Performance Evaluation of Parallel, Distributed and Grid Systems

A Dissertation Presented

by

Milton J. Jackson

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Doctor of Philosophy

in

Electrical Engineering

Stony Brook University

August 2008

Stony Brook University

The Graduate School

Milton J. Jackson

We, the dissertation committee for the above candidate for the

Doctor of Philosophy degree, hereby recommend

acceptance of this dissertation.

Thomas Robertazzi – Dissertation Advisor
Professor Electrical and Computer Engineering

Sangjin Hong – Chairperson of Defense
Professor Electrical and Computer Engineering

Vera Gorfinkel, Professor Electrical and Computing Engineering

David Ferguson, Professor Technology and Society

This dissertation is accepted by the Graduate School

Lawrence Martin
Dean of Graduate School

Abstract of the Dissertation

Performance Evaluation of Parallel, Distributed and Grid Systems

by

Milton J. Jackson

Doctor of Philosophy

Stony Brook University

2008

The intent of this work is to improve the performance evaluation of a number of different systems, by developing new methods and performance metrics. A technique is proposed where load sharing is modeled through the use of a set of differential equations. As examples, these equations are used to describe the flow of data between processors connected in a linear daisy chain and two dimensional mesh configurations. The processors have heterogeneous link speeds, processing speeds, and loads. The equations are designed to model the balancing of these loads over a period of time, by distributing the loads between the processors, allowing the amount at any processor's load to be accurately calculated at any point in time. A new and novel performance metric, utilization, is developed using distributed load theory. Four cases are used to determine utilization. All cases are sequentially distributed tree networks. Two cases have staggered starts, one with a root that does processing and the other with a root that does no processing. The other two cases have simultaneous starts, with a processing and a non-processing root, as above. The speedup of all cases is determined, and the performance metric of the speedup and utilization are compared. Using the same four cases as previously stated a method of performing a signature search, i.e. pattern recognition is developed. Depending on the type of sequentially distributed load, individual sets of equations are developed. These equation give the percent of the load a processor searches within a given time interval. The time interval is determined by communication delays. The utilization for two different types of linear daisy chains is found, one with a staggered start without a front end processor and the other with a simultaneous start with a front end processor. For the linear daisy chain it was necessary to develop a set of recursive equations to find the ratios between the loads on the processors before finding the utilization. These equations are not necessary with a tree network. The speedup for the linear daisy chains are found and compared with the utilization.

Table of Contents

List of Symbols

α_i : Load fraction of load assign to a link

$\alpha_i w_i T_{cp}$: Computation time of α_i on the i^{th} processor

$\alpha_i z_i T_{cm}$: Communication delay of the i^{th} link

AvgU: Average utilization

dt: Time interval

EQs: Equation set

Hyb: Hybrid Strategy

k: Ratio of loads

L: load

maxspeed: Cumulative speed of processors

N: total number of processors

$N_{99.6\%}$: Processor performance metric, 99.6% decrease

$N_{speedup}$: Processor performance metric, speedup

$N_{TN,Tf}$: Processor performance metric, last communication delay, finish time

N_{util}: Processor performance metric, utilization

P: Processor

RDP: Root does processing

RDOP: Root does no processing

Speedup: Performance metric

t: Time axis

T: Communication delay

T_{cm}: Communication intensity

T_f: Finish time

T_{cp}: Computation intensity

T_N: Last communication delay

U: Utilization

v: inverse link speed in the vertical direction

w: inverse processor speed

WFE: Processor with front end processing

WOFE: Processor with out front end processing

z: inverse link speed in vertical direction

List of Figures

List of Tables

Chapter 1

1. Introduction

Parallel processing has been used in many instances for problem solving throughout history. Now its principles are being applied to modern day computers. A grid search is a good example of this. When people are tasked to search a large area, each person is assigned a sector to search. This method greatly decreases the amount of time it takes to conduct the search. The idea behind this approach is to break a job down into manageable units. On a computer with more than one processor to achieve a benefit from parallel processing it is necessary to assign tasks to different processors. The key factor is communication, the ability to work at the same time without interfering with ongoing activities. For this to happen there must be a proper allocation of resources. To accomplish this flexible operating systems are needed. This is a task easily performed by the human brain. When sound and sight among other things are simultaneously processed and a decision is made based on the input and information in the memory. This is the goal of parallel processing.

1.1 Literature

A large literature on divisible load theory has been generated since the original work in 1988. Speedup calculations and linear daisy chains are cover in [9]. In [10] a load sharing problem is presented, where n processors are connected through a bus. The processors are controlled by sensors. It was found that a minimum solution time is achieved if all computations cease at the same time. In [11] closed form solutions for large symmetric tree networks are investigated. The processors in these tree networks may or may not have front end processing. Infinite networks are discussed in [12], where either closed form or numerical methods can be used to find the equivalent processor speed of a network with an infinite number of processors. Here the equivalent processor speed of a linear daisy chain is found. Computation in a two dimensional mesh composed of communicating processors is studied in [13]. A three dimensional mesh is considered in [21], where an optimum distributions that finds the minimum solutions among the processors in this mesh is investigated. Here the load is distributed from a parent processor in a node to node communication network, where the communication delays are incorporated into the process. Parallel distributed systems are used in [14] to distribute load in the least possible time. The purpose is to solve a problem of scheduling a divisible load where the communication and computation times are considered. This method is applied to a two dimensional mesh of processors. In [15] the front ends of communicating processors are used to minimize the finish time in a distributed linear network. For a single level tree network, with or without front end processing, the load is distributed in multiple installments with the result of minimizing the solution time in [16]. In [17] load distribution in linear networks, with and without front end processing is studied. Divisible load theory is introduced in this paper [18] by summarizing past research and highlighting its accomplishments. In [19] and [20] tree networks composed of processor with and without front end processing are investigated. Signature searching is covered in [22], where linear daisy chains and single level three networks are investigated by finding the expected time to perform a search for single and multiple files in parallel databases. In [23], a search is performed on a flat file, and the expected time for a search for single and multiple signature is found.

Closed form solutions are found in [24]. Here a linear network of communicating processors is studies. The processors having front end processing or not is of no consequence. Nor does it matter if the nodes receive load at the boundary or the interior the nodes of the network. The closed form solutions are valid. In [25] the instances when load originate at the parent node of a linear daisy chain as well as the interior nodes are studied. A linear daisy chain [26] with divisible load is investigated with the intention of reducing two or more loads into a single processor. The foundation of divisible load theory is laid out in [27]. Here the advantages of using divisible load theory are discussed, some of which are tractability and scalability. Divisible load theory can also be used with a variety of topologies. In [28] an analogy between superposition in and electrical network and superposition in a network using divisible load theory (DLT) is drawn. The linearity of DLT is shown by showing that superposition in DLT works. In [29] it was shown that a minimum solution time is achieved when all processors stop computing at the same time. Divisible load theory is used in [30] to model a grid system. Data from the STAR experiment facilities at the RHIC at Brookhaven National Labs is used in simulations to test the model validity.

1.2 Load Balancing

In chapter 2 Modeling load balancing through the use of differential equation is developed. In this chapter techniques are developed for load sharing. The loads are balanced using sets of differential equations. These equations use the concentration of load at a node, and the concentration of loads on the adjacent nodes to decide if load is to be transmitted or received. The loads will move from higher concentrations to lower concentrations until all the loads have reached zero.

First the linear daisy chain configuration is developed for a chain of nodes of indefinite length. The linear daisy chain is a topology that once developed can be used to create two dimensional m by n meshes. The sets of equations and the programs written for them make it possible to monitor the loads as they dissipated. At any point in time more load can be added to any node in the network. What this means is that an underutilized node can have more load directed to it. The processor speeds and link speeds are heterogeneous. Load can be directed to a node with a faster process speed, or one with a faster link speed. However it is not solely these speeds that determine which node will diminish the fastest. It is a combination of the processor and link speeds, and the concentrations on the node, and its adjacent nodes. These techniques can be used in any situation in which there is a load moving from one location to another. The load can be information or any concentration traveling through a medium. The load must be divisible. If the amount of load is known, how fast the concentrations are being processed, and how fast the concentrations are traveling through the medium. Then the equations in chapter 2 can be applied.

1.3 Sequentially Distributed Loads

In chapter 3 using divisible load theory, sequentially distributed loads are analyzed. A new performance metric, utilization is developed. The equations for utilization and speedup are derived from a Gantt like timing diagram. The diagrams describe single level tree networks that are composed of a parent processor also called the root, and the children processors. Four different sequential distributions are investigated. Case 1 has staggered start and a root that does no processing. Case 2 has staggered start and a root that does processing. Case 3 has simultaneous start and a root that does processing. Case 4 has simultaneous start and a root that does no processing. The four cases were compared using utilization and speedup. The performance metrics are use to explore the characteristics of the different cases. Utilization is a new performance metric and when used with speedup a more traditional metric it offers a useful comparisons.

1.4 Signature Searching

In chapter 4 the same sequentially distributed cases are investigated. The speedups are the same so the information gain in chapter 3 can be used here. Now the performance metrics are use to determine which network performs the better signature search or patter recognition. A different set of equations are developed for each of the timing diagram. How fast a search is being performed is based on parallel processing. Each processor is looking at different portions of a load that has been divided. So even though the processors are processing at the same speed the cumulative effect is that the job is accomplished faster. This is because when there are more processor looking for certain marker the chance of finding it faster are greatly increases. Besides the utilization and speedup performance metrics an additional method that uses the finish time and communication delay is developed to help validate the utilization and speedup metrics. The sought after patterns in these searches can range from radar signatures, DNA and finger prints to a host of data intensive problems.

1.5 Utilization of Linear Daisy Chains

In chapter 5 the utilization and speedup of two different types of linear daisy chain configurations are investigated. The first has a staggered start without front end processing. The second has a hybrid strategy with front end processing. For the second case the hybrid strategy means that the processor behaves as one with a staggered start when it is receiving information. It cannot receive information and compute at the same time. However it can perform computations while transmitting information at the same time. Sets of iterative equations were developed to find the ratios of the loads, as they could not be found using the previous methods. A description of how information is being processed as it moves through the network is provided for the linear daisy chain configurations and the timing diagrams. The timing diagrams describe the information in a manner that makes it easy to calculate the utilization and the speedup. also a different aspect of the utilization is use to make a determination about the networks. This time the average utilization is useful, as well as the utilization and the speedup.

Chapter 2

2. Modeling Load Balancing Through Differential Equations

2.1 Introduction

In this paper a method is advanced that uses differential equation to model load sharing. The differential equation are the decisions makers, and use the amount of load on adjacent processors, to determine if a task (load) is to be passed to another processor. The exchange of data between processors in two different topologies is demonstrated. These are the linear daisy chain and mesh configurations. Both configurations have heterogeneous link speeds, with processors that function at different rates. The loads of the linear daisy chain and mesh configurations are divisible, and have amounts that vary. The decisions made by the differential equations balance these loads, through a process of redistribution of the loads among the processors. The load is considered balanced when all the processors have reached zero. The exact amount of load at any processor can be determined precisely at any point in time.

2.2 State of The Art

There is a large body of literature on load balancing. Most of which is dedicated to indivisible jobs. Here some representative works will be considered that also encompass divisible jobs. Load balancing techniques are often components of an application. Their purpose is to make the applications perform better. Different techniques are better suited for certain applications. The types of load balancing techniques and applications vary. However the common thread in load balancing is the redistributions of tasks. The methods are diverse as in [1] where a work stealing algorithm is used that distributes workloads in a parallel system so that underutilized processors seek out work from other processors. Mitzenmacher demonstrates the usefulness of this modeling technique. When a processor is idle it tries to steal a task from a processor selected at random with uniform distribution. If the chosen processor has more than one task, a task is stolen. This model employs differential equations. The differential equations rely on the expected change in the behavior of the system over small periods of time. That is based on arrivals or departures of tasks. A core term of these equations is similar to a term in the equations in this paper, but used differently.

Another technique is diffusive load balancing [8]. In diffusive load balancing if a processor has a quantity of tasks greater than its neighbor it moves a portion of its tasks to the neighboring processor. The decision on rather to move some of the tasks is based solely on local information, and the amount of tasks transferred is in proportion to the differential between the number of tasks on the two processors. Furthermore only the number of tasks at each node is subject to attempts of equalization. This method of having nodes with greater loads transfer load to nodes with lesser loads is used in this paper. Other load balancing techniques are [3], where a high energy physics application uses a dynamic load balancing technique to balance the point of imbalance occurring in queues.

An application detailed in [5], where a dynamic load balancing technique is used as part of an Asynchronous Iterations-Asynchronous Communication (AIAC) model. Here the load balancing is not based on the amount of data, but the residual i.e. the max norm of the difference between a current value and two consecutive iterations. In [6] a dynamic load balancing method is developed for parallel applications. That has the purpose of evenly distributing work to processors. To ensure that processors are not idle while other are busy. For simple data structures the application divides the work into groups of equal size and distributes the groups among the processors. For complex data structures as in unstructured mashes, partitioning methods are used.

A similar method is in [7], where mesh adaption is shown to be an effective application for unstructured-grid computation. However a load imbalance is produced among the processors when used on parallel nodes. To solve this problem dynamic load balancing is used. The technique employs partitioning and remapping to balance the load. In [2] a load balancing algorithm for a distributed system is used. The algorithm has the purpose of minimizing the expected turnaround time. To prevent a situation in which a task waits for one particular processor while there is another idle processor that could process the task. The performance of the system is increased, when the instantaneous load on the multiple processors network is balanced. Presented in [4] is an agent based load balancing technique that is used in a homogeneous min-grid to achieve a uniform distribution of task to nodes. This solves the scheduling problem on the grid.

All of the techniques above have in part some bearing on this paper. This model is a load balancing technique, and could be placed as a component in some of the applications mentioned above. With this technique underutilized or idle processor are not used as part of the load balancing technique, but it would immediately detect these conditions. With changes to the program that uses this technique, the information on underutilized or idle nodes could be used to increase the rate at which all nodes reach a desired point. This technique would also detect an imbalance in queues. This technique would be extremely useful in the detection of imbalances in a system, since it knows the load at every node, at every time interval.

2.3 Model Description

Both the linear daisy chain and two dimensional mesh configurations use the same method for distributing loads. Each node is capable of sharing its load with its adjacent neighbors. The loads move from the nodes with larger loads to those with lesser loads. This process continues until a solution is reached, in other words, the loads reach zero at all nodes.

2.4 Variables For Chapter 1

w(i): the inverse processor speed

z(i): the inverse link speed in the horizontal directions

v(i): the inverse link speed in the vertical directions

L: the load

dt: the time interval

sgn(x): the signum function

N: the total number of processors

2.5 Linear Daisy Chain Configuration

In a linear daisy chain configuration, Figure 2.1, the following set of differential equations describes this process.

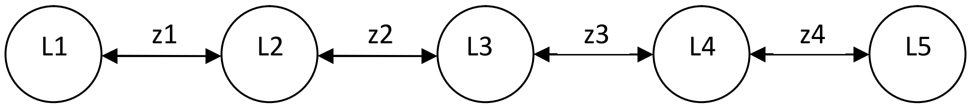


Figure 2. 1 Five loads connected in a linear daisy chain configuration, by four links.

For the linear daisy chain:
i: the number of processors between the first and last processor

$$dL(1) = dt\left[-\frac{1}{w(1)} + \frac{1}{z(1)}sgn(L(2) - L(1))\right] \quad (2.1)$$

$$dL2(i) = dt\left[-\frac{1}{w(i)} + \frac{1}{z(i)}sgn(L(i+1) - L(i)) + \frac{1}{z(i-1)}sgn(L(i-1) - L(i))\right] \quad (2.2)$$

$$dL3(N) = dt\left[-\frac{1}{w(N)} + \frac{1}{z(N-1)}sgn(L(N-1) - L(N))\right] \quad (2.3)$$

$$L(1) = L(1) + dL1(1) \qquad L(1) \geq 0 \tag{2.4}$$
$$L(i) = L(i) + dL2(i) \qquad L(i) \geq 0 \tag{2.5}$$
$$L(N) = L(N) + dL3(N) \qquad L(N) \geq 0 \tag{2.6}$$

The derivative of the load (dL) with respect to time is divided into three equations. The derivative dL1(1) represents the load on the first node, and the dL3(n) represents the load on the last node. The derivative dL2(i) represents the loads on interior nodes between the first and last nodes.

The inverse processor speed (w) has a minus sign to insure that the act of processing always serves to decrease the load when added to the inverse link speed. The sign of the inverse link speed is determined by the difference between the loads. If the difference is greater than zero the sign is positive, and if it is less than zero it is negative. Load is transported from nodes with larger loads to adjacent nodes with lesser loads. In equation (2.1), the inverse link speed (z1) and the inverse processor speed (w1) are fixed.

The differential equation dL1(1) uses the difference between the first load(L1) and, its nearest neighbor the second load(L2), and the inverse link speed between them in its calculations. In equation (2.1) “one” is used for the load at the first node, and “two” for the load at the second node instead of i. The reason will become clear in the description of dL2(i). The derivative dL3(N) works similar to dL1(1), the only difference being that dL1(1) uses the node to its right and dL3(N) used the node to its left in the calculations. The derivative dL2(i) takes into account its neighbors on either side. The variable i start at 2 to prevent division by zero in the term $\frac{1}{z(i-1)}$ of equation (2.2). Other than this dL2(i) functions in the same sense as dL1(1) and dL3(N).Once the derivatives of the loads are found they are added to the original loads, L(1), L(i) and L(N) creating a new set of loads. The process repeats until the loads are balanced, i.e. each node's load equals zero. Loads are forced to be no less than zero. When a load reaches zero it remains at zero unless a new load is added to that particular node. The load at a node may also be increased at anytime in the process. Furthermore, the number of nodes may be increased to any size n.

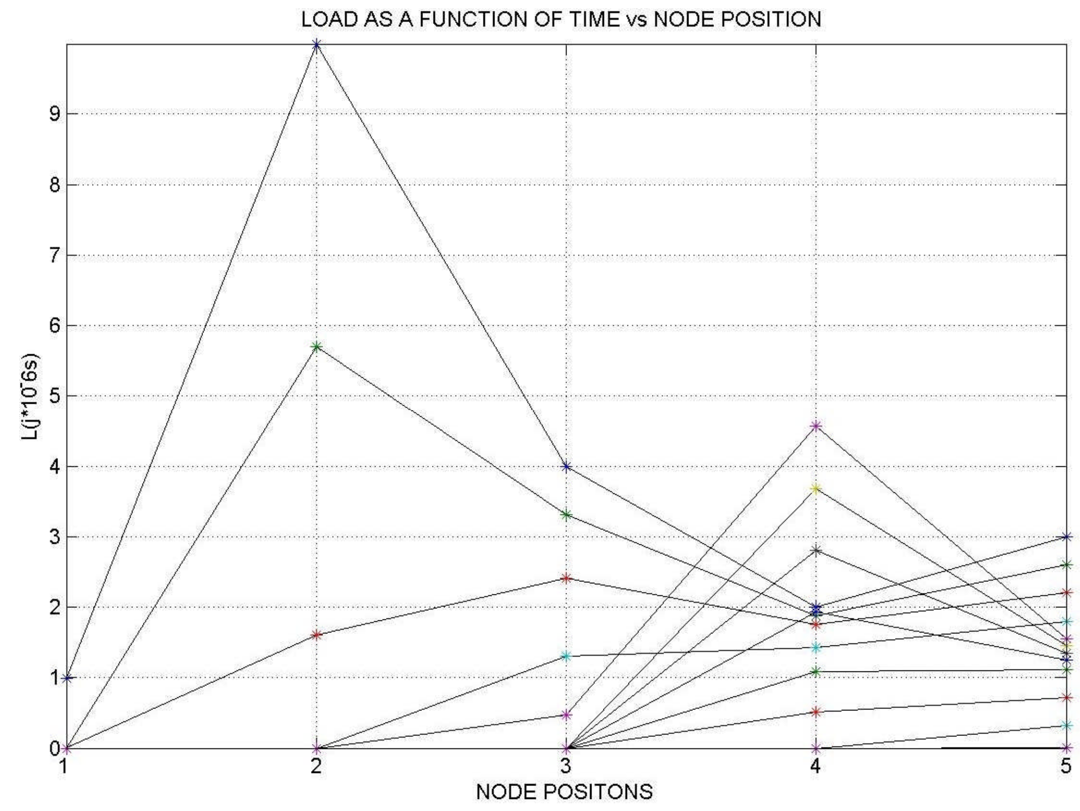


Figure 2.2. Load as a function of time vs. node position. There are five nodes, and the loads on these nodes are measured at time, $j \times (1 \times 10^6 s)$ where j represents the number of iterations of equations (2.1) through (2.6).

In Figure 2.2, the loads as a function of time are plotted against the node position. The amount of load on each node at a specific time is designated by the function $L(j \times dt(s))$, where j is the number of iterations of equations (2.1) through (2.6), and dt is the time interval, the iteration occurred in, here 1×10^{-6} s. As can be seen in Figure 2.2, the initial load on nodes one through five are respectively [1 10 4 2 3]. They diminish with time in accordance with equations (2.1) through (2.6). However at a given time when j = 3500 and $dt = 1 \times 10^6 s$, the load at node 4 is increased by 5, L(j × 10^{-6} s) = L(0.0035 s) = 5. The increase can be seen at node 4 on the graph above. Note in Figure 2.2 the increase appears to be 4.5 and not 5. The discrepancy is due to the fact that the information in the graph is plotted every 1000 iterations. The difference is the amount the load has decreased, from the time the load is input to the time the data is updated. This increases the time it will take for this node to reach zero.

2.6 Mesh Configuration

In the mesh configuration Figure 2.3, the nodes are extended in the vertical and horizontal directions forming a five by five mesh. In practice any m by n mesh is possible. There are 25 nodes (N), 24 inverse links speeds (z) in the horizontal directions, and 20 inverse link speeds (v) in the vertical directions. This mesh processes loads the same as the linear daisy chain, with an extra dimension added. The process is governed by nine equations that will work for any m by n mesh. However for the equations to work properly the mesh must be labeled as shown in Figure 2.3. Other labeling schemes are possible, but result in somewhat different equations. In the nine equations that describe the process of transferring loads between nodes, m is the number of rows and n is the number of columns. The equations are numbered (2.7) through (2.15).

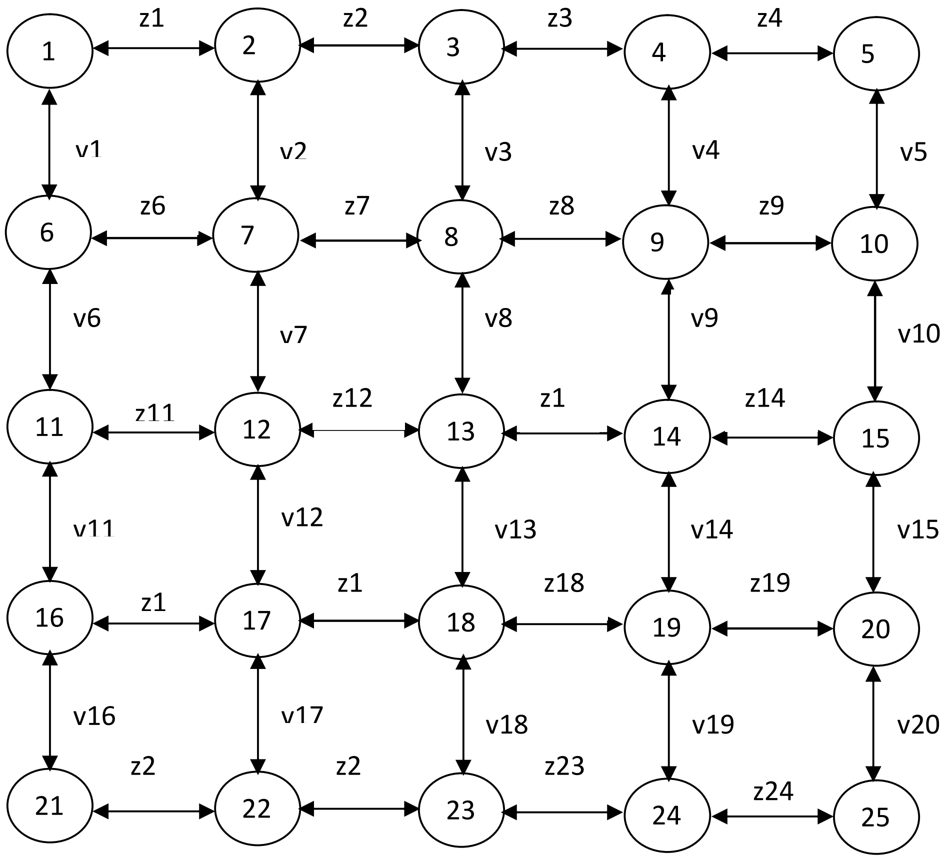


Figure2.3. A 5 by 5 mesh configuration, composed of 25 nodes, 20 horizontal links and 20 vertical links, here N is the total number nodes, m is the number of rows and n is the number of columns.

For the mesh configuration:

i: the number of processors from 1 to n

for i = i1 = 1

$$dL1(i) = dt\left[-\frac{1}{w(i)} + \frac{1}{z(i)}sgn(L(i+1)-L(i)) + \frac{1}{v(i)}sgn(L(i+n)-L(n))\right] \quad (2.7)$$

for i = i2 = 2 to n-1, increments of 1

$$dL2(i) = dt\left[-\frac{1}{w(i)} + \frac{1}{z(i)}sgn(L(i+1)-L(i)) + \frac{1}{z(i-1)}sgn(L(i-1)-L(i)) + \frac{1}{v(i)}sgn(L(i+n)-L(n))\right] \quad (2.8)$$

for i = i3 = n

$$dL3(i) = dt\left[-\frac{1}{w(i)} + \frac{1}{z(i-1)}sgn(L(i-1)-L(i)) + \frac{1}{v(i)}sgn(L(i+n)-L(i))\right] \quad (2.9)$$

for i = i4 = n+1 to (N-n+1) –n, in increments of n

$$dL4(i) = dt\left[-\frac{1}{w(i)} + \frac{1}{z(i)}sgn(L(i+1)-L(i)) + \frac{1}{v(i)}sgn(L(i+n)-L(i)) + \frac{1}{v(i-n)}sgn(L(i-n)-L(i))\right] \quad (2.10)$$

for rows = m-2

g = 0 to rows -1
i = i5 = [(n + 2) : in increments of 1: to (2n-1)] + gn

$$dL5(i) = dt\left[-\frac{1}{w(i)} + \frac{1}{z(i)}sgn(L(i+1)-L(i)) + \frac{1}{z(i-1)}sgn(L(i-1)-L(i)) + \frac{1}{v(i)}sgn(L(i+n)-L(i)) + \frac{1}{v(i-n)}sgn(L(i-n)-L(i))\right] \quad (2.11)$$

for i = i6 = 2n: in increments of n: to (N-n)

$$dL6(i) = dt\left[-\frac{1}{w(i)} + \frac{1}{z(i-1)}sgn(L(i-1)-L(i)) + \frac{1}{v(i)}sgn(L(i+n)-L(i)) + \frac{1}{v(i-n)}sgn(L(i-n)-L(i))\right] \tag{2.12}$$

for i = i7 = (N-n) +1

$$dL7(i) = dt\left[-\frac{1}{w(i)} + \frac{1}{z(i)}sgn(L(i+1)-L(i)) + \frac{1}{v(i-n)}sgn(L(i-n)-L(i))\right] \tag{2.13}$$

for i = i8 = (N-n) + 2: in increments of 1: to (N – 1)

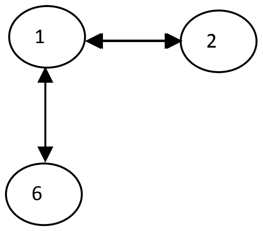
$$dL8(i) = dt\left[-\frac{1}{w(i)} + \frac{1}{z(i)}sgn(L(i+1)-L(i)) + \frac{1}{z(i-1)}sgn(L(i-1)-L(i)) + \frac{1}{v(i-n)}sgn(L(i-n)-L(i))\right] \tag{2.14}$$

for i = i9 = N

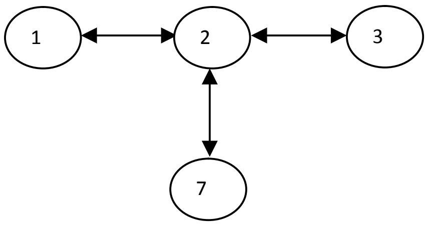
$$dL9(i) = dt\left[-\frac{1}{w(N)} + \frac{1}{z(N-1)}sgn(L(N-1)-L(N)) + \frac{1}{v(N-n)}sgn(L(N-n)-L(N))\right] \tag{2.15}$$

$$L(i1) = L(i1) + dL1(i1) \qquad L(i1) \geq 0 \tag{2.16}$$
$$L(i2) = L(i2) + dL2(i2) \qquad L(i2) \geq 0 \tag{2.17}$$
$$L(i3) = L(i3) + dL3(i3) \qquad L(i3) \geq 0 \tag{2.18}$$
$$L(i4) = L(i4) + dL4(i4) \qquad L(i4) \geq 0 \tag{2.19}$$
$$L(i5) = L(i5) + dL5(i5) \qquad L(i5) \geq 0 \tag{2.20}$$
$$L(i6) = L(i6) + dL6(i6) \qquad L(i6) \geq 0 \tag{2.21}$$
$$L(i7) = L(i7) + dL7(i7) \qquad L(i7) \geq 0 \tag{2.22}$$
$$L(i8) = L(i8) + dL8(i8) \qquad L(i8) \geq 0 \tag{2.23}$$
$$L(i9) = L(i9) + dL9(i9) \qquad L(i9) \geq 0 \tag{2.24}$$

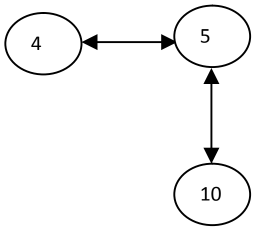
The mesh in Figure 2.3 is described by nine structures shown individually in Figure 2.4. Each equation corresponds to a structure, and each structure has one or more components. Furthermore each component has a center node. In figure 2.4 only the center node of the first component is shown when there is more than one component. In the 5 by 5 mesh of Figure 2.3, there is a total of 25 components. The equation numbers (2.7) to (2.15) correspond to the structure numbers (7) to (15). The structures representing the upper left and right hand corners, (7) and (9) respectively, and the lower left and right hand corners (13) and (15) respectively each have one component, which is composed of three nodes and two links. Structure (8) has three components corresponding to the number of nodes (columns) between the first and last node of the first row. Each component is made up of four nodes and three links. Structure (14) is the complement of (8). This structure is found in the last row of the mesh, and contains the same number of components, nodes and links as (8), for the same reasons. Structure (10) contains three component, based on the number of nodes (rows) found between the first and last rows of the mesh. It has four nodes and three links. The final structure, (11) has nine components each composed of five nodes and four links.



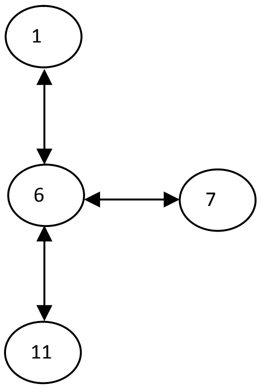
(7)



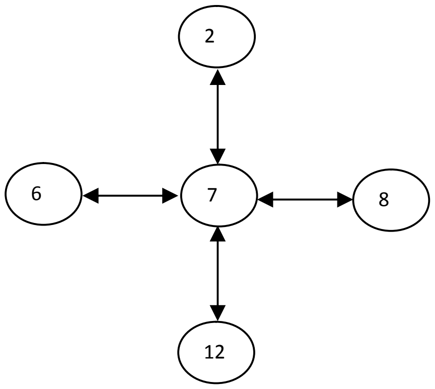
(8)



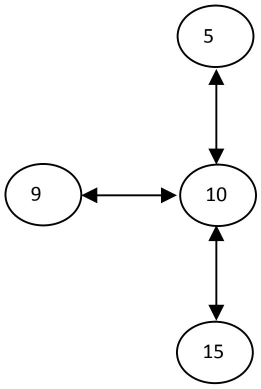
(9)



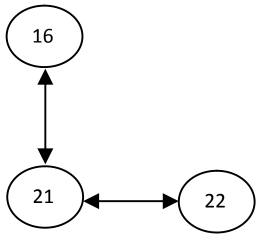
(10)



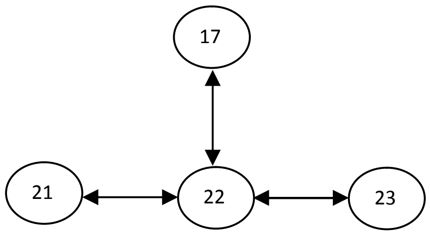
(11)



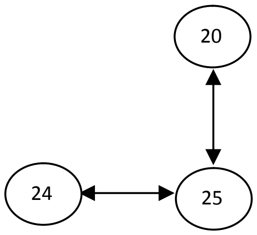
(12)



(13)



(14)



(15)

Figure 2.4. The nine structures describe any m by n mesh. The structure's number is the equation's number with the 2 removed from the front of the equations designation. For a 5 by 5 mesh as in Figure 2.3, the structures (7), (9), (13) and (15) each have one component. The structures (8), (10), (12) and (14) each have three components and structure (11) has nine components. The center nodes mark the position of the first component of each structure.

Once the derivative of a load at a particular node is found it is added to the current load of that node. This produces a new value for that load. This process is seen in equations (2.16) through (2.24). As with the linear daisy chain configuration, the load at each node in the five by five mesh is plotted as a function of load versus time, and time versus the node position, Figure 2.5.

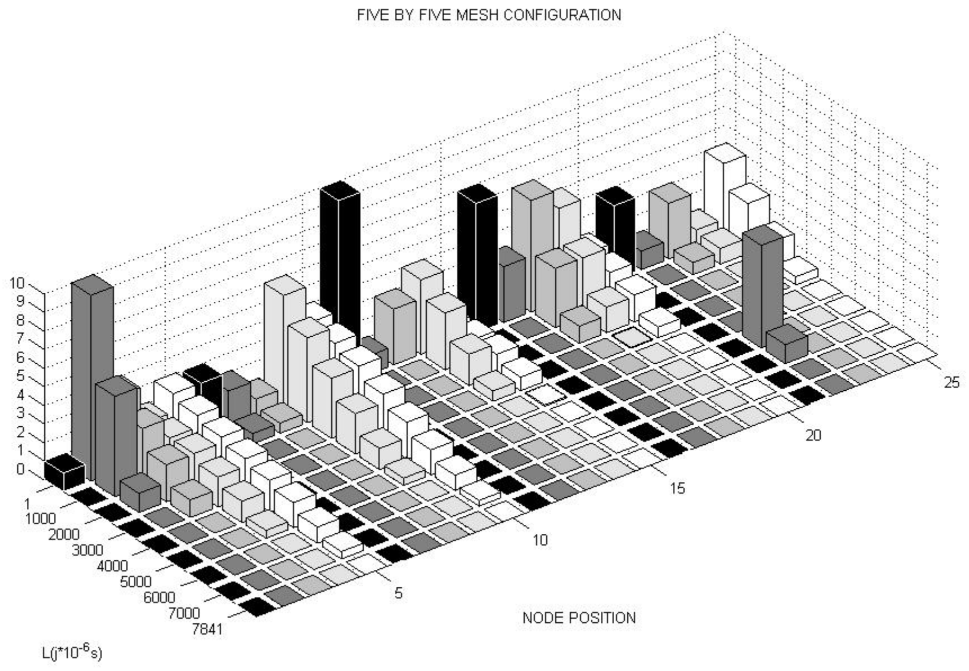


Figure 2.5. Plot of the loads versus the loads as a function of time versus the node position. The x axis gives the nodes position, the y axis the load as a function of time, and the z axis the load in mega bytes.

The rows in Figure 2.5, starting at j equals one represents the initial input loads on the nodes of the 5 by 5 mesh in Figure 2.3. The 5 by 5 mesh has been transformed into an array of length 25, and each element of the array 1 through 25 corresponds to the node number and load of the 5 by 5 mesh. Every 1000 iterations of equations (2.7) through (2.24) produces a new row, until all load reach zero at j = 7841. At j = 5000, the load at node position 22, $L(j \times 10^{-6}s)$ is set equal to 10. This node rapidly approaches zero because there are no load bearing nodes adjacent to it. Equation (2.14) governs this process.

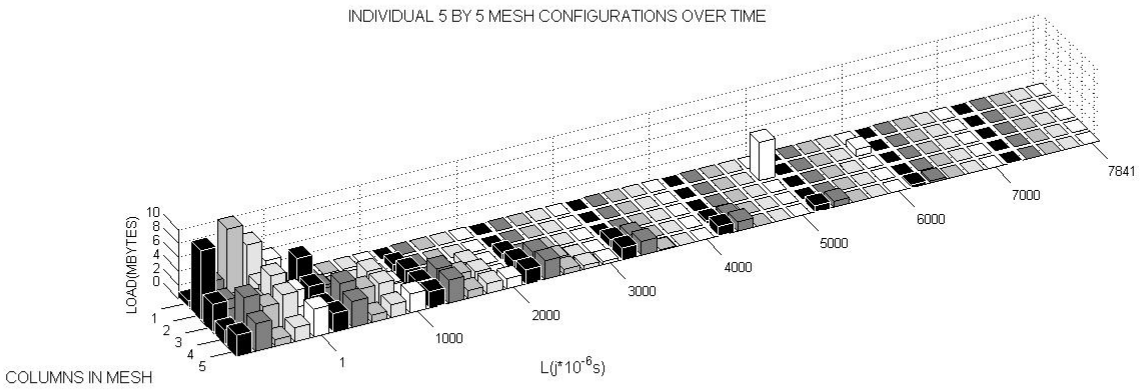


Figure 2.6. Displays the 5 by 5 meshes as time increases. The y axis shows the number of columns, and the x axis the rows, listed every 1000 iterations.

With the nodes adjacent to node 22 being zero, the four terms of equation (2.14) are negative, generating a large negative number. When dL8(i8) is subtracted from L(i8) in equation (2.23), the load goes to zero within 2000 iterations.

A geometrically clear picture of the surrounding nodes can be seen i figure 2.6. After the initial 5 by 5 mesh is plotted, with one on the x axis marking the last row of the 5 by 5 mesh. The mesh is plotted again after 1000 iterations. This process continues until all nodes in the mesh are zero.

The previous graphs have been generated by programming the given equations. For the one dimensional daisy chain equations (2.1) through (2.6) were used, and for the two dimensional mesh equations (2.7) through (2.24). The two dimensional m by n mesh can be extended into three dimensions by an arbitrary length p, producing and m by n by p mesh. A 5 by 5 by 5 mesh would require one additional parameter, the link along the z axis h. This cube would be composed of the following structures. Eight corner structures each consisting of one component having three links and four nodes. Twelve edge structures, where each structure has three components made of four links and five nodes. Six face structure comprised of nine components consisting of five links and six nodes. The last structure is for the internal nodes of the cube. This structure has 27 components, each having six links, and seven nodes. The center nodes of the components do not lie on the edges or surfaces of the cube.

2.7 Conclusion And Extensions of Chapter 2

For initial equal loads, processors speeds and links speed, if there are no shared nodes in the structures or if the adjacent nodes are zero the structures works as would be intuitively expected. The number of links dictates how fast the load decreases. The structure with the most links decreases the fastest. This decrease is imperceptible in graphs such as figure 2.6 and 2.7. However a decrease in the second and third decimal place is readily seen in the numbers used to generate the graphs, when the parameters are changed to observe smaller steps, i.e. when the values for the loads are listed every 10 iterations, instead of increments of 1000 iterations. When the processor speeds, and link speeds are chosen such that the processor speed is one third that of the link speed, and the loads are chosen randomly between 1 and 10, the data derived from the equations demonstrates that the equations perform as intended. In other words the loads tend to be balanced over time.

It should be noted that this methodology can be used to model other protocols involving thresholds for allowing load transfers, different speeds for different amounts at the loads, and time varying processor and links speeds. Modeling chemical diffusion through a cell and across the cell membrane is another potential future application.

Chapter 3

3. Sequentially Distributed Loads

3.1 Introduction

Divisible load theory (DLT) was developed in response to the need to handle ever increasing large amounts of data. In DLT a load can be arbitrarily divided an assigned to different processors and links. With DLT linear mathematical models are created which allow the investigation of performance metrics [9]. In this paper the focus is on utilization and speedup. The equations for utilization and speedup are derived, and analyzed for four different cases. These are: case 1, sequential distribution, staggered start, with a root that does no processing; case 2 sequential distribution, staggered start, with a root that does processing; case 3 sequential distribution, simultaneous start, with a root that does processing; and case 4 sequential distribution, simultaneous start, with a root that does no processing.

3.2 Performance Metric Description

Utilization is the ratio of useful time to total time. It is calculated for each processor in a network, and averaged over the entire network. In the literature variables needed to understand performance characteristics such as utilization and speedup are as follows.

3.3 Variables For Chapter 3

N: the number of processor

α_i [load]: The fraction of the load assigned to the link. The sum of the load fractions α_i equals one, the unit load.

z_i [second/load]: The inverse link speed.

w_i [second/load]: The inverse processor speed.

T_{cm}: The communication intensity, it is a dimensionless quantity used to increase or decrease the link speed for a particular job.

T_{cp}: The computation intensity, it is a dimensionless quantity used to increase or decrease the processor speed for a particular job.

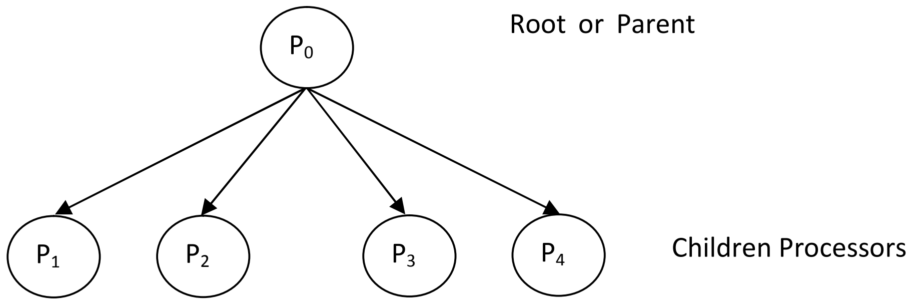
$\alpha_i z_i T_{cm}$ [second]: The time required to transmit the load fraction α_i over the ith link.

$\alpha_i w_i T_{cp}$ [second]: The time required to process the load fraction α_i of the entire load on the ith processor

T_f [second]: The finish time or makespan. This is the time it take the last processor to complete it computation.

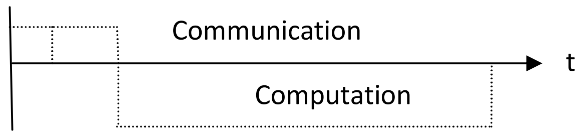
Speedup is the ratio of the time it takes to complete a computation on one processor, to the time it takes to complete a computation on an entire tree with N children processors. Speedup is a dimensionless number that gives the performance gained by using multiple processors.

.



(a)

Single level tree network with one parent processor P_0, and four children processors $P_1, \ldots, P_4$



(b)

Building block of a Gantt chart-like timing diagram, information above the time axis is dedicated to communication and that below to computation

Figure 3.1.

3.4 Descriptions of Categories Sited in Cases 1 through 4

Sequential Distribution: In sequential distribution the root first transmits the entire load to the child processor P_1, then to all other child processors in sequential order.

Staggered Start: When a processor has received the entire load it has been assigned. The processor begins processing, and continues processing until the assigned load is exhausted.

Simultaneous Start: The processor immediately starts processing the load as it is received.

With Front End Processing: Processors with front end processing have the ability to send and receive load at the same time.

Without Front End Processing: Processors without front end processing can either send or received load, but not both at the same time.

3.5 CASE 1
SEQUENTIAL DISTRIBUTION
STAGGERED START
ROOT DOES NO PROCESSING

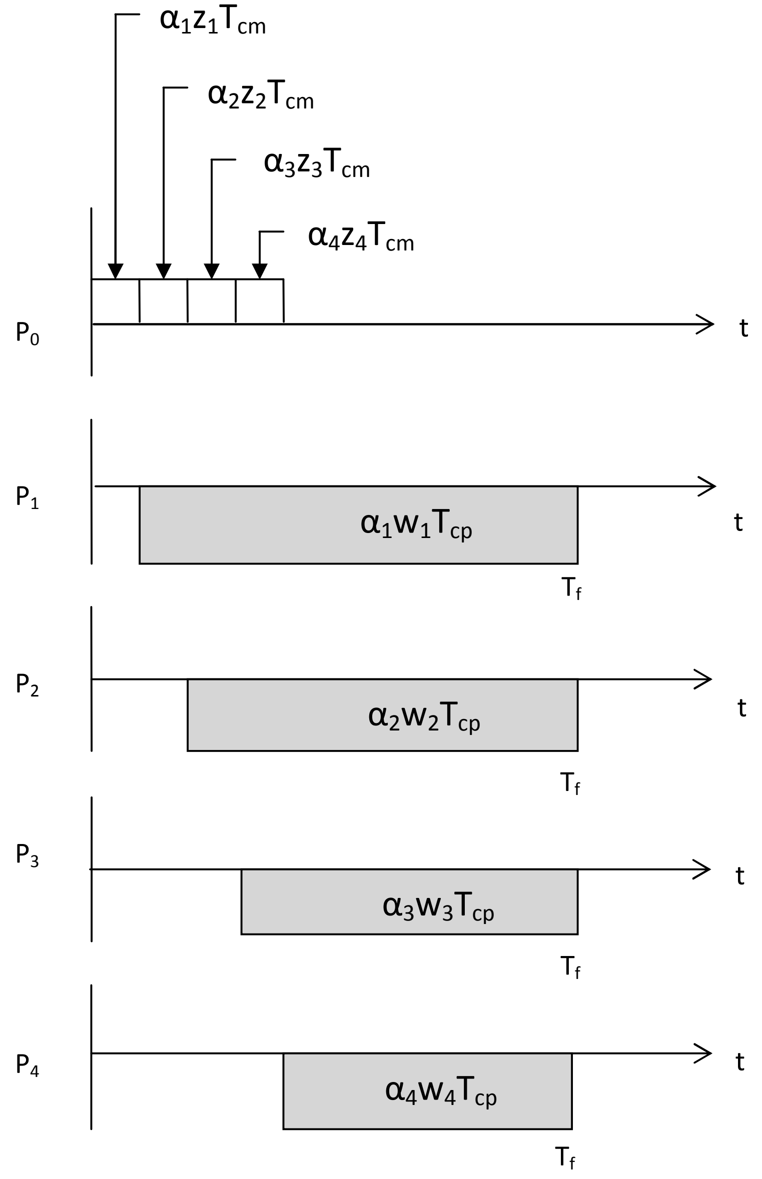


Figure 3.2. Timing diagram of single level tree with sequential distribution, staggered start and a root that does no processing

In Figure 3.2 the loads are sequentially distributed by the parent processor to the children processors. The parent processor cannot process load. The children processors do not have front end processing.

From Figure 3.2

$$T_{f1} = \alpha_1 z_1 T_{cm} + \alpha_1 w_1 T_{cp} \tag{3.1}$$

$$U_1 = \frac{\alpha_1 w_1 T_{cp}}{T_{f1}} \tag{3.2}$$

$$U_1 = \frac{\alpha_1 w_1 T_{cp}}{\alpha_1 z_1 T_{cm} + \alpha_1 w_1 T_{cp}} \tag{3.3}$$

$$U_1 = \frac{\frac{\alpha_1 w_1 T_{cp}}{\alpha_1 w_1 T_{cp}}}{\frac{\alpha_1 z_1 T_{cm}}{\alpha_1 w_1 T_{cp}} + \frac{\alpha_1 w_1 T_{cp}}{\alpha_1 w_1 T_{cp}}} \tag{3.4}$$

$$U_1 = \frac{1}{1 + \frac{\alpha_1 z_1 T_{cm}}{\alpha_1 w_1 T_{cp}}} \tag{3.5}$$

For $z = z_i$

$$U_1 = \frac{1}{1 + \frac{z T_{cm}}{w_1 T_{cp}}} \tag{3.6}$$

Equation (3.6) gives the utilization of processor P_1

From Figure 3.2

$$T_{f2} = \alpha_1 z_1 T_{cm} + \alpha_2 z_2 T_{cm} + \alpha_2 w_2 T_{cp} \tag{3.7}$$

$$U_2 = \frac{\alpha_2 w_2 T_{cp}}{T_{f2}} \tag{3.8}$$

$$U_2 = \frac{\alpha_2 w_2 T_{cp}}{\alpha_1 z_1 T_{cm} + \alpha_2 z_2 T_{cm} + \alpha_2 w_2 T_{cp}} \tag{3.9}$$

$$U_2 = \frac{\frac{\alpha_2 w_2 T_{cp}}{\alpha_2 w_2 T_{cp}}}{\frac{\alpha_1 z_1 T_{cm}}{\alpha_2 w_2 T_{cp}} + \frac{\alpha_2 z_2 T_{cm}}{\alpha_2 w_2 T_{cp}} + \frac{\alpha_2 w_2 T_{cp}}{\alpha_2 w_2 T_{cp}}} \tag{3.10}$$

$$U_2 = \frac{1}{1 + \frac{\alpha_1 z_1 T_{cm}}{\alpha_2 w_2 T_{cp}} + \frac{\alpha_2 z_2 T_{cm}}{\alpha_2 w_2 T_{cp}}} \tag{3.11}$$

$$U_2 = \frac{1}{1 + \left(\frac{\alpha_1}{\alpha_2} z_1 + \frac{\alpha_2}{\alpha_2} z_2\right) \frac{T_{cm}}{w_2 T_{cp}}} \tag{3.12}$$

For $z_i = z$

$$U_2 = \frac{1}{\left(\frac{\alpha_1}{\alpha_2} + \frac{\alpha_2}{\alpha_2}\right) \frac{z T_{cm}}{w_2 T_{cp}}} \tag{3.13}$$

Removing the load fraction α_i

From Figure 3.2

$$\alpha_2 z_2 T_{cm} + \alpha_2 w_2 T_{cp} = \alpha_1 w_1 T_{cp} \tag{3.14}$$

For $z_i = z$

$$\alpha_2 \left(z T_{cm} + w_2 T_{cp}\right) = \alpha_1 w_1 T_{cp} \tag{3.15}$$

$$\alpha_2 = \frac{w_1 T_{cp}}{z T_{cm} + w_1 T_{cp}} \alpha_1 \tag{3.16}$$

$$k_1 = \frac{w_1 T_{cp}}{z T_{cm} + w_1 T_{cp}} \tag{3.17}$$

$$\alpha_2 = k_1 \alpha_1 \tag{3.18}$$

Substituting $\frac{\alpha_1}{\alpha_2} = \frac{1}{k1}$ into equation (3.13)

$$U_2 = \frac{1}{1 + \left(\frac{1}{k_1} + 1\right) \frac{zT_{cm}}{w_2 T_{cp}}} \tag{3.19}$$

Equation (3.19) gives the utilization of processor P_2

From Figure 3.2

$$T_{f3} = \alpha_1 z_1 T_{cm} + \alpha_2 z_2 T_{cm} + \alpha_3 z_3 T_{cm} + \alpha_3 w_3 T_{cp} \tag{3.20}$$

$$U_3 = \frac{\alpha_3 w_3 T_{cp}}{T_{f3}} \tag{3.21}$$

$$U_3 = \frac{\alpha_3 w_3 T_{cp}}{\alpha_1 z_1 T_{cm} + \alpha_2 z_2 T_{cm} + \alpha_3 z_3 T_{cm} + \alpha_3 w_3 T_{cp}} \tag{3.22}$$

$$U_3 = \frac{\frac{\alpha_3 w_3 T_{cp}}{\alpha_3 w_3 T_{cp}}}{\frac{\alpha_1 z_1 T_{cm}}{\alpha_3 w_3 T_{cp}} + \frac{\alpha_2 z_2 T_{cm}}{\alpha_3 w_3 T_{cp}} + \frac{\alpha_3 z_3 T_{cm}}{\alpha_3 w_3 T_{cp}} + \frac{\alpha_3 w_3 T_{cp}}{\alpha_3 w_3 T_{cp}}} \tag{3.23}$$

$$U_3 = \frac{1}{\frac{\alpha_1 z_1 T_{cm}}{\alpha_3 w_3 T_{cp}} + \frac{\alpha_2 z_2 T_{cm}}{\alpha_3 w_3 T_{cp}} + \frac{\alpha_3 z_3 T_{cm}}{\alpha_3 w_3 T_{cp}} + 1} \tag{3.24}$$

$$U_3 = \frac{1}{\left(\frac{\alpha_1}{\alpha_3} z_1 + \frac{\alpha_2}{\alpha_3} z_2 + \frac{\alpha_3}{\alpha_3} z_3\right) \frac{T_{cm}}{w_3 T_{cp} + 1}} \tag{3.25}$$

For $z_i = z$

$$U_3 = \frac{1}{\left(\frac{\alpha_1}{\alpha_3} + \frac{\alpha_2}{\alpha_3} + \frac{\alpha_3}{\alpha_3}\right) \frac{zT_{cm}}{w_3 T_{cp} + 1}} \tag{3.26}$$

From Figure 3.2

$$\alpha_3 z_3 T_{cm} + \alpha_3 w_3 T_{cp} = \alpha_2 w_2 T_{cp} \tag{3.27}$$

For $z_i = z$

$$\alpha_3 \left(zT_{cm} + w_3 T_{cp}\right) = \alpha_2 w_2 T_{cp} \tag{3.28}$$

$$\alpha_3 = \left(\frac{w_2 T_{cp}}{zT_{cm} + w_3 T_{cp}}\right) \alpha_2 \tag{3.29}$$

$$k_2 = \frac{w_2 T_{cp}}{z T_{cm} + w_3 T_{cp}} \tag{3.30}$$

$$\alpha_3 = k_2 \alpha_2 \tag{3.31}$$

Using equation (3.18)

$$\alpha_3 = k_1 k_2 \alpha_1 \tag{3.32}$$

Substituting $\frac{\alpha_1}{\alpha_3} = k_1 k_2$ from equation(3.32) and $\frac{\alpha_2}{\alpha_3} = \frac{1}{k2}$ from equation

(3.31) into equation (3.26) yields

$$U_3 = \frac{1}{\left(\frac{1}{k_1 k_2} + \frac{1}{k_2} + 1\right) \frac{z T_{cm}}{w_3 T_{cp}}} \tag{3.33}$$

Equation (3.33) gives the utilization of processor P_3
From Figure 3.2

$$T_{f4} = \alpha_1 z_1 T_{cm} + \alpha_2 z_2 T_{cm} + \alpha_3 z_3 T_{cm} + \alpha_4 z_4 T_{cp} + \alpha_4 w_4 T_{cp} \tag{3.34}$$

$$U_4 = \frac{\alpha_4 w_4 T_{cp}}{T_{f4}} \tag{3.35}$$

$$U_4 = \frac{\alpha_4 w_4 T_{cp}}{\alpha_1 z_1 T_{cm} + \alpha_2 z_2 T_{cm} + \alpha_3 z_3 T_{cm} + \alpha_4 z_4 T_{cp} + \alpha_4 w_4 T_{cp}} \tag{3.36}$$

$$U_4 = \frac{\frac{\alpha_4 w_4 T_{cp}}{\alpha_4 w_4 T_{cp}}}{\frac{\alpha_1 z_1 T_{cm}}{\alpha_4 w_4 T_{cp}} + \frac{\alpha_2 z_2 T_{cm}}{\alpha_4 w_4 T_{cp}} + \frac{\alpha_3 z_3 T_{cm}}{\alpha_4 w_4 T_{cp}} + \frac{\alpha_4 z_4 T_{cp}}{\alpha_4 w_4 T_{cp}} + \frac{\alpha_4 w_4 T_{cp}}{\alpha_4 w_4 T_{cp}}} \tag{3.37}$$

$$U_4 = \frac{1}{\frac{\alpha_1 z_1 T_{cm}}{\alpha_4 w_4 T_{cp}} + \frac{\alpha_2 z_2 T_{cm}}{\alpha_4 w_4 T_{cp}} + \frac{\alpha_3 z_3 T_{cm}}{\alpha_4 w_4 T_{cp}} + 1} \tag{3.38}$$

$$U_4 = \frac{1}{1 + \left(\frac{\alpha_1}{\alpha_4} z_1 + \frac{\alpha_2}{\alpha_4} z_2 + \frac{\alpha_3}{\alpha_4} z_3\right) \frac{T_{cm}}{w_4 T_{cP}}} \tag{3.39}$$

for $z_i = z$

$$U_4 = \frac{1}{1+\left(\frac{\alpha_1}{\alpha_4}+\frac{\alpha_2}{\alpha_4}+\frac{\alpha_3}{\alpha_4}\right)\frac{zT_{cm}}{w_4T_{cP}}} \tag{3.40}$$

From Figure 3.2

$$\alpha_4 z_4 T_{cm} + \alpha_4 w_4 T_{cp} = \alpha_3 w_3 T_{cp} \tag{3.41}$$

$$\alpha_4\left(z_4T_{cm} + w_4T_{cp}\right) = \alpha_3 w_3 T_{cp} \tag{3.42}$$

$$\alpha_4 = \frac{w_3T_{cp}}{z_4T_{cm}+w_4T_{cp}}\alpha_3 \tag{3.43}$$

$$k_3 = \frac{w_3T_{cp}}{z_4T_{cm}+w_4T_{cp}} \tag{3.44}$$

$$\alpha_4 = k_3\alpha_3 \tag{3.45}$$

Using equation (3.32) gives

$$\alpha_4 = k_1k_2k_3\alpha_1 \tag{3.46}$$

From equation (3.31) $\alpha_3 = k_2\alpha_2$ and equation (3.45) $\alpha_4 = k_3\alpha_3$

$$\frac{\alpha_2}{\alpha_4} = \frac{1}{k_2k_3} \tag{3.47}$$

Substituting equations (3.45), (3.46) and (3.47) into equation (3.40) yields

$$U_4 = \frac{1}{1+\left(\frac{1}{k_1k_2k_3}+\frac{1}{k_2k_3}+\frac{1}{k_3}\right)\frac{zT_{cm}}{w_4T_{cP}}} \tag{3.48}$$

Equation (3.48) is the utilization of Processor P_4

The general form of the utilization equation for case 1 is

$$U_i = \frac{1}{1+\left[\sum_{j=1}^{i}\prod_{m=i}^{i-1}k_m\right]\frac{zT_{cm}}{w_iT_{cp})}} \qquad \text{for } i = 1, 2, ..., N \tag{3.49}$$

The general form for the average utilization is

$$AvgU = \frac{1}{N}\sum_{i=1}^{N} U_i \qquad \text{for } i = 1, 2, ..., N \tag{3.50}$$

Here N is the number of children processors.

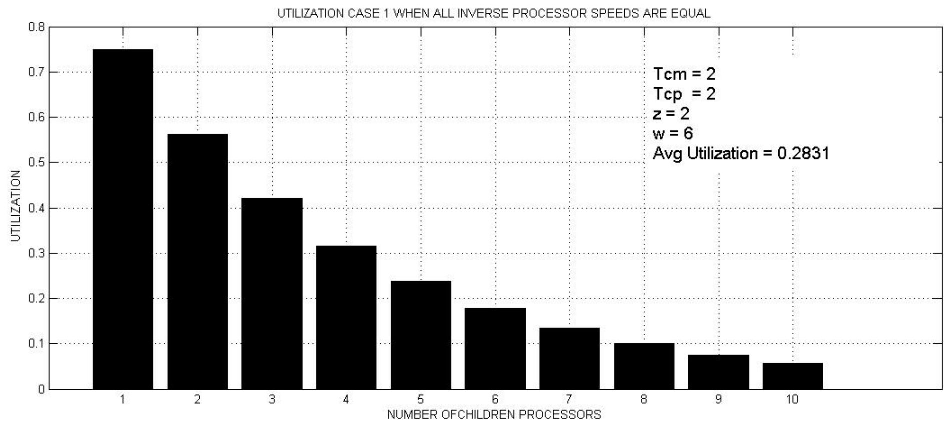


Figure 3.3. Utilization versus the number of children processors, all inverse processor speeds are equal

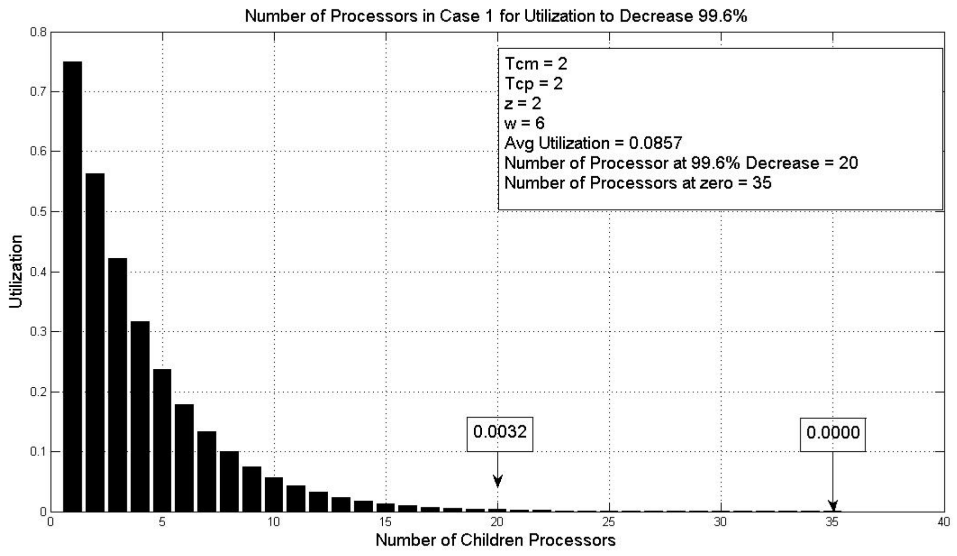


Figure 3.4. The number of children processors added to a network that will cause the utilization to decrease to 99.6% of its original value.

3.5.1 Speedup Case 1

Using equations (3.18), (3.32) and (3.46) listed below respectively

$$\alpha_2 = k_1\alpha_1$$
$$\alpha_3 = k_1k_2\alpha_1$$
$$\alpha_4 = k_1k_2k_3\alpha_1$$

And the fact that $k_1 = k_2 = k_N = k$

The loads of α_i for i = 2 to N, are found

$$\alpha_i = \prod_{m=1}^{i-1} k_m \alpha_{i-(i-1)} \tag{3.51}$$

$$\alpha_{i-(i-1)} = \alpha_1 \tag{3.52}$$

The i-(i-1) term assures that α_1 always appears on the right hand side of the equation (3.51). The normalization equation for case 1 where the root does no processing is

$$\alpha_1 + \alpha_2 + \alpha_3 + \ldots + \alpha_N = 1 \tag{3.53}$$

rewriting the normalized equation (3.53)

$$\alpha_1 + \sum_{i=2}^{N} \alpha_i = 1 \qquad \text{for } i = 1, 2, \ldots N \tag{3.54}$$

Substituting equations (3.51) and (3.52) in equation (3.54) for α_i

$$\alpha_1 + \sum_{i=2}^{N} \prod_{m=1}^{i-1} k_m \alpha_1 = 1 \qquad \text{for } i = 1, 2, ..., N \tag{3.55}$$

$$\alpha_1 \left[1 + \sum_{i=2}^{N} \prod_{m=1}^{i-1} k_m \right] = 1 \tag{3.56}$$

The value of α_1 for this network is

$$\alpha_1 = \frac{1}{\left[1 + \sum_{i=2}^{N} \prod_{m=1}^{i} k_m\right]} \qquad \text{for } i = 1, 2, ..., N \tag{3.57}$$

Equation (3.1) with $z_i = z$, $w_i = w$, and $\alpha_1 = 1$ becomes;

$$T_{f1} = zT_{cm} + wT_{cp} \tag{3.58}$$

Equation (3.58) is the computation time on a single processor.

Extending equation (3.1) to represent N processors yields

$$T_{fN} = \alpha_1 z T_{cm} + \alpha_1 w T_{cp} \tag{3.59}$$

Equation (3.59) is the computation time on the entire tree with N processors.

Speedup is the ratio of T_{f1} / $T_{fN.}$

$$\frac{T_{f1}}{T_N} = \frac{zT_{cm} + wT_{cp}}{\alpha_1(zT_{cm} + wT_{cp})} \tag{3.60}$$

Dividing the numerator and the denominator by wT_{cp} and using $\sigma = \frac{zT_{cm}}{w_{T\,cp}}$

Equation (3.60) becomes

$$\frac{T_{f1}}{T_N} = \frac{\sigma + 1}{\alpha_1(\sigma + 1)} \tag{3.61}$$

$$speedup = \frac{T_{f1}}{T_N} = \frac{1}{\alpha_1} \tag{3.62}$$

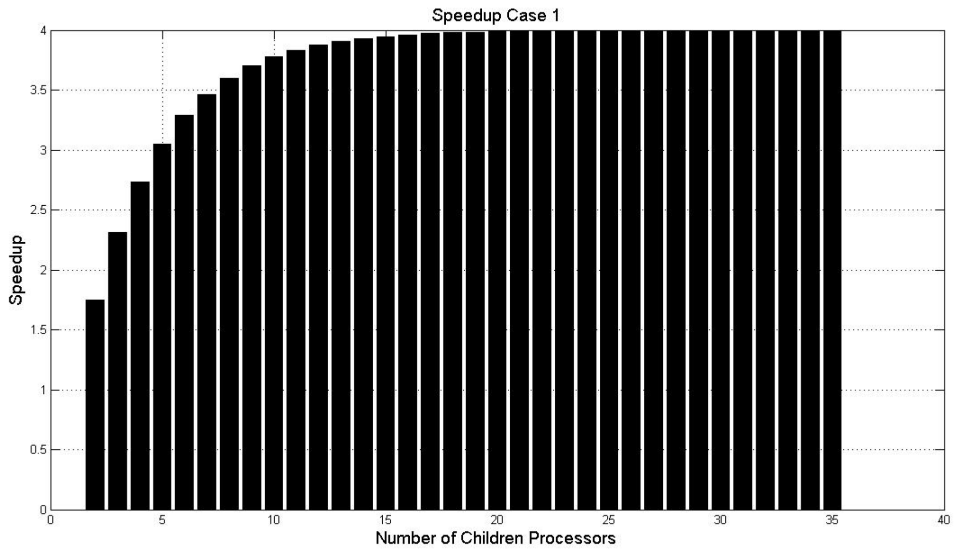


Figure 3.5. Speedup for case 1 as the number of processors are increased the speedup approaches a constant here a speedup of 4.0 w =6, z = 2, $T_{cm} = T_{cp} = 2$

Speedup saturates at 4.0, 99.6% of this is 3.984 this occurs at processor 12

3.6 CASE 2

SEQUENTIAL DISTRIBUTION
STAGGERED START
ROOT DOES PROCESSING

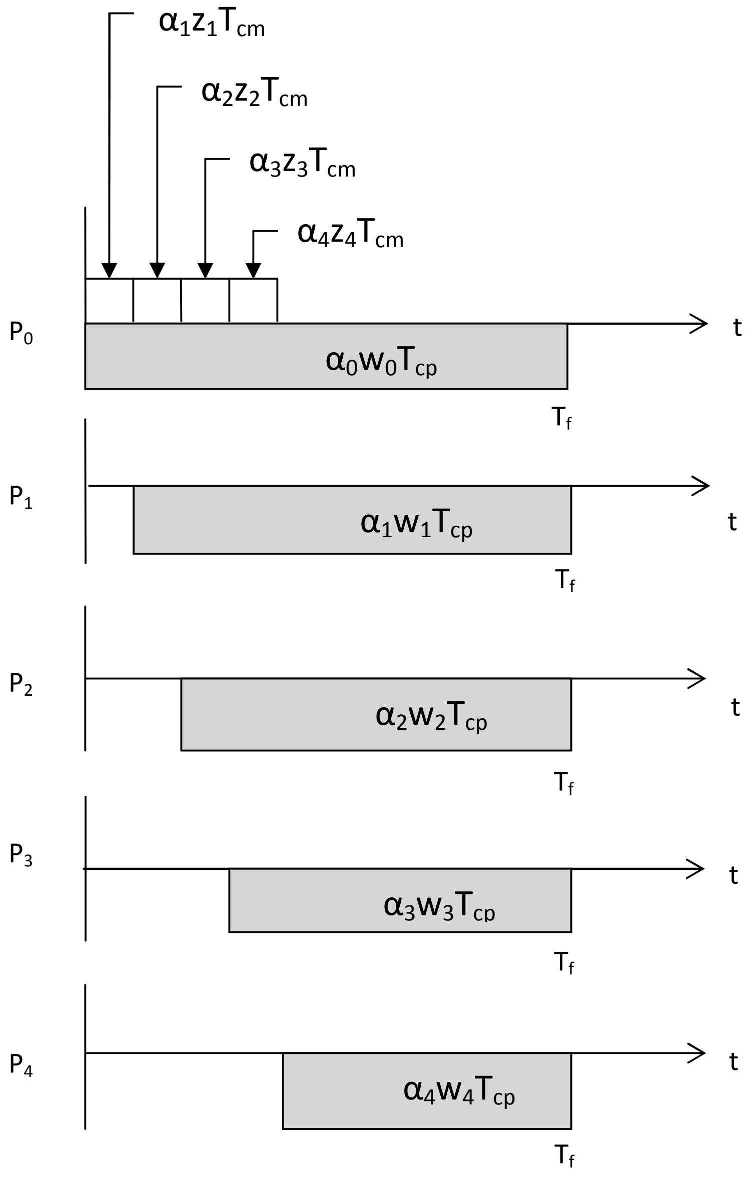


Figure 3.6. Timing diagram a of single level tree with sequential distribution, staggered start, and a root that does processing

In Figure 3.6 the loads are sequentially distributed by the parent processor. The parent processor has a front end processor. The children processors have no front end processors.

Computation is at 100% in processors P_0 therefore $U_0 = 1$

$$T_{f0} = \alpha_0 w_0 T_{cp} \tag{3.63}$$

$$U_1 = \frac{T_{F0} - \alpha_1 w_1 T_{cp}}{T_{f0}} \tag{3.64}$$

$$U_1 = 1 - \frac{\alpha_1 z_1 T_{cm}}{\alpha_0 w_0 T_{cp}} \tag{3.65}$$

From Figure 3.6

$$\alpha_1 z_1 T_{cm} + \alpha_1 w_1 T_{cp} = \alpha_0 w_0 T_{cp} \tag{3.66}$$

$$\alpha_1 (z_1 T_{cm} + w_1 T_{cp}) = \alpha_0 w_0 T_{cp} \tag{3.67}$$

$$\alpha_1 = \frac{w_0 T_{cp}}{(z_1 T_{cm} + w_1 T_{cp})} \alpha_0 \tag{3.68}$$

$$k_0 = \frac{w_0 T_{cp}}{(z_1 T_{cm} + w_1 T_{cp})} \tag{3.69}$$

$$\alpha_1 = k_0 \alpha_0 \tag{3.70}$$

Removing the load α_i

For $z_i = z$

$$U_1 = 1 - \frac{\alpha_1 z T_{cm}}{\alpha_0 w_0 T_{cp}} \tag{3.71}$$

$$U_1 = 1 - \left(\frac{\alpha_1}{\alpha_0}\right) \frac{z T_{cm}}{w_0 T_{cp}} \tag{3.72}$$

Substituting equation (3.70) into equation (3.72) gives

$$U_1 = 1 - k_0\left(\frac{zT_{cm}}{w_0T_{cp}}\right) \tag{3.73}$$

Equation (3.73) is the utilization of processor P_1

From Figure 3.6

$$T_{f2} = \alpha_0 w_0 T_{cp} \tag{3.74}$$

$$U_2 = \frac{T_{f2} - (\alpha_1 z_1 T_{cm} + \alpha_2 z_2 T_{cm})}{T_{f2}} \tag{3.75}$$

$$U_2 = 1 - \frac{\alpha_1 z_1 T_{cm} + \alpha_2 z_2 T_{cm}}{\alpha_0 w_0 T_{cp}} \tag{3.76}$$

From Figure 3.6

$$\alpha_2 z_2 T_{cm} + \alpha_2 w_2 T_{cp} = \alpha_1 w_2 T_{cp} \tag{3.77}$$

$$\alpha_2\left(z_2 T_{cm} + w_2 T_{cp}\right) = \alpha_1 w_2 T_{cp} \tag{3.78}$$

$$\alpha_2 = \frac{w_1 T_{cp}}{z_2 T_{cm} + w_2 T_{cp}}\alpha_1 \tag{3.79}$$

$$k_1 = \frac{w_1 T_{cp}}{z_2 T_{cm} + w_2 T_{cp}} \tag{3.80}$$

$$\alpha_2 = k_1 \alpha_1 \tag{3.81}$$

Substituting equation (3.70) into equation (3.81)

$$\alpha_2 = k_0 k_1 \alpha_0 \tag{3.82}$$

For $z_i = z$

$$U_2 = 1 - \frac{\alpha_1 z T_{cm} + \alpha_2 z T_{cm}}{\alpha_0 w_0 T_{cp}} \tag{3.83}$$

$$U_2 = 1 - \left(\frac{\alpha_1}{\alpha_0} + \frac{\alpha_2}{\alpha_0}\right)\frac{zTcm}{w_0 T_{cp}} \tag{3.84}$$

Substituting equations (3.70) and (3.84) into equation (3.84) gives

$$U_2 = 1 - (k_0 + k_0 k_1)\left(\frac{zTcm}{w_0 T_{cp}}\right) \tag{3.85}$$

Equation (3.85) is the utilization of processor P_2

From Figure 3.6

$$T_{f3} = \alpha_0 w_0 T_{cp} \tag{3.86}$$

$$U_3 = \frac{T_{f3} - (\alpha_1 z_1 T_{cm} + \alpha_2 z_2 T_{cm} + \alpha_3 z_3 T_{cm})}{T_{f3}} \tag{3.87}$$

$$U_3 = \frac{\alpha_0 w_0 T_{cp} - (\alpha_1 z_1 T_{cm} + \alpha_2 z_2 T_{cm} + \alpha_3 z_3 T_{cm})}{\alpha_0 w_0 T_{cp}} \tag{3.88}$$

For $z_i = z$

$$U_3 = 1 - \frac{\alpha_1 z T_{cm} + \alpha_2 z T_{cm} + \alpha_3 z T_{cm}}{\alpha_0 w_0 T_{cp}} \tag{3.89}$$

$$U_3 = 1 - \left(\frac{\alpha_1}{\alpha_0} + \frac{\alpha_2}{\alpha_0} + \frac{\alpha_3}{\alpha_0}\right)\left(\frac{zT_{cm}}{w_0 T_{cp}}\right) \tag{3.90}$$

From Figure 2.6

$$\alpha_3 z_3 T_{cm} + \alpha_3 w_3 T_{cp} = \alpha_2 w_2 T_{cp} \tag{3.91}$$

$$\alpha_3 (z_3 T_{cm} + w_3 T_{cp}) = \alpha_2 w_2 T_{cp} \tag{3.92}$$

$$\alpha_3 = \frac{w_2 T_{cp}}{z_3 T_{cm} + w_3 T_{cp}} \alpha_2 \tag{3.93}$$

$$k_2 = \frac{w_2 T_{cp}}{z_3 T_{cm} + w_3 T_{cp}} \alpha_2 \tag{3.94}$$

$$\alpha_3 = k_2\alpha_2 \tag{3.95}$$

Substituting equation (3.82) into equation (3.95)

$$\alpha_3 = k_0k_1k_2\alpha_0 \tag{3.96}$$

Substituting equations (3.70), (3.82) and (3.96) into equation (3.90)

$$U_3 = 1 - (k_0 + k_0k_1 + k_0k_1k_2)\left(\frac{zT_{cm}}{w_0T_{cp}}\right) \tag{3.97}$$

Equation (3.97) is the utilization at processor P_3

From Figure 3.6

$$T_{f4} = \alpha_0 w_0 T_{cp} \tag{3.98}$$

$$U_4 = \frac{T_{f4} - (\alpha_1 z_1 T_{cm} + \alpha_2 z_2 T_{cm} + \alpha_3 z_3 T_{cm} + \alpha_4 z_4 T_{cm})}{T_{f4}} \tag{3.99}$$

$$U_4 = 1 - \frac{(\alpha_1 z_1 T_{cm} + \alpha_2 z_2 T_{cm} + \alpha_3 z_3 T_{cm} + \alpha_4 z_4 T_{cm})}{\alpha_0 w_0 T_{cp}} \tag{3.100}$$

For $z_i = z$

$$U_4 = 1 - \frac{(\alpha_1 z T_{cm} + \alpha_2 z T_{cm} + \alpha_3 z T_{cm} + \alpha_4 z T_{cm})}{\alpha_0 w_0 T_{cp}} \tag{3.101}$$

$$U_4 = 1 - \left(\frac{\alpha_1}{\alpha_0} + \frac{\alpha_2}{\alpha_0} + \frac{\alpha_3}{\alpha_0} + \frac{\alpha_4}{\alpha_0}\right)\left(\frac{zT_{cm}}{w_0T_{cp}}\right) \tag{3.102}$$

From Figure 3.6

$$\alpha_4 z_4 T_{cm} + \alpha_4 w_4 T_{cp} = \alpha_3 w_3 T_{cp} \tag{3.103}$$

$$\alpha_4(z_4 T_{cm} + w_4 T_{cp}) = \alpha_3 w_3 T_{cp} \tag{3.104}$$

$$\alpha_4 = \frac{w_3 T_{cp}}{z_4 T_{cm} + w_4 T_{cp}} \alpha_3 \tag{3.105}$$

$$k_3 = \frac{w_3 T_{cp}}{z_4 T_{cm} + w_4 T_{cp}} \tag{3.106}$$

$$\alpha_4 = k_3 \alpha_3 \tag{3.107}$$

Substituting equation (3.96) into equation (3.107)

$$\alpha_4 = k_0 k_1 k_2 k_3 \alpha_0 \tag{3.108}$$

Substituting equations (3.70), (3.82), (3.96) and (3.108) into

equation (3.102) yields

$$U_4 = 1 - (k_0 + k_0 k_1 + k_0 k_1 k_2 + k_0 k_1 k_2 k_3) \left(\frac{z T_{cm}}{w_0 T_{cp}} \right) \tag{3.109}$$

Equation (3.109) is the utilization of processor P_4

The general from of the utilization is

$$U_i = 1 - \left[\sum_{j=1}^{i} \prod_{m=0}^{i-1} k_m \right] \frac{z T_{cm}}{w_0 T_{cp}} \qquad \text{for } i = 1, 2, ..., N \tag{3.110}$$

From equation (3.50) the average utilization is

$$AvgU = \frac{1}{N} \sum_{i=1}^{N} U_i \qquad \text{for } i = 1, 2, ..., N \tag{3.111}$$

The number of children processor is represented by N

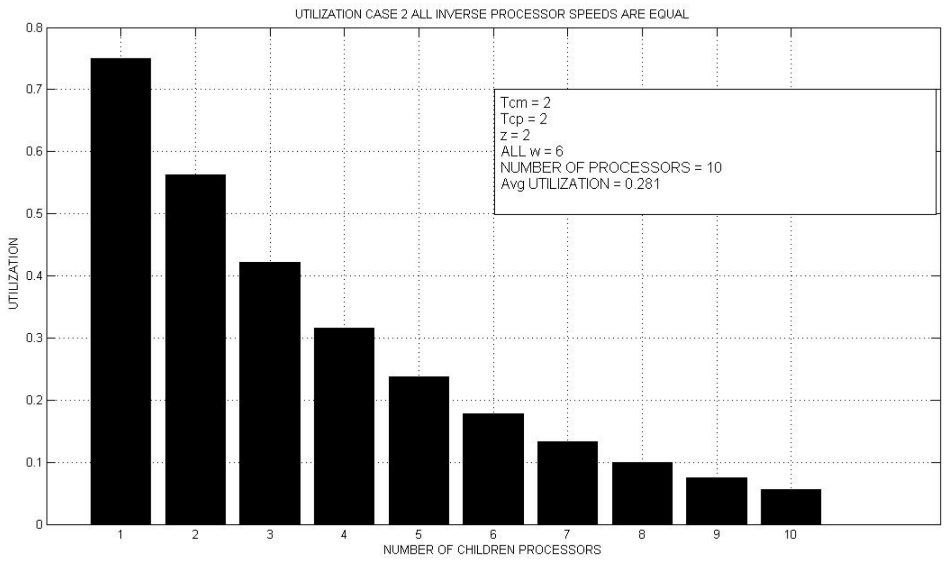


Figure 3.7. Utilization versus the number of children processors, all inverse processors are equal.

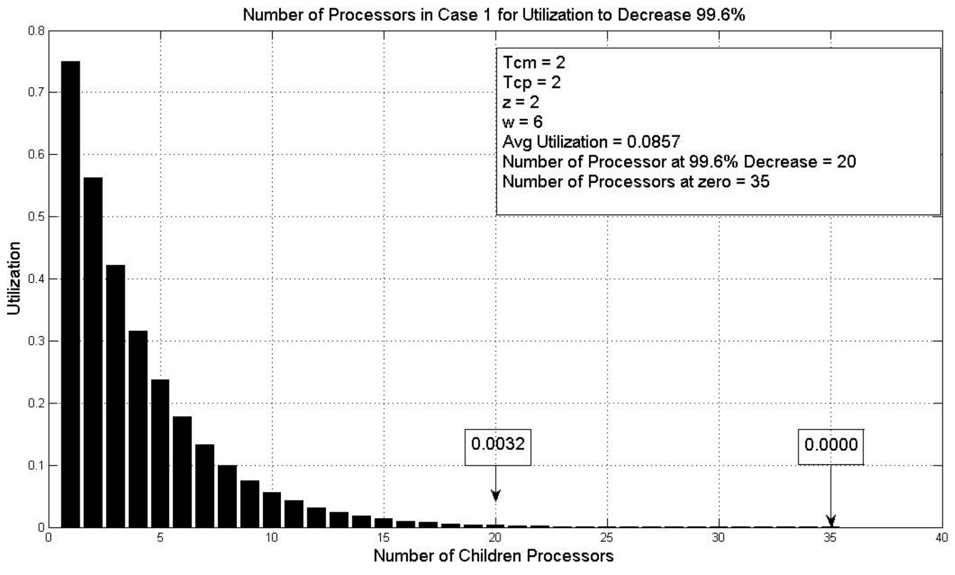


Figure 3.8. The number of children processors added to a network that will cause the utilization to decrease to 99.6% of its original value, here it is 20.

3.6.1 Speedup Case 2

Using equations (3.70), (3.82), (3.96) and (3.108), listed below

$$\begin{aligned} \alpha_1 &= k_0\alpha_0 \\ \alpha_2 &= k_0k_1\alpha_0 \\ \alpha_3 &= k_0k_1k_2\alpha_0 \\ \alpha_4 &= k_0k_1k_2k_3\alpha_0 \end{aligned}$$

A general equation for α_i is developed

Note $k_i = k$.

The loads of α_i for i = 1 to N, are found

$$\alpha_i = \prod_{m=0}^{i-1} k_m \alpha_{(i-(i-1)-1)} \qquad \text{for } i = 1, 2, ..., N \tag{3.112}$$

$$\alpha_{i-(i-1)-1)} = \alpha_0 \tag{3.113}$$

The i-(i-1)-1 term assures that α_0 always appears on the right hand side of the equation (3.112) .

The normalization equation for case 2 that has a root that does processing is given below.

$$\alpha_0 + \alpha_1 + \alpha_2 + \alpha_3 + ... + \alpha_N = 1 \tag{3.114}$$

rewriting the normalized equation (3.114)

$$\alpha_0 + \alpha_1 + \sum_{i=2}^{N} \alpha_i = 1 \tag{3.115}$$

From equation (3.70)

$$\alpha_0 = \frac{\alpha_1}{k_0} \tag{3.116}$$

Substituting (3.116) into equation (3.115)

$$\frac{\alpha_1}{k_0} + \alpha_1 + \sum_{i=2}^{N} \alpha_i = 1 \tag{3.117}$$

Substituting equations (3.12) and (3.13) into equation (3.117)

$$\frac{\alpha_1}{k_0} + \alpha_1 + \sum_{i=2}^{N} \prod_{m=0}^{i-1} k_m \alpha_0 = 1 \tag{3.118}$$

Substituting again for α_0

$$\frac{\alpha_1}{k_0} + \alpha_1 + \sum_{i=2}^{N} \prod_{m=0}^{i-1} k_m \left(\frac{1}{k_0}\right) = 1 \tag{3.119}$$

$$\alpha_1 \left[\frac{1}{k_0} + 1 + \frac{1}{k_0} \sum_{i=2}^{N} \prod_{m=0}^{i-1} k_m\right] = 1 \tag{3.120}$$

The value of α_1 for this network is

$$\alpha_1 = \frac{1}{\left[\frac{1}{k_0} + 1 + \frac{1}{k_0} \sum_{i=2}^{N} \prod_{m=0}^{i-1} k_m\right]} \tag{3.121}$$

Equation (3.63) with $w_i = w$ and $\alpha_0 = 1$ becomes

$$T_{f0} = wT_{cp} \tag{3.122}$$

Equation (3.122) is the computation time on a single processor.

$$T_{f0} = \alpha_0 w_0 T_{cp} = \alpha_1 w_1 T_{cp} + \alpha_1 z_1 T_{cp} \tag{3.123}$$

Extending equation (3.123) to represent N processors yields

For $w_i = w$ and $z_i = z$

$$T_{fN} = \alpha_1 z T_{cm} + \alpha_1 w T_{cp} \tag{3.124}$$

Equation (3.124) is the computation time on the entire tree with N processors.

Speedup is the ratio of $T_{f0} / T_{fN.}$

$$\frac{T_{f0}}{T_N} = \frac{wT_{cp}}{\alpha_1(zT_{cm} + wT_{cp})} \tag{3.125}$$

Dividing by wT_{cp} and using $\sigma = \frac{zT_{cm}}{w_{Tcp}}$

Equation (3.125) becomes

$$\frac{T_{f0}}{T_N} = \frac{1}{\alpha_1(\sigma + 1)} \tag{3.126}$$

$$speedup = \frac{T_{f0}}{T_N} = \frac{1}{\alpha_1(\sigma + 1)} \tag{3.127}$$

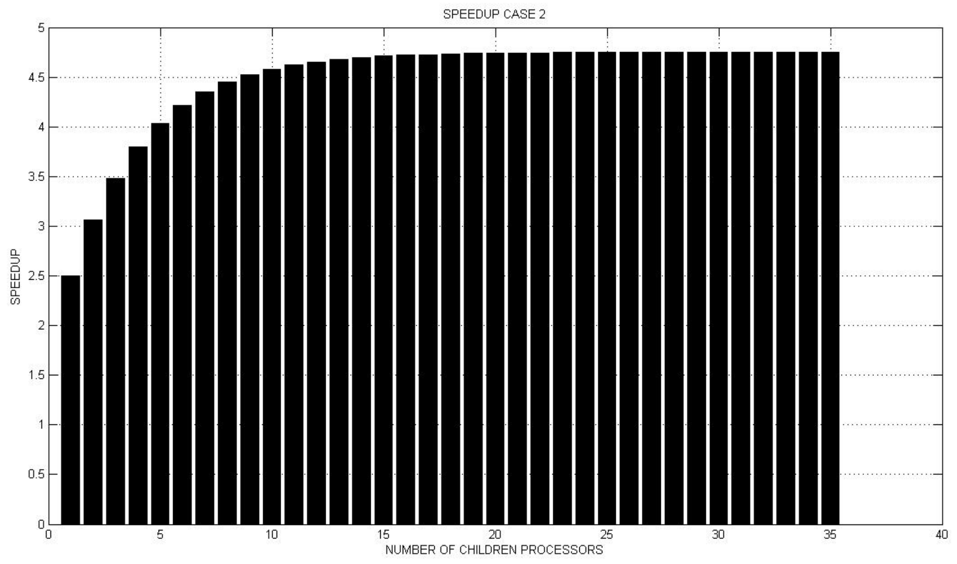


Figure 3.9. Speedup for case 2 as the number of processors is increased the speedup approaches a constant here a speedup of 4.7500.

3.7 CASE 3

SEQUENTIAL DISTRIBUTION
SIMULTANEOUS START
ROOT DOES PROCESSING

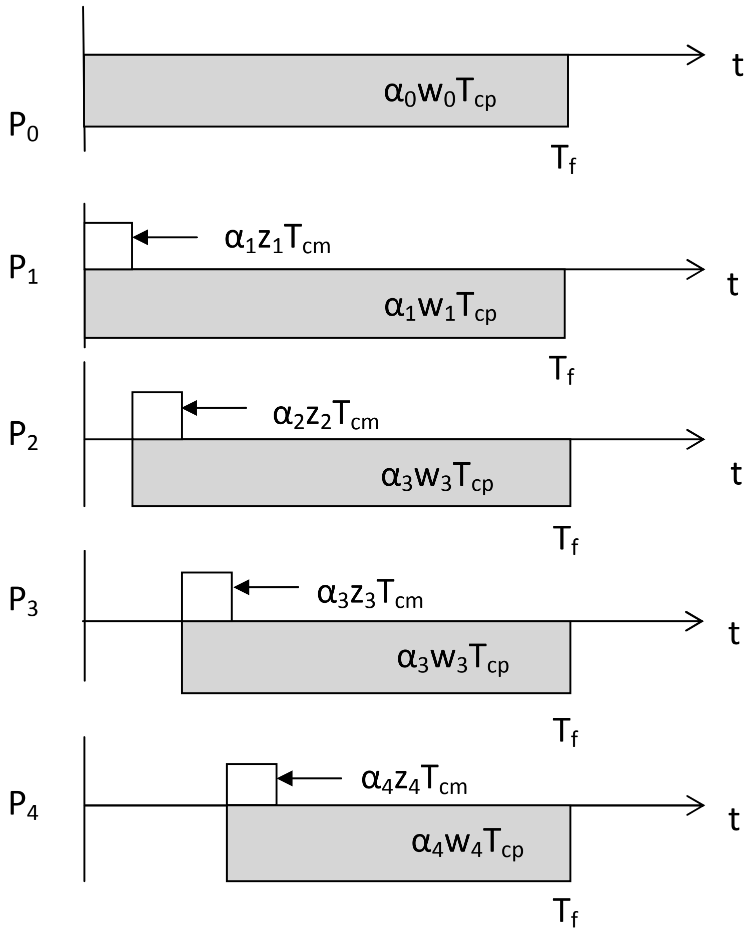


Figure 3.10. Sequential distribution, simultaneous start, and a root that does processing

The four boxes in figure 2.10, represented by $\alpha_1 z_1 T_{cm,}$ $\alpha_2 z_2 T_{cm,}$ $\alpha_3 z_3 T_{cm,}$ and $\alpha_4 z_4 T_{cm}$ could just as well been placed side by side on the top portion of

P_0's time axis. The diagrams are equivalent. These loads are distributed sequentially from the parent processor to the children processors.

The parent processor and the children processors have front end processing.

Computation a 100% processors P_0, and P_1 therefore $U_0 = 1$ and $U_1 = 1$.

$$T_{f2} = \alpha_1 z_1 T_{cm} + \alpha_2 w_2 T_{cp} \tag{3.128}$$

$$U_2 = \frac{\alpha_2 w_2 T_{cp}}{\alpha_1 z_1 T_{cp} + \alpha_2 w_2 T_{cp}} \tag{3.129}$$

$$U_2 = \frac{\frac{\alpha_2 w_2 T_{cp}}{\alpha_2 w_2 T_{cp}}}{\frac{\alpha_1 z_1 T_{cp}}{\alpha_2 w_2 T_{cp}} + \frac{\alpha_2 w_2 T_{cp}}{\alpha_2 w_2 T_{cp}}} \tag{3.130}$$

$$U_2 = \frac{1}{1 + \left(\frac{\alpha_1}{\alpha_2}\right) \frac{z_1 T_{cp}}{w_2 T_{cp}}} \tag{3.131}$$

From Figure 3.10

$$\alpha_1 z_1 T_{Cm} + \alpha_2 w_2 T_{cp} = \alpha_1 w_1 T_{cp} \tag{3.132}$$

$$\alpha_1 z_1 T_{Cm} - \alpha_1 w_1 T_{cp} = -\alpha_2 w_2 T_{cp} \tag{3.133}$$

$$\alpha_2 w_2 T_{cp} = \alpha_1 w_1 T_{cp} - \alpha_1 z_1 T_{Cm} \tag{3.134}$$

$$\alpha_2 w_2 T_{cp} = \alpha_1 (w_1 T_{cp} - z_1 T_{Cm}) \tag{3.135}$$

$$\alpha_2 = \frac{(w_1 T_{cp} - z_1 T_{Cm})}{w_2 T_{cp}} \alpha_1 \tag{3.136}$$

$$k_1 = \frac{(w_1 T_{cp} - z_1 T_{Cm})}{w_2 T_{cp}} \tag{3.137}$$

$$\alpha_2 = k_1\alpha_1 \tag{3.138}$$

Substituting equation (3.138) into equation (3.131)

$$U_2 = \frac{1}{1 + \left(\frac{1}{k_1}\right)\frac{z_1T_{cp}}{w_2T_{cp}}} \tag{3.139}$$

Equation (3.1390 is the utilization of processor P_2

From Figure 3.10

$$T_{f3} = \alpha_1 z_1 T_{cm} + \alpha_2 z_2 T_{cm} + \alpha_3 w_3 T_{cp} \tag{3.140}$$

$$U_3 = \frac{\alpha_3 w_3 T_{cp}}{\alpha_1 z_1 T_{cm} + \alpha_2 z_2 T_{cm} + \alpha_3 w_3 T_{cp}} \tag{3.141}$$

$$U_3 = \frac{\frac{\alpha_3 w_3 T_{cp}}{\alpha_3 w_3 T_{cp}}}{\frac{\alpha_1 z_1 T_{cm}}{\alpha_3 w_3 T_{cp}} + \frac{\alpha_2 z_2 T_{cm}}{\alpha_3 w_3 T_{cp}} + \frac{\alpha_3 w_3 T_{cp}}{\alpha_3 w_3 T_{cp}}} \tag{3.142}$$

$$U_3 = \frac{1}{1 + \frac{\alpha_1 z_1 T_{cm}}{\alpha_3 w_3 T_{cp}} + \frac{\alpha_2 z_2 T_{cm}}{\alpha_3 w_3 T_{cp}}} \tag{3.143}$$

$$U_3 = \frac{1}{1 + \left(\frac{\alpha_1}{\alpha_3}\right)\frac{z_1 T_{cm}}{w_3 T_{cp}} + \left(\frac{\alpha_2}{\alpha_3}\right)\frac{z_2 T_{cm}}{w_3 T_{cp}}} \tag{3.144}$$

From Figure 3.10

$$\alpha_2 z_2 T_{cm} + \alpha_3 w_3 T_{cp} = \alpha_2 w_2 T_{cp} \tag{3.145}$$

$$\alpha_2 z_2 T_{cm} - \alpha_2 w_2 T_{cp} = -\alpha_3 w_3 T_{cp} \tag{3.146}$$

$$\alpha_3 w_3 T_{cp} = \alpha_2 w_2 T_{cp} - \alpha_2 z_2 T_{cm} \tag{3.147}$$

$$\alpha_3 w_3 T_{cp} = \alpha_2 (w_2 T_{cp} - z_2 T_{cm}) \tag{3.148}$$

$$\alpha_3 = \frac{(w_2 T_{cp} - z_2 T_{cm})}{w_3 T_{cp}}\alpha_2 \tag{3.149}$$

$$k_2 = \frac{w_2T_{cp} - z_2T_{cm}}{w_3T_{cp}} \tag{3.150}$$

$$\alpha_3 = k_2\alpha_2 \tag{3.151}$$

Substituting equation (3.138) into equation (3.151) gives

$$\alpha_3 = k_1k_2\alpha_1 \tag{3.152}$$

Substituting equations (3.151) and (3.152) into equation (3.144) yields

$$U_3 = \frac{1}{1 + \left(\frac{1}{k_1k_2}\right)\frac{z_1T_{cm}}{w_3T_{cp}} + \left(\frac{1}{k_2}\right)\frac{z_2T_{cm}}{w_3T_{cp}}} \tag{3.153}$$

Equation (3.153) is the utilization at processor P$_3$

From Figure (3.10)

$$T_{f4} = \alpha_1z_1T_{cm} + \alpha_2z_2T_{cm} + \alpha_3z_3T_{cm} + \alpha_4w_4T_{cp} \tag{3.154}$$

$$U_3 = \frac{\frac{\alpha_4w_4T_{cp}}{\alpha_4w_4T_{cp}}}{\frac{\alpha_1z_1T_{cm}}{\alpha_4w_4T_{cp}} + \frac{\alpha_2z_2T_{cm}}{\alpha_4w_4T_{cp}} + \frac{\alpha_3z_3T_{cm}}{\alpha_4w_4T_{cp}} + \frac{\alpha_4w_4T_{cp}}{\alpha_4w_4T_{cp}}} \tag{3.155}$$

$$\frac{1}{1 + \left(\frac{\alpha_1}{\alpha_4}\right)\frac{z_1T_{cm}}{w_4T_{cp}} + \left(\frac{\alpha_2}{\alpha_4}\right)\frac{z_2T_{cm}}{w_4T_{cp}} + \left(\frac{\alpha_3}{\alpha_4}\right)\frac{z_3T_{cm}}{w_4T_{cp}}} \tag{3.156}$$

From Figure 3.10

$$\alpha_3z_3T_{cm} + \alpha_4w_4T_{cp} = \alpha_3w_3T_{cp} \tag{3.157}$$

$$\alpha_3z_3T_{cm} - \alpha_3w_3T_{cp} = -\alpha_4w_4T_{cp} \tag{3.158}$$

$$\alpha_4w_4T_{cp} = \alpha_3w_3T_{cp} - \alpha_3z_3T_{cm} \tag{3.159}$$

$$\alpha_4w_4T_{cp} = \alpha_3(w_3T_{cp} - z_3T_{cm}) \tag{3.160}$$

$$\alpha_4 = \frac{(w_3 T_{cp} - z_3 T_{cm})}{w_4 T_{cp}} \alpha_3 \tag{3.161}$$

$$k_3 = \frac{w_3 T_{cp} - z_3 T_{cm}}{w_4 T_{cp}} \tag{3.162}$$

$$\alpha_4 = k_3 \alpha_3 \tag{3.163}$$

Substituting equation (3.151) into equation (3.163)

$$\alpha_4 = k_2 k_3 \alpha_2 \tag{3.164}$$

Substituting equation(3.152) into equation(3.163) gives

$$\alpha_4 = k_1 k_2 k_3 \alpha_1 \tag{3.165}$$

Substituting equations (3.163), (3.164) and 3.165) into equation (3.156)

yields

$$U_4 = \frac{1}{1 + \left(\frac{1}{k_1 k_2 k_3}\right) \frac{z_1 T_{cm}}{w_4 T_{cp}} + \left(\frac{1}{k_2 k_3}\right) \frac{z_2 T_{cm}}{w_4 T_{cp}} + \left(\frac{1}{k_3}\right) \frac{z_3 T_{cm}}{w_4 T_{cp}}} \tag{3.166}$$

Equation (3.166) is the utilization of processor P$_4$

The general form of the equation is

$$U_i = \frac{1}{1 + \left[\sum_{j=1}^{i-1} \prod_{m=j}^{i-1} \frac{1}{k_m}\right] \frac{w_i T_{cp}}{z T_{cm}}} \qquad \text{for } i = 2, 3, ...N \tag{3.167}$$

The general form for the average utilization is

$$AvgU = \frac{1}{N} \sum_{i=1}^{N} U_i \tag{3.168}$$

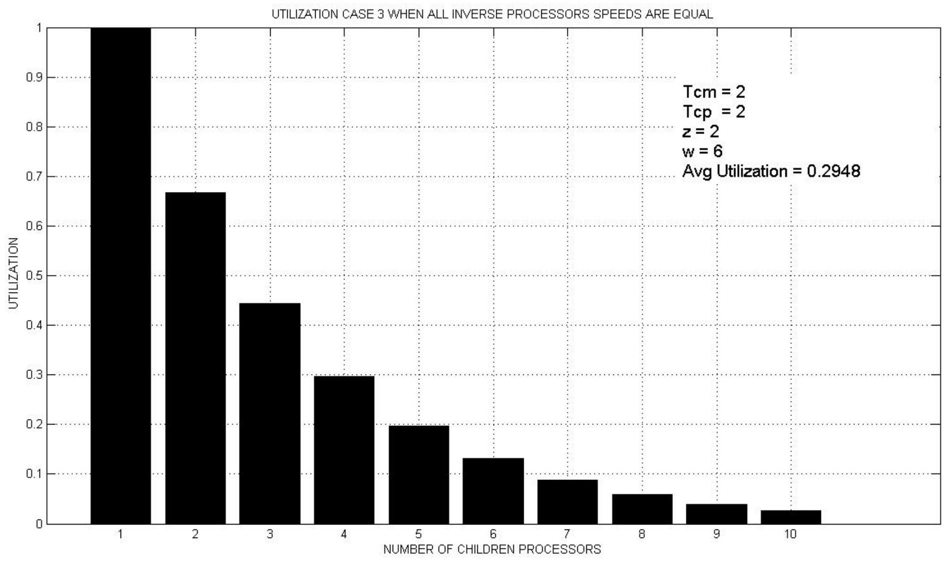


Figure 3.11. Utilization versus the number of children processor, all inverse processors speeds (w) are equal here w=6

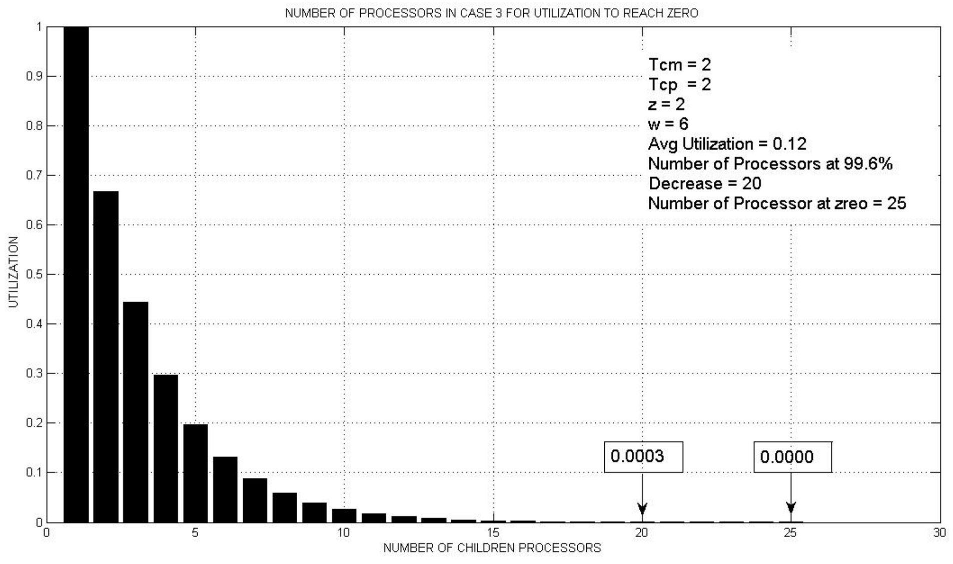


Figure 3.12. The number of children processors added to the tree that will force the utilization to zero, here 23 processors

3.7.1 SPEEDUP CASE 3

Using equations (3.138), (3.152) ,(3.165), and the conditions

$w_i = w$, $T_{cp} = T_{cm}$, $z_i = z$ and $k_i = k$

A general expression for α_i is formed.

$$\alpha_2 = k_1\alpha_1$$
$$\alpha_3 = k_1k_2\alpha_1$$
$$\alpha_4 = k_1k_2k_3\alpha_1$$

The loads of α_i for i = 2 to N, are found

$$\alpha_i = \prod_{m=0}^{i-1} k_m\alpha_{i-(i-1)} \qquad \text{for } i = 1, 2, ..., N \tag{3.169}$$

$$\alpha_{i-(i-1)} = \alpha_1 \tag{3.170}$$

The i-(i-1) term assures that α_1 always appears on the right hand side of the equation (3.169) .

The normalization equation for case 3 that has a root that does processing is given below.

$$\alpha_0 + \alpha_1 + \alpha_2 + \alpha_3 + ... + \alpha_N = 1 \tag{3.171}$$

From Figure 3.10

$$T_{f0} = \alpha_0 w_0 T_{cp} \tag{3.172}$$

$$\alpha_0 w_0 T_{cp} = \alpha_1 w_1 T_{cp} \tag{3.173}$$

$$\alpha_1 = \frac{w_0 T_{cp}}{w_1 T_{cp}} \alpha_0 \tag{3.174}$$

$$\alpha_1 = k_0 \alpha_0 \tag{3.175}$$

$$\alpha_0 = \frac{\alpha_1}{k_0} \tag{3.176}$$

$$k_0 = \frac{\alpha_1}{\alpha_0} = \frac{w_0}{w_1} \tag{3.177}$$

rewriting the normalized equation (3.171)

$$\alpha_0 + \alpha_1 + \sum_{i=2}^{N} \alpha_i = 1 \tag{3.178}$$

Substituting (3.176) into equation (3.178)

$$\frac{\alpha_1}{k_0} + \alpha_1 + \sum_{i=2}^{N} \alpha_i = 1 \tag{3.179}$$

Substituting equations (3.169) and (3.170) into equation (3.179)

$$\frac{\alpha_1}{k_0} + \alpha_1 + \sum_{i=2}^{N} \prod_{m=1}^{i-1} k_m \alpha_1 = 1 \tag{3.180}$$

$$\alpha_1 \left[\frac{1}{k_0} + 1 + \sum_{i=2}^{N} \prod_{m=1}^{i-1} k_m \right] = 1 \tag{3.181}$$

The value of α_1 for this network is

$$\alpha_1 = \frac{1}{\left[\frac{1}{k_0} + 1 + \sum_{i=2}^{N} \prod_{m=1}^{i-1} k_m\right]} \tag{3.182}$$

Equation (3.172) with $w_i = w$ and $\alpha_0 = 1$ becomes;

$$T_{f0} = wT_{cp} \tag{3.183}$$

Equation (3.183) is the computation time on a single processor.

From(3.173)

$$T_{f1} = \alpha_1 w_1 T_{cp} \tag{3.184}$$

Extending equation (3.184) to represent N processors yields

For $w_i = w$ and $z_i = z$

$$T_{fN} = \alpha_1 w T_{cp} \tag{3.185}$$

Equation (3.185) is the computation time on the entire tree with N processors.

Speedup is the ratio of $T_{f0} / T_{fN.}$

$$\frac{T_{f0}}{T_N} = \frac{wT_{cp}}{\alpha_1 w T_{cp}} = \frac{1}{\alpha_1} \tag{3.186}$$

$$speedup = \frac{T_{f0}}{T_N} = \frac{1}{\alpha_1} \tag{3.187}$$

$$speedup = \left[\frac{1}{k_0} + 1 + \sum_{i=2}^{N} \prod_{m=1}^{i-1} k_m\right] \tag{3.188}$$

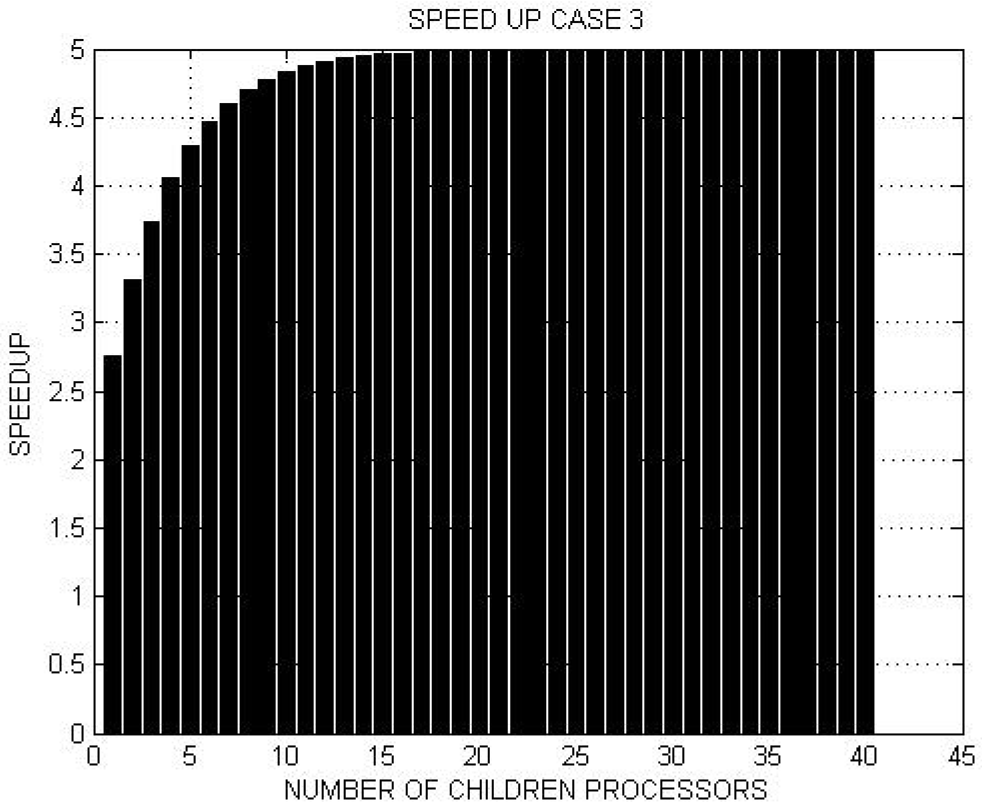


Figure 3.13. Speedup for case 3 as the number of processors is increased the speedup approaches a constant here a speedup of 5.0

3.8 CASE 4

SEQUENTIAL DISTRIBUTION
SIMULTANEOUS START
ROOT DOES NO PROCESSING

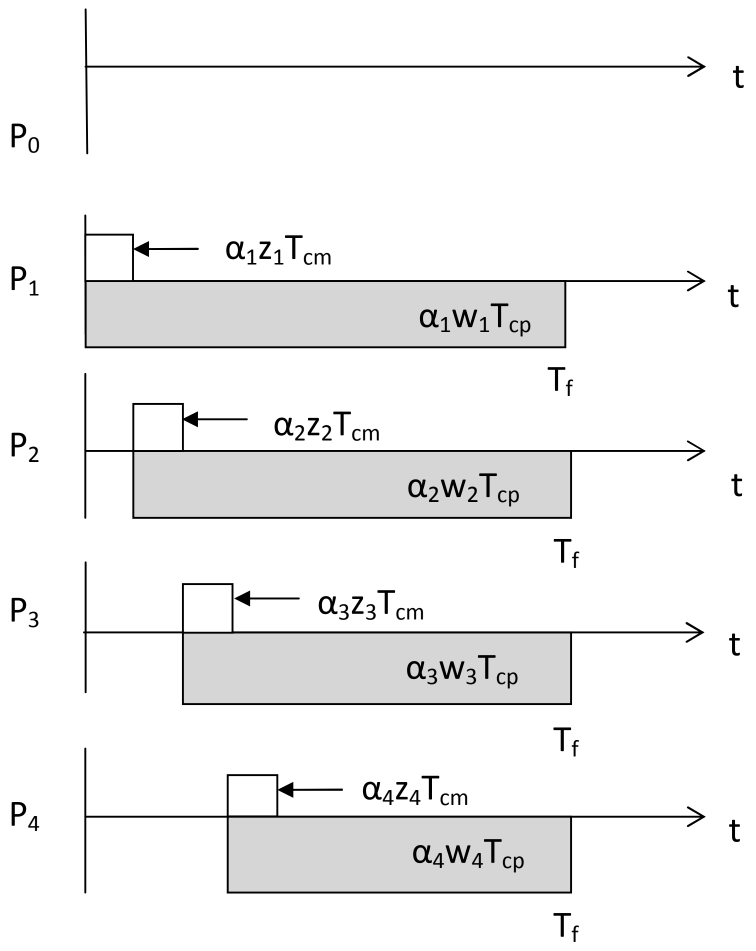


Figure 3.14. Sequential distribution, simultaneous start, and a root that does no processing.

Just as in Figure 3.13, the four boxes in figure 3.14, represented by $\alpha_1 z_1 T_{cm}$, $\alpha_2 z_2 T_{cm}$, $\alpha_3 z_3 T_{cm}$, and $\alpha_4 z_4 T_{cm}$ could just as well been placed side by side on the top portion of P_0's time axis. The diagrams are equivalent.

These loads are distributed sequentially from the parent processor to the children processors. The parent processor does no processing. The children processors have front end processors.

The equations for utilization in case 4 are the same as in case 3. Comparing figure 3.10 and figure 3.14, the only difference is that figure 3.10 has a root that does processing, denoted as $\alpha_0 w_0 T_{cp}$. This term does not enter into the formularization of the utilization equations of case 3. The derivations of the utilization equations for both case 3 and case 4 start at the first child processor P_1. From this point on the figure are identical, producing the same set of equations. The utilization equations are repeated here without their derivations. The original equations numbers have been updated. The graphs in Figure 3.11 and 3.12 would be identical in case 4. These graphs are not repeated.

$$U_2 = \frac{1}{1 + \left(\frac{1}{k_1}\right)\frac{z_1 T_{cp}}{w_2 T_{cp}}} \tag{3.189}$$

$$U_3 = \frac{1}{1 + \left(\frac{1}{k_1 k_2}\right)\frac{z_1 T_{cm}}{w_3 T_{cp}} + \left(\frac{1}{k_2}\right)\frac{z_2 T_{cm}}{w_3 T_{cp}}} \tag{3.190}$$

$$U_4 = \frac{1}{1 + \left(\frac{1}{k_1 k_2 k_3}\right)\frac{z_1 T_{cm}}{w_4 T_{cp}} + \left(\frac{1}{k_2 k_3}\right)\frac{z_2 T_{cm}}{w_4 T_{cp}} + \left(\frac{1}{k_3}\right)\frac{z_3 T_{cm}}{w_4 T_{cp}}} \tag{3.191}$$

$$U_i = \frac{1}{1 + \left[\sum_{j=1}^{i-1} \prod_{m=j}^{i-1} \frac{1}{k_m}\right] \frac{w_i T_{cp}}{z T_{cm}}} \qquad \text{for } i = 2, 3, ...N \quad (3.192)$$

$$AvgU = \frac{1}{N} \sum_{i=1}^{N} U_i \quad (3.193)$$

3.8.1 SPEEDUP CASE 4

Unlike utilization, the speedup in case 4 will be different from case 3.

From figure 3.14

$$T_{f1} = \alpha_1 w_1 T_{cp} \quad (3.194)$$

$$T_{f2} = \alpha_1 z_1 T_{cm} + \alpha_2 w_2 T_{cp} \quad (3.195)$$

As in case 3 using equations (3.138), (3.153), (3.165), listed below

$$\alpha_2 = k_1 \alpha_1$$
$$\alpha_3 = k_1 k_2 \alpha_1$$
$$\alpha_4 = k_1 k_2 k_3 \alpha_1$$

A general expression for α_i is formed, using the conditions $w_i = w$, $T_{cp} = T_{cm}$,

$z_i = z$ and $k_i = k$

$$\alpha_i = \prod_{m=0}^{i-1} k_m \alpha_{i-(i-1)} \qquad \text{for } i = 1, 2, ..., N \quad (3.196)$$

$$\alpha_{i-(i-1)} = \alpha_1 \quad (3.197)$$

Here the normalization equation is

$$\alpha_1 + \alpha_2 + \alpha_3 + ... + \alpha_N = 1$$

Rewriting the normalization equation

$$\alpha_1 + \sum_{i=2}^{N} \alpha_i = 1 \tag{3.198}$$

Substituting equations (3.196) and (3.197) into equation (3.178)

$$\alpha_1 + \sum_{i=2}^{N} \prod_{m=1}^{i-1} k_m \alpha_1 = 1 \tag{3.199}$$

$$\alpha_1 \left[1 + \sum_{i=2}^{N} \prod_{m=1}^{i-1} k_m \right] = 1 \tag{3.200}$$

The value of α_1 for this network is

$$\alpha_1 = \frac{1}{\left[1 + \sum_{i=2}^{N} \prod_{m=1}^{i-1} k_m \right]} \tag{3.201}$$

Equation (3.194) with $w_i = w$ and $\alpha_1 = 1$ becomes;

$$T_{f1} = wT_{cp} \tag{3.202}$$

Equation (3.184) is the computation time on a single processor

Extending equation (3.194) to represent N processors yields

For $w_i = w$

$$T_{fN} = \alpha_1 w T_{cp} \tag{3.203}$$

Equation (3.185) is the computation time on the entire tree with N processors.

Speedup is the ratio of T_{f1} / T_{fN}.

$$\frac{T_{f1}}{T_N} = \frac{wT_{cp}}{\alpha_1 w T_{cp}} = \frac{1}{\alpha_1} \tag{3.204}$$

$$speedup = \frac{T_{f1}}{T_N} = \frac{1}{\alpha_1} \tag{3.205}$$

$$speedup = \left[1 + \sum_{i=2}^{N} \prod_{m=1}^{i-1} k_m\right] \tag{3.206}$$

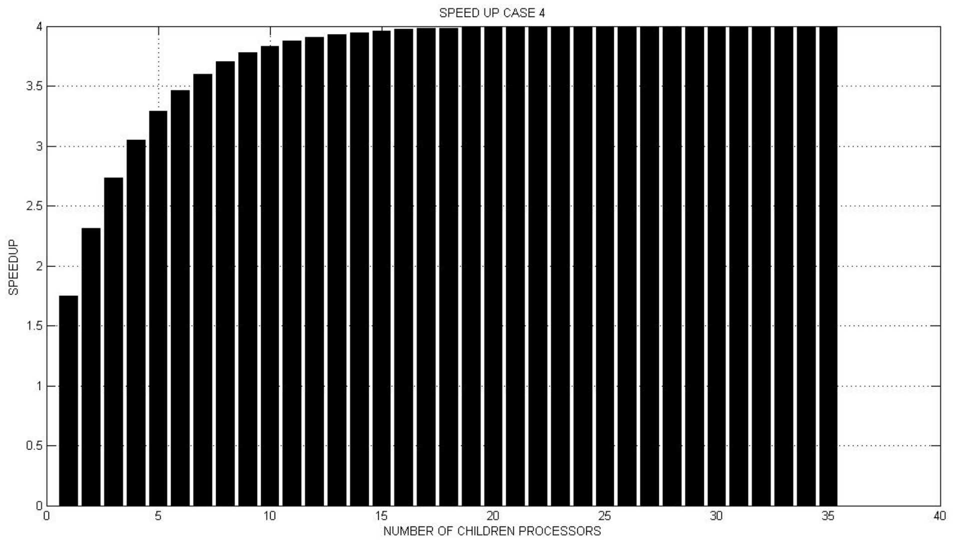


Figure 3.15. Speedup for case 4 as the number of processors is increased the speedup approaches a constant here a speedup of 4.0

3.9 ANALYSIS OF RESULTS/CONCLUSION OF CHAPTER 3

The utilization results in case 1 and case 2 are unexpected. Comparing Figure 3.3 and Figure 3.4 of case 1 with Figure 3.6 and Figure 3.7 of case 2, shows that the results are identical. The equations are derived from different sources using different methods. As can be seen in the timing diagrams of Figure 3.2 and Figure 3.6, in Figure 3.6 the parent processor has a front end processor. In Figure 3.2 the parent processor has no such processor. As a result there is an additional k term, k_0 in the equations derived for case 2. Comparing equations (3.6) of case1 and (3.73) for $z_i = z_1$, the two equations reduce to an equivalent expression.

For U_1 in case 1 with $z_i = z$

$$U_1 = \frac{1}{1 + \frac{zT_{cm}}{w_1 T_{cp}}} \tag{3.207}$$

$$U_1 = \frac{1}{\frac{w_1 T_{cp} + zT_{cm}}{w_1 T_{cp}}} \tag{3.208}$$

$$U_1 = \frac{w_1 T c_p}{w_1 T_{cp} + zT_{cm}} \tag{3.209}$$

Using equation (3.69) with the above condition yields

$$k_0 = \frac{w_0 T_{cp}}{(zT_{cm} + w_1 T_{cp})} \tag{3.210}$$

Substituting equation (3.210) into equation (3.210)

For U_1 in case 2

$$U_1 = 1 - k_0 \left(\frac{zT_{cm}}{w_0 T_{cp}}\right) \tag{3.211}$$

$$U_1 = 1 - \left(\frac{w_0 T_{cp}}{zT_{cm} + w_1 T_{cp}}\right)\left(\frac{zT_{cm}}{w_0 T_{cp}}\right) \tag{3.212}$$

$$U_1 = 1 - \left(\frac{zT_{cm}}{zT_{cm} + w_1 T_{cp}}\right) \tag{3.213}$$

$$U_1 = \left(\frac{zT_{cm} + w_1 T_{cp} - zT_{cm}}{zT_{cm} + w_1 T_{cp}}\right) \tag{3.214}$$

$$U_1 = \frac{w_1 T_{cp}}{zT_{cm} + w_1 T_{cp}} \tag{3.215}$$

Equations (3.209) and (3.215) are equal

The reason equation (3.211) reduces to an equivalent expression is that in the equations of case 2 the total time is given by $\alpha_0 w_o T_{cp.}$ The utilization for the children processors are written in terms of this total time. This produces equations that have a constant term (zT_{cm}/w_0T_{cp}) that multiplies each equation derived in case 2, as can be seen in equation (3.110). The denominator of this term, which represents the inverse processor speed of the parent processor's cancels like terms in the only term that the equations share that is different, k_0. This means that the parent processor having the ability to send and receive load at the same time does not improve utilization.

In equation (3.85)

$$U_2 = 1 - (k_0 + k_0 k_1)\left(\frac{zTcm}{w_0 T_{cp}}\right)$$

(3.216)

The term $(k_0 + k_0k_1) = k + k^2$ is a power series

$1/(1 - x) = 1 + x + x^2 + ... + x^N$ for $|x| < 1$

Equation (3.216) is represented by a power series times a constant which is then subtracted from 1. This produces the exponentially decreasing utilization curves. It has already been shown that equation (3.73) can be rewritten as (3.6).

From the data used to generate Figure 3.5, the speedup in case 1 saturates at 4.0. The same data shows that the first occurrence of 4.0 corresponds to 39 children processors. This indicates that the network could be expanded to 39 children processors and still benefit from the advantages of parallel processing. When viewing the graph in Figure 3.5, a value of 19 children processors on the x axis is a good visual estimation. Using the numerical data this value is found to be 99.6 % of the saturation value of 4.0. This percentage is used to find the number of children processors that the network can expand to in all other speedup calculations. Four decimal places are used in order to decide which value is actually the closest to 99.6%.

In case 2 the speedup saturates at 4.7500, 4.7500×0.996 = 4.731, comparing this value to the numerical data corresponds to 17 children processors. In case 3 the speedup saturates at 5.0 following the same process gives an extension of 10 children processors. In case 4 the speedup saturation is 4.0. This equates to 18 children processors.

There is no fix percentage for speedup used; values range from 95% and up. The initial speedup saturation values such as 4.0 are found through computer computations. Here a strict definition of saturation is defined as a value that does not increase regardless of how many additional children processors are added.

For a network of 10 children processor as shown in Figure 3.3, if the strict definition of saturation is used, it implies that the network could expand to 39 processors and still benefit from parallelism. However the utilization curve in Figure 3.4 shows that the utilization at 35 is zero. In this situation the strict definition is not valid; the estimation of 99.6 % corresponds to a realistic value in Figure3.4.

Utilization is a useful performance metric , and compliments the traditional performance metrics such as speedup and finish time (makespan)

3.10 RESEARCH GOALS

To calculate the utilization of a linear daisy chain and mesh configurations. Furthermore to investigate new and novel performance metrics that can be applied to divisible load scheduling.

Chapter 4

4. Signature Searching

4.1 Introduction

Signature searching or pattern recognition is a process by which the task of searching large amounts of data is accomplished by distributing the load among a number of processors and having each processor search its assigned load for a distinct marker, for example the occurrence of a certain voltage. In this chapter signature searching is applied to the sequentially distributed single level tree networks of chapter 3.

Four different types of networks are investigated, in order to find which is better suited for an optimal search of a divisible load. These are staggered start with and without a root that does processing, and simultaneous start with and without a root that does processing. The results are compared with utilization and speedup results from chapter 3. A third metric is developed using the finish time and the last communication delay of the network.

4.2 CASE 1

SEQUENTIAL DISTRIBUTED
STAGGERED START
ROOT DOES NO PROCESSING

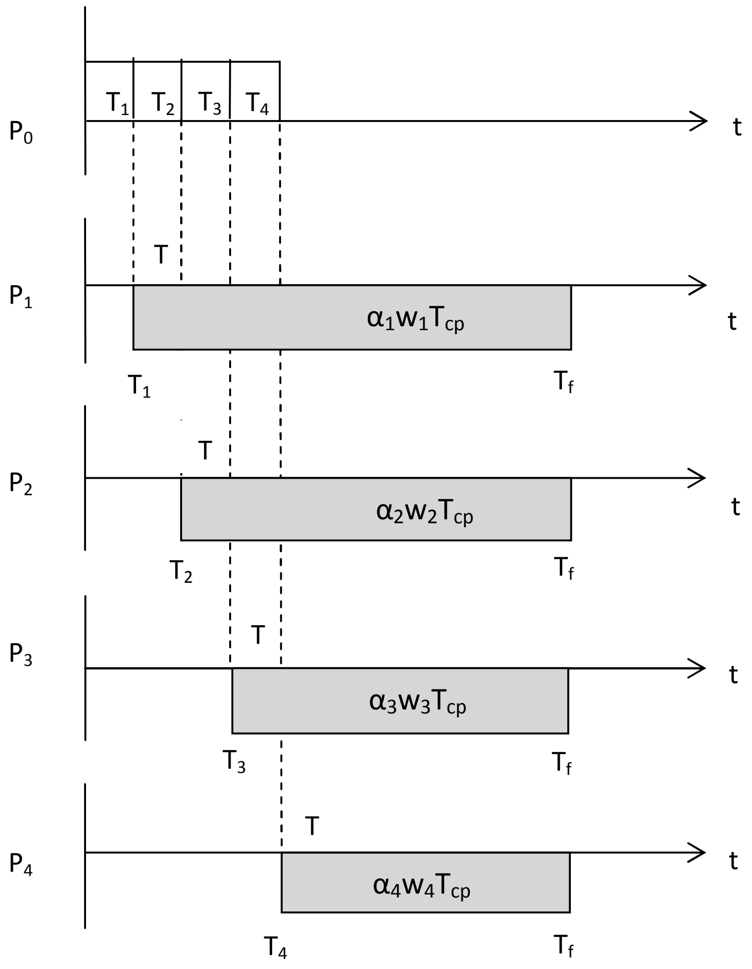


Figure 4.1. Timing diagram of a single level tree with sequential distribution, staggered start, and a root that does no processing.

In Figure 4.1, T_1 through T_4 represent the loads that are sequentially distributed, similar to Figure 3.2. Here T represents the amount of load searched by a particular processor within the time intervals determined by the communication delays, i.e. T_1 through T_4. Since this distribution has a staggered start, the loads are completely transmitted before the processors begin computations. So the load α_1 of T_1 is entirely transmitted over link z_1 in the time $\alpha_1 z_1 T_{cm}$ before processor P_1 starts processing at time T_1. This process continues until all values from T_1 to T_N are transmitted.

$$T_1 = (\alpha_1 z_1) T_{cm} \tag{4.1}$$

$$T_2 = (\alpha_1 z_1 + \alpha_2 z_2) T_{cm} \tag{4.2}$$

$$T_3 = (\alpha_1 z_1 + \alpha_2 z_2 + \alpha_3 z_3) T_{cm} \tag{4.3}$$

$$T_4 = (\alpha_1 z_1 + \alpha_2 z_2 + \alpha_3 z_3 + \alpha_4 z_4) T_{cm} \tag{4.4}$$

$$T_N = (\alpha_1 z_1 + \alpha_2 z_2 + \alpha_3 z_3 + \alpha_4 z_4 + ... + \alpha_N z_N) T_{cm} \tag{4.5}$$

Each processor P_1 through P_4 all of which are children processors finish processing at the same time, the finish time T_f.

$$T_f = \alpha_1 z_1 T_{cm} + \alpha_1 w_1 T_{cp} \tag{4.6}$$

The computation speed of the processor is given by $\alpha_N w_N T_{cp}$. Where α_N is the load transmitted to processor N, w_N is the inverse processing speed, and T_{cp} is a dimensionless number used to increase or decrease the computation speed. The variable T_{cm} performs a function similar to that of T_{cp}, but for the communication speed.

The percent of the load processors P_1 through P_4 searchers within an interval is defined as αT%, and given by the following equations.

$$T_1 \leq T \leq T_2 \quad \frac{T-T_1}{T_f-T_1}\alpha_1 \tag{4.7}$$

$$T_2 \leq T \leq T_3 \quad \frac{T-T_1}{T_f-T_1}\alpha_1 + \frac{T-T_2}{T_f-T_2}\alpha_2 \tag{4.8}$$

$$T_3 \leq T \leq T_4 \quad \frac{T-T_1}{T_f-T_1}\alpha_1 + \frac{T-T_2}{T_f-T_2}\alpha_2 + \frac{T-T_3}{T_f-T_3}\alpha_3 \tag{4.9}$$

$$T_4 \leq T \leq T_f \quad \frac{T-T_1}{T_f-T_1}\alpha_1 + \frac{T-T_2}{T_f-T_2}\alpha_2 + \frac{T-T_3}{T_f-T_3}\alpha_3 + \frac{T-T_4}{T_f-T_4}\alpha_4 \tag{4.10}$$

For $i = 1, 2, ..., N$

$$\alpha T\%_i = \sum_{m=1}^{i} \frac{T-T_i}{T_f-T_i}\alpha_i \tag{4.11}$$

Where T is in the interval of

$$T_i \leq T \leq T_{i+1} \qquad \text{For } i = 1, 2, ..., N-1 \tag{4.12}$$

And T is in the interval of

$$T_N \leq T \leq T_f \qquad \text{For } i = N \tag{4.13}$$

The total time is equal to the combined intervals of equation (4.12) and (4.13).

In chapter 3 finding α_1 was sufficient for all calculations. However in chapter 4 to explain the behavior of equation (4.11), and Figure 4.2, the values of α_1 through α_N are necessary. Since case 1 of chapter 3 and case 1 of chapter 4 use the same conditions and timing diagrams, the values of α_1 is found using equation (3.57) repeated below.

$$\alpha_1 = \frac{1}{1 + \sum_{i=2}^{N} \prod_{m=1}^{i-1} k_m} \tag{4.14}$$

With each value of N a different α_1 is found. In a network with four processors the value of α_1 is 0.3657. This value is used to calculate the remaining alpha's of which α_1 is the largest.

The normalized equation is equation(3.53)

$$\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = 1 \tag{4.15}$$

Knowing this and using the following conditions:

$w_i = 6$, $z_i = 2$, $T_{cm} = T_{cp} = 2$, $N = 4$ and $\alpha_1 = 0.3657$

The ratios of the alpha's are found

For i = 1 to N-1

$$k_i = \frac{w_i T_{cp}}{z_{i+1} T_{cm} + w_{i+1} T_{cp}} = 0.75 \tag{4.16}$$

The number of k's is equal to N-1.

Equation (3.51) below

$$\alpha_i = \prod_{m=1}^{i-1} k_m \alpha_{i-(i-1)} \tag{4.17}$$

for i = 2 to N is used to produce equations (3.18), (3.32) and (3.46), listed below respectively:

$$\alpha_2 = k_1\alpha_1$$

$$\alpha_3 = k_1k_2\alpha_1$$

$$\alpha_4 = k_1k_2k_3\alpha_1$$

Using the above conditions the k’s equated to:

$$k_1 = k = 0.75$$

$$k_1k_2 = k^2 = 0.5625$$

$$k_1k_2k_3 = k^3 = 0.4219$$

An array with the first element set to 1 is created.

$$k_A = \begin{bmatrix} 1 & k & k^2 & k^3 \end{bmatrix} = \begin{bmatrix} 1 & 0.75 & 0.5625 & 0.4219 \end{bmatrix} \tag{4.18}$$

$$\alpha = \alpha_1 k_A = \begin{bmatrix} 0.3657 & 0.2743 & 0.2057 & 0.1543 \end{bmatrix} \tag{4.19}$$

The sum of the elements of α must equal one.

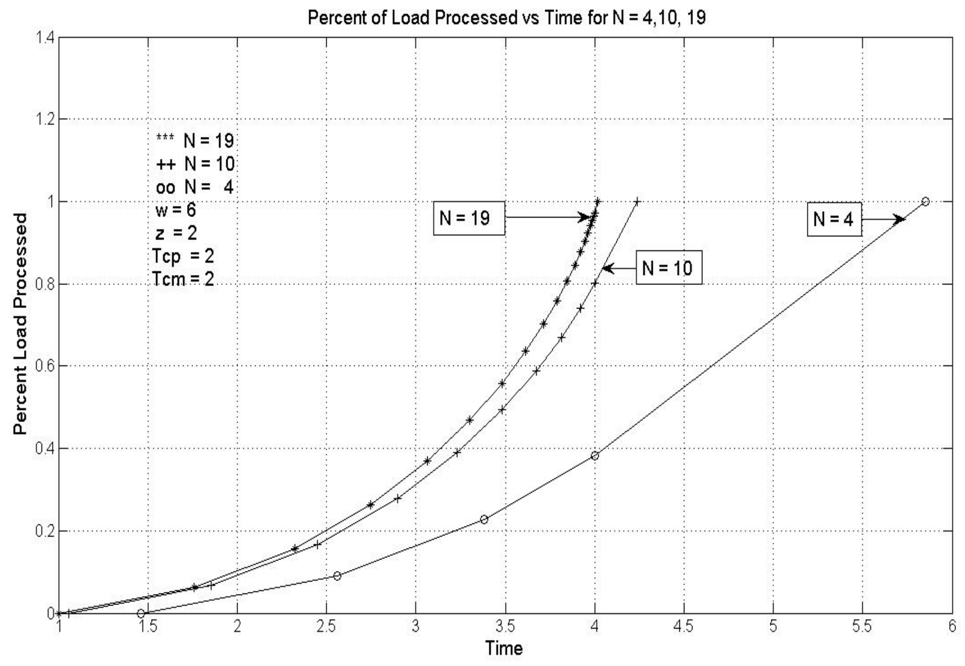


Figure 4.2. Percent of load processed (αT%) versus time for three different values of N. For staggered start with a root that does no processing, case 1.

In the calculation of αT% for equations (4.7) through (4.10) the variable T is replaced by the term on the right hand side of the interval before the equations in order to find the end points. The equations also allow for the exact percentage of the load searched at any time T in the interval to be determined. When the end points of αT% are plotted versus time as determined by T_1 through T_N plus T_f, Figure 4.2, it is found that for homogeneous link speeds and processors speeds the slope from point to point increases and the time between the points decreases, with the exception of the interval $T_N \leq T \leq T_f$. The time between the points decreases because the alpha’s of equation (4.5) decrease as demonstrated by equation (4.19).

The slope between the points in Figure 4.2 is an indicator of processor speed. Each successive processor is helped by the previous processor. For P_1 one processor is processing, for P_2 both processors are processing. At the final processor all processors in the network are processing load. So even thou the individual processors speeds are the same the cumulative effect is that load is being processed faster in the network as a whole. This can be seen by the increase in the slope over time. The slope between T_N and T_f does not follow the same pattern as the slopes from T_1 to T_N. On the plots in figure 4.2, $\alpha T\%$ must reach 100% at the finish time, if not the network is not optimized because it is possible for a processor to continue processing. The finish time of equation (4.6) depends on α_1, and α_1 depends on the number of processors, when z_1T_{cm} and w_1T_{cp} are held constant. For example with $N = 4$, $\alpha_1 = 0.3657$ and $T_f = 5.814$. If $N = 10$, $\alpha_1 = 0.2449$ and $T_f = 4.2387$. This shows that the finish time decreases as processors are added. As the network increases in size, T_f approaches T_N, this can be seen on Figure 4.2. Here the value of T_N is fixed at the same value for all plots. This is because in equation (4.6) the alpha’s sum to one which means that the value of T_N is determined by T_{cm} and z. In these plots $T_N = 4$. When T_f converges to a value that is closest to the margin of T_N + 0.4% of T_N the number of processors this occurs at is noted and considered the maximum number of processor allowed for this type of distribution with the given conditions. Here it is N = 19. If $T_f = T_N$ the load would be distributed to the processor but no computation would take place as the finish time would occur as the distribution was completed. If $T_f < T_N$ the transmission would cease at the finish time, and would not be fully distributed. Furthermore no computation would take place. This says that the network had too many processors for the given conditions. In Figure 4.3 the processor speeds are plotted against the number of processors. Here as above the processor speeds are homogenous. The slope is constant from processor to processor as expected. The maximum speed for this particular network for the given conditions is the processor speed at processor 19, which is 3.1667.

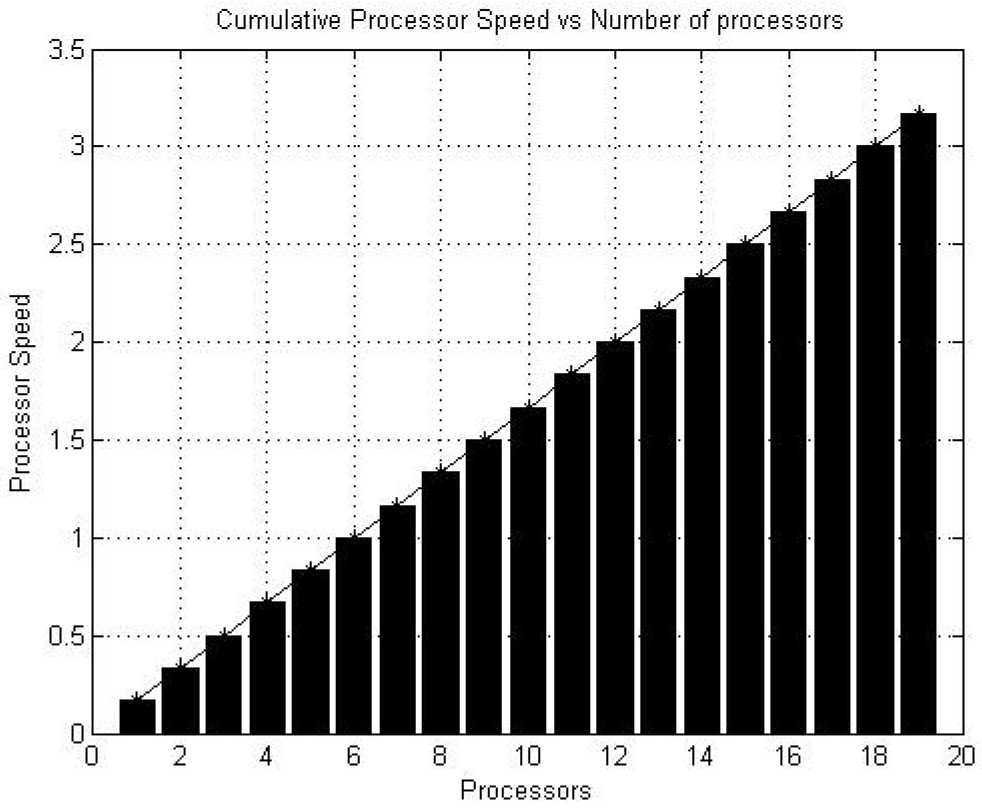


Figure 4.3. Processor speeds versus the number of processor in the network, where the root does no processing, case 1. The speed on a single processor is $\frac{1}{w}$, where w = 6. All processor speeds are the are the same.

4.3 CASE 2
SEQUENTIAL DISTRIBUTED
STAGGERED START
ROOT DOES PROCESSING

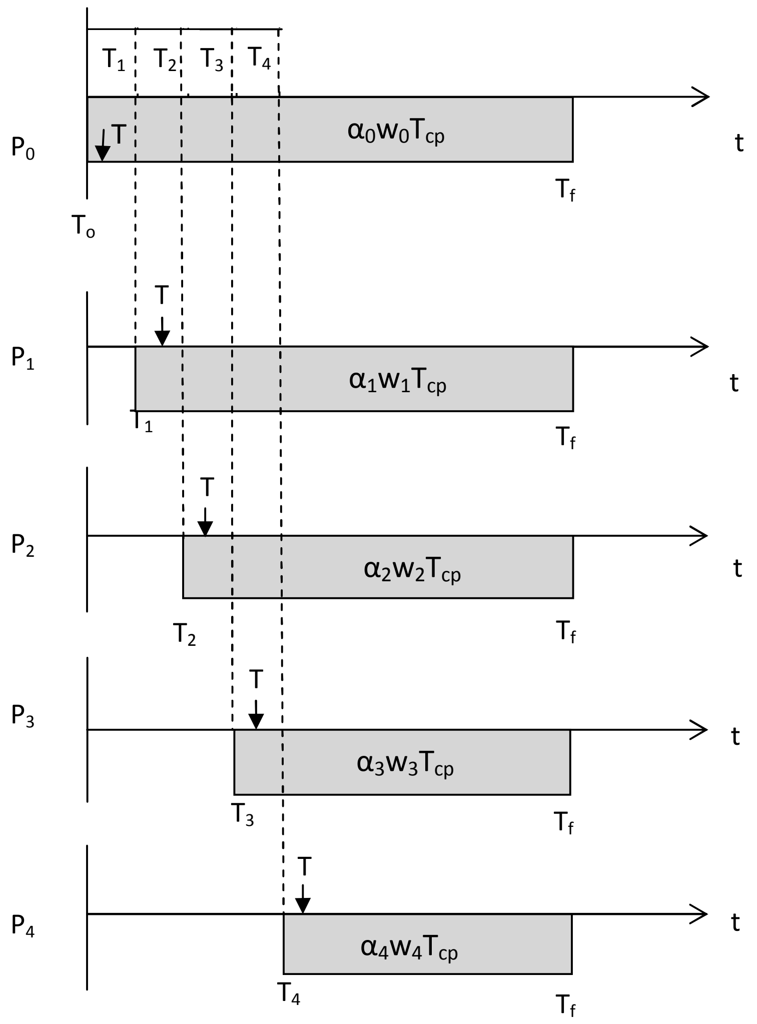


Figure 4.4. Timing diagram of a single level tree with sequential distribution, staggered start, and a root that does processing.

In Figure 4.4, T_1 through T_4 represents the loads that are sequentially distributed, similar to figure 3.6. Figure 4.4 is different from Figure 4.1 in that the root does processing. As in Figure 4.1, T represents the amount of load searched by the processor in an interval defined by the communication delays, T_0 through T_4, where the communication delay of T_0 is zero. In Figure 4.4, the root P_o has front end processing and can process and transmit at the same time. As a result the root begins processing at $T_{o,}$ and transmitting the load α_1 of T_1 to processor P_1 over link z_1 at time T_o. Just like in Figure 4.1 the load being transmitted to P_1 will not start processing until the entire load has been transmitted. From this point on the timing diagrams of Figures 4.1 and Figure 4.4 are identical and perform the same.

$$T_0 = 0 \tag{4.20}$$

$$T_1 = (\alpha_1 z_1)T_{cm} \tag{4.21}$$

$$T_2 = (\alpha_1 z_1 + \alpha_2 z_2)T_{cm} \tag{4.22}$$

$$T_3 = (\alpha_1 z_1 + \alpha_2 z_2 + \alpha_3 z_3)T_{cm} \tag{4.23}$$

$$T_4 = (\alpha_1 z_1 + \alpha_2 z_2 + \alpha_3 z_3 + \alpha_4 z_4)T_{cm} \tag{4.24}$$

$$T_N = (\alpha_1 z_1 + \alpha_2 z_2 + \alpha_3 z_3 + \alpha_4 z_4 + ... + \alpha_N z_N)T_{cm} \tag{4.25}$$

All processors P_0 through P_4 finish processing at the same time T_f.

$$T_f = \alpha_0 w_0 T_{cp} \tag{4.26}$$

The percent of the load searched by processors P_0 through P_4 within an interval is given by αT%, and is defined by the following equation.

$T_0 \leq T \leq T_1$

$$\frac{T-T_0}{T_f-T_0}\alpha_0 \tag{4.27}$$

$T_1 \leq T \leq T_2$

$$\frac{T-T_0}{T_f-T_0}\alpha_0 + \frac{T-T_1}{T_f-T_1}\alpha_1 \tag{4.28}$$

$T_2 \leq T \leq T_3$

$$\frac{T-T_0}{T_f-T_0}\alpha_0 + \frac{T-T_1}{T_f-T_1}\alpha_1 + \frac{T-T_2}{T_f-T_2}\alpha_2 \tag{4.29}$$

$T_3 \leq T \leq T_4$

$$\frac{T-T_0}{T_f-T_0}\alpha_0 + \frac{T-T_1}{T_f-T_1}\alpha_1 + \frac{T-T_2}{T_f-T_2}\alpha_2 + \frac{T-T_3}{T_f-T_3}\alpha_3 \tag{4.30}$$

$T_4 \leq T \leq T_f$

$$\frac{T-T_0}{T_f-T_0}\alpha_0 + \frac{T-T_1}{T_f-T_1}\alpha_1 + \frac{T-T_2}{T_f-T_2}\alpha_2 + \frac{T-T_3}{T_f-T_3}\alpha_3 + \frac{T-T_4}{T_f-T_4}\alpha_4 \tag{4.31}$$

For $i = 0, 1, 2, ..., N$

$$\alpha T\%_i = \sum_{m=0}^{i} \frac{T - T_i}{T_f - T_i} \alpha_i \tag{4.32}$$

Where T is in the interval of

$$T_i \leq T \leq T_{i+1} \qquad \text{For } i = 0, 1, 2, ..., N-1 \tag{4.33}$$

And T is in the interval of

$$T_N \leq T \leq T_f \qquad \text{For } i = N \tag{4.34}$$

The total time is equal to the combined intervals of equation (4.33) and (4.34).

Case 2 of chapter 3 and case 2 of chapter 4 use the same timing diagram. The normalized equation from is equation(3.114)

$$\alpha_0 + \alpha_1 + ... + \alpha_N = 1 \tag{4.35}$$

From equation (3.116)

$$\alpha_0 = \frac{\alpha_1}{k_0} \tag{4.36}$$

Here α_1 is found using equation (3.121) repeated here as

$$\alpha_1 = \frac{1}{\frac{1}{k_0} + 1 + \frac{1}{k_0} \sum_{i=2}^{N} \prod_{m=0}^{i=1} k_m} \tag{4.37}$$

As before in case 1 each value of N produces a different α_1, the α_1 used depends on the size N of the network. Here processors P_1 to P_4 are children processors and N refers to the number of children processors. In case 2 for

N = 4, α_1 = 0.2458 and α_0 = 0.3278. The ratios of the alpha's, the k's are found using equation (4.16), with i = 0 to N-1. The equation that shows the relationship between the alpha's and the k's equation (3.112) is repeated below.

$$\alpha_i = \prod_{m=0}^{i-1} k_m \alpha_{i-(i-1)-1} \qquad \text{for } i = 1, 2, ..., N \tag{4.38}$$

Using the same conditions as in case 1 the values of the k's are the as before same, k_i = k = 0.75, and the number of k's is equal to N. Equation (4.36) reproduces equations (3.70), (3.82), (3.96) and (3.108) listed below.

$$\alpha_1 = k_0 \alpha_0$$

$$\alpha_2 = k_0 k_1 \alpha_0$$

$$\alpha_3 = k_0 k_1 k_2 \alpha_0$$

$$\alpha_4 = k_0 k_1 k_2 k_3 \alpha_0$$

Using the above conditions the k's equate to:

$$k_0 = k = 0.75$$

$$k_0 k_1 = k^2 = 0.5621$$

$$k_0 k_1 k_2 = k^3 = 0.4219$$

$$k_0 k_1 k_2 k_3 = k^4 = 0.3164$$

As in case 1 an array K_A is formed with the first element set to 1.

$$k_A = \begin{bmatrix} 1 & k & k^2 & k^3 & k^4 \end{bmatrix} = \begin{bmatrix} 1 & 0.75 & 0.5625 & .4219 & 0.3164 \end{bmatrix} \tag{4.39}$$

$$\alpha = \alpha_0 k_A = \left[\begin{array}{ccccc} 0.3657 & 0.2459 & 0.1844 & 0.1383 & 0.1037 \end{array} \right] \tag{4.40}$$

The sum of the elements of α must equals one.

In the calculation of αT% for equations (4.27) to (4.31) the variable T is replaced by the term on the right hand side of the interval above the equation. This as in case 1 provides the end points. The end points are plotted versus time in a similar fashion as in the previous case, and under the same conditions, Figure 4.5.

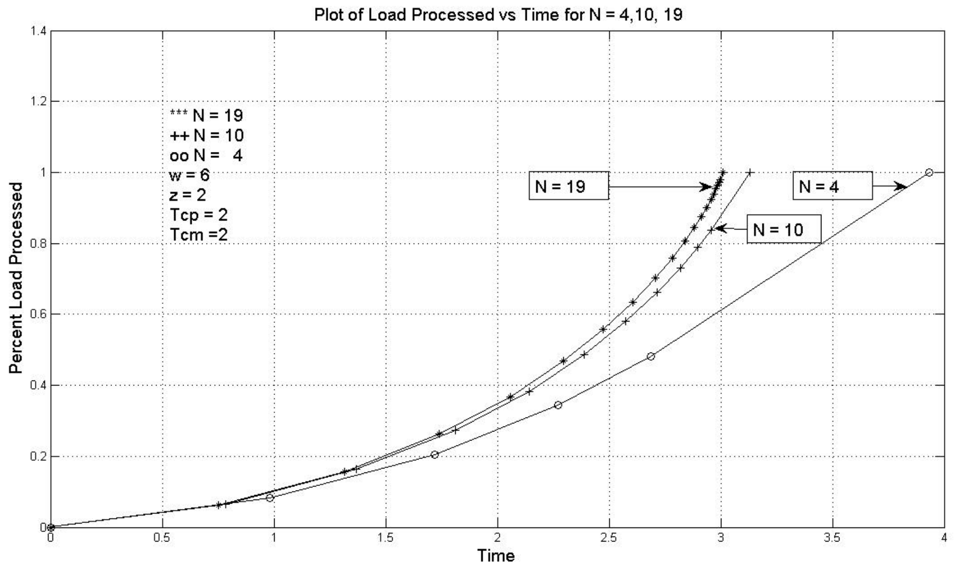


Figure 4.5. Percent of load processed (αT%) versus time for three different values of N. For staggered start with a root that does processing, case 2.

The slopes increase from point to point, and the time between them decreases in interval $T_0 \leq T \leq T_N$. In the interval $T_N \leq T \leq T_f$ for N = 4 the slope increases with respect to the previous slope on the same plot, however not the steady increase seen up to this point. The time between T_N

and T_f is at its greatest for any value of N larger than four. For N = 10 the slope increases by a greater degree, and the time decreases. At N = 19 the slope and the time now conform to a pattern consistent with that established in the interval $T_0 \leq T \leq T_N$.

Unlike case 1 the value of T_N is not the same for all plots. This is because in equation (4.23) the alpha's do not sum to one. In this case the normalized equation (4.33) includes α_0, a term not found in equation (4.33). With each new value of N the sum of α_1 to α_N increases. At the same time T_f which depends on α_0 decreases, because α_0 decreases with each new value of N. This is due to the redistribution of the load, which must sum to one, while incorporating more alpha's. For this type of network the first alpha, α_0 remains the largest after redistribution. This creates a situation where T_f and T_N are moving toward each other. When T_f converges to a value that is closest to the margin of T_N + 0.4% of T_N, the processor this occurs at is noted and considered to be the maximum number of children processors for this distribution and conditions. Again this occurred at N = 19, as in case 1. The same pronouncement for the parameters of $T_f = T_N$ and $T_{f\,<}\ T_N$ in case 1 hold for case 2.

In Figure 4.6 the processors speeds are plotted versus the number of processors. The processors speeds are homogenous, and as in case 1 the slope is constant from processor to processor. However in this case the processor P_0 is included. At N = 19 the maximum speed for this type of network for the given condition is 3.3333. However there are 20 processors working at the same speed that produces this result.

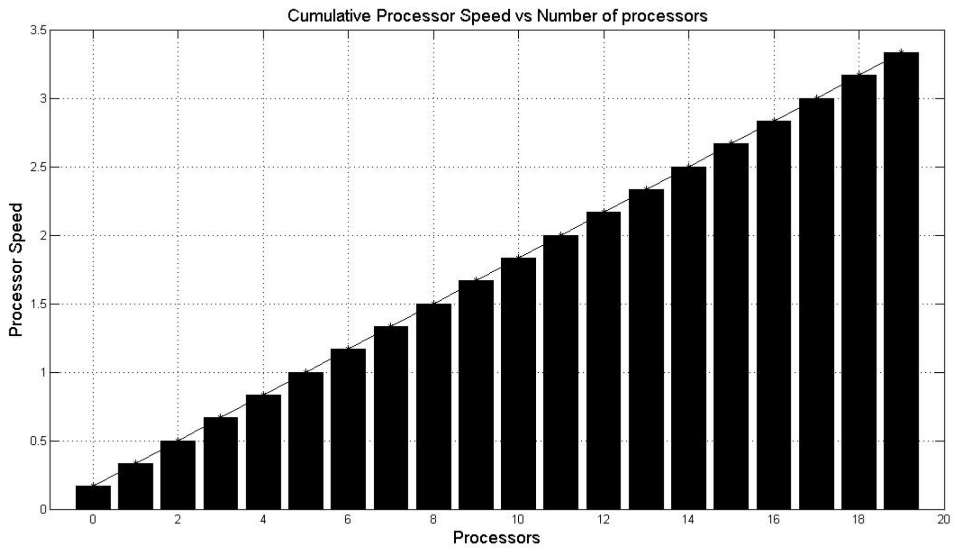


Figure 4.6. Processor speeds versus the number of processors in the network, where the root does processing, case 2. The speed on a single processor is $\frac{1}{w}$, where w = 6. All processor speeds are the same.

4.4 CASE 3 SEQUENTIALLY DISTRIBUTED SIMULTANEOUS START ROOT DOES PROCESSING

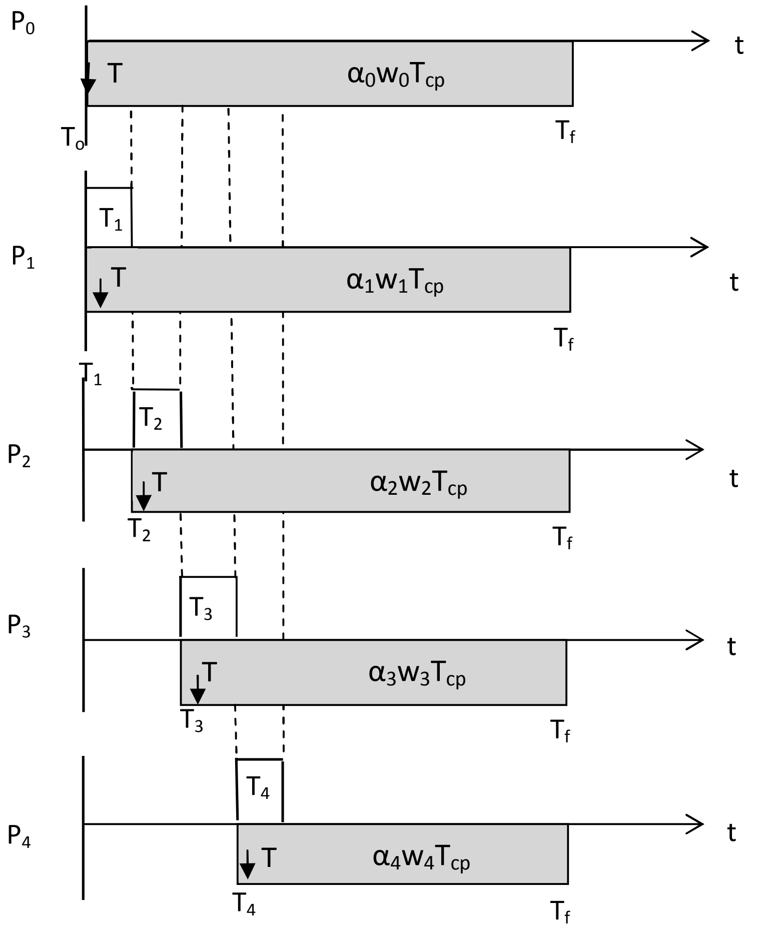


Figure 4.7. Timing diagram of a single level tree with sequential distribution, simultaneous start and a root that does processing

In Figure 4.7 the communication delays are represented by T_1 to T_4 shown in boxes above the t axis. The additional T_1 through T_4, shown below the t axis marks the beginning of the communication delays for the same variable. First the load of T_1 is transmitted to processor P_1 it starts processing as soon as part of it is received because the processors have simultaneous start. When the transmission is complete, the load of T_2 is transmitted, this process continues until all the loads are transmitted. The equations below describe the working of Figure 4.7.

$$T_0 = 0 \tag{4.41}$$

$$T_1 = 0 \tag{4.42}$$

$$T_2 = (\alpha_1 z_1)T_{cm} \tag{4.43}$$

$$T_3 = (\alpha_1 z_1 + \alpha_2 z_2)T_{cm} \tag{4.44}$$

$$T_4 = (\alpha_1 z_1 + \alpha_2 z_2 + \alpha_3 z_3)T_{cm} \tag{4.45}$$

$$T_N = (\alpha_1 z_1 + \alpha_2 z_2 + \alpha_3 z_3 + ... + \alpha_N z_N)T_{cm} \tag{4.46}$$

All processors P_0 through P_4 finish processing at the same time T_f.

$$T_f = \alpha_0 w_0 T_{cp} \tag{4.47}$$

The percent of the load searched by processors P_0 through P_4 within an interval is given by αT%. The same set of equations for the calculation of αT% in case 2, (4.31) to (4.27) applies to case 3, repeated below.

$T_0 \leq T \leq T_1$

$$\frac{T-T_0}{T_f-T_0}\alpha_0 \tag{4.48}$$

$T_1 \leq T \leq T_2$

$$\frac{T-T_0}{T_f-T_0}\alpha_0 + \frac{T-T_1}{T_f-T_1}\alpha_1 \tag{4.49}$$

$T_2 \leq T \leq T_3$

$$\frac{T-T_0}{T_f-T_0}\alpha_0 + \frac{T-T_1}{T_f-T_1}\alpha_1 + \frac{T-T_2}{T_f-T_2}\alpha_2 \tag{4.50}$$

$T_3 \leq T \leq T_4$

$$\frac{T-T_0}{T_f-T_0}\alpha_0 + \frac{T-T_1}{T_f-T_1}\alpha_1 + \frac{T-T_2}{T_f-T_2}\alpha_2 + \frac{T-T_3}{T_f-T_3}\alpha_3 \tag{4.51}$$

$T_4 \leq T \leq T_f$

$$\frac{T-T_0}{T_f-T_0}\alpha_0 + \frac{T-T_1}{T_f-T_1}\alpha_1 + \frac{T-T_2}{T_f-T_2}\alpha_2 + \frac{T-T_3}{T_f-T_3}\alpha_3 + \frac{T-T_4}{T_f-T_4}\alpha_4 \tag{4.52}$$

For $i = 0, 1, 2, ..., N$

$$\alpha T\%_i = \sum_{m=0}^{i} \frac{T-T_i}{T_f-T_i}\alpha_i \tag{4.53}$$

Where T is in the interval of

$$T_i \leq T \leq T_{i+1} \qquad \text{For } i = 0, 1, 2, ..., N-1 \tag{4.54}$$

And T is in the interval of

$$T_N \leq T \leq T_f \qquad \text{For } i = N \tag{4.55}$$

The total time is equal to the combined intervals of equation (4.54) and (4.55). Case 3 of chapter 3 and case 3 of chapter 4 use the same timing diagram. The normalized equation from (3.171) is

$$\alpha_0 + \alpha_1 + ... + \alpha_N = 1 \tag{4.56}$$

From equation (3.175)

$$\alpha_0 = \frac{\alpha_1}{k_0} \tag{4.57}$$

And from equation (3.177)

$$k_0 = \frac{w_0}{w_1} \tag{4.58}$$

The alpha's are calculated using equation (3.182) repeated below

$$\alpha_1 = \frac{1}{\frac{1}{k_0} + 1 + \sum_{i=2}^{N} \prod_{m=1}^{i-1} k_m} \tag{4.59}$$

As in the previous cases for each new value of N a different α_1 is found.

The ratios of the k's are found using a generalized form of equation

(3.162)

$$k_i = \frac{w_i T_{cp} - z_i T_{cm}}{w_{i+1} T_{cp}} \qquad i = 1, 2, ..., N \tag{4.60}$$

For homogenous processor speeds and link speeds, which include both parent and children processors $w_i = w$ and $z_i = z$. The intensity parameters are set at $T_{cp} = T_{cm} = 2$. Using the above conditions in equation (4.60),

$k_i = k = 0.6667$, for i = 1, 2, ..., N.

In case 3 for N = 4, $\alpha_1 = 0.2935$ since $w_i = w$, equation (4.58) yields $k_0 = 1$.

From equation (4.57) $\alpha_1 = \alpha_0 = 0.2935$.

The equation that shows the relationship between the alpha's and the k's (3.169) is repeated below.

$$\alpha_i = \prod_{m=1}^{i-1} k_m \alpha_{i-(i-1)} \qquad i = 2, 3, ..., N \tag{4.61}$$

This equation reproduces equation (3.138), (3.178) and (3.165) listed below.

$$\alpha_2 = k_1 \alpha_1$$

$$\alpha_3 = k_1 k_2 \alpha_1$$

$$\alpha_4 = k_1 k_2 k_3 \alpha_1$$

Using the above conditions the k's equate to:

$$k_1 = k = 0.6667$$

$$k_1 k_2 = k^2 = 0.4445$$

$$k_1 k_2 k_3 = k^3 = 0.2963$$

In case 2 the first element of the array is set to 1, however in this case 3 the first two elements are set to 1. This is due to the fact that $\alpha_1 = \alpha_0$.

$$k_A = \left[\begin{array}{ccccc} 1 & 1 & k & k^2 & k^3 \end{array}\right] = \left[\begin{array}{ccccc} 1 & 1 & 0.6667 & 0.4445 & 0.2963 \end{array}\right] \tag{4.62}$$

$$\alpha = \alpha_1 k_A = \left[\begin{array}{ccccc} 0.2935 & 0.2935 & 0.1957 & 0.13042 & 0.0870 \end{array}\right] \tag{4.63}$$

In agreement with the normalized equation (4.56), the elements of α array sum to one.

The percent of load processed (αT%) is found for equations (4.48) to (4.52) following the same procedure as in case 2. Similarly the end points are determined and plotted versus time, Figure 4.8.

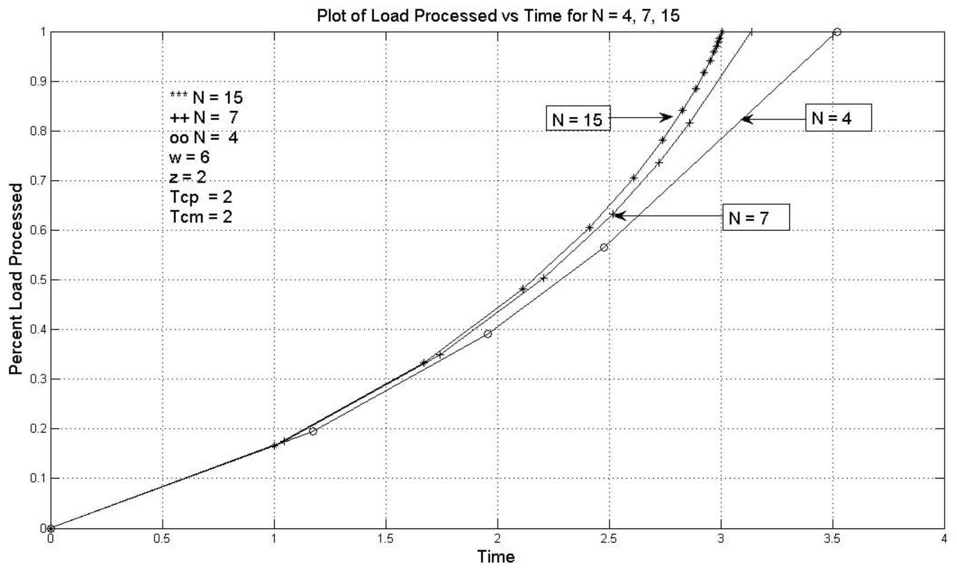


Figure 4.8. Percent of load processed (αT%) versus time for three different values of N, for simultaneous start with a root that does processing, case 3.

As in case 2 the slope from point to point increases, and the time between them decreases in interval $T_0 \leq T \leq T_N$. In interval $T_N \leq T \leq T_f$, for N = 4 the finish time is at its greatest value as compared with any other finish time for a N of greater value. Furthermore the slope has increased, but not with at the same rate as the previous slopes on the plot. Here as in case 2, T_f and T_N are moving toward each other, because the alpha's of equation (4.46) do not sum to one. For N = 7, the same pattern as in case 2 appears. As the size of the network increases the slope increases and the finish time decreases. At, so the actual number of processor is the parent plus the children processor totaling 16. The same reasoning for the parameters of $T_f = T_N$ and $T_f < T_N$ in case 1 and case 2 hold for case 3.

In Figure N = 15, the processor at which the finish time converges to a value nearest the margin of T_N + 0.4% of T_N is reached. Here N = 15 is the maximum number of children processor for this distribution and conditions. As stated before the parent processor aids in increasing the overall speed of the network4.9 the processors speeds are plotted versus the number of processors. The processor speeds are homogenous as in the previous cases. The root does processing so the parent is included. At N = 15 the maximum speed for this type of network for the given condition is 2.6667.

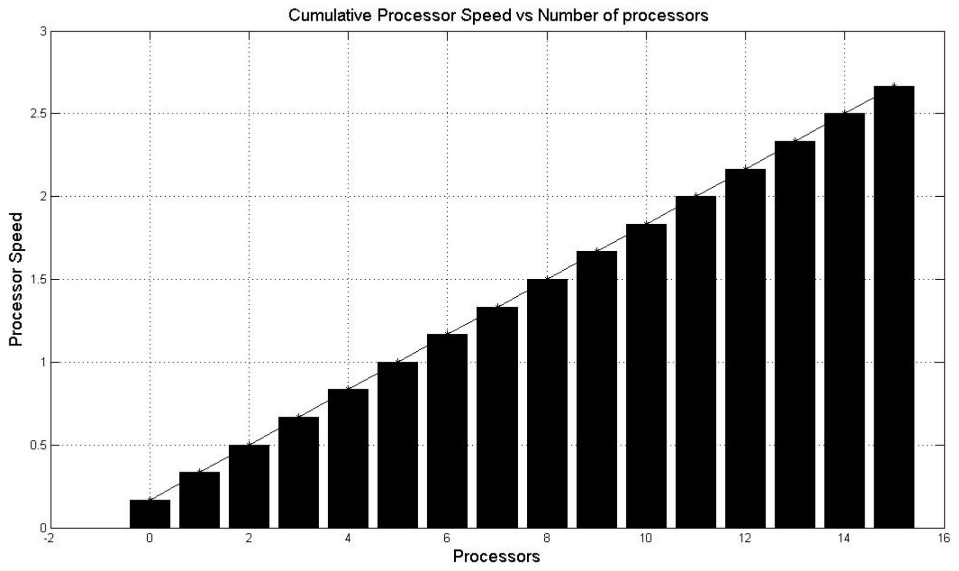


Figure 4.9. Processor speeds versus the number of processor in the network, where there is simultaneous start and the root does processing, case 3. The speed on a single processor is $\frac{1}{w}$ where w = 6. All processors are equal.

4.5 CASE 4

SEQUENTIALLY DISTRIBUTED
SIMULTANEOUS START
ROOT DOES NO PROCESSING

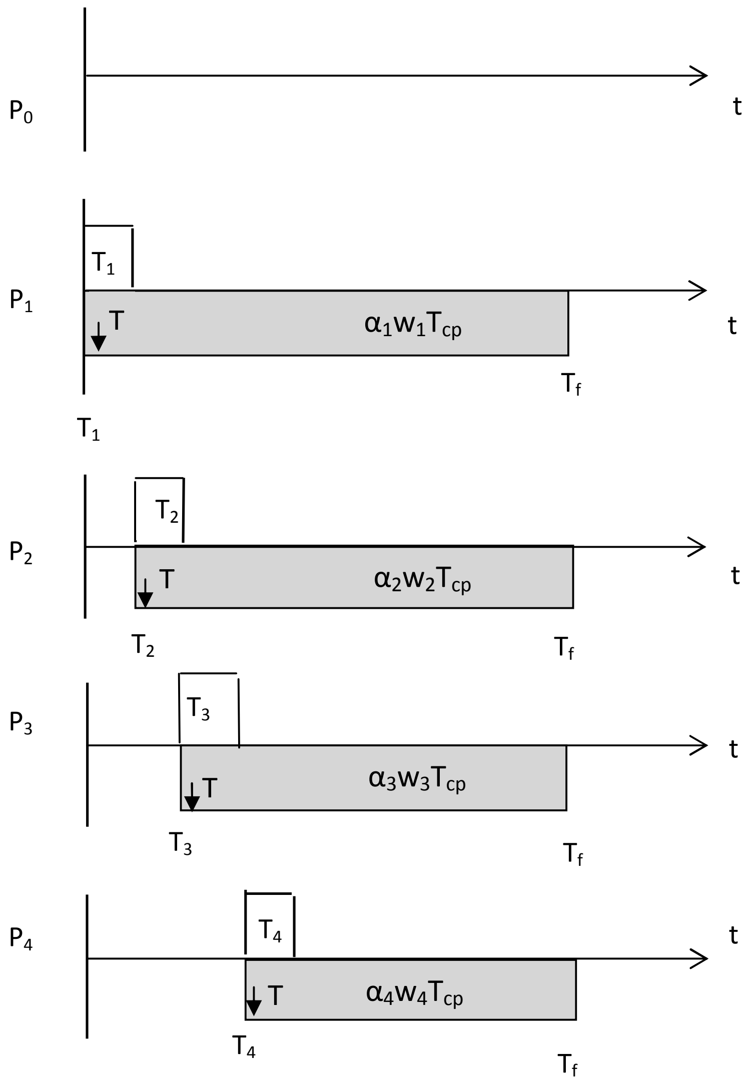


Figure 4.10 Timing diagram of a single level tree with sequential distribution, simultaneous start and a root that does no processing.

In Figure 4.10 the communication delays are depicted as boxes numbered T_1 to T_4 on top of the axis. Below the axis the same T_1 through T_4 mark the beginning of the communication delays. The computation portion of the processor is also below the t axis. The T besides the downward pointing arrow is an indicator of what percent of the processor is being searched in a given interval. Here the loads are transmitted sequentially, as in case 3. As soon as any part of the load is received computation starts. The computation starts in processor P_1 at T_1, and at processor P_2 at T_2. From this point on both processor P_1 and P_2

are computing. When all the loads are transmitted each processor will be engaged. Upon reaching the finish time T_f all processor cease computations. The equations that described this timing diagram are listed below.

$$T_1 = 0 \tag{4.64}$$

$$T_2 = (\alpha_1 z_1) T_{cm} \tag{4.65}$$

$$T_3 = (\alpha_1 z_1 + \alpha_2 z_2) T_{cm} \tag{4.66}$$

$$T_4 = (\alpha_1 z_1 + \alpha_2 z_2 + \alpha_3 z_3) T_{cm} \tag{4.67}$$

$$T_N = (\alpha_1 z_1 + \alpha_2 z_2 + \alpha_3 z_3 + \alpha_4 z_4 + ... + \alpha_N z_N) T_{cm} \tag{4.68}$$

Each processor P_1 through P_4 all of which are children processors finish processing at the same time, the finish time T_f.

$$T_f = \alpha_1 w_1 T_{cp} \tag{4.69}$$

The percent load searched by processor P_1 to P_4 within a given interval is defined as αT%. These are in fact the same set of equations from case 1, (4.7) to (4.10), and (4.12) and (4.13) repeated below.

$$T_1 \le T \le T_2 \quad \frac{T - T_1}{T_f - T_1}\alpha_1 \tag{4.70}$$

$$T_2 \le T \le T_3 \quad \frac{T - T_1}{T_f - T_1}\alpha_1 + \frac{T - T_2}{T_f - T_2}\alpha_2 \tag{4.71}$$

$$T_3 \le T \le T_4 \quad \frac{T - T_1}{T_f - T_1}\alpha_1 + \frac{T - T_2}{T_f - T_2}\alpha_2 + \frac{T - T_3}{T_f - T_3}\alpha_3 \tag{4.72}$$

$$T_4 \le T \le T_f \quad \frac{T - T_1}{T_f - T_1}\alpha_1 + \frac{T - T_2}{T_f - T_2}\alpha_2 + \frac{T - T_3}{T_f - T_3}\alpha_3 + \frac{T - T_4}{T_f - T_4}\alpha_4 \tag{4.73}$$

For $i = 1, 2, ..., N$

$$\alpha T\%_i = \sum_{m=1}^{i} \frac{T - T_i}{T_f - T_i}\alpha_i \tag{4.74}$$

Where T is in the interval of

$$T_i \le T \le T_{i+1} \qquad \text{For } i = 1, 2, ..., N-1 \tag{4.75}$$

And T is in the interval of

$$T_N \le T \le T_f \qquad \text{For } \quad i = N \tag{4.76}$$

The total time is equal to the combined intervals of equation (4.75) and (4.76).

Case 4 of chapter 3 and case 4 of chapter 4 use the same timing diagram. The portion of the load α_1 is found using equation (3.201) repeated below.

$$\alpha_1 = \frac{1}{1 + \sum_{i=2}^{N} \prod_{m=1}^{i-1} k_m} \tag{4.77}$$

As in all previous cases each new value of N produces a different α_1.

The ratio of the α's, the k's using equation (4.60) repeated below

$$k_i = \frac{w_i T_{cp} - z_i T_{cm}}{w_{i+1} T_{cp}} \qquad i = 1, 2, ..., N \tag{4.78}$$

The equation that shows the relationship between the alpha's and the k's is equation (3.220) shown below

$$\alpha_i = \prod_{m=1}^{i-1} k_m \alpha_{i-(i-1)} \qquad i = 2, 3, ..., N \tag{4.79}$$

Using the conditions $w_i = w = 6$, $z_i = z = 2$, and $T_{cp} = T_{cm} = 2$. The value of the k's are $k_i = k = 0.6667$. For N = 4 the number of k's is N-1.

Equation (4.79) reproduces equations (3.138), (3.153) and (3.165), derived in chapter 3, they are listed below respectively.

$$\alpha_2 = k_1 \alpha_1$$

$$\alpha_3 = k_1 k_2 \alpha_1$$

$$\alpha_4 = k_1 k_2 k_3 \alpha_1$$

Using the above conditions the k's equate to:

$$k_1 = k = 0.6667$$

$$k_1 k_2 = k^2 = 0.4445$$

$$k_1 k_2 k_3 = k^3 = 0.2963$$

As in case 1 an array K_A is formed with the first element set to 1.

$$k_A = \left[\begin{array}{cccc} 1 & k & k^2 & k^3 \end{array}\right] = \left[\begin{array}{cccc} 1 & 0.6667 & 0.4445 & 0.2963 \end{array}\right] \quad (4.80)$$

$$\alpha = \alpha_1 k_A = \left[\begin{array}{cccc} 0.4154 & 0.2769 & 0.1864 & 0.1230 \end{array}\right] \quad (4.81)$$

The elements of α sum to one, as required by the normalization equation (4.15).

The percent load processed ($\alpha T\%$) is found as is case 1. Using equation (4.70) to (4.73), the end points are determined and plotted in a similar fashion. In this figure T_N from equation (4.68) is not stationary. This is because $T_1 = 0$ equation (4.64) and the first usage of α_1 appears in T_2 equation (4.65). Since the normalized equation (4.15) is

$\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = 1$, for $N = 4$. The sum of the alpha's of equation (4.67) is not equal to one. This means that T_N or T_4 here does not solely depend on the link speed (z) and the communication intensity (T_{cm}), which for homogeneous link speeds would hold the value of T_N constant for all values of N.

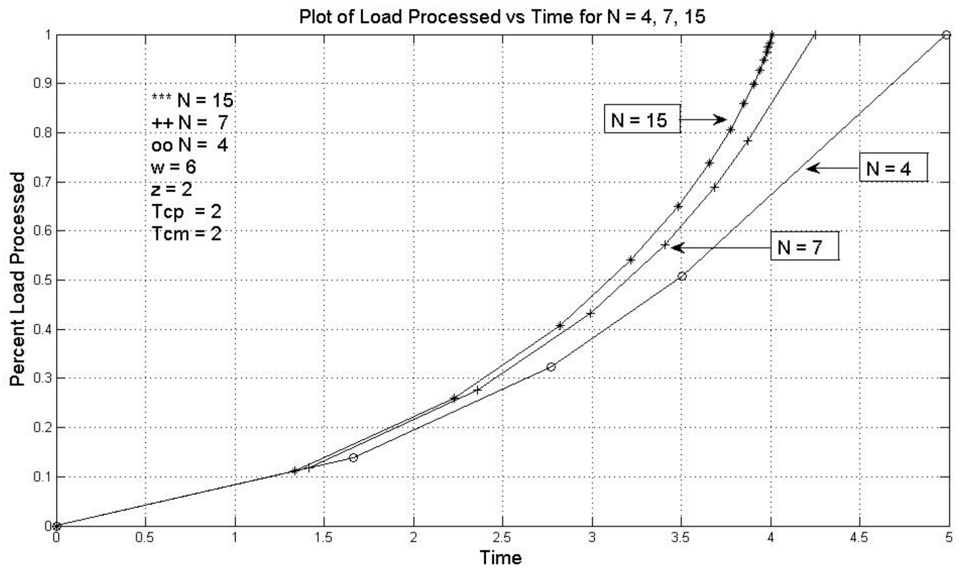


Figure 4.11. Percent of load processed (αT%) versus time for three different values of N for simultaneous start with a root that does no processing, case 4.

As expected from the previous cases, in Figure 4.11 the slopes increase from point to point and the time between them decreases in the interval $T_1 \leq T \leq T_N$. In the interval $T_N \leq T \leq T_f$, increases in the size of the network cause the finish time decrease. The number of processors required for the finish time to close within a value that is closest to the margin determined by T_N + .4% of T_N is, N = 15 for this type of network and conditions. The same reasoning for the parameters of $T_f = T_N$ and $T_f < T_N$ in the previous cases applied to case 4.

In Figure 4.12 the processor speeds are plotted versus the number of processor. At N = 15 the maximum speed for this type of network for the given conditions is 2.5.

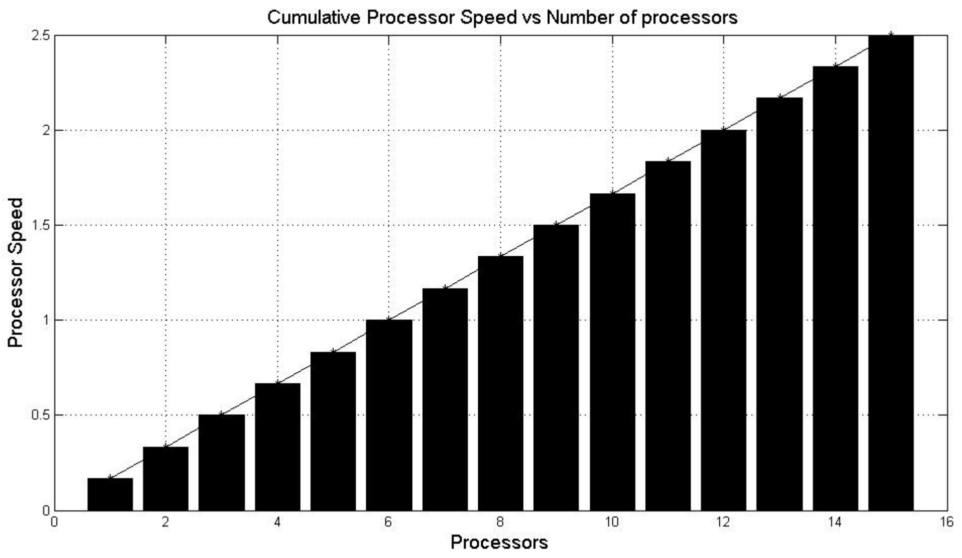


Figure 4.12. Processor speeds versus the number of processors in the network, that uses simultaneous start and a root that does no processing, case 4. The speed on a single processor is $\frac{1}{w}$, where w = 6. All processor speeds are equal.

4.6 ANALYSIS OF RESULTS/ CONCLUSION OF CHAPTER 4

Two sets of equations and their intervals are used to find the percent load processed. The type of parent processor determines which set is used.

Set 1 and set 2 are described basically by the same equation. Set 1 is for a root that does no processing (RDNP) equations (4.11) and (4.74) $\alpha T\%_i = \sum_{m=1}^{i} \frac{T-T_i}{T_f-T_i}\alpha_i$, for i = 1 to N. Set 2 is for a root that does processing (RDP) equations (4.32) and (4.53), $\alpha T\%_i = \sum_{m=0}^{i} \frac{T-T_i}{T_f-T_i}\alpha_i$ for i = 0 to N.

The upper limit of the summation i represents the number of computing processors, and m takes its value from the first value of i. In the four cases i = 1 to N for a root that does no processing, and i = 0 to N for a root that does processing.

The intervals are stated below:

$T_i \leq T \leq T_{i+1}$ for i = 1 to N-1 equations (4.12) and (4.75)

for i = 0 to N-1 equations (4.33) and (4.54)

$T_N \leq T \leq T_f$ for i = N equations (4.13), (4.32), (4.53) and (4.75)

The limits of the interval are controlled by the communication delays and finish time.

The speedup and utilization of chapter 3, along with performance metrics are compiled in Table 1.

case	Type of start	RDP RDNP	N T_N,T_f	max speed	T_N	T_f	EQs	N util	N speedup
1	Stag	RDNP	19	3.1667	4.0000	4.0170	1	20	19
2	Stag	RDP	19	3.3333	2.9968	3.0095	2	20	18
3	Sim	RDP	15	2.6667	2.9949	3.0051	2	14	17
4	Sim	RDNP	15	2.5000	3.9954	4.0092	1	14	18

Table 1. Complication of parameters from chapter 3 and 4.

Stag = Staggered start
Sim = Simultaneous start
RDP = Root does processing
RDNP = Root does no processing
$N_{TN,Tf}$ = Performance metric
max speed = Maximum cumulative processor speed
T_N = Last communication delay in a network
T_f = Finish time
EQs = Equation set
N_{Util} = Performance metric
$N_{speedup}$ = Performance metric
w = The inverse processor speed
z = The inverse link speed

When a root does processing the max speed is greater for processors in set 2 as compared with set 1. This is reasonable due to the cumulative effect of the processors, the more processors processing information the faster the overall network performance.

The number of processors a particular network can have is based on three different performance metrics. In Table 1, $N_{TN,Tf}$ is the number of processors a network can have based on the finish time and T_N the last communication delay in the network. Here $N_{TN,Tf}$ is the processor at which the finish time converges to a value closest to the margin of T_N + 0.4% of T_N. When the processor speeds versus the number of processor, including the root are plotted. The result is the maximum speed for the network, Figures 4.3, 4.6, 4.9, and 4.12.

The performance metric N_{Util} is based on how many processors it would take to force the utilization to zero. Here zero is defined as when the first processor's utilization has declined 99.6% from its original value. The number of the processor this occurs at is considered to be the number of processors the network can sustain, Figures 3.4, 3.8, and 3.12.

The final performance metric, $N_{speedup}$ is the number of processors that can be added to a network and still befit from parallel processing. It is found by taking 99.6% of the number at which speedup saturates, and matching it with the number of the processor it occurs at.

Speedup is an established performance metric, utilization was developed in chapter 3, and using the finish time and the T_N is another way found to check the validity of utilization. The performance metric N_{Util} is consistently one off from $N_{TN,Tf}$. For staggered start it is one greater, and for simultaneous start it is one less. The speedup is higher for simultaneous start. Overall for the four cases N_{Util} and $N_{TN,Tf}$ are a better match then speedup under these particular conditions.

Case 3 has the best finish time and uses the fewest processors. It would process the same load as case 2 with fewer processors and at a slightly smaller finish time. The maximum speed is higher in case 2, but the max speed is based on how many processors there are. So even though the maximum speed of case 3 is less than that of case 2 it accomplishes more with fewer processors. This is readily seen in Figure 4.5 case 2 and figure 4.8 case 3, although the exact values of the finish times cannot be discerned from the plots.

For a divisible load that can be partitions as that of α in equation (4.63). This is a result of having homogeneous processor and link speeds. A network that is sequentially distributed has simultaneous start, and a root that does processing will perform the best signature search.

Chapter 5

5. Linear Daisy Chain Configurations

5.1 Introduction

Linear daisy chains are the basic building blocks of network topologies, with the exception of a single node. The smallest linear daisy chain (LDC) the equations in this chapter can work with is one compose of three nodes (processor) or more. Understanding how the LDC's work makes it possible to create more complicated structure. For example in chapter 1 first a linear daisy chain of five nodes was designed. This was expanded to a mesh, and can be further expanded into three dimensions.

In this chapter two types of linear daisy chain are investigated. One is with staggered start without front end processing, the other has a hybrid strategy and front end processing. Staggered start means that when loads are transmitted to a processor all loads must arrive at that processor before processing can start. The hybrid strategy means that it has properties of staggered state and simultaneous start, depending on rather it is transmitting or receiving loads. The simultaneous starts can process information and transmit at the same time.

5.2 UTILIZATION OF LINEAR DAISY CHAIN
STAGGERED START
WITHOUT FRONT END PROCESSING

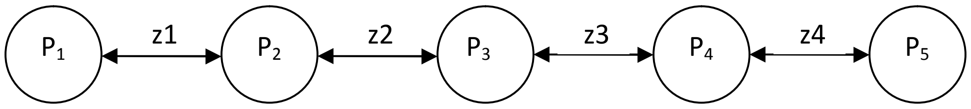


Figure 5.1. Five processors connected in a linear daisy chain configuration

In Figure 5.1 the loads on processor P_1 are α_2, α_3, α_4 and α_5. Processor P_1 does not have front end processing, it can transmit loads but not process them. The loads on P_1 are transmitted to P_2 over link z_1. Because the linear daisy chain has staggered start after all the loads have arrived at P_2, processing can begin. The load α_2 is absorbed by P_2, meaning that P_2 is processing α_2. The remaining loads α_3, α_4 and α_5 are transmitted to P_3 over link z_2. Here α_3 is absorbed by P_3 and loads α_4 and α_5 are transmitted to P_4 over link z_3. Following the same procedure load α_4 is absorbed by processor $P_{4,}$ and α_5 is transmitted to P_5 over link z_4 where it is absorbed.

The communication delays of Figure 5.2 and the loads within them are listed below.

$$T_2 = (\alpha_2 + \alpha_3 + \alpha_4 + \alpha_5) z_1 T_{cm} \tag{5.2}$$
$$T_3 = (\alpha_3 + \alpha_4 + \alpha_5) z_2 T_{cm} \tag{5.3}$$
$$T_4 = (\alpha_4 + \alpha_5) z_3 T_{cm} \tag{5.4}$$
$$T_5 = (\alpha_5) z_4 T_{cm} \tag{5.5}$$

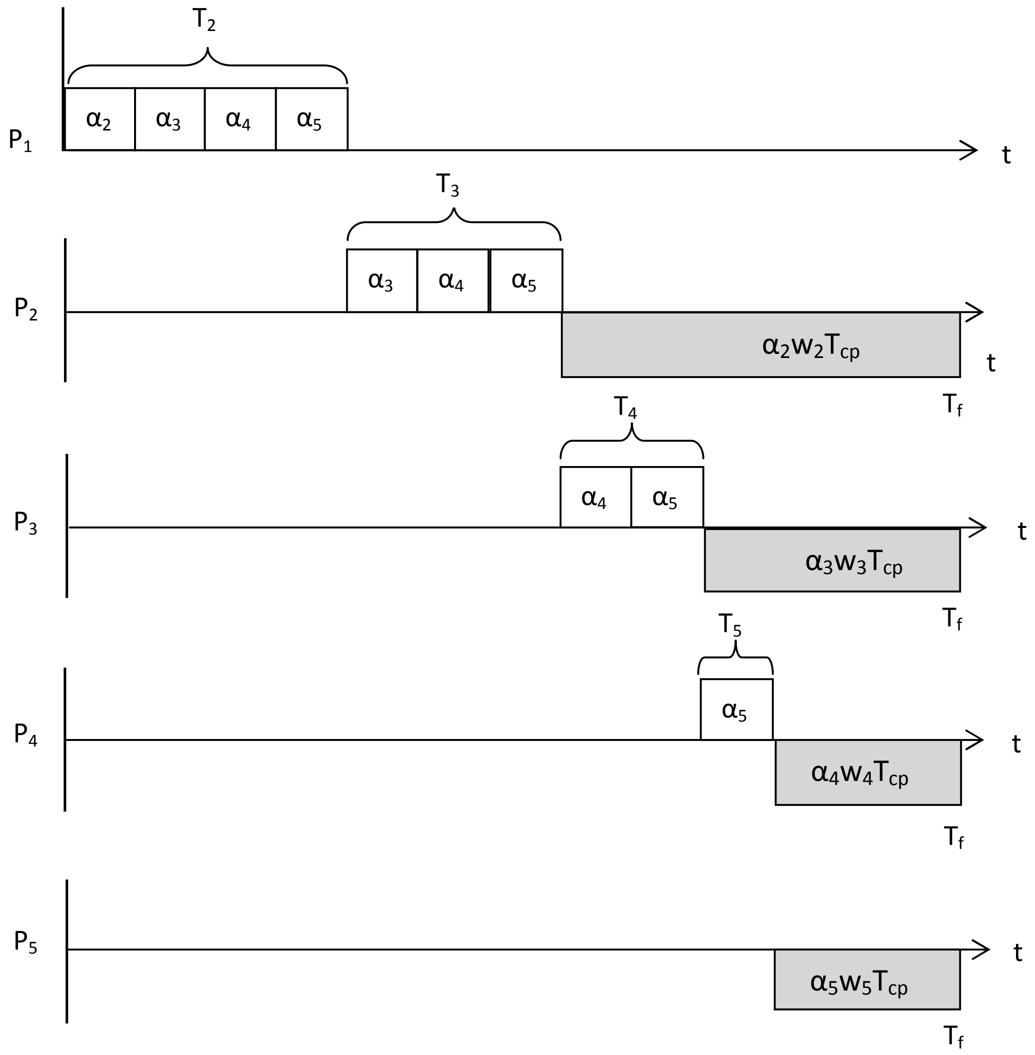


Figure 5.2. Timing diagram of a linear daisy chain with staggered start and no front end processing.

The timing diagram in Figure 5.2 is another way of representing the linear daisy chain configuration of Figure 5.1. The function is the same.

Only now it has been put in a form that makes it easy to calculate the utilization and speedup. The discussion is the same. First T_2 is transmitted to processor P_2 over link z_2. Processor P_2 absorbs α_2 from T_2 and transmitted α_3, α_4, and α_5 to processor P_3, over link z_2 by means of T_3. Processor P_3 now absorbs α_3 from T_3, and the remaining loads α_4 and α_5 are transmitted over link z_3 to Processor P_4 by means of T_4. Processor P_4 absorbs α_4 from T_4, and transmits α_5 to processor P_5 over link z_4, using T_5. Processor P_5 absorbs α_5 from T_5 completing the process.

The utilization of the linear daisy chain is calculated as in chapter 3, useful time divided by total time.

The total time of P_2 is given by

$$\begin{aligned} T_{f2} = {} & \alpha_2 z_1 T_{cm} + \alpha_3 z_1 T_{cm} + \alpha_4 z_1 T_{cm} + \alpha_5 z_1 T_{cm} \\ & + \alpha_3 z_2 T_{cm} + \alpha_4 z_{2cm} + \alpha_5 z_2 T_{cm} + \alpha_2 w_2 T_{cp} \end{aligned} \tag{5.6}$$

$$U_2 = \frac{\alpha_2 w_2 T_{cp}}{T_{f2}} \tag{5.7}$$

$$U_2 = \frac{\alpha_2 w_2 T_{cp}}{(\alpha_2 + \alpha_3 + \alpha_4 + \alpha_5) z_1 T_{cm} + (\alpha_3 + \alpha_4 + \alpha_5) z_2 T_{cm} + \alpha_2 w_2 T_{cp}} \tag{5.8}$$

$$U_2 = \frac{\frac{\alpha_2 w_2 T_{cp}}{\alpha_2 w_2 T_{cp}}}{\frac{(\alpha_2+\alpha_3+\alpha_4+\alpha_5)}{\alpha_2} \frac{z_1 T_{cm}}{w_2 T_{cp}} + \frac{(\alpha_3+\alpha_4+\alpha_5)}{\alpha_2} \frac{z_2 T_{cm}}{w_2 T_{cp}} + \frac{\alpha_2 w_2 T_{cp}}{\alpha_2 w_2 T_{cp}}} \tag{5.9}$$

$$U_2 = \frac{1}{\frac{(\alpha_2+\alpha_3+\alpha_4+\alpha_5)}{\alpha_2}\frac{z_1T_{cm}}{w_2T_{cp}} + \frac{(\alpha_3+\alpha_4+\alpha_5)}{\alpha_2}\frac{z_2T_{cm}}{w_2T_{cp}} + 1} \tag{5.10}$$

The utilization of processor P_2 is

$$U_2 = \frac{1}{1 + (\frac{\alpha_2}{\alpha_2} + \frac{\alpha_3}{\alpha_2} + \frac{\alpha_4}{\alpha_2} + \frac{\alpha_5}{\alpha_2})\frac{z_1T_{cm}}{w_2T_{cm}} + (\frac{\alpha_3}{\alpha_2} + \frac{\alpha_4}{\alpha_2} + \frac{\alpha_5}{\alpha_2})\frac{z_2T_{cm}}{w_2T_{cp}}} \tag{5.11}$$

The Total time of Processor P_3 is given by

$$\begin{aligned} T_{f3} = \alpha_2z_1T_{cm} + \alpha_3z_1T_{cm} + \alpha_4z_1T_{cm} + \alpha_5z_1T_{cm} \\ + \alpha_3z_2T_{cm} + \alpha_4z_{2cm} \quad + \alpha_5z_2T_{cm} \\ + \alpha_4z_3T_{cm} + \alpha_5z_3 + \alpha_3w_3T_{cp} \end{aligned} \tag{5.12}$$

$$U_3 = \frac{\alpha_3w_3T_{cm}}{T_{f3}} \tag{5.13}$$

$$U_3 = \frac{\alpha_3w_3T_{cp}}{(\alpha_2+\alpha_3+\alpha_4+\alpha_5)z_1T_{cm} + (\alpha_3+\alpha_4+\alpha_5)z_2T_{cm} + (\alpha_4+\alpha_5)z_3T_{cm} + \alpha_3w_3T_{cp}} \tag{5.14}$$

$$U_3 = \frac{\frac{\alpha_3w_3T_{cp}}{\alpha_3w_3T_{cp}}}{\frac{(\alpha_2+\alpha_3+\alpha_4+\alpha_5)}{\alpha_3}\frac{z_1T_{cm}}{w_3T_{cp}} + \frac{(\alpha_3+\alpha_4+\alpha_5)}{\alpha_3}\frac{z_2T_{cm}}{w_3T_{cp}} + \frac{(\alpha_4+\alpha_5)}{\alpha_3}\frac{z_3T_{cm}}{w_3T_{cp}} + \frac{\alpha_3w_3T_{cp}}{\alpha_3w_3T_{cp}}} \tag{5.15}$$

$$U_3 = \frac{1}{\frac{(\alpha_2+\alpha_3+\alpha_4+\alpha_5)}{\alpha_3}\frac{z_1T_{cm}}{w_3T_{cp}} + \frac{(\alpha_3+\alpha_4+\alpha_5)}{\alpha_3}\frac{z_2T_{cm}}{w_3T_{cp}} + \frac{(\alpha_4+\alpha_5)}{\alpha_3}\frac{z_3T_{cm}}{w_3T_{cp}} + 1} \tag{5.16}$$

The utilization of processor P_3 is

$$U_3 = \frac{1}{1 + (\frac{\alpha_2}{\alpha_3} + \frac{\alpha_3}{\alpha_3} + \frac{\alpha_4}{\alpha_3} + \frac{\alpha_5}{\alpha_3})\frac{z_1T_{cm}}{w_3T_{cp}} + (\frac{\alpha_3}{\alpha_3} + \frac{\alpha_4}{\alpha_3} + \frac{\alpha_5}{\alpha_3})\frac{z_2T_{cm}}{w_3T_{cp}} + (\frac{\alpha_4}{\alpha_3} + \frac{\alpha_5}{\alpha_3})\frac{z_3T_{cm}}{w_3T_{cp}}} \tag{5.17}$$

The total time of processor P$_4$ is given by

$$\begin{aligned} T_{f4} = \alpha_2 z_1 T_{cm} + \alpha_3 z_1 T_{cm} + \alpha_4 z_1 T_{cm} + \alpha_5 z_1 T_{cm} \\ + \alpha_3 z_2 T_{cm} + \alpha_4 z_{2cm} \quad + \alpha_5 z_2 T_{cm} \\ + \alpha_4 z_3 T_{cm} + \alpha_5 z_3 T_{cm} \\ + \alpha_5 z_4 Tc_m + \alpha_4 w_4 T_{cp} \end{aligned} \tag{5.18}$$

$$U_4 = \frac{\alpha_4 w_4 T_{cp}}{T_{f4}} \tag{5.19}$$

$$U_4 = \frac{\alpha_4 w_4 T_{cp}}{(\alpha_2+\alpha_3+\alpha_4+\alpha_5)z_1T_{cm} + (\alpha_3+\alpha_4+\alpha_5)z_2T_{cm} + (\alpha_4+\alpha_5)z_3T_{cm} + (\alpha_5)z_4T_{cm} + \alpha_4w_4T_{cp}} \tag{5.20}$$

$$U_4 = \frac{\frac{\alpha_4 w_4 T_{cp}}{\alpha_4 w_4 T_{cp}}}{\frac{(\alpha_2+\alpha_3+\alpha_4+\alpha_5)}{\alpha_4}\frac{z_1T_{cm}}{w_4T_{cp}} + \frac{(\alpha_3+\alpha_4+\alpha_5)}{\alpha_4}\frac{z_2T_{cm}}{w_4T_{cp}} + \frac{(\alpha_4+\alpha_5)}{\alpha_4}\frac{z_3T_{cm}}{w_4T_{cp}} + \frac{(\alpha_5)}{\alpha_4}\frac{z_4T_{cm}}{w_4T_{cp}} + \frac{\alpha_4w_4T_{cp}}{\alpha_4w_4T_{cp}}} \tag{5.21}$$

$$U_4 = \frac{1}{\frac{(\alpha_2+\alpha_3+\alpha_4+\alpha_5)}{\alpha_4}\frac{z_1T_{cm}}{w_4T_{cp}} + \frac{(\alpha_3+\alpha_4+\alpha_5)}{\alpha_4}\frac{z_2T_{cm}}{w_4T_{cp}} + \frac{(\alpha_4+\alpha_5)}{\alpha_4}\frac{z_3T_{cm}}{w_4T_{cp}} + \frac{(\alpha_5)}{\alpha_4}\frac{z_4T_{cm}}{w_4T_{cp}} + 1} \tag{5.22}$$

The utilization of processor P$_4$ is

$$U_4 = \frac{1}{1 + (\frac{\alpha_2}{\alpha_4}+\frac{\alpha_3}{\alpha_4}+\frac{\alpha_4}{\alpha_4}+\frac{\alpha_5}{\alpha_4})\frac{z_1T_{cm}}{w_4T_{cp}} + (\frac{\alpha_3}{\alpha_4}+\frac{\alpha_4}{\alpha_4}+\frac{\alpha_5}{\alpha_4})\frac{z_2T_{cm}}{w_4T_{cp}} + (\frac{\alpha_4}{\alpha_4}+\frac{\alpha_5}{\alpha_4})\frac{z_3T_{cm}}{w_4T_{cp}} + (\frac{\alpha_5}{\alpha_4})\frac{z_4T_{cm}}{w_4T_{cp}}} \tag{5.23}$$

The total time of processor P$_5$ is

$$\begin{aligned} T_{f5} = \alpha_2 z_1 T_{cm} + \alpha_3 z_1 T_{cm} + \alpha_4 z_1 T_{cm} + \alpha_5 z_1 T_{cm} \\ + \alpha_3 z_2 T_{cm} + \alpha_4 z_{2cm} \quad + \alpha_5 z_2 T_{cm} \\ + \alpha_4 z_3 T_{cm} + \alpha_5 z_3 T_{cm} \\ + \alpha_5 z_4 T_{cm} + \alpha_5 w_5 T_{cp} \end{aligned} \tag{5.24}$$

$$U_5 = \frac{\alpha_5 w_5 T_{cp}}{T_{f5}} \tag{5.25}$$

$$U_5 = \frac{\alpha_5 w_5 T_{cp}}{(\alpha_2+\alpha_3+\alpha_4+\alpha_5)z_1T_{cm} + (\alpha_3+\alpha_4+\alpha_5)z_2T_{cm} + (\alpha_4+\alpha_5)z_3T_{cm} + (\alpha_5)z_4T_{cm} + \alpha_5 w_5 T_{cp}} \quad (5.26)$$

$$U_5 = \frac{\frac{\alpha_5 w_5 T_{cp}}{\alpha_5 w_5 T_{cp}}}{\frac{(\alpha_2+\alpha_3+\alpha_4+\alpha_5)}{\alpha_5}\frac{z_1T_{cm}}{w_5T_{cp}} + \frac{(\alpha_3+\alpha_4+\alpha_5)}{\alpha_5}\frac{z_2T_{cm}}{w_5T_{cp}} + \frac{(\alpha_4+\alpha_5)}{\alpha_5}\frac{z_3T_{cm}}{w_4T_{cp}} + \frac{(\alpha_5)}{\alpha_5}\frac{z_4T_{cm}}{w_4T_{cp}} + \frac{\alpha_5 w_5 T_{cp}}{\alpha_5 w_5 T_{cp}}} \quad (5.27)$$

$$U_5 = \frac{1}{\frac{(\alpha_2+\alpha_3+\alpha_4+\alpha_5)}{\alpha_5}\frac{z_1T_{cm}}{w_5T_{cm}} + \frac{(\alpha_3+\alpha_4+\alpha_5)}{\alpha_5}\frac{z_2T_{cm}}{w_5T_{cp}} + \frac{(\alpha_4+\alpha_5)}{\alpha_5}\frac{z_3T_{cm}}{w_4T_{cp}} + \frac{(\alpha_5)}{\alpha_5}\frac{z_4T_{cm}}{w_4T_{cp}} + 1} \quad (5.28)$$

The utilization of processor P_5 is given by

$$U_5 = \frac{1}{1 + \left(\frac{\alpha_2}{\alpha_5}+\frac{\alpha_3}{\alpha_5}+\frac{\alpha_4}{\alpha_5}+\frac{\alpha_5}{\alpha_5}\right)\frac{z_1T_{cm}}{w_5T_{cm}} + \left(\frac{\alpha_3}{\alpha_5}+\frac{\alpha_4}{\alpha_5}+\frac{\alpha_5}{\alpha_5}\right)\frac{z_2T_{cm}}{w_5T_{cp}} + \left(\frac{\alpha_4}{\alpha_5}+\frac{\alpha_5}{\alpha_5}\right)\frac{z_3T_{cm}}{w_5T_{cp}} + \left(\frac{\alpha_5}{\alpha_5}\right)\frac{z_4T_{cm}}{w_5T_{cp}}} \quad (5.29)$$

In chapter 3 the ratios of the alpha's the k's, are found by writing an expressing based on the timing diagram. The alpha's are isolated and set equal to the remaining parameters in the expression. They are the link speed $\left(\frac{1}{z}\right)$, processor speeds $\left(\frac{1}{w}\right)$, and the intensity parameters associated with the link speed and the processor speed, T_{cm} and T_{cp} respectively. This ratio is then set equal to a parameter k. It is possible to reduce all expressions in the timing diagrams of chapter 3 to a ratio of two alpha's and a k. However in Figure 5.2 there are more than two alpha's in all but one of the expression that can be derived from the timing diagram. This one expression is a result of the computation times of processors P_4 and P_5 being equal.

From Figure 5.2 processors P_4 and P_5 yield

$$\alpha_4 w_4 T_{cp} = \alpha_5 w_5 T_{cp} \tag{5.30}$$

$$\frac{\alpha_4}{\alpha_5} = \frac{w_5 T_{cp}}{w_4 T_{cp}} = \frac{w_5}{w_4} = k_5 \tag{5.31}$$

From processors P_3 and P_4

$$\alpha_3 w_3 Tc_p = \alpha_5 z_4 T_{cm} + \alpha_4 w_4 T_{cp} \tag{5.32}$$

$$\frac{\alpha_3 w_3 Tc_p}{\alpha_4 w_4 T_{cp}} = \frac{\alpha_5 z_4 T_{cm}}{\alpha_4 w_4 T_{cp}} + \frac{\alpha_4 w_4 T_{cp}}{\alpha_4 w_4 T_{cp}} \tag{5.33}$$

$$\frac{\alpha_3}{\alpha_4} = \left(\frac{\alpha_5}{\alpha_4} z_4\right) \frac{T_{cm}}{w_3 T_{cp}} + \frac{w_4}{w_3} = k_4 \tag{5.34}$$

From processors P_2 and P_3

$$\alpha_2 w_2 T_{cp} = \alpha_4 z_3 T_{cm} + \alpha_5 z_4 T_{cm} + \alpha_3 w_3 T_{cp} \tag{5.35}$$

$$\frac{\alpha_2 w_2 T_{cp}}{\alpha_3 w_2 T_{cp}} = \frac{\alpha_4 z_3 T_{cm}}{\alpha_3 w_2 T_{cp}} + \frac{\alpha_5 z_4 T_{cm}}{\alpha_3 w_2 T_{cp}} + \frac{\alpha_3 w_3 T_{cp}}{\alpha_3 w_2 T_{cp}} \tag{5.36}$$

$$\frac{\alpha_2}{\alpha_3} = \left(\frac{\alpha_4}{\alpha_3} z_3 + \frac{\alpha_5}{\alpha_3} z_4\right) \frac{T_{cm}}{w_2 T_{cp}} + \frac{w_2}{w_2} = k_3 \tag{5.37}$$

Rewriting equations (5.36), (5.33) and (5.30) yields

$$\alpha_2 = k_3 \alpha_3 \tag{5.38}$$
$$\alpha_3 = k_4 \alpha_4 \tag{5.39}$$
$$\alpha_4 = k_5 \alpha_5 \tag{5.40}$$

A general method for reproducing equations (5.37), (5.38), and (5.59) is

$$\alpha_i = k_{i+1}\alpha_{i+1} \qquad \text{for } i = 2, ..., N-1 \tag{5.41}$$

The general form for finding the k's is

$$k_{i+1} = \sum_{m=i+2}^{N} \left(\frac{\alpha_m}{\alpha_{i+1}} z_{m-1} \right) \frac{T_{cm}}{w_i T_{cp}} + \frac{w_{i+1}}{w_i} = \frac{\alpha_i}{\alpha_{i+1}} \quad \text{for } i = 2, 3, ..., N-1 \tag{5.42}$$

The ratio of the loads on the last two processors in the network is found first, in equation (5.30). The inverse processors speeds w_i is known so the exact value of k_5 known. Using an iterative process the values of the rest of the k's in the network are found.

In a network with N = 5 processors , with i = N - 1 to 2, w_i = w and z_i = z the values of k_{i+1} are :

$$\text{for } i = N-1, ..., 2$$

$$k_{4+1} = k_5 = \sum_{m=i+2=6}^{N=5} \left(\frac{\alpha_m}{\alpha_{i+1}} z_{m-1} \right) \frac{T_{cm}}{w_i T_{cp}} + \frac{w_{i+1}}{w_i} = \frac{\alpha_i}{\alpha_{i+1}} \tag{5.43}$$

$$= \frac{w_5}{w_4} = \frac{\alpha_4}{\alpha_5}$$

From equation (5.30) $k_5 = \frac{\alpha_4}{\alpha_5}$

The inverse is placed in the next equation, giving a numerical value for

$$k_4 = \frac{\alpha_3}{\alpha_4}$$

$$k_{3+1} = k_4 = \sum_{m=i+2=5}^{N=5} \left(\frac{\alpha_m}{\alpha_{i+1}} z_{m-1}\right) \frac{T_{cm}}{w_i T_{cp}} + \frac{w_{i+1}}{w_i} = \frac{\alpha_i}{\alpha_{i+1}}$$

$$= \left(\frac{\alpha_5}{\alpha_4} z_4\right) \frac{T_{cm}}{w_3 T_{cp}} + \frac{w_4}{w_3} = \frac{\alpha_3}{\alpha_4} \tag{5.44}$$

From equation (5.33) $k_4 = \frac{\alpha_3}{\alpha_4}$

$$k_{2+1} = k_3 = \sum_{m=i+2=4}^{N=5} \left(\frac{\alpha_m}{\alpha_{i+1}} z_{m-1}\right) \frac{T_{cm}}{w_i T_{cp}} + \frac{w_{i+1}}{w_i} = \frac{\alpha_i}{\alpha_{i+1}}$$

$$= \left(\frac{\alpha_4}{\alpha_3} z_3 + \frac{\alpha_5}{\alpha_3} z_4\right) \frac{T_{cm}}{w_2 T_{cp}} + \frac{w_3}{w_2} = \frac{\alpha_2}{\alpha_3} \tag{5.45}$$

From equation (5.36) $k_3 = \frac{\alpha_2}{\alpha_3}$

Using equations (5.38) and (5.39) $\alpha_3 = k_4\alpha_4 = k_4k_5\alpha_5$, and $\frac{\alpha_5}{\alpha_3} = \frac{1}{k_4k_5}$

both k_4 and k_5 are known from the previous equations.

The general expression for Utilization is

$$U_i = \frac{1}{1 + \sum_{j=2}^{i+1} \left(\sum_{m=j}^{N} \left(\frac{\alpha_m}{\alpha_i}\right) \frac{z_{j-1} T_{cm}}{w_i T_{cp}}\right)} \qquad \text{for } i = 2, 3, ..., N \tag{5.46}$$

Note at N = 5 the outer summation's upper index is 6. This means that m in the inner summation takes on values from 2 to 6. When the value of m = 6 the summation will equal zero and the process will terminate.

The conditions for U_i are as follows:

For the ratio $\left(\frac{\alpha_m}{\alpha_i}\right)$ found in the summation of equation (5.45)

1. if $\alpha_m = \alpha_i \quad \rightarrow \quad \left(\frac{\alpha_m}{\alpha_i}\right) = 1$
2. if $\alpha_{i+2} \geq \alpha_m \quad \rightarrow \quad \left(\frac{\alpha_m}{\alpha_i}\right) = \prod_{c=i+1}^{m} \frac{1}{k_c}$
3. if $\alpha_{m+2} \geq \alpha_i \quad \rightarrow \quad \left(\frac{\alpha_m}{\alpha_i}\right) = \prod_{c=m+1}^{i} k_c$

Equation(5.45) for i = 2 to N, produces equations (5.10), (5.16), (5.22) and (5.28). They are reproduced on the first line of equations (5.46) to (5.48) for comparison. Applying the above conditions produces the second line in equations (5.46) to (5.48).

$$\begin{aligned} U_2 &= \frac{1}{1 + (\frac{\alpha_2}{\alpha_2} + \frac{\alpha_3}{\alpha_2} + \frac{\alpha_4}{\alpha_2} + \frac{\alpha_5}{\alpha_2})\frac{z_1 T_{cm}}{w_2 T_{cp}}} \\ &= \frac{1}{1 + (1 + \frac{1}{k_3} + \frac{1}{k_3 k_4} + \frac{1}{k_3 k_4 k_5})\frac{z_1 T_{cm}}{w_2 T_{cp}}} \end{aligned} \tag{5.47}$$

$$\begin{aligned} U_3 &= \frac{1}{1 + (\frac{\alpha_2}{\alpha_3} + \frac{\alpha_3}{\alpha_3} + \frac{\alpha_4}{\alpha_3} + \frac{\alpha_5}{\alpha_3})\frac{z_1 T_{cm}}{w_3 T_{cp}} + (\frac{\alpha_3}{\alpha_3} + \frac{\alpha_4}{\alpha_3} + \frac{\alpha_5}{\alpha_3})\frac{z_2 T_{cm}}{w_3 T_{cp}} + (\frac{\alpha_4}{\alpha_3} + \frac{\alpha_5}{\alpha_3})\frac{z_3 T_{cm}}{w_3 T_{cp}}} \\ &= \frac{1}{1 + (k_3 + 1 + \frac{1}{k_4} + \frac{1}{k_4 k_5})\frac{z_1 T_{cm}}{w_3 T_{cp}} + (1 + \frac{1}{k_4} + \frac{1}{k_4 k_5})\frac{z_2 T_{cm}}{w_3 T_{cp}} + (\frac{1}{k_4} + \frac{1}{k_4 k_5})\frac{z_3 T_{cm}}{w_3 T_{cp}}} \end{aligned} \tag{5.48}$$

$$
\begin{aligned}
U_4 &= \frac{1}{1+\left(\frac{\alpha_2}{\alpha_4}+\frac{\alpha_3}{\alpha_4}+\frac{\alpha_4}{\alpha_4}+\frac{\alpha_5}{\alpha_4}\right)\frac{z_1T_{cm}}{w_4T_{cp}}+\left(\frac{\alpha_3}{\alpha_4}+\frac{\alpha_4}{\alpha_4}+\frac{\alpha_5}{\alpha_4}\right)\frac{z_2T_{cm}}{w_4T_{cp}}+\left(\frac{\alpha_4}{\alpha_4}+\frac{\alpha_5}{\alpha_4}\right)\frac{z_3T_{cm}}{w_4T_{cp}}+\left(\frac{\alpha_5}{\alpha_4}\right)\frac{z_4T_{cm}}{w_4T_{cp}}} \\
&= \frac{1}{1+\left(k_3k_4+k_4+1+\frac{1}{k_5}\right)\frac{z_1T_{cm}}{w_4T_{cp}}+\left(k_4+1+\frac{1}{k_5}\right)\frac{z_2T_{cm}}{w_4T_{cp}}+\left(1+\frac{1}{k_5}\right)\frac{z_3T_{cm}}{w_4T_{cp}}+\left(\frac{1}{5}\right)\frac{z_4T_{cm}}{w_4T_{cp}}}
\end{aligned}
\tag{5.49}
$$

$$
\begin{aligned}
U_5 &= \frac{1}{1+\left(\frac{\alpha_2}{\alpha_5}+\frac{\alpha_3}{\alpha_5}+\frac{\alpha_4}{\alpha_5}+\frac{\alpha_5}{\alpha_5}\right)\frac{z_1T_{cm}}{w_5T_{cp}}+\left(\frac{\alpha_3}{\alpha_5}+\frac{\alpha_4}{\alpha_5}+\frac{\alpha_5}{\alpha_5}\right)\frac{z_2T_{cm}}{w_5T_{cp}}+\left(\frac{\alpha_4}{\alpha_5}+\frac{\alpha_5}{\alpha_5}\right)\frac{z_3T_{cm}}{w_5T_{cp}}+\left(\frac{\alpha_5}{\alpha_5}\right)\frac{z_4T_{cm}}{w_5T_{cp}}} \\
&= \frac{1}{1+\left(k_3k_4k_5+k_4k_5+k_5+1\right)\frac{z_1T_{cm}}{w_5T_{cp}}+\left(k_4k_5+k_5+1\right)\frac{z_2T_{cm}}{w_5T_{cp}}+\left(k_5+1\right)\frac{z_3T_{cm}}{w_5T_{cp}}+\left(1\right)\frac{z_4T_{cm}}{w_5T_{cp}}}
\end{aligned}
\tag{5.50}
$$

The utilizations for N = 5 and N = 22 are plotted in Figure 5.3 and Figure 5.4 respectively.

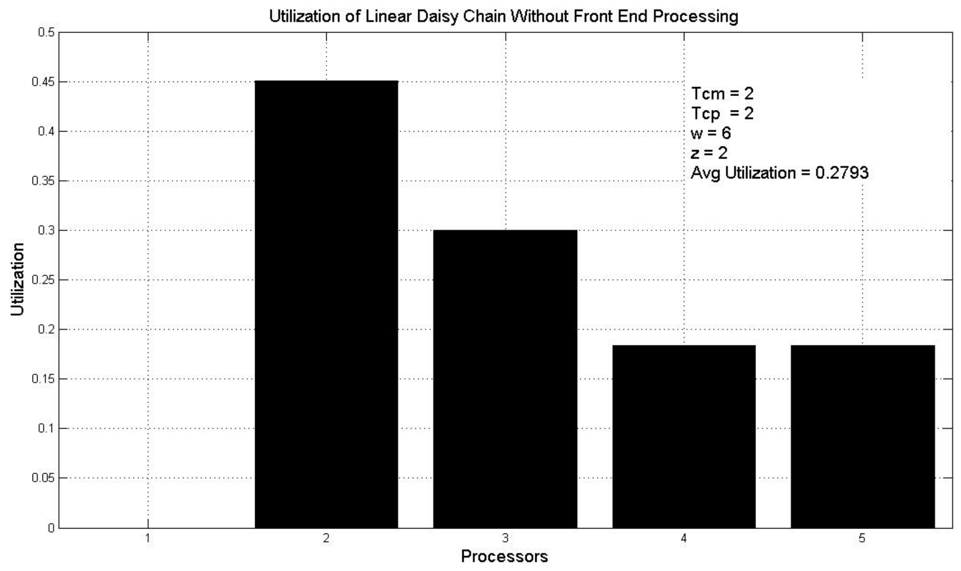


Figure 5.3. The utilization of a linear daisy chain without front end processing

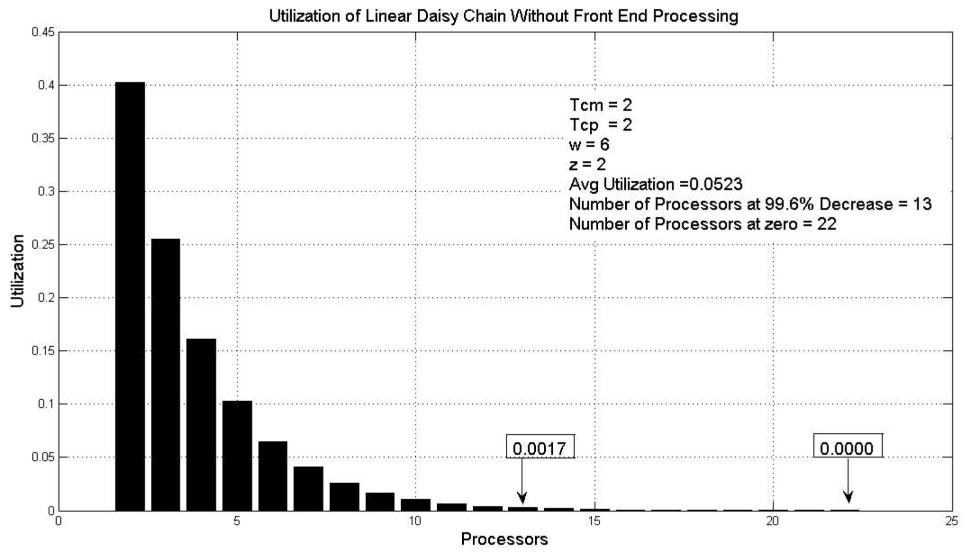


Figure 5.4. The utilization of a linear daisy chain without front end processing. At N = 13 the utilization has decrease 99.6% from its original value . At N = 22 the utilization is zero.

The speedup for the network is found by taking the ratio of the time it takes to finish a computation on one processor divided by the time it takes to complete computations on the entire network.

T_P : is the time to complete computations on a single processor

T_{PN} : is the time to complete computations on an entire network

$$speedup = \frac{T_P}{T_{PN}} \tag{5.51}$$

Using equations (5.37), (5.38) and (5.39) repeated below respectively

$$\alpha_2 = k_3\alpha_3$$
$$\alpha_3 = k_4\alpha_4$$
$$\alpha_4 = k_5\alpha_5$$

Rewriting them yields

$$\alpha_3 = \frac{1}{k_3} \tag{5.52}$$

$$\alpha_4 = \frac{1}{k_4}\alpha_3 \tag{5.53}$$

$$\alpha_5 = \frac{1}{k_5}\alpha_4 \tag{5.54}$$

A general expression is found for α_i

$$\alpha_i = \prod_{m=3}^{i} \frac{1}{k_m}\alpha_{i-(i-2)} \qquad \text{for } i = 3, 4, ..., N \tag{5.55}$$

$$\alpha_{i-(i-2)} = \alpha_2 \tag{5.56}$$

For the above conditions in equation (5.54)

The normalized equation for this network is

$$\alpha_2 + \alpha_3 + ... + \alpha_N = 1 \tag{5.57}$$

The load on the first processor $\alpha_1 = 0$, because the processor P_1 does not retain any of the load, it transmits the entire load to processor P_2. rewriting the normalized equation (5.56)

$$\alpha_2 + \sum_{i=3}^{N} \alpha_i = 1 \tag{5.58}$$

Substituting equations (5.54) and (5.55) in equation (5.56) for α_i

$$\alpha_2 + \sum_{i=3}^{N} \prod_{m=3}^{i} \frac{1}{k_m}\alpha_2 = 1 \qquad \text{for } i = 3, ..., N \tag{5.59}$$

$$\alpha_2 \left[1 + \sum_{i=3}^{N} \prod_{m=3}^{i} \frac{1}{k_m} \right] = 1 \tag{5.60}$$

The value of α_2 for this network is

$$\alpha_2 = \frac{1}{\left[1 + \sum_{i=3}^{N} \prod_{m=3}^{i} \frac{1}{k_m} \right]} \qquad \text{for } i = 3, 4, ..., N \tag{5.61}$$

The time to perform computations on a single processor is

$$T_{P2} = \alpha_2 w_2 T_{cp} \tag{5.62}$$

For $\alpha_2 = 1$ and $w_i = w$ equation (5.61) becomes

$$T_{P2} = w T_{cp} \tag{5.63}$$

The time to perform computations on the entire network is

$$T_{PN} = \alpha_2 w_2 T_{cp} \tag{5.64}$$

When $w_i = w$ equation (5.63) becomes

$$T_{PN} = \alpha_2 w T_{cp} \tag{5.65}$$

$$speedup = \frac{T_{P2}}{T_{PN}} = \frac{w T_{cp}}{\alpha_2 w T_{cp}} = \frac{1}{\alpha_2} \tag{5.66}$$

$$speedup = \left[1 + \sum_{i=3}^{N} \prod_{m=3}^{i} \frac{1}{k_m} \right] \qquad \text{for } i = 3, 4, ..., N \tag{5.67}$$

Speedup is plotted versus the number of processors, Figure 5.5. A network of N = 22 is used because this is the number of processors used for utilization in Figure 5.4. The speedup saturates at 2.7321. The number of processors that can be added to a network and still receive a benefit from parallel processing is found by taking 99.6% of the saturation value. In this case it is 99.6% of 2.731 this equates to 2.27. The number of the processor this occurs at is N = 11.

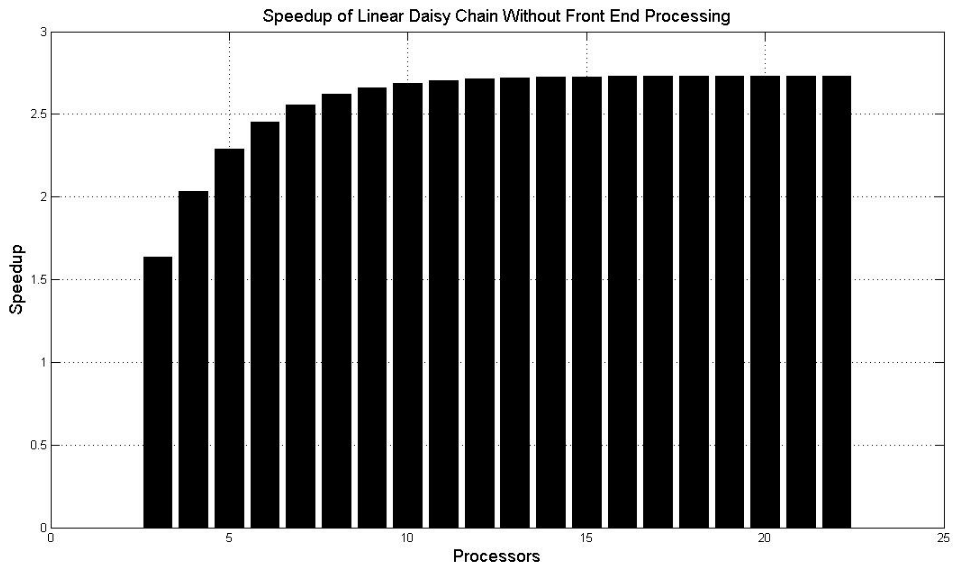


Figure 5.5. Speedup of a linear daisy chain without front end processing

5.3 UTILIZATION OF LINEAR DAISY CHAIN USING A HYBRID STRATEGY WITH FRONT END PROCESSING

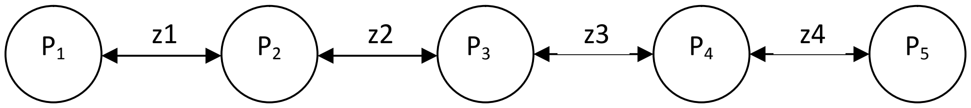


Figure 5.6. Five processors connected in a linear daisy chain configuration

The linear daisy configuration of Figure 5.6 and the timing diagram of Figure 5.7, implement a hybrid strategy. In that they contain aspects of both staggered start and simultaneous start. Behavior characterized by a simultaneous start is that the processor can transmit and process information. This is a front end property. They have staggered start behavior because the processors can receive information but cannot process the information at the same time. This describes a processor without front end processing.

In Figure 5.6 the loads on processor P_1 are α_1, α_2, α_3, α_4, and α_5. Absorption (processing) of α_1 by processor P_1 starts immediately because there is no need to transmit α_1 over a link. At the same time processor P_1 is absorbing α_1, it sequentially distributes the loads α_2, α_3, α_4, and α_5 by transmitting them over link z_1 to processor P_2. When all the loads have been received by processor P_2 absorption of α_2 can begin. The loads α_3, α_4, and α_5 are transmitted over link z_2 to processor P_3 at the same time absorption starts. After the arrival of all loads at processor P_3 absorption of α_3 begins, and the transmission of α_4 and α_5 over link z_3 to processor P_4 starts. When α_4 and α_5 have arrived at processor P_4, the absorption of α_4 starts ,and α_5 is transmitted to processor P_5 over link z_4. Upon complete arrival of α_5 at processor P_5 absorption of α_5 by processor P_5 starts.

The timing diagram if Figure 5.7 is an alternative representation of the linear daisy chain configuration. The description is the similar. Processor P_1 absorption of α_1 starts immediately , the loads α_2, α_3, α_4, and α_5 are sequentially distributed to processor P_2 over link z_2 by means of T_2. After the arrival of all loads at P_2, absorption of α_2 can start. When the absorption starts loads α_3, α_4, and α_5 are transmitted to processor P_3 over link z_3 by means of T_3. After all the loads have been received by processor P_3, absorption of α_3 starts. With the absorption of α_3, the remaining loads α_4, and α_5 are transmitter over link z_3 by mean of T_4 to processor P_4. After the complete arrival of α_4 and α_5 at Processor P_4, absorption of α_4 by processor P_4 starts. When absorption starts the final load α_5 is transmitted over link z_4 to processor P_5. After being received α_5 is absorbed by processor P_5. The communication delays of Figure 5.7 and the loads within them are listed below.

$$T_2 = (\alpha_2 + \alpha_3 + \alpha_4 + \alpha_5) z_1 T_{cm} \quad (5.68)$$
$$T_3 = (\alpha_3 + \alpha_4 + \alpha_5) z_2 T_{cm} \quad (5.69)$$
$$T_4 = (\alpha_4 + \alpha_5) z_3 T_{cm} \quad (5.70)$$
$$T_5 = (\alpha_5) z_4 T_{cm} \quad (5.71)$$

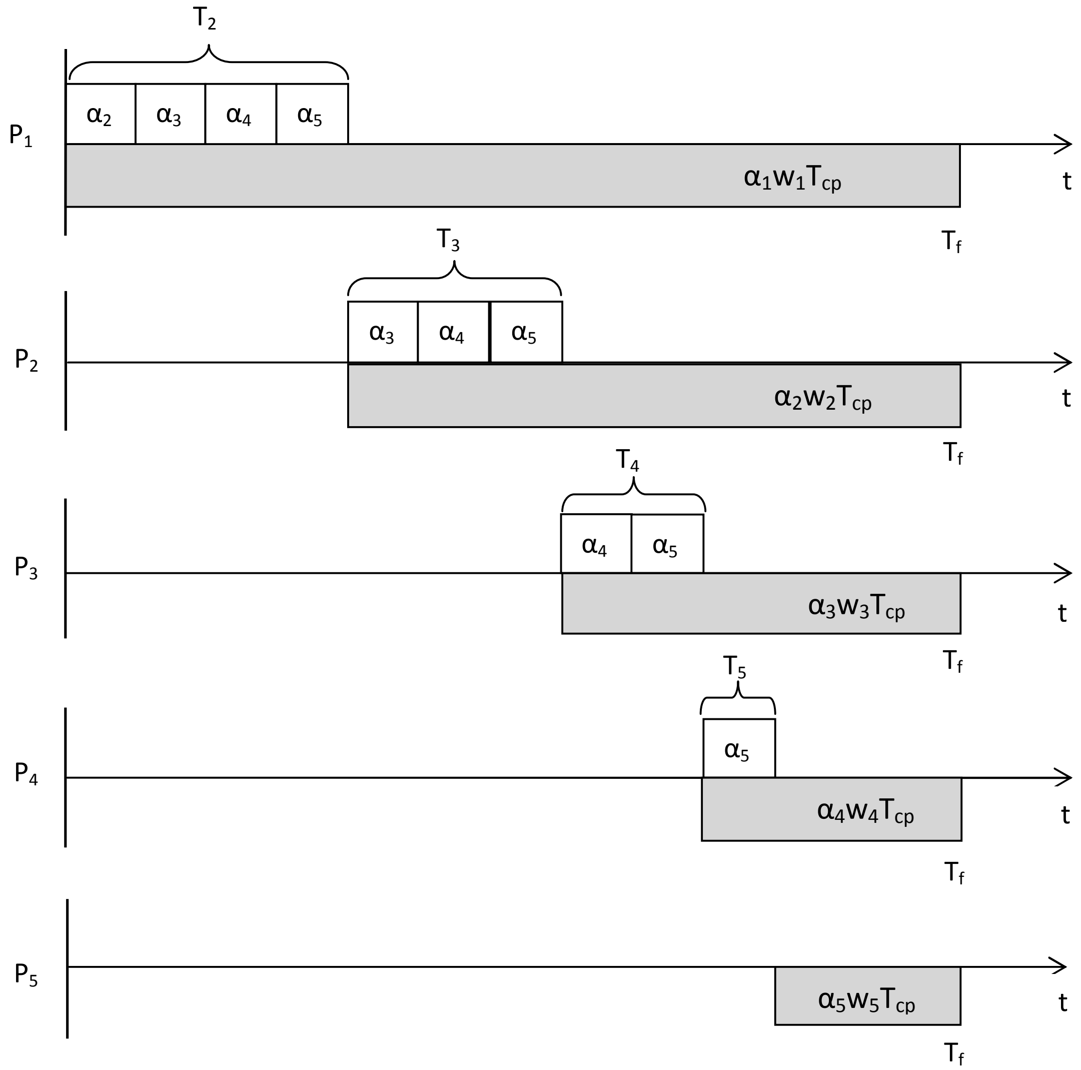


Figure 5.7. Timing diagram of a linear daisy chain that implements a hybrid strategy, and front end processing

The utilization of the linear daisy chain is calculated the same as in chapter 5.2, by finding the ratio of useful time to total time.

The total time is given by

$$T_{f2} = \alpha_2 z_1 T_{cm} + \alpha_3 z_1 T_{cm} + \alpha_4 z_1 T_{cm} + \alpha_5 z_1 T_{cm} + \alpha_2 w_2 T_{cp} \tag{5.72}$$

$$U_2 = \frac{\alpha_2 w_2 T_{cp}}{T_{f2}} \tag{5.73}$$

$$U_2 = \frac{\alpha_2 w_2 T_{cp}}{\alpha_2 z_1 T_{cm} + \alpha_3 z_1 T_{cm} + \alpha_4 z_1 T_{cm} + \alpha_5 z_1 T_{cm} + \alpha_2 w_2 T_{cp}} \tag{5.74}$$

$$U_2 = \frac{\frac{\alpha_2 w_2 T_{cp}}{\alpha_2 w_2 T_{cp}}}{\frac{(\alpha_2+\alpha_3+\alpha_4+\alpha_5)}{\alpha_2}\frac{z_1 T_{cm}}{w_2 T_{cp}} + \frac{\alpha_2 w_2 T_{cp}}{\alpha_2 w_2 T_{cp}}} \tag{5.75}$$

The utilization of processor P$_2$ is

$$U_2 = \frac{1}{1 + (\frac{\alpha_2}{\alpha_2} + \frac{\alpha_3}{\alpha_2} + \frac{\alpha_4}{\alpha_2} + \frac{\alpha_5}{\alpha_2})\frac{z_1 T_{cm}}{w_2 T_{cm}}} \tag{5.76}$$

The total time of processor 3 is given by

$$\begin{aligned} T_{f3} = {} & \alpha_2 z_1 T_{cm} + \alpha_3 z_1 T_{cm} + \alpha_4 z_1 T_{cm} + \alpha_5 z_1 T_{cm} \\ & + \alpha_3 z_2 T_{cm} + \alpha_4 z_{2cm} \quad + \alpha_5 z_2 T_{cm} + \alpha_3 w_3 T_{cp} \end{aligned} \tag{5.77}$$

$$U_3 = \frac{\alpha_3 w_3 T_{cm}}{T_{f3}} \tag{5.78}$$

$$U_3 = \frac{\alpha_3 w_3 T_{cp}}{(\alpha_2 + \alpha_3 + \alpha_4 + \alpha_5) z_1 T_{cm} + (\alpha_3 + \alpha_4 + \alpha_5) z_2 T_{cm} + \alpha_3 w_3 T_{cp}} \tag{5.79}$$

$$U_3 = \frac{\frac{\alpha_3 w_3 T_{cp}}{\alpha_3 w_3 T_{cp}}}{\frac{(\alpha_2+\alpha_3+\alpha_4+\alpha_5)}{\alpha_3}\frac{z_1 T_{cm}}{w_3 T_{cp}} + \frac{(\alpha_3+\alpha_4+\alpha_5)}{\alpha_3}\frac{z_2 T_{cm}}{w_3 T_{cp}} + \frac{\alpha_3 w_3 T_{cp}}{\alpha_3 w_3 T_{cp}}} \tag{5.80}$$

The utilization of Processor P_3 is

$$U_3 = \frac{1}{1 + (\frac{\alpha_2}{\alpha_2} + \frac{\alpha_3}{\alpha_2} + \frac{\alpha_4}{\alpha_2} + \frac{\alpha_5}{\alpha_2})\frac{z_1 T_{cm}}{w_3 T_{cm}} + (\frac{\alpha_3}{\alpha_2} + \frac{\alpha_4}{\alpha_2} + \frac{\alpha_5}{\alpha_2})\frac{z_2 T_{cm}}{w_3 T_{cp}}} \tag{5.81}$$

The total time for processor P_4 is given by

$$\begin{aligned} T_{f4} = \alpha_2 z_1 T_{cm} + \alpha_3 z_1 T_{cm} + \alpha_4 z_1 T_{cm} + \alpha_5 z_1 T_{cm} \\ + \alpha_3 z_2 T_{cm} + \alpha_4 z_{2cm} \quad + \alpha_5 z_2 T_{cm} \\ + \alpha_4 z_3 T_{cm} + \alpha_5 z_3 + \alpha_4 w_4 T_{cp} \end{aligned} \tag{5.82}$$

$$U_4 = \frac{\alpha_4 w_4 T_{cp}}{T_{f4}} \tag{5.83}$$

$$U_4 = \frac{\alpha_4 w_4 T_{cp}}{(\alpha_2+\alpha_3+\alpha_4+\alpha_5) z_1 T_{cm} + (\alpha_3+\alpha_4+\alpha_5) z_2 T_{cm} + (\alpha_4+\alpha_5) z_3 T_{cp} + \alpha_4 w_3 T_{cp}} \tag{5.84}$$

$$U_4 = \frac{\frac{\alpha_4 w_4 T_{cp}}{\alpha_4 w_4 T_{cp}}}{\frac{(\alpha_2+\alpha_3+\alpha_4+\alpha_5)}{\alpha_4}\frac{z_1 T_{cm}}{w_4 T_{cp}} + \frac{(\alpha_3+\alpha_4+\alpha_5)}{\alpha_4}\frac{z_2 T_{cm}}{w_4 T_{cp}} + \frac{(\alpha_4+\alpha_5)}{\alpha_4}\frac{z_3 T_{cm}}{w_4 T_{cp}} + \frac{\alpha_4 w_4 T_{cp}}{\alpha_4 w_4 T_{cp}}} \tag{5.85}$$

The utilization of processor P_4 is

$$U_4 = \frac{1}{1 + (\frac{\alpha_2}{\alpha_4} + \frac{\alpha_3}{\alpha_4} + \frac{\alpha_4}{\alpha_4} + \frac{\alpha_5}{\alpha_4})\frac{z_1 T_{cm}}{w_4 T_{cp}} + (\frac{\alpha_3}{\alpha_4} + \frac{\alpha_4}{\alpha_4} + \frac{\alpha_5}{\alpha_4})\frac{z_2 T_{cm}}{w_4 T_{cp}} + (\frac{\alpha_4}{\alpha_4} + \frac{\alpha_5}{\alpha_4})\frac{z_3 T_{cm}}{w_4 T_{cp}}} \tag{5.86}$$

The total time of processor P_5 is given by

$$\begin{aligned} T_{f5} = \alpha_2 z_1 T_{cm} + \alpha_3 z_1 T_{cm} + \alpha_4 z_1 T_{cm} + \alpha_5 z_1 T_{cm} \\ + \alpha_3 z_2 T_{cm} + \alpha_4 z_{2cm} \quad + \alpha_5 z_2 T_{cm} \\ + \alpha_4 z_3 T_{cm} + \alpha_5 z_3 T_{cm} \\ + \alpha_5 z_4 T_{cm} + \alpha_5 w_5 T_{cp} \end{aligned} \tag{5.87}$$

$$U_5 = \frac{\alpha_5 w_5 T_{cp}}{T_{f5}} \tag{5.88}$$

$$U_5 = \frac{\alpha_5 w_5 T_{cp}}{(\alpha_2 + \alpha_3 + \alpha_4 + \alpha_5) z_1 T_{cm} + (\alpha_3 + \alpha_4 + \alpha_5) z_2 T_{cm} + (\alpha_4 + \alpha_5) z_3 T_{cm} + (\alpha_5) z_4 T_{cm} + \alpha_5 w_5 T_{cp}} \tag{5.89}$$

$$U_5 = \frac{\frac{\alpha_5 w_5 T_{cp}}{\alpha_5 w_5 T_{cp}}}{\frac{(\alpha_2+\alpha_3+\alpha_4+\alpha_5)}{\alpha_5} \frac{z_1 T_{cm}}{w_5 T_{cp}} + \frac{(\alpha_3+\alpha_4+\alpha_5)}{\alpha_5} \frac{z_2 T_{cm}}{w_5 T_{cp}} + \frac{(\alpha_4+\alpha_5)}{\alpha_5} \frac{z_3 T_{cm}}{w_4 T_{cp}} + \frac{(\alpha_5)}{\alpha_5} z_4 T_{cm} + \frac{\alpha_5 w_5 T_{cp}}{\alpha_5 w_5 T_{cp}}} \tag{5.90}$$

The utilization of P$_5$ is

$$U_5 = \frac{1}{1 + \left(\frac{\alpha_2}{\alpha_5} + \frac{\alpha_3}{\alpha_5} + \frac{\alpha_4}{\alpha_5} + \frac{\alpha_5}{\alpha_5}\right) \frac{z_1 T_{cm}}{w_5 T_{cm}} + \left(\frac{\alpha_3}{\alpha_5} + \frac{\alpha_4}{\alpha_5} + \frac{\alpha_5}{\alpha_5}\right) \frac{z_2 T_{cm}}{w_5 T_{cp}} + \left(\frac{\alpha_4}{\alpha_5} + \frac{\alpha_5}{\alpha_5}\right) \frac{z_3 T_{cm}}{w_5 T_{cp}} + \left(\frac{\alpha_5}{\alpha_5}\right) \frac{z_5 T_{cm}}{w_5 T_{cp}}} \tag{5.91}$$

The procedure for finding the ratio of the alpha’s the k’s is the same as in the previous section 5.2. An expression is derived from the timing diagram. From Figure 5.7 processors P$_4$ and P$_5$

$$\alpha_4 w_4 T_{cp} = \alpha_5 z_{Tcm} + \alpha_5 w_5 T_{cp} \tag{5.92}$$

$$\frac{\alpha_4 w_4 T_{cp}}{\alpha_5 w_4 T_{cp}} = \frac{\alpha_5 z_4 T_{cm}}{\alpha_5 w_4 T_{cp}} + \frac{\alpha_5 w_5 T_{cp}}{\alpha_5 w_4 T_{cp}} \tag{5.93}$$

$$\frac{\alpha_4}{\alpha_5} = \left(\frac{\alpha_5}{\alpha_5}\right) \frac{z_4 T_{cm}}{w_4 T_{cp}} + \frac{w_5 T_{cp}}{w_4} = k_5 \tag{5.94}$$

From processors P$_3$ and P$_4$

$$\alpha_3 w_3 T_{cp} = \alpha_4 z_3 T_{cm} + \alpha_5 z_3 T_{cm} + \alpha_4 w_4 T_{cp} \tag{5.95}$$

$$\frac{\alpha_3 w_3 T_{cp}}{\alpha_4 w_4 T_{cp}} = \frac{\alpha_4 z_3 T_{cm}}{\alpha_4 w_3 T_{cp}} + \frac{\alpha_5 z_3 T_{cm}}{\alpha_4 w_3 T_{cp}} + \frac{\alpha_4 w_4 T_{cP}}{\alpha_4 w_3 T_{cP}} \tag{5.96}$$

$$\frac{\alpha_3}{\alpha_4} = \left(\frac{\alpha_4}{\alpha_4} + \frac{\alpha_5}{\alpha_4}\right) \frac{z_3 T_{cm}}{w_3 T_{cp}} + \frac{w_4}{w_3} = k_4 \tag{5.97}$$

From processor P_2 and P_3

$$\alpha_2 w_2 T_{cp} = \alpha_3 z_2 T_{cm} + \alpha_4 z_2 T_{cm} + \alpha_5 z_2 T_{cm} + \alpha_3 w_3 T_{cp} \tag{5.98}$$

$$\frac{\alpha_2 w_2 T_{cp}}{\alpha_3 w_2 T_{cp}} = \frac{\alpha_3 z_2 T_{cm}}{\alpha_3 w_2 T_{cp}} + \frac{\alpha_4 z_2 T_{cm}}{\alpha_3 w_2 T_{cp}} + \frac{\alpha_5 z_2 T_{cm}}{\alpha_3 w_2 T_{cp}} + \frac{\alpha_3 w_3 T_{cp}}{\alpha_3 w_2 T_{cp}} \tag{5.99}$$

$$\frac{\alpha_2}{\alpha_3} = \left(\frac{\alpha_3}{\alpha_3} + \frac{\alpha_4}{\alpha_3} + \frac{\alpha_5}{\alpha_3}\right) \frac{z_2 T_{cm}}{w_2 T_{cp}} + \frac{w_3}{w_2} = k_3 \tag{5.100}$$

Rewriting equation (5.99), (5.96) and (5.93) yields

$$\alpha_2 = k_3 \alpha_3 \tag{5.101}$$
$$\alpha_3 = k_4 \alpha_4 \tag{5.102}$$
$$\alpha_4 = k_5 \alpha_5 \tag{5.103}$$

Equation (5.40) is general method for reproducing these equations

repeated below

$$\alpha_i = k_{i+1} \alpha_{i+1} \qquad \text{for } i = 2, ..., N-1 \tag{5.104}$$

Here $i = 1, ..., N-1$ can also be used, but there is no need to find $\alpha_1 = k_2 \alpha_2$. The utilization is one for processor P_1. This can be seen from Figure 5.7, the computation time goes from zero the finish time Tf, 100%.

The general form for finding the k's is developed using equation (5.93), (5.96) and (5.99).

$$k_{i+1} = \sum_{m=i+1}^{N} \left(\frac{\alpha_m}{\alpha_{i+1}}\right) \frac{z_i T_{cm}}{w_i T_{cp}} + \frac{w_{i+1}}{w_i} = \frac{\alpha_i}{\alpha_{i+1}} \qquad \text{for } i = 2, 3, ..., N-1 \tag{5.105}$$

As in section 5.2 the ratio of the loads on the last two processors in the network is found. In this section it is equation (5.93). The inverse processor speeds (w_i) are known so the exact value of k_5 is known. Using the same iterative process as is 5.2 the remaining k's are found.

In a network with N = 5 processors, with i = N-1 to 2, w_i = w and z_i=z, and know values of T_{cm} and T_{cp} the values of the k_{i+1}'s are:

$$\text{for } i = N-1, ..., 2$$

$$k_{4+1} = k_5 = \sum_{m=i+1=5}^{N=5} \left(\frac{\alpha_m}{\alpha_{i+1}}\right) \frac{z_i T_{cm}}{w_i T_{cp}} + \frac{w_{i+1}}{w_i} = \frac{\alpha_i}{\alpha_{i+1}} \tag{5.106}$$

$$= \left(\frac{\alpha_5}{\alpha_5}\right) \frac{z_4 T_{cm}}{w_4 T_{cp}} + \frac{w_5}{w_4} = \frac{\alpha_4}{\alpha_5}$$

The inverse of $\frac{\alpha_4}{\alpha_5}$ is placed in the next equation change

$$k_{3+1} = k_4 = \sum_{m=i+1=4}^{N=5} \left(\frac{\alpha_m}{\alpha_{i+1}}\right) \frac{z_i T_{cm}}{w_i T_{cp}} + \frac{w_{i+1}}{w_i} = \frac{\alpha_i}{\alpha_{i+1}}$$

$$= \left(\frac{\alpha_4}{\alpha_4} + \frac{\alpha_5}{\alpha_4}\right) \frac{z_3 T_{cm}}{w_3 T_{cp}} + \frac{w_4}{w_3} = \frac{\alpha_3}{\alpha_4} \tag{5.107}$$

The inverse of $\frac{\alpha_3}{\alpha_4}$ is placed in the next equation

$$k_{2+1} = k_3 = \sum_{m=i+1=3}^{N=5} \left(\frac{\alpha_m}{\alpha_{i+1}}\right) \frac{z_4 T_{cm}}{w_i T_{cp}} + \frac{w_{i+1}}{w_i} = \frac{\alpha_i}{\alpha_{i+1}}$$

(5.108)

$$= \left(\frac{\alpha_3}{\alpha_3} + \frac{\alpha_4}{\alpha_3} + \frac{\alpha_5}{\alpha_3}\right) \frac{z_2 T_{cm}}{w_2 T_{cp}} + \frac{w_3}{w_2} = \frac{\alpha_2}{\alpha_3}$$

From equation (5.101), $\alpha_3 = k_4\alpha_4$ and (5.102) $\alpha_4 = k_5\alpha_5$

$$\alpha_3 = k_4\alpha_4 = k_4 k_5 \alpha_5 \tag{5.109}$$

$$\frac{\alpha_5}{\alpha_3} = \frac{1}{k_4 k_5} \tag{5.110}$$

Since all other parameters in equation(5.107) are known, and k_4 and k_5 have been calculated above $k_3 = \frac{\alpha_2}{\alpha_3}$ is also known.

The general expression for utilization is

$$U_i = \frac{1}{1 + \sum_{j=2}^{i} \left(\sum_{m=j}^{N} \left(\frac{\alpha_m}{\alpha_i}\right) \frac{z_{j-1} T_{cm}}{w_i T_{cp}} \right)} \qquad \text{for } i = 2, 3, ..., N \tag{5.111}$$

This is the same as equation (5.45), except for a change in the upper index of summation on the outer summation, $i+1$ has been change to i.

The conditions for Ui are the same as in section 5.2

For the ratio $\left(\frac{\alpha_m}{\alpha_i}\right)$ found in the summation of equation (5.110)

1. if $\alpha_m = \alpha_i \quad \rightarrow \quad \left(\frac{\alpha_m}{\alpha_i}\right) = 1$

2. if $\alpha_{i+2} \geq \alpha_m \quad \rightarrow \quad \left(\frac{\alpha_m}{\alpha_i}\right) = \prod_{c=i+1}^{m} \frac{1}{k_c}$

3. if $\alpha_{m+2} \geq \alpha_i \quad \rightarrow \quad \left(\frac{\alpha_m}{\alpha_i}\right) = \prod_{c=m+1}^{i} k_c$

Equation (5.110) for i=2 to N, produces equations (5.75), (5.80) and (5.85). They are reproduced on the first line of equation(5.111) to (5.114) for comparison. Applying the above conditions produces the second line in equations (5.111) to (5.114).

$$\begin{aligned} U_2 &= \frac{1}{1 + (\frac{\alpha_2}{\alpha_2} + \frac{\alpha_3}{\alpha_2} + \frac{\alpha_4}{\alpha_2} + \frac{\alpha_5}{\alpha_2})\frac{z_1 T_{cm}}{w_2 T_{cp}}} \\ &= \frac{1}{1 + (1 + \frac{1}{k_3} + \frac{1}{k_3 k_4} + \frac{1}{k_3 k_4 k_5})\frac{z_1 T_{cm}}{w_2 T_{cp}}} \end{aligned} \tag{5.112}$$

$$\begin{aligned} U_3 &= \frac{1}{1 + (\frac{\alpha_2}{\alpha_3} + \frac{\alpha_3}{\alpha_3} + \frac{\alpha_4}{\alpha_3} + \frac{\alpha_5}{\alpha_3})\frac{z_1 T_{cm}}{w_3 T_{cp}} + (\frac{\alpha_3}{\alpha_3} + \frac{\alpha_4}{\alpha_3} + \frac{\alpha_5}{\alpha_3})\frac{z_2 T_{cm}}{w_3 T_{cp}}} \\ &= \frac{1}{1 + (k_3 + 1 + \frac{1}{k_4} + \frac{1}{k_4 k_5})\frac{z_1 T_{cm}}{w_3 T_{cp}} + (1 + \frac{1}{k_4} + \frac{1}{k_4 k_5})\frac{z_2 T_{cm}}{w_3 T_{cp}}} \end{aligned} \tag{5.113}$$

$$\begin{aligned} U_4 &= \frac{1}{1 + (\frac{\alpha_2}{\alpha_4} + \frac{\alpha_3}{\alpha_4} + \frac{\alpha_4}{\alpha_4} + \frac{\alpha_5}{\alpha_4})\frac{z_1 T_{cm}}{w_4 T_{cp}} + (\frac{\alpha_3}{\alpha_4} + \frac{\alpha_4}{\alpha_4} + \frac{\alpha_5}{\alpha_4})\frac{z_2 T_{cm}}{w_4 T_{cp}} + (\frac{\alpha_4}{\alpha_4} + \frac{\alpha_5}{\alpha_4})\frac{z_3 T_{cm}}{w_4 T_{cp}}} \\ &= \frac{1}{1 + (k_3 k_4 + k_4 + 1 + \frac{1}{k_5})\frac{z_1 T_{cm}}{w_4 T_{cp}} + (k_4 + 1 + \frac{1}{k_5})\frac{z_2 T_{cm}}{w_4 T_{cp}} + (1 + \frac{1}{k_5})\frac{z_3 T_{cm}}{w_4 T_{cp}}} \end{aligned} \tag{5.114}$$

$$U_5 = \frac{1}{1+\left(\frac{\alpha_2}{\alpha_5}+\frac{\alpha_3}{\alpha_5}+\frac{\alpha_4}{\alpha_5}+\frac{\alpha_5}{\alpha_5}\right)\frac{z_1 T_{cm}}{w_5 T_{cp}}+\left(\frac{\alpha_3}{\alpha_5}+\frac{\alpha_4}{\alpha_5}+\frac{\alpha_5}{\alpha_5}\right)\frac{z_2 T_{cm}}{w_5 T_{cp}}+\left(\frac{\alpha_4}{\alpha_5}+\frac{\alpha_5}{\alpha_5}\right)\frac{z_3 T_{cm}}{w_5 T_{cp}}+\left(\frac{\alpha_5}{\alpha_5}\right)\frac{z_4 T_{cm}}{w_5 T_{cp}}}$$
$$= \frac{1}{1+\left(k_3k_4k_5+k_4k_5+k_5+1\right)\frac{z_1 T_{cm}}{w_5 T_{cp}}+\left(k_4k_5+k_5+1\right)\frac{z_2 T_{cm}}{w_5 T_{cp}}+\left(k_5+1\right)\frac{z_3 T_{cm}}{w_5 T_{cp}}+\left(1\right)\frac{z_4 T_{cm}}{w_5 T_{cp}}} \quad (5.115)$$

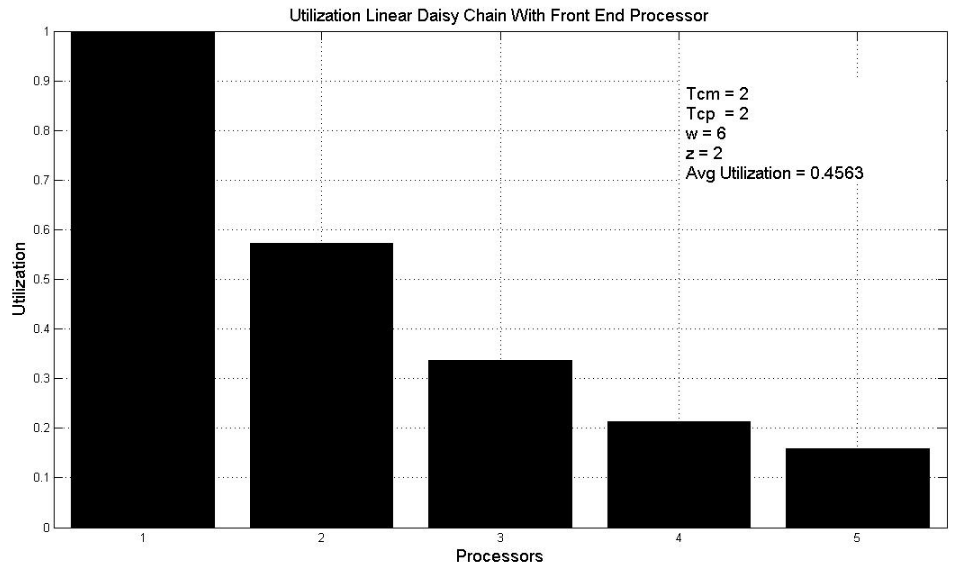


Figure 5.8. The utilization of a linear daisy chain with front end processing

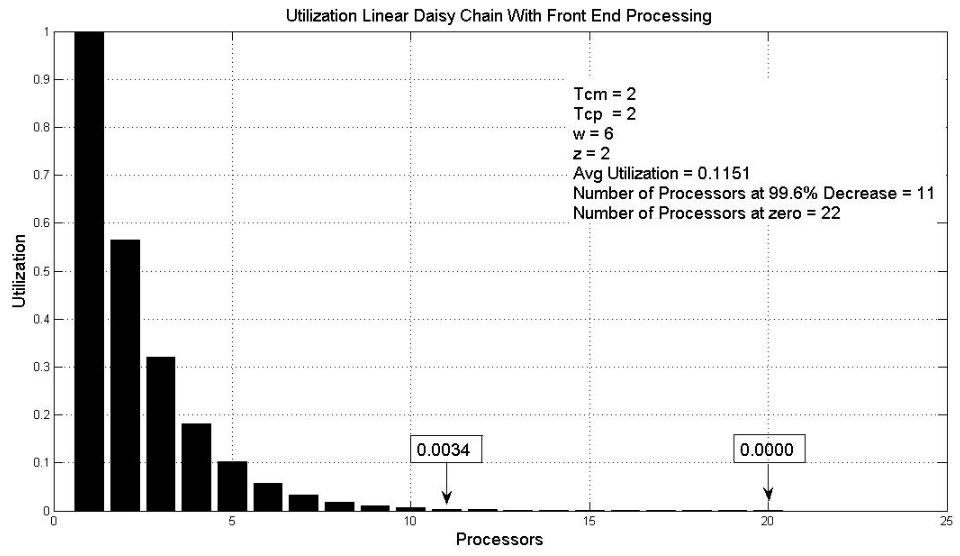


Figure 5.9. The utilization of a linear daisy chain with front end processing. At N = 11 the utilization has decreased 99.6% from its original value. At N = 20 the utilization is zero.

Speedup for the network is found as in section 5.2, by taking the ratio of the time it takes to finish computation on one processor divided by the time it takes to complete computations on the entire network.

T_P : is the time to complete computations on a single processor

T_{PN} : is the time to complete computations on an entire network

$$speedup = \frac{T_P}{T_{PN}} \tag{5.116}$$

Using equations (5.100), (5.101) and (5.102) repeated below respectively

$$\alpha_2 = k_3\alpha_3$$
$$\alpha_3 = k_4\alpha_4$$
$$\alpha_4 = k_5\alpha_5$$

Rewriting them yields

$$\alpha_3 = \frac{1}{k_3} \tag{5.117}$$

$$\alpha_4 = \frac{1}{k_4}\alpha_3 \tag{5.118}$$

$$\alpha_5 = \frac{1}{k_5}\alpha_4 \tag{5.119}$$

A general expression is found for α_i , it is the same equation as in section 5.2 with slight modifications of the index to accommodate the addition processor, add because there is now front end processing.

$$\alpha_i = \prod_{m=2}^{i} \frac{1}{k_m}\alpha_{i-(i-1)} \qquad \text{for } i = 2, ..., N \tag{5.120}$$

$$\alpha_{i-(i-1)} = \alpha_1 \tag{5.121}$$

For the above conditions in equation (5.119)

The normalized equation for this network is

$$\alpha_1 + \alpha_2 + \alpha_3 + ... + \alpha_N = 1 \tag{5.122}$$

rewriting the normalized equation (5.121)

$$\alpha_1 + \sum_{i=2}^{N} \alpha_i = 1 \tag{5.123}$$

Substituting equations (5.119) and (5.120) in equation (5.122)

$$\alpha_1 + \sum_{i=2}^{N} \prod_{m=2}^{i} \frac{1}{k_m} \alpha_1 = 1 \qquad \text{for } i = 2, 3, ..., N \tag{5.124}$$

$$\alpha_1 \left[1 + \sum_{i=2}^{N} \prod_{m=2}^{i} \frac{1}{k_m} \right] = 1 \tag{5.125}$$

The value of α_1 for this network is

$$\alpha_1 = \frac{1}{\left[1 + \sum_{i=2}^{N} \prod_{m=2}^{i} \frac{1}{k_m} \right]} \qquad \text{for } i = 2, 3, ..., N \tag{5.126}$$

The time to perform computations on a single processor is

$$T_{P1} = \alpha_1 w_1 T_{cp} \tag{5.127}$$

For $\alpha_1 = 1$ and $w_i = w$ equation (5.126) becomes

$$T_{P1} = w T_{cp} \tag{5.128}$$

The time to perform computations on the entire network is

$$T_{PN} = \alpha_1 w_1 T_{cp} \tag{5.129}$$

When $w_i = w$ equation (5.128) becomes

$$T_{PN} = \alpha_1 w T_{cp} \tag{5.130}$$

$$speedup = \frac{T_{P1}}{T_{PN}} = \frac{wT_{cp}}{\alpha_1 w T_{cp}} = \frac{1}{\alpha_1} \tag{5.131}$$

$$speedup = \left[1 + \sum_{i=2}^{N} \prod_{m=2}^{i} \frac{1}{k_m}\right] \qquad \text{for } i = 2, 3, ..., N \tag{5.132}$$

As in section 5.2 speedup is plotted versus the number of processors, Figure 5.10. A network of N = 20 is used because this is the number of processor used for the utilization in Figure 5.9. Here the speedup saturates at 2.3028. Taking 99.6% of the saturation value gives 2.29. This equates to 9 processors.

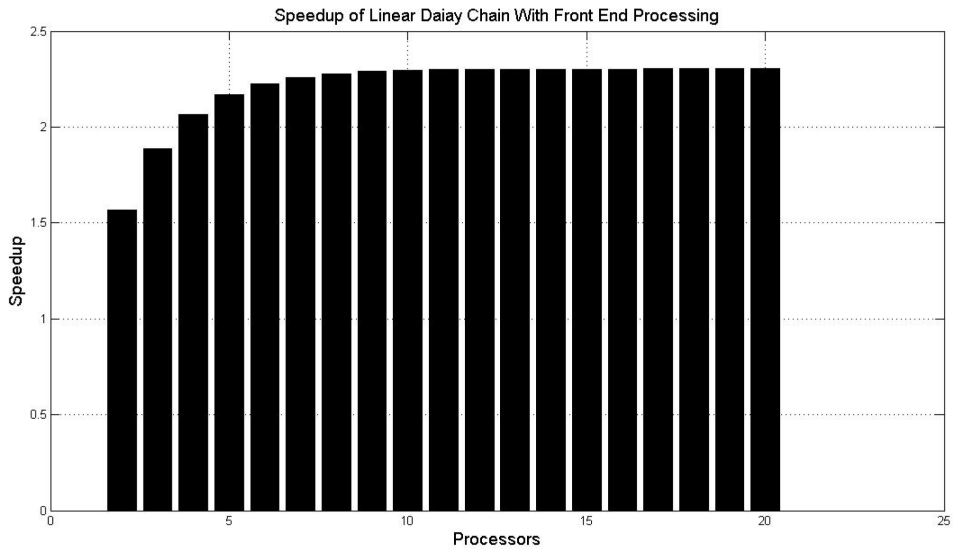


Figure 5.10. Speedup of a linear daisy chain with front end processing.

5.4 ANALYSIS OF RESULTS/CONCLUSION OF CHAPTER 5

The performance metric $N_{99.6\%}$ and speedup are used to determine if a particular network performs better than another, Table 1.

	WOFE stag	WFE Hyb
$N_{99.6\%}$	13	11
Speedup	11	9

Table 2. Compilation of data from Figures 5.4, 5.5, 5.9, and 5.10
$W_i = w = 6$, $z_i = z = 2$, $T_{cm} = T_{cp} = 3$

$N_{99.6\%}$ = The processor at which the utilization had decreased 99.6% from its original value
Speedup = The processor at which 99.6% of the saturation value is found
WOFE stag = Daisy chain without front end processing and staggered start
WFE Hyb = Daisy chain with front end processing and Hybrid strategy
W = Inverse processor speed
z = Inverse speed
T_{cp} = Computation intensity
T_{cm} = Communication intensity

This data shows that WOFE can add two more processors and still benefit from parallel processing. The performance metric $N_{99.6\%}$ shows how many processors can be added to a network before the utilization drops to a point at which utilization is consider too low for the entire network. The numbers are close, but seem to suggest a slight advantage to having a network based of WOFE.

However for the linear daisy chain configuration this is misleading. When the average utilization is considered for a network with N = 5 processors. The linear daisy chain with front end processing has an average utilization that is about 23% better than that of WOFE.

For large network with N= 30 processors WFE is about 5% better, Table 2.

N	5	8	9	11	13	20	30
WOFE AvgU	0.2238	0.1301	0.1138	0.0911	0.0759	0.0479	0.0314
WFE AvgU	0.4563	0.2878	0.2558	0.2093	0.1775	0.1145	0.0768
%WFE$_{larg}$	23.3	15.8	14.2	11.8	9.6	6.7	4.5
%inc	103.9	121.2	124.8	129.7	133.9	139.0	144.6

Table 3. Average utilizations for linear daisy chain with and without front end processing. The values are taken for different size networks.

AvgU = Average utilization
WOFE = Without front end processing
WFE = With front end processing.
%WFE$_{larg}$ = Percent WFE is larger than WOFE
%inc = Percent increase of WFE over WOFE
W = Inverse processor speed
z = Inverse speed
T_{cp} = Computation intensity
T_{cm} = Communication intensity

A 100% utilization of processor P_1 in the linear daisy chain (LDC) with front end processing plays a role in its high average utilization. The utilization at P_1 is removed. That is easily done because U_1 is set to one in the program. The average utilization is still 4% better for small networks, N = 5. The average utilization is lower when there is no front end processing because the staggered star also play role in the average utilization. In Figure 5.2 the total time for processor P_2 is found by summing $T_2 + T_3 + \alpha_2 w_2 T_{cp}$, this is equal to $T_{f2,}$ equation (5.5). This total time is the denominator of the utilization. Comparing equation (510) which does not have front end processing and equation (5.75) which has front end processing, it is shown that summing the communication delays and the computation time as in equation (5.10) creates more terms in the denominator increasing its size and as a result lowering the utilization. In Figure 5.7 if the front end processing was removed and the total time for processor P_2 was found as above. It would be $T_2 + \alpha_2 w_2 T_{cp}$ the terms of T_3 would not increase the size of the dominator. As the networks increase in size, Table 2, the percent of the difference between the decreases. Although the average utilization of the linear daisy chain with front end process remains higher they converge. In chapter 3 this was shown that this decrease is that of a power series. The percent increase of WFE to WOFE increases as the number of processors N increases.

When comparing the two linear daisy chains if the goal is to have a better average utilization over the entire network the linear daisy chain with the front processor and the hybrid start would be preferred. If there is a need for the last two processors in a network to remain equal as the changes, the linear daisy without front end processing, and staggered start would be preferable.

In this chapter the utilization as a performance metric is shown to be versatile. When comparing the number of processor at which a certain event occurs does not give insight into the network, another aspect of the utilization can be used to get a better understanding of the network. It has shown that it is another performance metric that can be used in conjunction with speedup.

REFERENCES

[1] Michael Mitzenmacher
Analyses of Load stealing Models Based on Differential Equations
SPAA 98 Puerto Vallarta Mexico
ACM 1998 0-89791-989-0/98/6
Pages 212-221

[2] Miron Livny and Myron Melman
Load Balancing in Homogeneous Broadcast Distributed System
1982 ACM 0-89791-069-9/82/0400-0047
Pages 47-55

[3] Jagadeesh Kasaraneni, Theodore Johnson, and Paul Avery
Load Balancing in a Distributed Processing System for High-Energy Physics (UFMulti)
1995 ACM 0-89791-658-1 95 0002
Pages 177-181

[4] Jiming Liu, Xiaolong Jin, and Yuanshi Wang
Agent-Based Load Balancing on Homogeneous Mini-grids: Macroscopic Modeling and Characterization
IEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEM
VOL.6, NO.7, JULY 2005
Pages 586-598

[5] Jacques M. Bahi, Sylvain Contassort-Vivier, and Raphael Couturier
Dynamic Load Balancing and Efficient Load Estimators for Asynchronous Iterative Algorithms
IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEM
VOL.6, NO.4, APRIL 2005
Pages 289-299

[6] Karen Devine, Bruce Hendrickson, Erik Boman, Matthew St. John and Courtenay Vaughan
Design of Load-Balancing Tools for Parallel Applications
ICS 2000 Santa Fe, New Mexico, USA
Pages 110-118

[7] Leonid Oliker and Rupak Biswas
Efficient Load Balancing and Data Remapping for Adaptive Grid Calculations
SPAA 97 Newport, Rhode Island, USA
1997 ACM 0-89791-890-8/97/06
Pages 33-42

[8] Raghu Subramanian and Isaac D. Scherson
An Analysis of Diffusive Load Balancing
SPAA 94 -6/94 Cape May, N.J, USA
1994 ACM 0-89791-671-9/94/0006
Pages 220-225

[9] Thomas G. Robertazzi
Network and Grids Technology and theory
2007 Springer Science + Business Media LLC, New York

[10] Bataineh, S. and Robertazzi, T.G. (1991)
Bus oriented load sharing for a network of sensor driven processor. IEEE Transaction on systems, Man, and Cybernetics. 21, 5 (1991), 1202-1205

[11] Bataineh, S., Hsiung, T. and Robertazzi, T.G. (1994)
Closed form solutions for bus and tree network of processors load sharing divisible job. IEEE Transaction on Computers, 43, 10 (1994), 1184-1196.

[12] Bataineh, S.; Robertazzi, T.G.
Performance limits for processor network with divisible jobs. Aerospace and Electronic Systems, IEEE Transactions on volume 33, issue 4, Oct. 1997 page(s): 1189-1198

[13] Blazewicz, J., Drozdowski, M., and Wqglarz, P. J.
Performance limits of a two dimensional network of load sharing processor (1991). Foundation of Computing and Decision Science.

[14] Blazewicz, J., Drozdowski, M., Guinand, F., and Denis Trystram
Discrete Applied Mathematics volume 94,issue 1-3 (May 1999) page 35-50

[15] V. BHARADWAJ, D.GHOSE AND V. MANI
Department of Aerospace Engineering, Indian Institute of Science, Bangalore, 560012, India , September 1993

[16] BHARADWAJ, V., GHOSE, D. AND MANI, V. (1995)
Multi-installment load distribution in tree networks with delays. IEEE Transaction on Aerospace and Electronic Systems, 31,2, (1995), 555-566.

[17] V. Bharadwaj, D. Ghose, V. Mani and T. G. Robertazzi
Scheduling Divisible Loads in Parallel and Distributed Systems.
IEEE Computer Society Press, Los Alamitos, CA, 1996

[18] Bharadwaj, V., Ghose, D. and Robertazzi, T.G. (2003)
Divisible Load Theory: a new paradigm for load scheduling in distributed systems. Cluster Computing, 6,1,(2003), 7-17

[19] Cheng, Y.C. and Robertazzi, T.G. (1988)
Distributed computation with communication delays. IEEE Aerospace and Electronics Systems. 24,6,(1988) 700-712

[20] Cheng, Y.C. and Robertazzi, T.G. 1990
Distributed computation for a t tree network with communication delays. IEEE Transactions on Aerospace and Electronics Systems, 26,3,(1990), 511-16

[21] Drozdowski, M., Glazek, W.

Scheduling divisible loads in a three-dimensional mesh of processors.
Parallel Computing 25(4): 381-400 (1999)

[22] K. Ko and T.G. Robertazzi
Record search Time Evaluation (March 2000)
Department of Electrical and Computing Engineering
SUNY at Stony Brook, Stony Brook, NY 11794
e-mail: tom@ece.sunysb.edu

[23] Kwangil Ko and Thomas G. Robertazzi, Fellow, IEEE
Signature Searching Time Evaluation in Flat File Data bases.
IEEEE Transaction on Aerospace and Electronics Systems, vol. 44, no. 2, April 2008, pp. 493-502

[24] Mani, V., and Ghose , D. (1994)
Distributed Computation in Linear Networks: Closed- form solutions.
IEEE Transaction on Aerospace and Electronic Systems, 30,2,(1994), 471-83

[25] V. MANI
An Equivalent Tree Network Methodology for Efficient Utilization of Front-Ends in Linear Networks
Department of Aerospace Engineering, Indian Institute of Science
Bangalore, 56001 India, Cluster Computing 6,57-62, 2003

[26] Robertazzi, T.G.
Processor Equivalent for a Linear Daisy Chain of Load sharing Processors
IEEE Transactions of Aerospace and Electronics Systems, vol. 29, No.4, Oct. 1993, pp. 1216-1221

[27] Robertazzi, T.G. (2003)
Ten reason to use divisible load theory
Computer, 36,5,(2003),63-68.

[28] Thomas G. Robertazzi* and Dantong Yu**
Multi-Source Grid Scheduling for Divisible Loads
* Department of Electrical and Computer Engineering
Stony Brook University, stony Brook, NY 11794, USA
** Department of Physics Brookhaven National Laboratory
Upton, NY 11973, USA
tom@ece.sunysb.edu and dyyu@bnl.gov

[29] Sohn j. and Robertazzi, T.G. (1996)
Optimal divisible job load sharing for bus network
IEEE Transactions on Aerospace and Electronic Systems,
32,1,(1996), 34-39

[30] Danton Yu* and Thomas G. Robertazzi**
Divisible Load Scheduling for Grid Computing
* Department of Physics Department of Physics Brookhaven National Laboratory Upton, NY 11973, USA
** Department of Electrical and Computer Engineering
Stony Brook University, stony Brook, NY 11794, USA
dtyu@bnl.gov and tom@ece.sunysb.edu