors.

Figure 6 relates the expected finish time to the estimate of probability for \( M=100 \) and \( z_{ij} = 0.1, 0.5, 1, 1.5 \) and 2 (for \( i=1\ldots N \) and \( j=1\ldots M \)), assuming that \( w_j = 1 \) (for \( i=1\ldots N \) and \( j=1\ldots M \)), and \( T_{cp} = T_{cm} =1 \).

Figure 6 shows that as the number of available nodes in the grid increases, the expected finish time will decrease. For our parameters, there will be no significant improvement after \( M=150 \). The minimum expected finish time occurs when all the nodes are available to serve the grid loads. Comparing the results with the deterministic case, when the nodes devoted all power to one job, in other words when the probability estimate = 1, we notice that the expected finish time could be 5 times slower when the system has a large number of processors.

Figure 7 relates the estimate of probability for node \( k \), (\( k \) may be any node \( 1\ldots M \)), to its load fraction \( \alpha_{k} \) assigned from load source \( i \), for \( M=25, 50, 100 \), and 150. Assuming that the estimate of probability of each other available node is equal to one and that \( z_{ij} = \lambda_{ij} \) for \( i=1\ldots N \) and \( j=1\ldots M \), and \( T_{cp} = T_{cm} =1 \).

From Figure 7, nodes that are available for serving the grid loads...
most of the time will be assigned larger load fractions. This is exactly what is expected because we assume that all processors must stop at the same time to abide with the optimality principle.

Conclusion

Unlike all previous work, which using DLT, in this paper we propose a model where the processors speeds is a function of number of jobs that a node in the distributed system is in charge at time \( t \). We assumed that the availability of a node is based on certain probability and it varies with time. Of course, the probabilistic function should be derived from a certain realistic distribution.

The load is found in \( N \) sites and has to be distributed and assigned to \( M \) nodes such that total load is executed in a minimum time. We derived closed form solution for the load fraction of each node, and the minimum expected finish time of the total load. The result is shown and it measures the variation of execution time against the availability of processors for different system parameters. Our next step is to find a distribution function that can apt the system behavior. This will generate an accurate estimate for the probabilities to precisely reflect the availability of the processors in the system.

References


