

# Probabilistic Approach to Scheduling Divisible Load on Network of Processors

Manar Arafat<sup>1</sup>, Sameer Bataineh<sup>2\*</sup> and Issa Khalil<sup>3</sup>

<sup>1</sup>Department of Computer Science, An-Najah National University, Nablus, Palestine

<sup>2</sup>Faculty of Computer and Information Technology, Jordan University of Science and Technology, Jordan

<sup>3</sup>Qatar 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

$S_i$ : Data source (site).  $i=1 \dots N$

$P_i$ : Available nodes to process the grid tasks.  $i=1 \dots M$

$L$ : Total load at the originating load node

$cs_i$ : Computational speed of node  $i$  in the network

$w_i$ : A constant that is inversely proportional to the computational speed of node  $i$ ,  $w_i = 1/cs_i$

$z_{ij}$ : A constant that is inversely proportional to the link speed between site  $i$  and node  $j$ .

$f_i$ : The fraction of the load  $L$  assigned to data source  $i=1 \dots N$

$L_i$ : The load assigned to data source  $i=1, \dots, N$

$\alpha_{ij}$ : The fraction of load assigned from site  $i$  to node  $j$ .

$T_{cp}$ : The time it takes the  $i^{th}$  node to process the entire load when  $w_i = 1$ . The entire load can be processed on the  $i^{th}$  processor in time  $w_i T_{cp}$ .

$T_{cm}$ : The time it takes to transmit the entire load over a link when  $z_i = 1$ . The entire load can be transmitted over the  $i^{th}$  link in time  $z_i T_{cm}$ .

$T_{ij}$ : Time to finish processing  $\alpha_{ij}$ .

$P_j(t)$ : The probability of finding processor  $j$  is willing to contribute its whole computing power at time  $t$  during  $\alpha_{ij} w_j T_{cp}$ , where  $(i=1 \dots N)$  and  $(j=1 \dots M)$ .

$T_{ij}^c$ : The time it takes to send  $\alpha_{ij}$ .

$T_{ij}^p$ : The time it takes to process  $\alpha_{ij}$ .

$T_{finish}$ : The minimum expected finish time for the total load.

$C_i$ : The equivalent communication power for site  $i$ , ( $i=1, 2, \dots, N$ ).

$C_T$ : 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

**\*Corresponding author:** Sameer Bataineh, Faculty of Computer and Information Technology, Jordan University of Science and Technology, Jordan, Tel: +96227201000; E-mail: [samir@just.edu.jo](mailto:samir@just.edu.jo)

**Received** October 28, 2015; **Accepted** November 24, 2015; **Published** November 27, 2015

**Citation:** Arafat M, Bataineh S, Khalil I (2015) Probabilistic Approach to Scheduling Divisible Load on Network of Processors. Sensor Netw Data Commun 4: 130. doi:10.4172/2090-4886.1000130

**Copyright:** © 2015 Arafat M, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

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)  $S_i, i=1, \dots, N$ , and M available nodes  $P_j, j=1, \dots, M$  to process the grid tasks. Nodes have different speeds  $w_j, j=1, \dots, M$ . The links that connect the data sources with the nodes have also different speeds  $z_{ij}, i=1, \dots, N, j=1, \dots, M$ .

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  $f_i, i=1 \dots N$ . It follows that the share of the load L to be assigned to data source i is given by  $L_i = f_i L, i=1 \dots N$ . Then, each data source will distribute its load  $L_i$  to the M available nodes for processing. The fraction of the load to be assigned to each node j from site i is  $\alpha_{ij}, (i=1, 2, \dots, N) \& (j=1, 2, \dots, M)$ .

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 speed of the links in the network. Second, we obtain the fraction of load that has to be assigned from each site to each of the available nodes in the network. The fractions depend on the speed of the links, the speed of the nodes and the dynamic availability of each node in the grid.

Our analytical model guarantees that the total load is executed in a minimum time. We derive closed form solution for the minimum expected finish time of the total load.

### System Equations

Using the Equivalent Processor and Communication Link Concept, which was first introduced in [29], we can replace the set of M nodes in Figure 1 with a single node, which has an equivalent computing power. In other words, if the M nodes are all removed and replaced by the equivalent node, the performance of the system will be exactly the same. We can also replace the M links from each data source by one link which has an equivalent communication power  $C_i, (i=1, 2, \dots, N)$  as depicted in Figure 2. The value of  $C_i$  is given by:

$$c_i = \sum_{j=1}^M \frac{1}{Z_{ij}} \quad i=1 \dots \dots N \tag{1}$$

The total communication power in the system is

$$c_T = \sum_{j=1}^M \frac{1}{Z_{1j}} + \sum_{j=1}^M \frac{1}{Z_{2j}} + \dots + \sum_{j=1}^M \frac{1}{Z_{Nj}} = \sum_{i=1}^N C_i \tag{2}$$

The distribution of the total load L from the originating node to the N sites depends on the links speed to the equivalent node. Each site  $S_i$  will be assigned a fraction that depends on the equivalent communication power  $C_i, (i=1, 2, \dots, N)$  and the total communication power in the system  $C_T$ . Consequently, the fraction of the load to be assigned to site i is  $f_i = \frac{C_i}{C_T}$  for  $i=1 \dots N$ , 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  $i=1 \dots N$  is given by:

$$L_i = f_i L \tag{3}$$

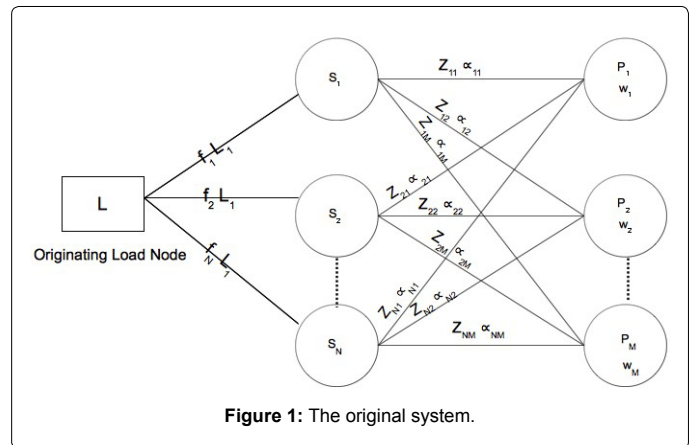


Figure 1: The original system.

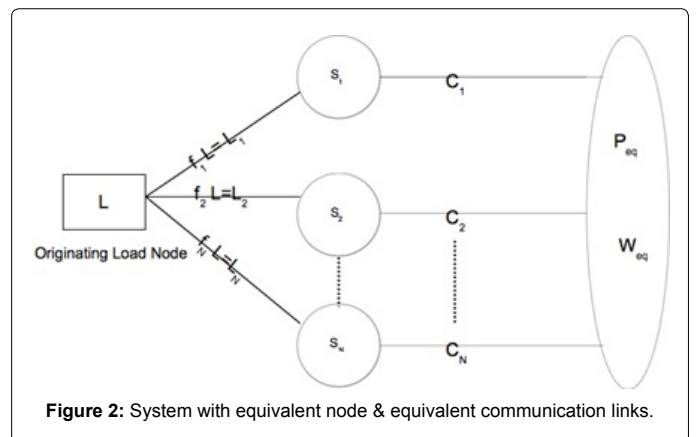


Figure 2: System with equivalent node & equivalent communication links.

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  $P$  ( $j=1..M$ ) 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  $T_{ij}^c$  and processing time  $T_{ij}^p$  divided by the probability of finding node  $j$  available during  $T_{ij}^p$

$$T_{ij} = T_{ij}^c + \frac{T_{ij}^p}{p_j(t)} \quad \text{Where } i=1,2,\dots, N; j=1,2,\dots, M \quad (4)$$

$$T_{ij} = \alpha_{ij} Z_{ij} T_{cm} + \frac{\alpha_{ij} w_j T_{cp}}{P_j(t)} \quad (5)$$

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

$$T_{ij} = \alpha_{ij} (Z_{1j} T_{cm} + \frac{w_j T_{cp}}{P_j(t)}) \quad j=1,2, \dots, M \quad (6)$$

Applying the optimality criterion that all processors should stop

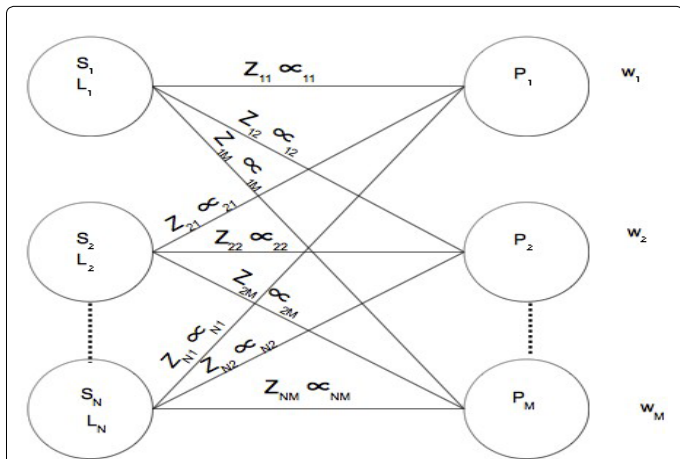


Figure 3: Grid with  $N$  data sources each has portion of load to be distributed to  $M$  heterogeneous nodes.

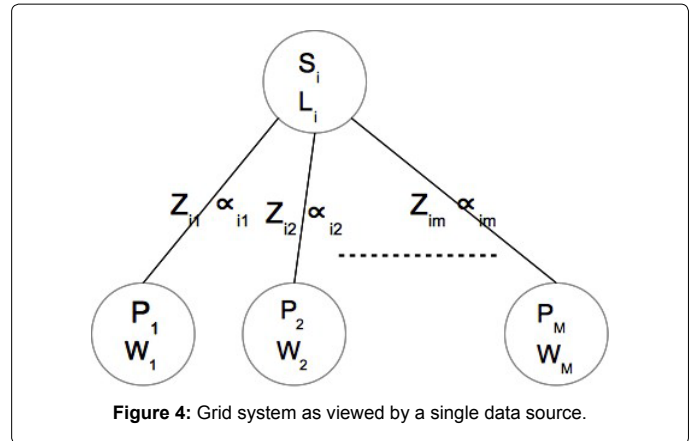


Figure 4: Grid system as viewed by a single data source.

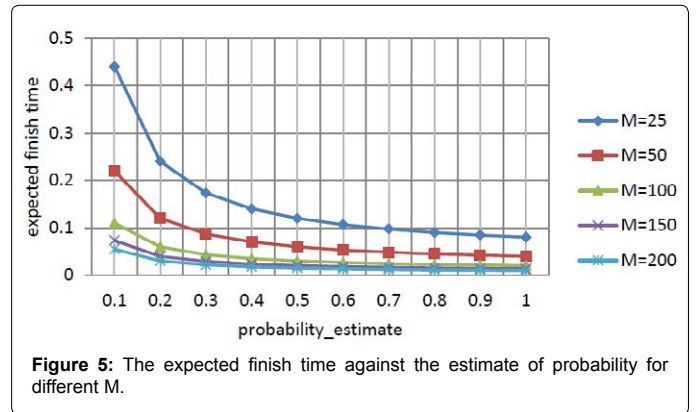


Figure 5: The expected finish time against the estimate of probability for different  $M$ .

computing at the same time [31,32]:

$$\text{Where } T_{ij} = T_{i(j+1)} \quad \text{where } j=1,2,\dots, M-1 \quad (7)$$

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

$$\alpha_{12} = \alpha_{11} \frac{P_2(t)(Z_{11} T_{cm} P_1(t) + w_1 T_{cp})}{P_1(t)(Z_{12} T_{cm} P_2(t) + w_2 T_{cp})} \quad (8)$$

$$\alpha_{1j} = \alpha_{11} \frac{P_j(t)(Z_{11} T_{cm} P_1(t) + w_1 T_{cp})}{P_1(t)(Z_{1j} T_{cm} P_j(t) + w_j T_{cp})} \quad (9)$$

.....

$$\alpha_{1(M-1)} = \alpha_{1M} \frac{P_{M-1}(t)(Z_{1M} T_{cm} P_M(t) + w_M T_{cp})}{P_M(t)(Z_{1(M-1)} T_{cm} P_{(M-1)}(t) + w_{(M-1)} T_{cp})} \quad (10)$$

Now we can write all the above equations as a function of  $\alpha_{11}$  and the parameters of the Grid as follows:

$$\alpha_{12} = \alpha_{11} \frac{P_2(t)(Z_{11} T_{cm} P_1(t) + w_1 T_{cp})}{P_1(t)(Z_{12} T_{cm} P_2(t) + w_2 T_{cp})} \quad (11)$$

$$\alpha_{13} = \alpha_{11} \frac{P_3(t)(Z_{11} T_{cm} P_1(t) + w_1 T_{cp})}{P_1(t)(Z_{13} T_{cm} P_3(t) + w_3 T_{cp})} \quad (12)$$

In general,

$$\alpha_{1j} = \alpha_{11} \frac{P_j(t)(Z_{11} T_{cm} P_1(t) + w_1 T_{cp})}{P_1(t)(Z_{1j} T_{cm} P_j(t) + w_j T_{cp})} \quad \text{Where } j=1,2,\dots, M \quad (13)$$

Obviously, the summation of all fractions of the load must equal 1.

$$\alpha_{i1} + \alpha_{i2} + \alpha_{i3} + \dots + \alpha_{iM} = 1 \tag{14}$$

Using equations (13) and (14), we can find the exact value of  $\alpha_{i1}$  as a function of grid parameter

$$\alpha_{i1} = \frac{1}{1 + \frac{Z_{11}T_{cm}P_1(t) + w_1T_{cp}}{P_1(t)} \left[ \sum_{j=2}^M \frac{p_j(T)}{(Z_{1j}T_{cm}P_j(t) + w_jT_{cp})} \right]} \tag{15}$$

Since  $\alpha_{i1}$  is found,  $\alpha_{i2}, \alpha_{i3}, \dots, \alpha_{iM}$  can be calculated using equation (13) and the minimum expected finish time can also be determined using equations (6) and (15):

$$T_{finish} = T_{i1} \frac{1}{\sum_{j=1}^M \frac{p_j(T)}{(Z_{1j}T_{cm}P_j(t) + w_jT_{cp})}} \tag{16}$$

### 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  $f_k$  ( $k=1 \dots N$ ) to the  $M$  available nodes in the network. The availability of each node  $i$  ( $i=1 \dots M$ ) varies with time and is based on certain probability  $P_j(t)$ . If  $P_j(t)=1$ , the total computing power of node  $j$  is available and if  $P_j(t)=0.5$ , half of the total computing power of node  $j$  can be lend to the newly arriving job. In general  $P_j(t)$  of the computing power of node  $j$  can be devoted to a newly arriving jobs. The probability function can be derived from a certain realistic distribution. An estimate of  $P_j(t)$  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

$$p_j(t) = \frac{1}{E_j(l)} = \frac{1 - \rho}{\rho}$$

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

$$p_j(t) = \sum_{k=0}^{k=n} k \times p_{kj}(t)$$

Where  $p_{kj}(t)$  is the probability that at time  $t$  there are  $k$  jobs in the system which can handle at most  $n$  jobs  $p_{kj}(t)$  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  $M$  available nodes in the network will have the same probability of availability to serve the grid loads ( $0 < \text{probability\_estimate} \leq 1$ ). Figure 5 relates the expected finish time to the estimate of probability, for  $M=25, 100, 150$  and  $200$ , assuming that  $z_{ij}=1$  and  $w_j=1$  (for  $i=1 \dots N$  and  $j=1 \dots M$ ), and  $T_{cp}=T_{cm}=1$ .

Figure 5 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 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  $i=1 \dots N$  and  $j=1 \dots M$ ), assuming that  $w_j=1$  (for  $j=1 \dots M$ ), and  $T_{cp}=T_{cm}=1$ .

Figure 6 shows that as each link speed increases, the expected finish will decrease.

Figure 7 relates the estimate of probability for node  $k$ , ( $k$ : may be any node  $1 \dots M$ ), to its load fraction  $\alpha_{ik}$  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  $z_{ij}=1$  and  $w_j=1$  (for  $i=1 \dots N$  and  $j=1 \dots M$ ), and  $T_{cp}=T_{cm}=1$ .

From Figure 7, nodes that are available for serving the grid loads

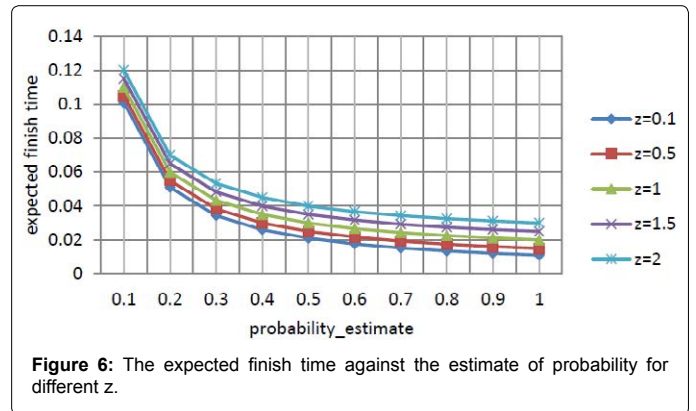


Figure 6: The expected finish time against the estimate of probability for different z.

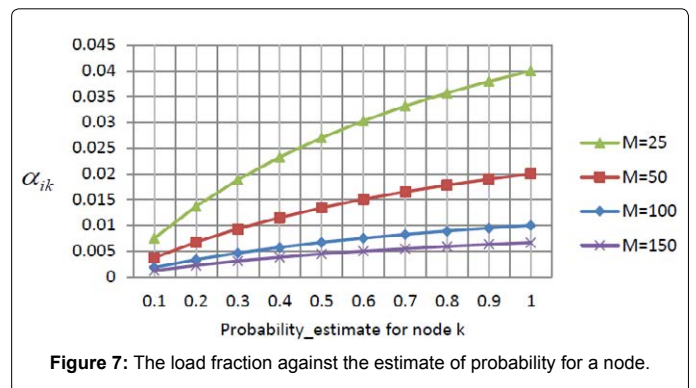


Figure 7: The load fraction against the estimate of probability for a node.



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

1. Armbrust M, Fox A, Griffith R, Joseph AD, Katz RH, et al. (2009) Above the Clouds: A Berkeley View of Cloud Computing. Technical Report No. U CB/EECS-2009-28.
2. Nandgaonkar SV, Raut AB (2014) A Comprehensive Study on Cloud Computing. *International Journal of Computer Science and Mobile Computing* 3: 733-738.
3. Kathrine GJW, Ilaghi MU (2012) Job Scheduling Algorithms in Grid Computing - Survey. *International Journal of Engineering Research & Technology* 1: 7.
4. Qureshi MB, Dehnavi MM, Allah NM, Qureshi MS, Hussain H, et al. (2014) Survey on Grid Resource Allocation Mechanisms. *Journal of Grid Computing* 12: 399-441.
5. Toma I, Iqbal K, Roman D, Strang T, Fensel D, et al. (2007) Discovery in grid and web services environments: A survey and evaluation. *Multiagent and Grid Systems* 3: 341-352.
6. Manvi SS, Birje MN (2009) Wireless Grid Computing: A Survey. *IETE Journal of Education* 50: 119-131.
7. Byuna EJ, Choia SJ, Baikb MS (2007) MJSA Markov job scheduler based on availability in desktop grid. *Future Generation Computer Systems* 23: 616-622.
8. Tian G, Xiao C, Xu X, Gao1 CQ, Nuslati, et al. (2010) Grid Workflow Scheduling Based on Time Prediction of Queuing Theory. *Proceedings of the 2010 IEEE, International Conference on Information and Automation* pp: 36-39.
9. Moges MA, Robertazzi TG (2009) Grid Scheduling Divisible Loads from Two Sources. *Computers and Mathematics with Applications* 58: 1081-1092.
10. Abdullah M, Othman M, Ibrahim H, Subramaniam S (2009) Closed form Solution for Scheduling Arbitrarily Divisible Load Model in Data Grid Applications: Multiple Sources. *American Journal of Applied Sciences* 6: 626-630.
11. Moges M, Robertazzi TG (2003) Optimal Divisible Load Scheduling and Markov Chain Models. *Proceedings of the 2003 Conference on Information Sciences and Systems*, The Johns Hopkins University, Baltimore, MD, USA.
12. Jia J, Veeravalli B (2010) Scheduling Multisource Divisible Loads on Arbitrary Networks. *IEEE Transactions On Parallel And Distributed Systems* 21: 520-531.
13. Li X, Veeravalli B (2010) PPDD: scheduling multi-site divisible loads in single-level tree networks. *Cluster Computing* 13: 31-46
14. Robertazzi TG, Yu D (2006) Multi-source grid scheduling for divisible loads. In: *Proceedings of the 40th Annual Conference on Information Sciences and Systems (CISS'06)* pp: 188-191.
15. Moise D, Moise I, Pop F, Cristea V (2008) Resource Co Allocation for Scheduling Tasks with Dependencies, in Grid. *The Second International Workshop on High Performance in Grid Middleware HiPerGRID*.
16. Fujimoto N, Hagihara K (2003) Near-optimal dynamic task scheduling of precedence constrained coarse-grained tasks onto a computational grid. *Second International Symposium on Parallel and Distributed Computing 2003 Proceedings* pp: 80-87.
17. Abdullah M, Othman M, Ibrahim H, Subramaniam S (2007) An integrated approach for scheduling divisible load on large scale data grids. In: *Computational Science and Its Applications—ICCSA, 4705: 748-757*.
18. Mamat A, Lu Y, Deogun J, Goddard S (2012) Scheduling real-time divisible loads with advance reservations. *Real-Time Systems* 48: 264-293.
19. Lin X, Mamat A, Lu Y, Deogun J, Goddard S (2010) Real-time scheduling of divisible loads in cluster computing environments. *Journal of Parallel and Distributed Computing* 70: 296-308.
20. Chuprat S (2010) Divisible Load scheduling of real-time task on heterogeneous clusters. In *Proceedings of the IEEE International Symposium on Information Technology (ITSim '10)* 2: 721-726.
21. Ghanbari S, Othman M, Leong WJ, Abu Bakar MR (2014) Multi-criteria based algorithm for scheduling divisible load. In: *Proceedings of the 1st International Conference on Advanced Data and Information Engineering (DaEng '13)*, Lecture Notes in Electrical Engineering, pp: 547-554.
22. Othman SM, Ibrahim H, Subramaniam S (2012) New method for scheduling heterogeneous multi-installment systems. *Future Generation Computer Systems* 28: 1205-1216.
23. Yu C, Marinescu DC (2010) Algorithms for divisible load scheduling of data-intensive applications. *Journal of Grid Computing* 8: 133-155.
24. Bataineh S (2008) Divisible Load Distribution in a Network of Processors. *Journal of Interconnection Networks* 9: 31-51.
25. Shokripour A, Othman M (2009) Survey on Divisible Load Theory and its Applications. *International Conference on Information Management and Engineering* pp: 300-304.
26. Shokripour A, Othman M (2009) Survey on Divisible Load Theory and Its Applications. *International Conference on Information Management and Engineering*.
27. Ghanbari S, Othman M (2014) Comprehensive Review on Divisible Load Theory: Concepts, Strategies, and Approach. *Mathematical Problems in Engineering* Volume 2014: 460354
28. Bataineh S, Khalil I (2013) Scheduling divisible load on Wireless Grid with communication Delay. *3rd International Conference on Wireless Communications and Mobile Computing (MIC- WCMC 2013)*, Valencia, Spain
29. RichardsonP, Sieh L, Elkateeb AM (2001) Fault-tolerant adaptive scheduling for embedded real-time systems. *IEEE Micro* 21: 41-51.
30. Kleinrock L (1975) *Queueing Systems: Theory*. Wiley, New York.
31. Sohn J, Robertazzi TG (1996) Optimal Load Sharing for a Divisible Job on a Bus Network. *IEEE Transactions on Aerospace and Electronic Systems* 32: 34-40.
32. Mingsheng S (2008) Optimal algorithm for scheduling large divisible workload on heterogeneous system. *Applied Mathematical Modelling* 32: 1682-1695.
33. Yu D, Robertazzi TG (2003) Divisible Load Scheduling for Grid Computing. In: *PDCS'2003, 15th Int'l Conf. Parallel and Distributed Computing and Systems*.
34. Robertazzi TG (2003) Ten reasons to use divisible load theory. *Computer* 36: 63-68.
35. Bharadwaj V, Ghose D, Robertazzi TG (2003) Divisible load theory: a new paradigm for load scheduling in distributed systems. *Cluster Comput.* 6: 7-17.
36. Wang X, Wang Y, Meng K (2014) Optimization Algorithm for Divisible Load Scheduling on Heterogeneous Star Networks. *JOURNAL OF SOFTWARE* 9: 1757:1766
37. Bharadwaj V, Ghose D, Mani V (1992) A study of optimality conditions for load distribution in tree networks with communication delays. *Dept. of Aerospace Engineering, Indian Institute of Science, Bangalore, India, Technical Report 423: 02-92*.
38. Bataineh S, Hsiung T, Robertazzi TG (1994) Closed Form Solutions for Bus and Tree Networks of Processors Load Sharing A Divisible Job. *IEEE Tran. on Computers* 43: 1184-1197.