Proceedings of 8th International Conference on Engineering and Technology Research
24 - 25 April 2014, Novotel World Trade Centre, Dubai, UAE, ISBN: 978-1-922069-49-8

Divisible Load Scheduling For Two Source Single Level Tree Networks

Manar Arafat and Sameer Bataineh*

This paper presents a divisible load scheduling for two-source heterogeneous single level tree network. Solutions for the optimal finish time and the optimal load allocation over processors in the network are obtained via linear programming. We have developed also closed form solutions for the optimal finish time and the optimal load fractions. The solutions have been derived with sequential communication strategy. Performance evaluation of a two-source homogeneous single level tree network with sequential communication is also presented.

Index Terms: Divisible load, Linear programming, Multiple sources, Optimal scheduling, Parallel computing, Tree network.

1. Introduction

The ever increasing data-intensive tasks have created a need for dividing the data amongst multiple processors to improve the speed of computation through parallel computing. Parallel computing is the most effective way of exploiting the processing capability of the entire network in order to achieve faster solution time. As an evident, most of recent applications such as signal and image processing, experimental data processing, Kalman filtering, cryptography, and genetic algorithms, all involve parallel and distributed computing in order to improve system performance [1]. A powerful tool for modeling data-exhaustive computational problems in distributed computing systems is the divisible load theory (DLT) [1-6].

All of the prior work on divisible load analysis involves an optimal load distribution strategy in order to minimize the finish (processing) time [1-4, 8-14]. Closed form solutions for the optimal load assignment to each processor for different network topologies were obtained [1, 2, 4, 8, 9, 11-14]. The majority of literature on divisible load modeling focused on networks with a single load originating (source) processor [1-4, 7-14]. However, load is generated from multiple load source processors in large-scale data intensive applications with geographically distributed resources. Unfortunately, there is a limited amount of literature focused on networks with multiple load originating processors [15-23]. Grid scheduling problem involving multiple sources under resource (buffer) constraints were examined in [16]. Moges, Yu and Robertazzi considered the scheduling of divisible loads in a single level tree networks from two load sources (roots) [18-20].

Manar Arafat, Department of Computer science, An-najah National University, P.O Box 7 Nablus, West Bank Palestine, Email: manar_arafat@najah.edu

Sameer Bataineh*, Department of Computer Engineering, College of Computer and Information Tech., Jordan University of Science and Tech., P.O Box 3030, Irbid 22110, Jordan, 00-962-79-5377999, Email:samir@just.edu.jo

In [20] solutions for the optimal load allocation from two load sources (roots) over heterogeneous processors (child processors) such that the finish time is optimum were achieved. The paper presents both: linear programming solutions and closed form solutions. The optimal solutions in [20] were achieved under two constraints: First, the load distribution strategy from a root is concurrent, that is, the root can communicate simultaneously with all the child processors. Second, a root node can communicate and process its load share at a time.

In this paper, considering a single level tree network with two sources (roots), we will extend the two source concurrent load distribution model in [20] to a sequential load distribution model. In a sequential distribution, the root can communicate with only one child at a time. It will be assumed that a root node can communicate and process at the same time. We will achieve solutions for the optimal processing time and the optimal load allocation over heterogonous processors in the network via linear programming. Closed form solutions will be also derived.

This paper is organized as follows. Section 2 presents the system model and the system parameters. The problem of finding an optimal finish time in a single level tree with two load originating processors and sequential communication is presented in section 3. The respective performance analysis results in terms of finish time and load assignment is presented in section 4. Finally section 5 concludes the paper.

2. System Model

Consider a network system of N heterogeneous nodes (processors) ($P_1, P_2 \ldots P_N$). The network topology is a tree network consisting of two root nodes (P_1, P_2) and N-2 child nodes ($P_3 \ldots P_N$) with $2(N-2)$ links as depicted in Fig.1. It will be assumed that the network contains at least four processors and the total processing load is divisible. Each root originates a part of the total load and divides it into fractions. Then each root will keep its own fraction and transfer the fractions assigned to each of the rest N-2 child nodes. The load distribution strategy from a root is sequential, that is, the root can communicate with only one child at a time, and a root node can communicate and process at the same time. Each child node will begin to process its share of load only when the load fraction has been completely received from either root node. It will be assumed also that for the same child, P_1 terminates communication before P_2.
System parameters:

L_i : The total load originated at root node i, (i = 1, 2).

α_i : The total fraction of load that is assigned to node i, (i=1…N).

$\alpha_{1,i}$: The fraction of load that is assigned to node i by the first root, (i=3…N).

$\alpha_{2,i}$: The fraction of load that is assigned to node i by the second root, (i=3…N).

$$\alpha_i = \alpha_{1,i} + \alpha_{2,i}, (i = 3 \ldots N)$$

w_i : A constant that is inversely proportional to the processing speed of the i^{th} node.

d_i : A constant that is inversely proportional to the speed of i^{th} link. (i=3…N).

$d_{1,i}$: A constant that is inversely proportional to the speed of link between the first root node and the i^{th} child node. ($i=3…N$).

$d_{2,i}$: A constant that is inversely proportional to the speed of link between the second root node and the i^{th} child node. ($i=3…N$).

T_{cm} : The time it takes the root node to transmit the entire load over a link when $d_i=1$.

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

T_i : The total time it takes the i^{th} node to complete its computation, communication, and also includes the idle time.

T_f : This is the time when the last processor finishes processing.

$$T_f = \max(T_1, T_2, ..., T_N)$$

In this paper, we assume a normalized processing load. Therefore, the fractions of load must sum to one:

$$\sum_{i=1}^{N} \alpha_i = 1$$

3. Optimal scheduling strategy

3.1 Linear Programming solutions

Consider the Single level tree system with two load originating nodes and sequential communication described in section 2. The system timing diagram is shown in Fig.2. At time $t=0$, all child nodes are idle. Child nodes begin to process their share of load once the load fraction has been completely received from either root node. It will be assumed that for the same child, P_1 terminates communication before P_2. The sequential load distribution strategy proceeds as follows: Each of the two Roots (P_1, P_2) originates a part of the total load L_1 and L_2 respectively. Then each Root i, ($i=1, 2$) keeps its own fraction α_i to process and distributes ($\alpha_{i,3} ... \alpha_{i,N}$) shares to child nodes ($P_3…P_N$) respectively. To obtain the optimal finish time, all the nodes involved in the processing of the parallel load must stop at the same time. The equations that capture the various parameters of the system can be written as follows:

$$T_1 = \alpha_1 \omega_1 T_{cp} \quad (1)$$

$$T_2 = \alpha_2 \omega_2 T_{cp} \quad (2)$$

$$T_3 = \alpha_{1,3} d_{1,3} T_{cm} + \alpha_3 \omega_3 T_{cp} \quad (3)$$

$$T_4 = \alpha_{1,3} d_{1,3} T_{cm} + \alpha_{1,4} d_{1,4} T_{cm} + \alpha_4 \omega_4 T_{cp} \quad (4)$$

:

$$T_N = \sum_{i=3}^{N} \left(\alpha_{1,i} d_{1,i} T_{cm} \right) + \alpha_N \omega_N T_{cp} \quad (5)$$

As mentioned earlier a normalized processing load is assumed:

$$\sum_{i=1}^{N} \alpha_i = 1 \qquad (6)$$

Where:

$$\alpha_i = \alpha_{1,i} + \alpha_{2,i}, (i = 3 \ldots N) \qquad (7)$$

From the timing diagram shown in Fig.2, all nodes stop processing at the same time, thus we have:

$$T_1 = T_2 = \ldots = T_N \qquad (8)$$

We can write the following set of N-1 equations based on the equations above:

$$\alpha_1 \omega_1 T_{cp} = \alpha_2 \omega_2 T_{cp} \qquad (9)$$

$$\alpha_2 \omega_2 T_{cp} = \alpha_{1,3} d_{1,3} T_{cm} + \alpha_3 \omega_3 T_{cp} \qquad (10)$$

$$\alpha_3 \omega_3 T_{cp} = \alpha_{1,4} d_{1,4} T_{cm} + \alpha_4 \omega_4 T_{cp} \qquad (11)$$

$$\alpha_4 \omega_4 T_{cp} = \alpha_{1,5} d_{1,5} T_{cm} + \alpha_5 \omega_5 T_{cp} \qquad (12)$$

:

$$\alpha_{N-1} \omega_{N-1} T_{cp} = \alpha_{1,N} d_{1,N} T_{cm} + \alpha_N \omega_N T_{cp} \qquad (13)$$

From the timing diagram, another set of *N-2* equations can be written for each of the child nodes:

$$\alpha_{2,3} d_{2,3} T_{cm} \le \alpha_{1,3} (d_{1,3} T_{cm} + \omega_3 T_{cp}) \qquad (14)$$

$$\alpha_{2,3} d_{2,3} T_{cm} + \alpha_{2,4} d_{2,4} T_{cm} \le \alpha_{1,3} d_{1,3} T_{cm} + \alpha_{1,4} (d_{1,4} T_{cm} + \omega_4 T_{cp}) \qquad (15)$$

$$\alpha_{2,3} d_{2,3} T_{cm} + \alpha_{2,4} d_{2,4} T_{cm} + \alpha_{2,5} d_{2,5} T_{cm} \le \alpha_{1,3} d_{1,3} T_{cm} + \alpha_{1,4} d_{1,4} T_{cm} + \alpha_{1,5} (d_{1,5} T_{cm} + \omega_5 T_{cp}) \qquad (16)$$

:

$$\sum_{i=3}^{N} \alpha_{2,i} d_{2,i} T_{cm} \le \sum_{i=3}^{N} \alpha_{1,i} d_{1,i} T_{cm} + \alpha_{1,N} \omega_N T_{cp} \qquad (17)$$

From above, the system has *2N-1* equations with *2N-2* unknowns. Such problems can be solved by linear programming [18]. The objective function of this linear programming problem is to minimize the total processing time needed to process the loads originated from two sources.

Consequently, the objective function will be, Minimize:

$$T_f = \alpha_1 \omega_1 T_{cp}$$

Subject to constraints to stop processors at the same time:

$$\alpha_1 \omega_1 T_{cp} - \alpha_2 \omega_2 T_{cp} = 0$$

$$\alpha_2 \omega_2 T_{cp} - \alpha_{1,3} d_{1,3} T_{cm} - \alpha_3 \omega_3 T_{cp} = 0$$

$$\alpha_3 \omega_3 T_{cp} - \alpha_{1,4} d_{1,4} T_{cm} - \alpha_4 \omega_4 T_{cp} = 0$$

:

$$\alpha_{N-1} \omega_{N-1} T_{cp} - \alpha_{1,N} d_{1,N} T_{cm} - \alpha_N \omega_N T_{cp} = 0$$

And subject to inequality constraints

$$\alpha_{2,3} d_{2,3} T_{cm} - \alpha_{1,3}(d_{1,3} T_{cm} + \omega_3 T_{cp}) \le 0$$

$$\alpha_{2,3} d_{2,3} T_{cm} + \alpha_{2,4} d_{2,4} T_{cm} - \alpha_{1,3} d_{1,3} T_{cm} - \alpha_{1,4}(d_{1,4} T_{cm} + \omega_4 T_{cp}) \le 0$$

:

$$\sum_{i=3}^{N} \alpha_{2,i} d_{2,i} T_{cm} - \sum_{i=3}^{N} \alpha_{1,i} d_{1,i} T_{cm} - \alpha_{1,N} \omega_N T_{cp} \le 0$$

All the load fractions should be positive

$$\alpha_i \ge 0$$

And the load is normalized

$$\sum_{i=1}^{N} \alpha_i - 1 = 0$$

And

$$\alpha_1 + \alpha_2 + \sum_{i=3}^{N} \alpha_{1,i} + \sum_{i=3}^{N} \alpha_{2,i} - 1 = 0 \quad (18)$$

Linear programming solutions are preferred to closed form solutions because fewer assumptions are involved.

3.2 Closed form solutions

Regarding the load distribution strategy, the following assumptions can be made in order to obtain closed form solutions for the optimal finish time and optimal load shares:

- The root can communicate with only one child at a time, and a root node can communicate and process at the same time.
- For the same child node, P_1 terminates communication before P_2.
- Each child node will begin to process its share of load once the load fraction has been completely received from either root node.
- For each child node *x=3 to N*, the communication time needed to distribute the respective fractions of load by P_2 to each child node *i=3 to x*, is equal to, the communication time needed to distribute the respective fractions of load by P_1 to each child node *i=3 to x* plus the time needed to process the fraction of load assigned to node *x* by P_1. Thus,

$$\sum_{i=3}^{x} \alpha_{2,i} d_{2,i} T_{cm} = \sum_{i=3}^{x} \alpha_{1,i} d_{1,i} T_{cm} + \alpha_{1,x} w_x T_{cp}$$

The system timing diagram under these assumptions is shown in Fig.3. From the figure, we can derive closed form solutions for the optimal finish time and optimal load shares in five steps:

First step: We can solve for the load fractions that will be assigned to each child *i=3 to N* by P_2 in terms of α_2 and other system parameters. From Fig.3,

$$\alpha_{2,3} \omega_3 T_{cp} = \alpha_{2,4}\left(d_{2,4} T_{cm} + \omega_4 T_{cp}\right) \quad (19)$$

$$\alpha_{2,4} \omega_4 T_{cp} = \alpha_{2,5}\left(d_{2,5} T_{cm} + \omega_5 T_{cp}\right) \quad (20)$$

:

$$\alpha_{2,N-1}\omega_{N-1}T_{cp} = \alpha_{2,N}\left(d_{2,N}T_{cm} + \omega_N T_{cp}\right) \quad (21)$$

$$\alpha_{2,N}\omega_N T_{cp} = \alpha_2\omega_2 T_{cp} - \sum_{i=3}^{N}\alpha_{2,i}d_{2,i}T_{cm} \quad (22)$$

In general, for *i=3 to N-1*

$$\alpha_{2,i} = \alpha_{2,i+1}\left(\frac{d_{2,i+1}T_{cm} + \omega_{i+1}T_{cp}}{\omega_i T_{cp}}\right) \quad (23)$$

Define

$$\beta_i = \left(\frac{d_{2,i+1}T_{cm} + \omega_{i+1}T_{cp}}{\omega_i T_{cp}}\right) \quad (24)$$

Rewrite (23) using (24)

$$\alpha_{2,i} = \alpha_{2,i+1}\beta_i \quad (25)$$

One can solve the set of equations (25) and (22) recursively, we get

$$\alpha_{2,N} = \alpha_2\left[\frac{\omega_2 T_{cp}}{\left[\left(\omega_N T_{cp} + d_{2,N}T_{cm}\right) + T_{cm}\sum_{i=3}^{N-1}\left(d_{2,i}\times\prod_{j=i}^{N-1}\beta_j\right)\right]}\right] \quad (26)$$

Define

$$\beta_N = \frac{\omega_2 T_{cp}}{\left[\left(\omega_N T_{cp} + d_{2,N}T_{cm}\right) + T_{cm}\sum_{i=3}^{N-1}\left(d_{2,i}\times\prod_{j=i}^{N-1}\beta_j\right)\right]} \quad (27)$$

Rewrite (26) using (27)

$$\alpha_{2,N} = \alpha_2\beta_N \quad (28)$$

One can solve the set of equations (25) and (28) recursively to get the load fractions that will be assigned to each child node *i=3 to N* by P_2 in terms of α_2 and other system parameters.

In general for i=3 to N

$$\alpha_{2,i} = \alpha_2 \prod_{j=i}^{N} \beta_j$$

where

$$\beta_i = \begin{cases} \left(\dfrac{d_{2,i+1} T_{cm} + \omega_{i+1} T_{cp}}{\omega_i T_{cp}} \right) & i = 3 to N - 1 \\ \left[\dfrac{\omega_2 T_{cp}}{\left[\left(\omega_N T_{cp} + d_{2N} T_{cm} \right) + T_{cm} \sum_{i=3}^{N-1} \left(d_{2,i} \times \prod_{j=i}^{N-1} \beta_j \right) \right]} \right] & i = N \end{cases} \quad (29)$$

To simplify, for i=3 to N

$$\alpha_{2,i} = \alpha_2 H_i$$

where (30)

$$H_i = \prod_{j=i}^{N} \beta_j$$

<u>Second step</u>: We can solve for the total load fraction assigned by P_1 and P_2 to the first child node in terms of α_2 and other system parameters. From Fig.3,

$$\alpha_{1,3}\left(\omega_3 T_{cp} + d_{1,3} T_{cm}\right) = \alpha_{2,3} d_{2,3} T_{cm} \quad (31)$$

From (7),

$$\alpha_{1,3} = \alpha_3 - \alpha_{2,3} \quad (32)$$

Substitute (32) in (31), we get

$$\alpha_3 = \alpha_{2,3} \left(\frac{d_{2,3} T_{cm} + \omega_3 T_{cp} + d_{1,3} T_{cm}}{\left(\omega_3 T_{cp} + d_{1,3} T_{cm}\right)} \right) \quad (33)$$

From (30),

$$\alpha_{2,3} = \alpha_2 H_3 \quad (34)$$

Substitute (34) in (33)

$$\alpha_3 = \alpha_2 H_3 \left(\frac{d_{2,3} T_{cm} + \omega_3 T_{cp} + d_{1,3} T_{cm}}{\left(\omega_3 T_{cp} + d_{1,3} T_{cm}\right)} \right) \quad (35)$$

From the equations above, the total load fraction assigned by P_1 and P_2 to the first child node in terms of α_2 and other system parameters is:

$$\alpha_3 = \alpha_2 C$$

where

$$C = H_3 \left(\frac{d_{2,3} T_{cm} + \omega_3 T_{cp} + d_{1,3} T_{cm}}{\left(\omega_3 T_{cp} + d_{1,3} T_{cm}\right)} \right) \quad (36)$$

Third step: We can solve for the load fraction α_1 in terms of α_2 and other system parameters. From Fig.3,

$$\alpha_1 = \frac{\alpha_2 \omega_2}{\omega_1} \qquad (37)$$

Fourth step: we can solve for the total load fraction assigned by P_1 and P_2 to child node *i* (*i=4 to N*) in terms of α_2 and α_{i-1} and other system parameters. From Fig.3,

$$\alpha_3 \omega_3 T_{cp} = \alpha_{1,4} d_{1,4} T_{cm} + \alpha_4 \omega_4 T_{cp} \qquad (38)$$

$$\alpha_4 \omega_4 T_{cp} = \alpha_{1,5} d_{1,5} T_{cm} + \alpha_5 \omega_5 T_{cp} \qquad (39)$$

:

$$\alpha_{N-1} \omega_{N-1} T_{cp} = \alpha_{1,N} d_{1,N} T_{cm} + \alpha_N \omega_N T_{cp} \qquad (40)$$

Using (7) rewrite (38)… (40)

$$\alpha_3 \omega_3 T_{cp} + \alpha_{2,4} d_{1,4} T_{cm} = \alpha_4 (d_{1,4} T_{cm} + \omega_4 T_{cp}) \qquad (41)$$

$$\alpha_4 \omega_4 T_{cp} + \alpha_{2,5} d_{1,5} T_{cm} = \alpha_5 (d_{1,5} T_{cm} + \omega_5 T_{cp}) \qquad (42)$$

:

$$\alpha_{N-2} \omega_{N-2} T_{cp} + \alpha_{2,N-1} d_{1,N-1} T_{cm} = \alpha_{N-1} (d_{1,N-1} T_{cm} + \omega_{N-1} T_{cp}) \qquad (43)$$

$$\alpha_{N-1} \omega_{N-1} T_{cp} + \alpha_{2,N} d_{1,N} T_{cm} = \alpha_N (d_{1,N} T_{cm} + \omega_N T_{cp}) \qquad (44)$$

In general, for *i=4 to N*

$$\alpha_{i-1} \omega_{i-1} T_{cp} + \alpha_{2,i} d_{1,i} T_{cm} = \alpha_i \left(d_{1,i} T_{cm} + \omega_i T_{cp}\right) \qquad (45)$$

Substitute (30) in (45), *i=4 to N*

$$\alpha_i = \alpha_{i-1} \left(\frac{\omega_{i-1} T_{cp}}{\left(d_{1,i} T_{cm} + \omega_i T_{cp}\right)} \right) + \alpha_2 \left(H_i\right) \left(\frac{d_{1,i} T_{cm}}{\left(d_{1,i} T_{cm} + \omega_i T_{cp}\right)} \right) \qquad (46)$$

From the equations above, the total load fraction assigned by P_1 and P_2 to child *i* (*i=4 to N*) in terms of α_2 and α_{i-1} is:

$$\alpha_i = \alpha_{i-1} x_i + \alpha_2 y_i \cdots\cdots\cdots\cdots i = 4 to N$$

where

$$x_i = \left(\frac{\omega_{i-1} T_{cp}}{d_{1,i} T_{cm} + \omega_i T_{cp}} \right)$$

and

$$y_i = \left(H_i\right) \left(\frac{d_{1,i} T_{cm}}{\left(d_{1,i} T_{cm} + \omega_i T_{cp}\right)} \right) \qquad (47)$$

Fifth step: We solve equations (36), (37) and (47) recursively, to find ($\alpha_4 ... \alpha_N$) in terms of α_2. We get:

$$\alpha_i = \alpha_2 (M_i) \cdots\cdots\cdots i = 1 to N$$
$$where$$
$$M_1 = \frac{\omega_2}{\omega_1}$$
$$M_2 = 1$$
$$M_3 = C$$
$$M_i = M_{i-1} x_i + y_i \ldots\ldots\ldots i = 4 to N \tag{48}$$

Finally, we can solve for α_2 by substituting equations (48) in the normalization equation (6) we get:

$$\alpha_2 = \frac{1}{\sum_{i=1}^{N} M_i} \tag{49}$$

To find the total load fraction assigned by P_1 and P_2 to each child node, one can substitute (49) in (48). To find the load fraction assigned by P_2 to each child node, one can substitute (49) in (30). Then one can use (7) to solve for the load fraction assigned by P_1 to each child node.
The minimum load processing time in the network can be given as:

$$T_f = \frac{\omega_2 T_{cp}}{\sum_{i=1}^{N} M_i} \tag{50}$$

4. Performance Analysis : Processing Finish Time Results

Consider a homogeneous tree network system of two root nodes. In this section we will use linear programming to study the effect of the number of processors, the processor computation speed, and the communication links speed on the total finish time. Plots of the finish time vs. the number of nodes in the network are shown in Fig.4, Fig.5 and Fig.6. In all the plots we will assume that the values of *Tcm* and *Tcp* to be one.
In Fig.4, we assume a fixed speed second root node communication links (*d2=1*). Each node is assumed to have fixed processing speed (*w=2*). The plot is obtained by varying the speed of the first root node communication links (*d1* varies from *0.5* to *2.5*).
In Fig.5, we assume a fixed speed first root node communication links (*d1=1*). Each node is assumed to have fixed processing speed (*w=2*). The plot is obtained by varying the speed of the second root node communication links (*d2* varies from *0.5* to *2.5*).
Fig.6 shows a plot of the finish time vs. the number of nodes for different node speed. We assume that *d1* and *d2* are set to be equal to *0.5*.

5. Conclusion

This paper presents a divisible load scheduling strategy for distributing processing load that originates from two sources in heterogeneous single level tree networks. Linear programming with objective function of minimizing the total finish time is used to obtain solutions for the optimum load assignment to each processor in the network. With some restrictive assumptions, closed form solutions for the optimal finish time and the optimal load fractions are obtained.

In this paper we present a performance study of homogeneous single level tree network with sequential communication load distribution strategy. We study the effect of increasing the number of processors in the network on the finish time. The effects of processors and links speed are also studied. The study shows that increasing the number of processors up to a certain point can improve the finish time. The results show that minimum time solutions do not require large number of processors in the network.

References

[1] Jeeho Sohn, Thomas G. Robertazzi and Serge Luryi (1998). Optimizing Computing Costs Using Divisible Load Analysis. IEEE transactions on parallel and distributed systems. 9(3), pp- 225-23.

[2] Thomas G.Robertazzi (2003). Ten Reasons to Use Divisible Load Theory. IEEE Computer. 36(5), pp- 63-68.

[3] Veeravalli Bharadwaj, Debasish Ghose and Thomas G. Robertazzi (2003). Divisible Load Theory: A New Paradigm for Load Scheduling in Distributed Systems. Cluster Computing. 6(1), pp- 7–17.

[4] Y-C. Cheng and T.G. Robertazzi (1988). Distributed Computation with Communication Delays. IEEE Trans. Aerospace and Electronic Systems. 24(6), pp- 700-712.

[5] Veeravalli, B., Ghose, D.,Mani, V., Robertazzi, T. (1996). Scheduling Divisible Loads in Parallel and Distributed Systems (eds.). IEEE Computer Society Press, Los Alamitos.

[6] Shirazi, B., Hurson, A., Kavi, K. (1995). Scheduling and Load Balancing in Parallel and Distributed Systems (eds.). IEEE Computer Society Press, Los Alamitos.

[7] Sameer Bataineh and Mohammad Al-Ibrahim (1994). Effect of fault tolerance and communication delay on response time in a multiprocessor system with a bus topology. Computer communications. 17(12), pp- 843- 851.

[8] V. Bharadwaj, D. Ghose, and V. Mani (1995). Multi-Installment Load Distribution in Tree Networks with Delays. IEEE Trans. Aerospace and Electronic Systems. 31(2), pp- 555-56.

[9] Sameer Bataineh (2008). Divisible Load Distribution in A Network Of Processors. Journal of Interconnection Networks. 09(01n02), pp- 31-51.

[10] M. Drozdowski and W. Glazek. (1999). Scheduling Divisible Loads in a Three-Dimensional Mesh of Processors. Parallel Computing. 25(4), pp- 381-404.

[11] V.Bharadwaj, T.Radhakrishnan (1996). Scheduling divisible loads in bus networks with arbitrary processor release times. Computers & Mathematics with Applications. 32(7), pp- 57-77.

[12] J. Blazewicz and M. Drozdowski. (1995). Scheduling divisible jobs on hypercubes. Parallel Computing. 21(12), pp. 1945–1956.

[13] S. Bataineh, T. Hsiung and T. G. Robertazzi, "Closed Form Solutions for Bus and Tree Networks of Processors Load Sharing A Divisible Job" in Proceedings of the 1993 International Conference on Parallel Processing, IEEE Computer Society Washington, DC, 1993, pp- 290-293.

[14] V. Bharadwaj, D. Ghose, and V. Mani. (1994). Optimal sequencing and arrangement in distributed single-level tree networks with communication delays. IEEE Trans. Parallel Distrib. Syst.. 5(9), pp. 968–976.

[15] K. Ko and T. Robertazzi, "Scheduling in an Environment of Multiple Job Submissions," in Proc. of the Conference on Information Sciences and Systems, Prinseton University, March 20-22, 2002.

[16] H. Wong, D. Yu, B. Veeravalli, and T. Robertazzi, "Data Intensive Grid Scheduling: Multiple Sources with Capacity Constraints," in IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS 2003), Marina del Rey, CA, Nov. 2003.

[17] Theron Lammie and Thomas G.Robertazzi, "A Linear Daisy Chain with Two Divisible Load Sources," presented at 2005 Conference on Information Sciences and Systems, The Johns Hopkins University, March 16-18, 2005.

[18] M. Moges and T. Robertazzi, "Grid Scheduling Divisible Loads from Multiple Sources via Linear Programming" in IASTED International Conference on Parallel and Distributed Computing and Systems(PDCS 2005), Cambridge, MA, USA, 2004, pp. 423-428.

[19] M. Moges, T. Robertazzi, and D. Yu, "Divisible Load Scheduling with Multiple Sources: Closed Form Solutions," in Proceedings of 2005 Conf. on Information Sciences and Systems, The Johns Hopkins University, Baltimore, MD, March 16-18, 2005.

[20] M. Moges, T. Robertazzi, and D. Yu. (2009). Grid scheduling divisible loads from two sources. Journal of Computers & Mathematics with Applications. 58(6), pp-1081-1092.

[21] Thomas G. Robertazzi and Dantong Yu, "Multi-Source Grid Scheduling for Divisible Loads," in the 40th annual conference on information sciences and systems, Princeton university, MARCH 22–24, 2006.

[22] Jingxi Jia and Bharadwaj Veeravalli. (2010). Scheduling Multisource Divisible Loads on Arbitrary Networks. IEEE Transactions On Parallel And Distributed Systems. 21(4), pp. 520-531.

[23] Xiaolin Li · Bharadwaj Veeravalli. (2010). PPDD: scheduling multi-site divisible loads in single-level tree Networks. Cluster Computing. 13(1), pp. 31–46.

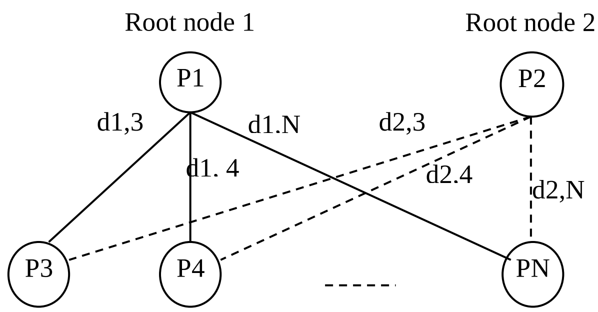

Fig.1. Single level tree network with two root nodes.

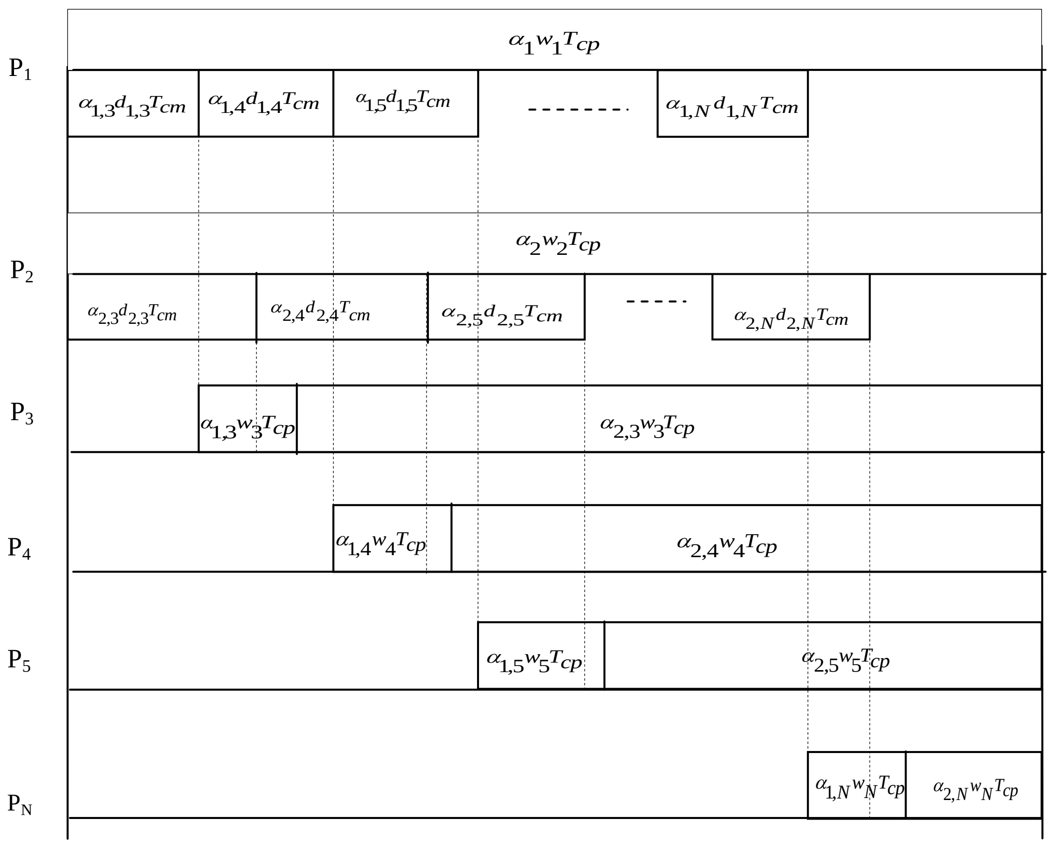

Fig.2. Timing diagram for a single level tree network with two root nodes and sequential communication.

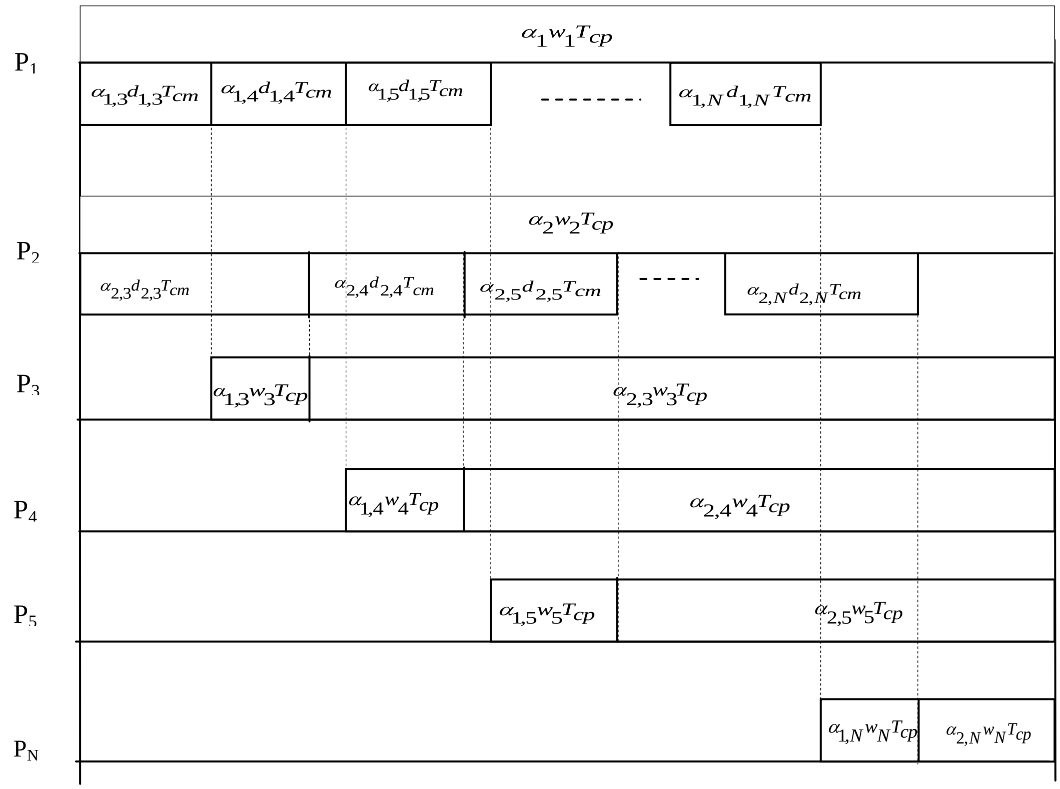

Fig.3. Timing diagram for a single level tree network with two root nodes and sequential communication: Scheduling for closed form solution.

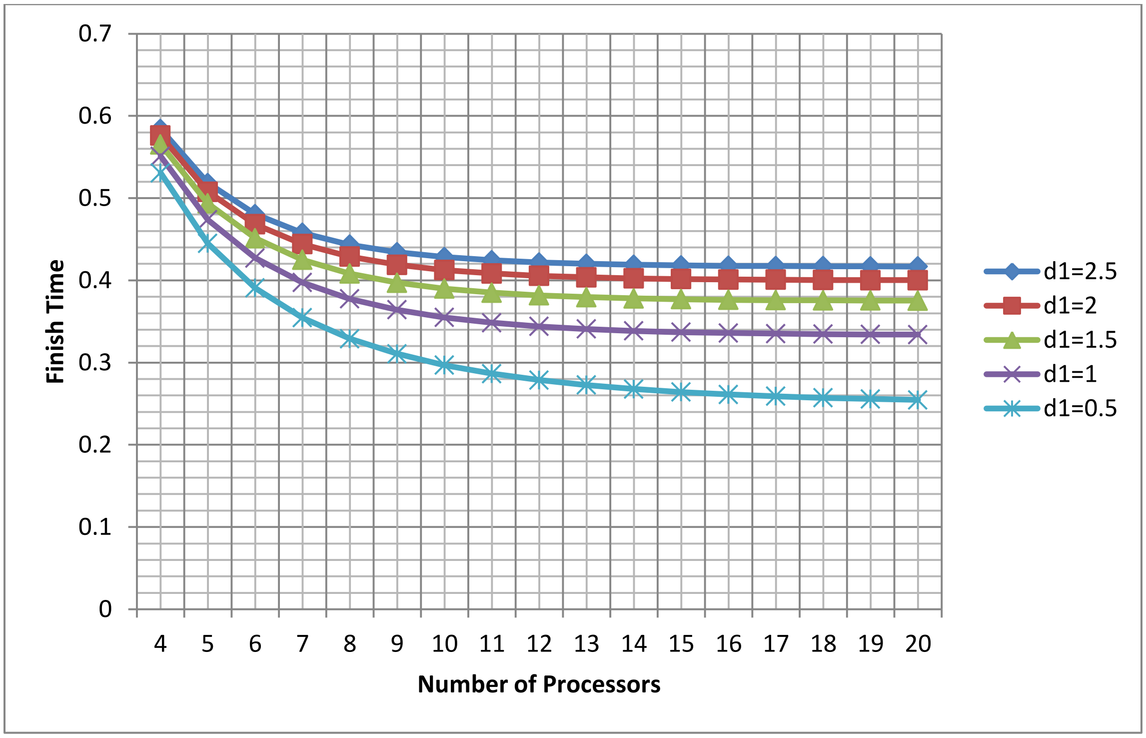

Fig.4. Finish time versus number of nodes, for two root sources single level homogeneous tree network and variable d1

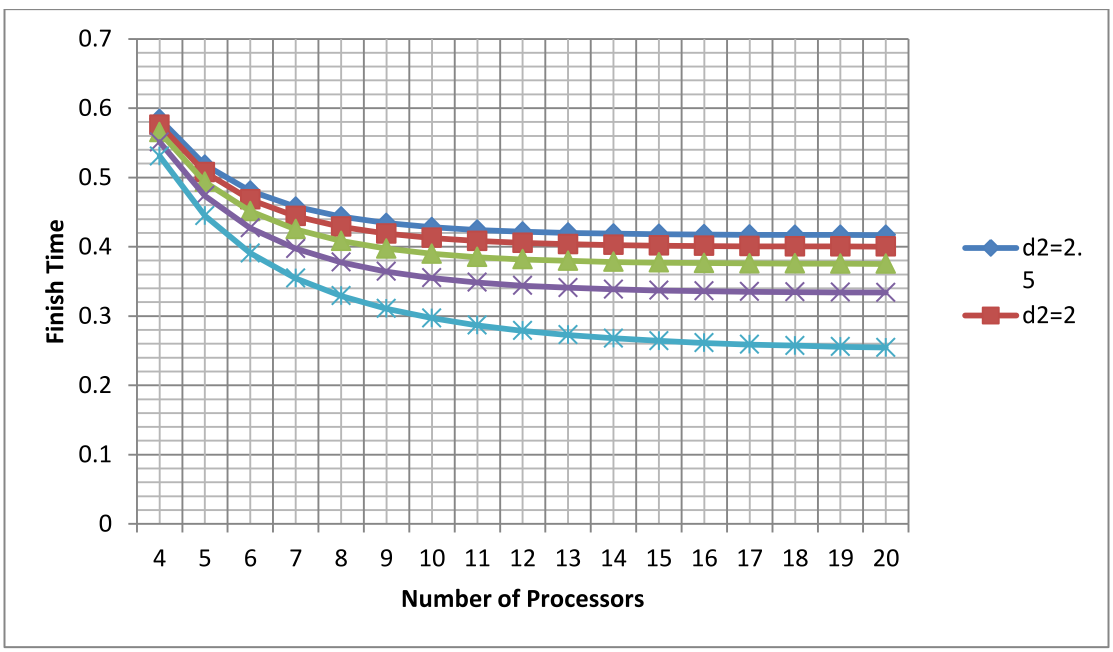

Fig.5. Finish time versus number of nodes, for two root sources single level homogeneous tree network and variable d2.

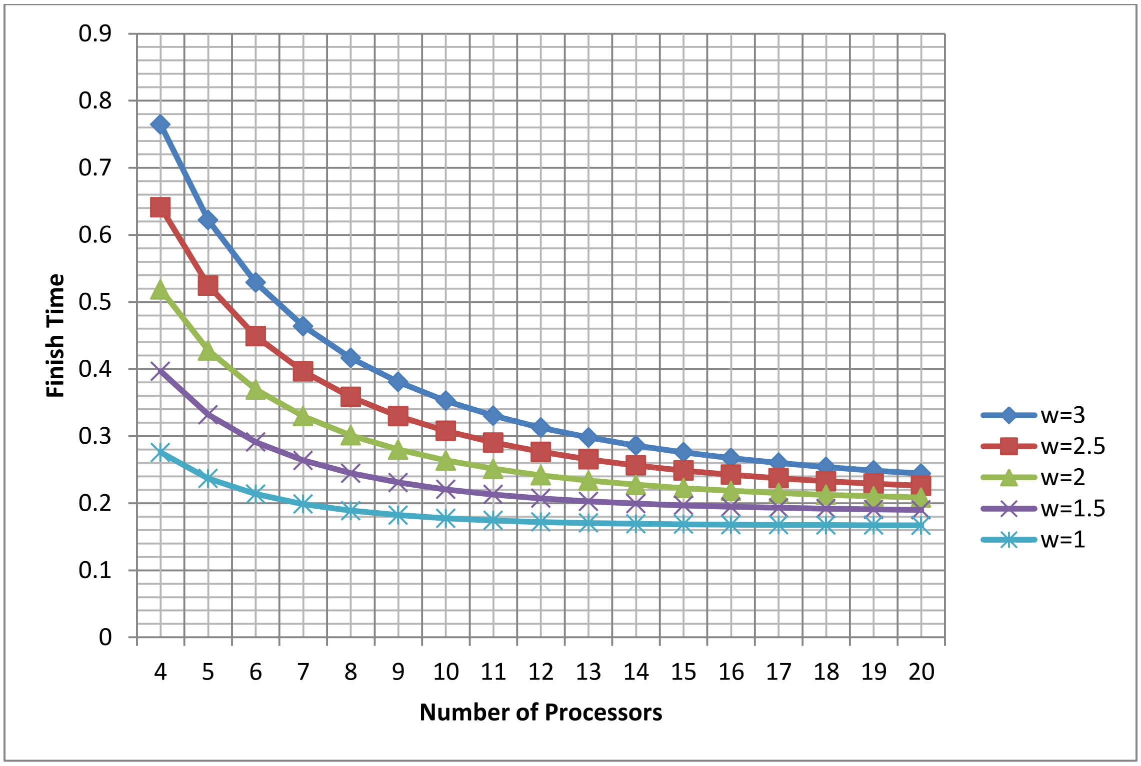

Fig.6. Finish time versus number of nodes, for two root sources single level homogeneous tree network and variable w.