Total System Cost Analysis and Comparison of DX-Tree and Star Ring Architectures in conjunction with Master-Slave Multi-Super-Hypercube topology

Hamid Abachi
Computer Engineering Department, King Saud University
Riyadh, 11543, Saudi Arabia
E-mail: habachi@ksu.edu.sa

Roger Lee
Department of Computer Science,Central Michigan University
U.S.A
E-mail: lee1ry@cmich.edu

Received 10 February 2013

Accepted 19 November 2013

Abstract

In todays advanced technology environment, we come across a number of research areas that require extensive data processing. These areas include but not limited to, modeling and simulation of accuracy, safety and reliability of nuclear weapons, global warming, weather forecasting, DNA computing, nanotechnology, immune cell system, optical computing. Since we constantly continue to explore new avenues in our research program, hence the demand for improvements in other scientific and engineering parameters are continuously on the rise. These scientific and engineering requirements include but not limited to speed, reliability, fault tolerance, availability, compatibility, scalability, flexibility, cost to name a few. In todays environment, utilisation of parallel processing not only satisfies increasing demand for extensive data processing, yet it relatively fulfils deficiencies that one would experience in these scientific and engineering requirements. To this effect the authors have introduced architectures based on Master-Slave Multi-Super-Hypercube DX-Tree architecture and Master-Slave Multi-Super-Hypercube Star Ring Architecture. For these architectures and the current message passing architectures, the total system costs are developed and through mathematical modeling and simulation are compared. In this comparison the merit and demerits of these architectures from the cost point of view are highlighted to enable the user to select an appropriate Message-Passing architecture which would best satisfy ones scientific requirements.

Keywords: Master slave, DX-Tree, Star-ring, Message passing architectures, parallel processing, Super hypercube.

1. Introduction

The need for achieving higher processing power to satisfy extensive data processing requirement as well as the advancement of semi-conductor technology, together with the software development and its contemporary applications, have resulted in the rapid development of the high performance computing. Evidently these technological achievements have had direct impact on improving criteria such as speed, reliability, fault tolerance and cost in general. One of the integral and critical aspects of the performance evaluation of any new parallel processing architecture is the cost analysis. With the budgetary constraints that we are currently facing, a proper cost analysis would determine as whether or not a new model is cost-effective and hence would it meet our scientific expectations?

Bearing this motivation in mind the authors have compared the cost analysis of two models which are coined as Master-Slave Multi-Super Hypercube DX-Tree ($(MS)^2HDX$ - *Tree*) and Master-Slave Multi-Super-Hypercube Star-Ring ($(MS)^2HS$ - *R*) architectures. Moreover the total system cost analysis for these two architectures together with the current message passing architectures are computed and analysed for comparison purposes.

2. MIMD Architectures

Multiple-instruction multiple-data (MIMD) stream parallel architectures are classified into two categories. The first category is known as shared memory organization where the second category is known as message passing organization. In the former scenario, processors communicate by reading and writing locations in a shared memory that is equally accessible by all processors. However, the latter case deals with the situation where each processor has its own memory attached to the processor [1]. Implementation of the message passing architectures will to some extent reduce the memory contention that normally exists in the shared memory architectures. Among the most common message passing architecture, one can include Torus, Binary Tree and Hypercube. However, X-Tree architecture is a derivative of Tree architecture which is configured by connecting rings at each Tree level. In this architecture, by providing alternate routes along each Tree level, message density can more evenly be distributed which leads to enhanced performance. In order to complete the modeling and system simulation of a message passing architecture, one needs to consider the major network performance metrics. As reported in [2], these network performance metrics can be defined as:

- Number of nodes (N_N): the accessible processing elements within the architecture.
- Number of links (N_L): the number of edges (links or channels) connected to a node.
- Diameter (R): is identified as the longest path between any two nodes.
  Note that for the diameter calculation, it was assumed that in Super-Hypercube, two indirect nodes communicate through a Router.
- Normalised System Cost (K_{STN}): the ratio of total system cost in units of processor cost divided by total number of nodes.

However, many authors have to some extent considered more general and interrelated parameters that would complement the above parameters and fulfill the performance evaluation requirements. These include hardware system, architecture schemes, operating system, language, program and algorithm [3], [4] which the authors believe, can be considered as further work in this area.

3. Cost Utilization Analysis

The cost of a multiprocessor system is a critical factor in determining its feasibility for a given application. In the context of multiprocessor systems, cost is a difficult parameter to define, especially given that component costs are highly dependent on economic conditions. This section creates a standard framework for relative cost comparison between different message-passing architectures based on normalized component costs.

4. Cost Metrics

As reported in [5], the overall total system cost $(OTS)_c$ is dominated by the total node-related cost $(TNR)_c$ and the total communication-link cost $(TCL)_c$ which leads to:

$$(OTS)_C = (TNR)_C + (TCL)_C \qquad (1)$$

On the other hand, the total node-related cost $(TNR)_C$ is the product of the unit node cost $(UN)_C$ and the number of nodes (N_N) that is;

$$(TNR)_C = (UN)_C \times N_N \qquad (2)$$

Nevertheless, as a general rule, we assume that each processing node consists of CPU, memory modules and I/O interfacing ports that provide connections between different functional units in the overall system configuration. The total communication-link cost $(TCL)_C$ is the product of the unit link cost $(UL)_C$ and the number of links (N_L),

$$(TCL)_C = (UL)_C \times N_L \qquad (3)$$

We assume that each link consists of some form of interconnection capability that facilitates joining nodes and receiver/transmitter pairs at the ends of each link in order to furnish any required signal conditioning. To this end we summarize the total system cost $(OTS)_C$ as being:

$$(OTS)_C = (UN)_C \times N_N + (UL)_C \times N_L \qquad (4)$$

However, a far more difficult task for the multi-processor system designer is to justify the suitability of a network over a range of component costs. In order to provide a meaningful tool for this justification, one can refine the issue by introducing a term called normalized overall total system cost function $(OTS)_C$[6]. We take $(UN)_C$ as the base cost, because it is a constant from a system designer's point of view, since $(UL)_C$ is likely to be a fraction of $(UN)_C$. This is accomplished by scaling $(OTS)_C$ down by $(UN)_C$.

$$K_{ST} = \frac{(OTS)_C}{(UN)_C} = N_N + K_L N_L \qquad (5)$$

where $K_{ST} = \frac{(UL)_C}{(UN)_C}$. Therefore, K_{ST} gives the total system cost in units of $(UN)_C$. In practice, K_L will vary from near zero for a tightly coupled multiprocessor system to somewhat near one for a loosely coupled or distributed computer network. Finally the minimum value of K_{ST} is N_N and is invariant for any network of a fixed size. This suggests that it is a base value that can be used to normalize K_{ST} which is coined as K_{STN} for a better comparison between different network architectures. To this end we can express K_{STN} as being:

$$K_{STN} = \frac{K_{ST}}{N_N} = 1 + \frac{K_L N_L}{N_N} \qquad (6)$$

The normalized cost function for message-passing architectures is summarized together with the rest of the parameters that are used for the system evaluation.

5. Message-Passing Architectures

5.1 *X-Tree Architecture*

Another possible solution to the problem of congestion that normally exists in the tree topology is to add rings at each level. This modification provides alternate routes along each tree level, which results in the reduction of the traffic congestion. Furthermore this would result in message density being more evenly distributed, which leads to an enhanced performance [7]. This scheme is called an X-Tree as shown in Figure 1.

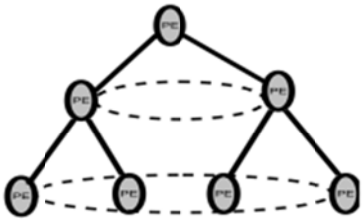


Figure 1: X-Tree Architecture

Provision of the alternate routes also improve reliabilitywhen processing nodes or links become faulty because message can circumnavigate faults by routing up/down and left/right. This is similar characteristic of the two dimensional tours and hexagonal networks.

5.2 *DX-Tree Architecture*

In this architecture we have added a mirror image of say three levels on the top half of the X-tree arrangement to the lower half part as can be seen in Figure 2.

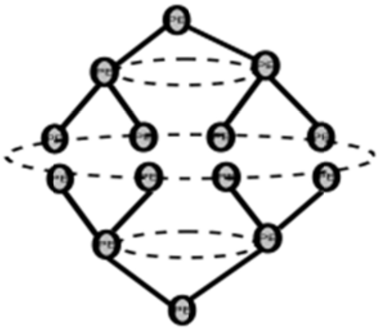


Figure 2: DX-Tree Architecture

6. Super-Hypercube Architecture

In order to overcome Hypercube limitations such as routing and expandability, a derivative version of the Hypercube architecture namely Super-Hypercube (SHP) is introduced [7]. This architecture includes applying a Router (R) to the basic Hypercube. This router acts as a crossbar switch, which can provide a communication path between two indirect PEs. Figure 3 shows the basic principle of this architecture.

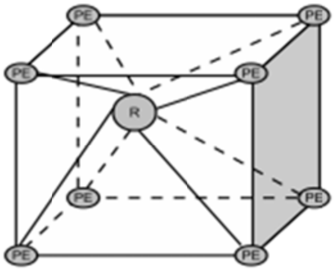
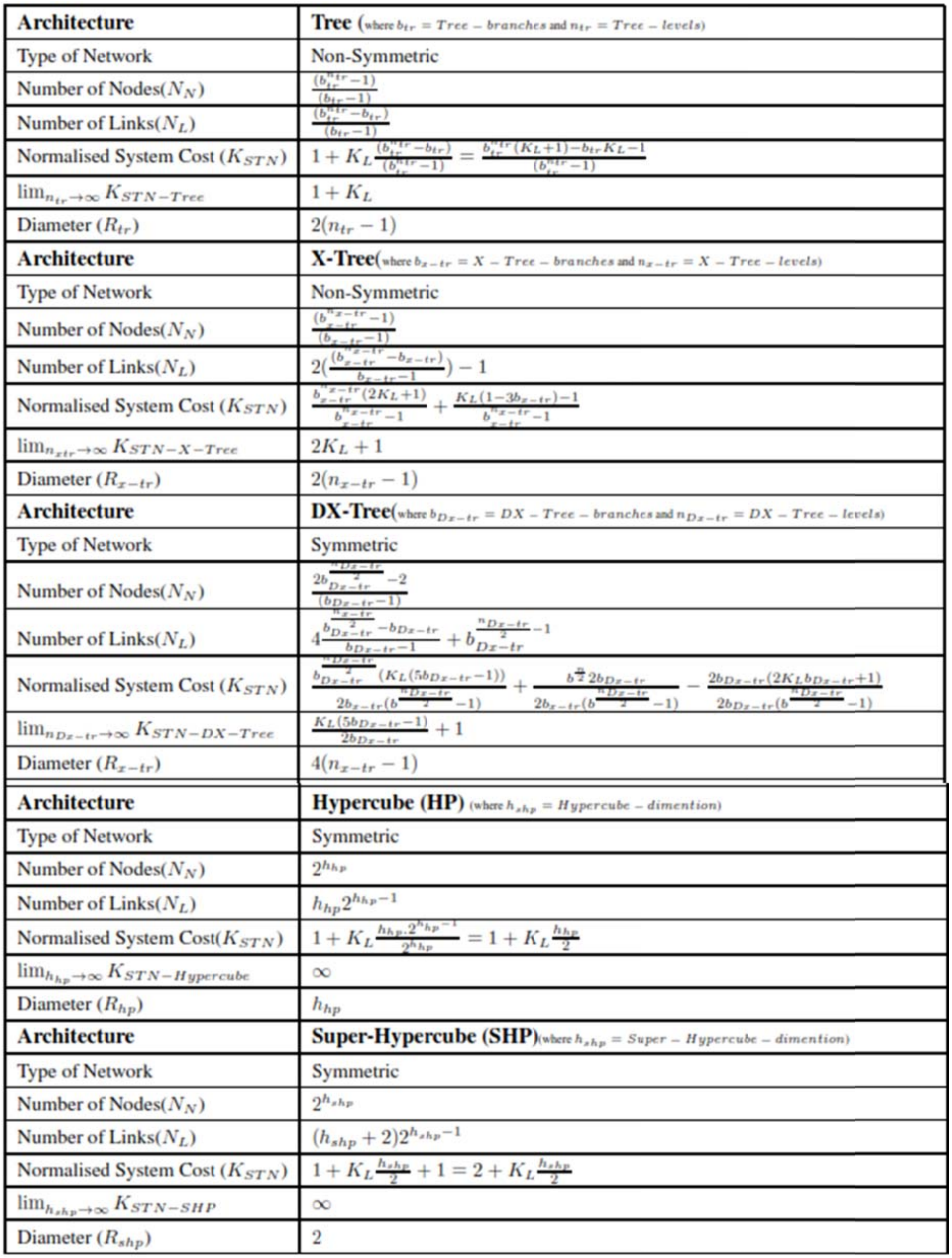


Figure 3: Super-Hypercube Architecture

7. Network Metrics for the Existing Message Passing Architecture

As reported in [7],[8], the results of the network metrics for message-passing architectures such as: Tree, X-Tree, Hypercube (HP), Super-Hypercube (SHP), Super-Hypercube-Array (SHA) and Tours are shown in Table 1.



Architecture	**Tree** (where $b_{tr} = Tree - branches$ and $n_{tr} = Tree - levels$)
Type of Network	Non-Symmetric
Number of Nodes(N_N)	$\frac{(b_{tr}^{n_{tr}}-1)}{(b_{tr}-1)}$
Number of Links(N_L)	$\frac{(b_{tr}^{n_{tr}}-b_{tr})}{(b_{tr}-1)}$
Normalised System Cost (K_{STN})	$1+K_L\frac{(b_{tr}^{n_{tr}}-b_{tr})}{(b_{tr}^{n_{tr}}-1)} = \frac{b_{tr}^{n_{tr}}(K_L+1)-b_{tr}K_L-1}{(b_{tr}^{n_{tr}}-1)}$
$\lim_{n_{tr}\to\infty} K_{STN-Tree}$	$1+K_L$
Diameter (R_{tr})	$2(n_{tr}-1)$
Architecture	**X-Tree**(where $b_{x-tr} = X - Tree - branches$ and $n_{x-tr} = X - Tree - levels$)
Type of Network	Non-Symmetric
Number of Nodes(N_N)	$\frac{(b_{x-tr}^{n_{x-tr}}-1)}{(b_{x-tr}-1)}$
Number of Links(N_L)	$2(\frac{(b_{x-tr}^{n_{x-tr}}-b_{x-tr})}{b_{x-tr}-1})-1$
Normalised System Cost (K_{STN})	$\frac{b_{x-tr}^{n_{x-tr}}(2K_L+1)}{b_{x-tr}^{n_{x-tr}}-1} + \frac{K_L(1-3b_{x-tr})-1}{b_{x-tr}^{n_{x-tr}}-1}$
$\lim_{n_{xtr}\to\infty} K_{STN-X-Tree}$	$2K_L+1$
Diameter (R_{x-tr})	$2(n_{x-tr}-1)$
Architecture	**DX-Tree**(where $b_{Dx-tr} = DX - Tree - branches$ and $n_{Dx-tr} = DX - Tree - levels$)
Type of Network	Symmetric
Number of Nodes(N_N)	$\frac{2b_{Dx-tr}^{\frac{n_{Dx-tr}}{2}}-2}{(b_{Dx-tr}-1)}$
Number of Links(N_L)	$4\frac{b_{Dx-tr}^{\frac{n_{x-tr}}{2}}-b_{Dx-tr}}{b_{Dx-tr}-1}+b_{Dx-tr}^{\frac{n_{Dx-tr}}{2}-1}$
Normalised System Cost (K_{STN})	$\frac{b_{Dx-tr}^{\frac{n_{Dx-tr}}{2}}(K_L(5b_{Dx-tr}-1))}{2b_{x-tr}(b^{\frac{n_{Dx-tr}}{2}}-1)} + \frac{b^{\frac{n}{2}}2b_{Dx-tr}}{2b_{x-tr}(b^{\frac{n_{Dx-tr}}{2}}-1)} - \frac{2b_{Dx-tr}(2K_Lb_{Dx-tr}+1)}{2b_{Dx-tr}(b^{\frac{n_{Dx-tr}}{2}}-1)}$
$\lim_{n_{Dx-tr}\to\infty} K_{STN-DX-Tree}$	$\frac{K_L(5b_{Dx-tr}-1)}{2b_{Dx-tr}}+1$
Diameter (R_{x-tr})	$4(n_{x-tr}-1)$
Architecture	**Hypercube (HP)** (where $h_{shp} = Hypercube - dimention$)
Type of Network	Symmetric
Number of Nodes(N_N)	$2^{h_{hp}}$
Number of Links(N_L)	$h_{hp}2^{h_{hp}-1}$
Normalised System Cost(K_{STN})	$1+K_L\frac{h_{hp}.2^{h_{hp}-1}}{2^{h_{hp}}} = 1+K_L\frac{h_{hp}}{2}$
$\lim_{h_{hp}\to\infty} K_{STN-Hypercube}$	∞
Diameter (R_{hp})	h_{hp}
Architecture	**Super-Hypercube (SHP)**(where $h_{shp} = Super - Hypercube - dimention$)
Type of Network	Symmetric
Number of Nodes(N_N)	$2^{h_{shp}}$
Number of Links(N_L)	$(h_{shp}+2)2^{h_{shp}-1}$
Normalised System Cost (K_{STN})	$1+K_L\frac{h_{shp}}{2}+1 = 2+K_L\frac{h_{shp}}{2}$
$\lim_{h_{shp}\to\infty} K_{STN-SHP}$	∞
Diameter (R_{shp})	2

Architecture	**Torus**
Type of Network	Symmetric
Number of Nodes(N_N)	$w_{trs}^{t_{trs}}$
Number of Links(N_L)	$t_{trs}w_{trs}^{t_{trs}}$
Normalised System Cost(K_{STN})	$1+K_L\frac{t_{trs}(w_{trs}^2)}{w_{trs}^2}=1+K_Lt_{trs}$
$\lim_{w_{trs}\to\infty} K_{STN-Torus}$	$1+K_Lt_{trs}$
Diameter(R_{trs})	$t_{trs}[\frac{w_{trs}}{2}]$

Table 1: Summary of network metrics

8. Master-Slave Multi-Super-Hypercube Star-Ring Architecture

A recently developed architecture which is coined as Master-Slave Multi-Super-Hypercube Star-Ring architecture is considered. Its cost parameter has been developed [9] and is used for comparison purposes.

This architecture which represents a true multiprocessing topology consists of the combination of star and ring topology. In this architecture as it is shown in Figure 4, the master processor is at the center of the ring and by having connections through Routers, it can provide fast and reliable communication access to each satellite node. This architecture is constructed to perform simultaneous and concurrent processing activities. The principal architecture of each satellite node is based on the Super-Hypercube (SHP) architecture as shown in Figure 3.

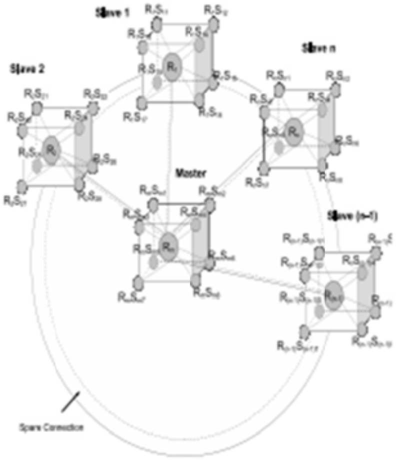


Figure 4: Master-Slave Multi-Super-Hypercube Star- Ring Architecture

8.1 *Mathematical modeling of Master-Slave Multiple Super-Hypercube Star-Ring Architecture*

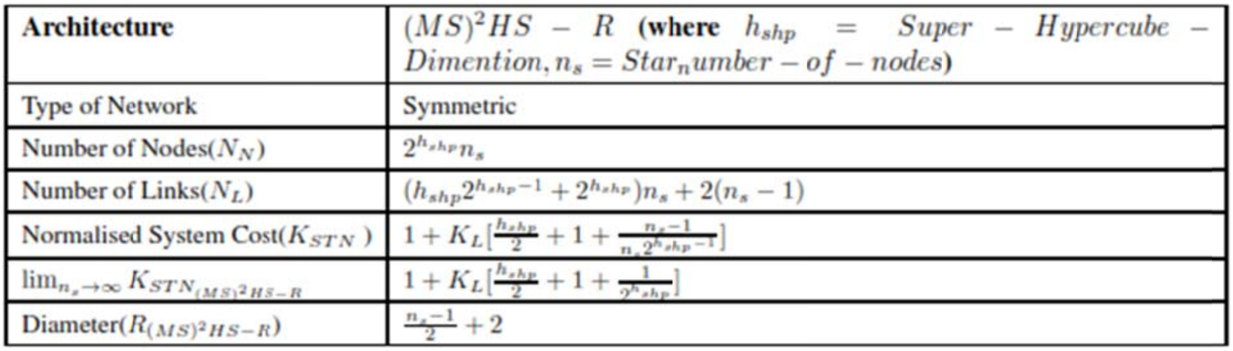
As reported in [9], Table 2 illustrates the networks metrics for the Master-Slave Multi-Super-Hypercube Star-Ring ($(MS)^2HS$ - R) architecture. It is worth mentioning in calculating the Normalised System Cost, we assumed that the Router cost compare with processor cost is negligible and hence was ignored.

9. Master-Slave Multi-Super- Hypercube X-Tree Architecture

This newly developed architecture which is depicted in Figure 5 utilizes Tree architecture with SHP as its processing elements. However the Master-Slave scheme is adopted in order to manage and control the overall system activities. In this architecture, the performance would be greatly degraded if there are communication failures with no spare connections between different levels such as level 1 and 2. This means a failure can occur between the Router R_1 and Router R_1R_{21} and/or Router R_1R_{22} which causes a major deficiency in the performance of the overall system. In order to partially overcome this shortcoming, one could include additional communication links (as alternative paths) between all the routers at all levels (not shown in the diagram).

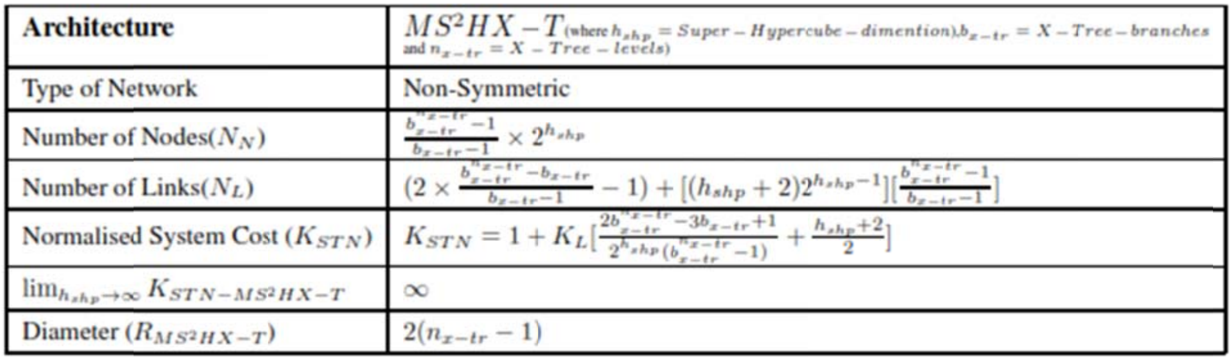
9.1 *Mathematical modeling of Master-Slave Multi-Super-Hypercube X-Tree Architecture*

This section addresses the mathematical models for the $(MS)^2HX$ - T architecture as reported in [10], which facilitates the mathematical modeling of an enhanced version of $(MS)^2HX$ - T model namely Master-Slave Multi-Super-Hypercube DX-Tree architecture. In this calculations, we have assumed that the $(MS)^2HX$ - T has b branches and n levels in which nodes have been replaced with one SHP and tabulated in Table 2. Furthermore, the author compares the cost analysis of



Architecture	$(MS)^2HS - R$ **(where** $h_{shp} = Super - Hypercube - Dimention, n_s = Star_number - of - nodes$**)**
Type of Network	Symmetric
Number of Nodes(N_N)	$2^{h_{shp}}n_s$
Number of Links(N_L)	$(h_{shp}2^{h_{shp}-1} + 2^{h_{shp}})n_s + 2(n_s - 1)$
Normalised System Cost(K_{STN})	$1 + K_L[\frac{h_{shp}}{2} + 1 + \frac{n_s-1}{n_s 2^{h_{shp}-1}}]$
$\lim_{n_s \to \infty} K_{STN_{(MS)^2HS-R}}$	$1 + K_L[\frac{h_{shp}}{2} + 1 + \frac{1}{2^{h_{shp}}}]$
Diameter$(R_{(MS)^2HS-R})$	$\frac{n_s-1}{2} + 2$

Table 2: $(MS)^2HS$-R network metrics



Architecture	$MS^2HX - T$ (where $h_{shp} = Super - Hypercube - dimention), b_{x-tr} = X - Tree - branches$ and $n_{x-tr} = X - Tree - levels)$
Type of Network	Non-Symmetric
Number of Nodes(N_N)	$\frac{b_{x-tr}^{n_{x-tr}} - 1}{b_{x-tr} - 1} \times 2^{h_{shp}}$
Number of Links(N_L)	$(2 \times \frac{b_{x-tr}^{n_{x-tr}} - b_{x-tr}}{b_{x-tr} - 1} - 1) + [(h_{shp} + 2)2^{h_{shp}-1}][\frac{b_{x-tr}^{n_{x-tr}} - 1}{b_{x-tr} - 1}]$
Normalised System Cost (K_{STN})	$K_{STN} = 1 + K_L[\frac{2b_{x-tr}^{n_{x-tr}} - 3b_{x-tr} + 1}{2^{h_{shp}}(b_{x-tr}^{n_{x-tr}} - 1)} + \frac{h_{shp}+2}{2}]$
$\lim_{h_{shp} \to \infty} K_{STN-MS^2HX-T}$	∞
Diameter (R_{MS^2HX-T})	$2(n_{x-tr} - 1)$

Table 3: $(MS)^2HX$ – T network metrics

this model with the existing message-passing architecture as described in this paper.

10. Operation of Master-Slave Multi-Super Hypercube DX-Tree Architecture

As an extension of $(MS)^2HX$ - T architecture, one could use DX-Tree topology where a mirror image of the X-Tree is used. This newly developed architecture is coined as Master-Slave Multi-Super-Hypercube DX-Tree architecture ($(MS)^2HDX$ -T) and is depicted in Figure 6. This architecture incorporates the combination of Master-slave Multi-Super-Hypercube in a DX-Tree environment. We have included a master and two co-master processors to manage and coordinate the allocation of the tasks and to perform the overall management of the system in general. The operation of this newly proposed architecture ($(MS)^2HDX$ - T) in a massively parallel processing system can best be explained as follows. The main role of the Master processor, is the task allocation and overall management and control of the system[11]. The master processor is intended to have the latest technology expected of an advanced processing element and the largest memory capacity in order to fulfill the load balancing, task allocation and overall system control requirements of the proposed massively parallel processing system. As the first step the main task is divided into multiple subtasks, which then it is placed in the main memory of the master processor. This is followed by first allocating these subtasks into the memory of the co masters processors (that is upper half of the system uses co master 1 whereas the lower half would use the co master 2) and then to the respective satellite slave processors for execution purposes. Upon completion of each subtask, the slave processor by sending an interrupt request initially to the respective co processor and then to the master processors will inform the master processor of the completion of its current subtasks and its readiness to accept the next available subtask. This process would continue until the entire task is completed. However, upon the existence of simultaneous multiple interrupt requests generated by two or more salve processors, there need to be a priority mechanism in place to take care of any conflict that may arise. Once the subtasks are executed the results are transferred from the local memory of each co-master

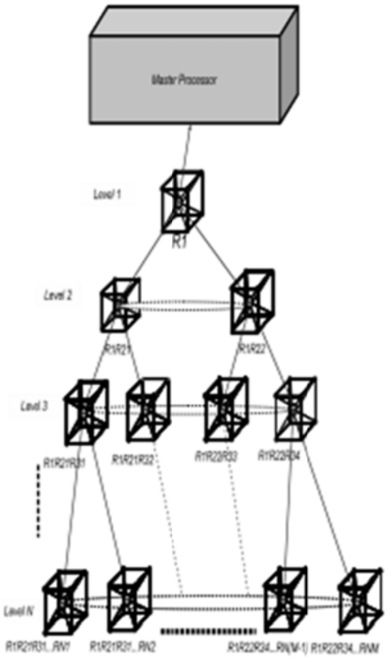


Figure 5: Master-Slave Multi-Super-Hypercube X- Tree Architecture

processor to the memory of the master processor for final processing purposes.

11. Mathematical Modeling of Master-Slave Multi-Super-Hypercube DX-Tree Architecture

This section presents the mathematical models and structural details for the proposed architecture [12], which would lead to a comparison between the model parameters of the proposed topology with those of the remaining message-passing architectures. The Master-Slave Multi-Super-Hypercube X-Tree is a XTree arrangement with b branches and n levels in which nodes have been replaced by a SHP. Furthermore, the number of nodes in Master-Slave multi-Super-Hypercube DX Tree ($N_{N_{(MS)^2HDX-T}}$) would be the number of nodes in DX-Tree multiply by the number of nodes in SHP which simply results in having:

$$N_{N_{(MS)^2HDX-T}} = \frac{2b_{Dx-tr}\frac{n_{Dx-tr}}{2} - 2}{b_{Dx-tr} - 1} \times 2^{h_{shp}}$$

where

$b_{Dx-tr} = DX - Tree - branches, n_{Dx-tr} = DX - Tree - levels$ and $h_{shp} = Super - Hypercube - dimension.$

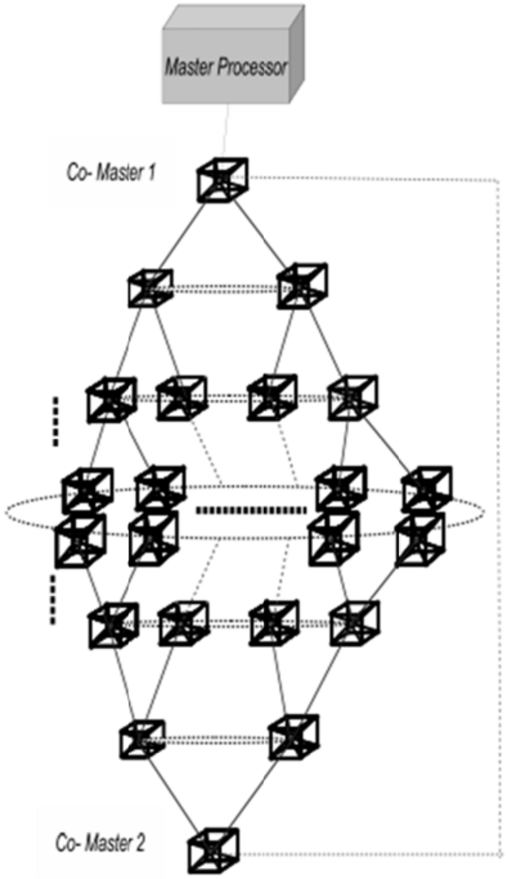


Figure 6: Master-Slave Multi-Super-Hypercube DXTree Architecture

Super-Hypercube DX-Tree would be calculated as follows:

$$N_{L_{(MS)^2HPDX-T}} = 4\left[\frac{\frac{n_{Dx-tr}}{2} - b_{Dx-tr} \; b_{Dx-tr}}{b_{Dx-tr} - 1}\right] + b_{Dx-tr}^{\frac{n_{Dx-tr}}{2}-1} +$$

$\left[(h_{shp} + 2)2^{h_{shp}-1}\right]\left[\frac{2b_{Dx-tr}-2}{b_{Dx-tr}-1}\right]$. Now, we proceed to compute the total system cost for $(MS)^2HDX$-T.As reported: $K_{STN} = 1 + \frac{K_L N_L}{N_N}$ where $K_L = \frac{C_L}{C_N} = \frac{unit-link-cost}{unit-node-cost}$.

Using the values of N_L and N_N from the above relationship results in having:

$$K_{STN} = 1 + K_L \left[\frac{5b_{Dx-tr}^{\frac{n_{Dx-tr}}{2}} - b_{Dx-tr}^{\frac{n_{Dx-tr}}{2}-1} - 4b_{Dx-tr}}{(4b_{Dx-tr}^{\frac{n_{Dx-tr}}{2}-1})h_{shp}} + \frac{h_{shp}+2}{2} \right].$$

Therefore,

$$\lim_{n_{(MS)^2HDX-T} \to \infty} K_{STN_{(MS)^2HDX-T}} = \infty$$

and the diameter is:

$$R_{(MS)^2HDX-T} = 4n^{Dx-tr}.$$

Figure 7 illustrates the Normalised System Cost (K_{STN}) for all described network architectures. These include existing message-passing architectures as well as proposed subclasses of Master-Slave Multi-Super-Hypercube. The latter includes $(MS)^2HX - T$, $(MS)^2HDX - T$ and $(MS)^2 HS - R$ architectures which are included for comparison purposes

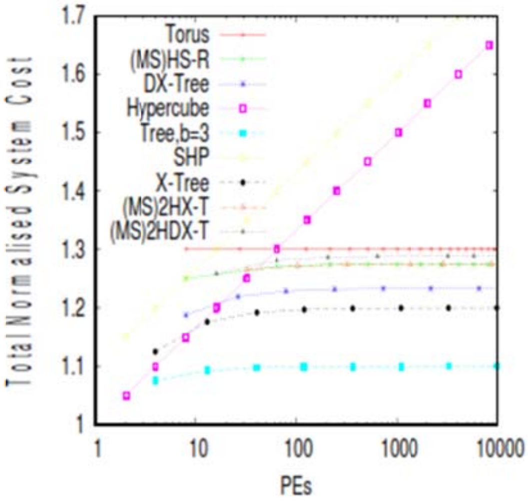


Figure 7: Normalised System Cost for different Networks

12. Conclusion

This paper examines and compares through mathematical modeling and simulation, the total system cost of a number of message-passing architectures. This comparison includes topologies such as: Master-Slave Multi-Super-Hypercube architectures with the existing message-passing architectures including: Tree, X-Tree, DX-Tree, Hypercube, Super-Hypercube and Torus. Analysis of graphical presentation depicted in Figure 7, reveals that except the Hypercube and Super-Hypercube, the remaining architectures enjoy a relatively constant variation of K_{STN} over a wide range of PEs. This undesirable scenario for Hypercube and Super-Hypercube, results from the fact that an increasing proportion of the system cost from a practical point of view is devoted to communication network overheads and not processing elements cost. A closer analysis of this graph also indicates that the Torus and to the lesser degree $(MS)^2HS - R$ has the closest markers distance for PEs greater than 1000 units. This behavior indicates that these architectures have a better flexibility, although their cost effectiveness are not as favorable as Tree and its derivatives.

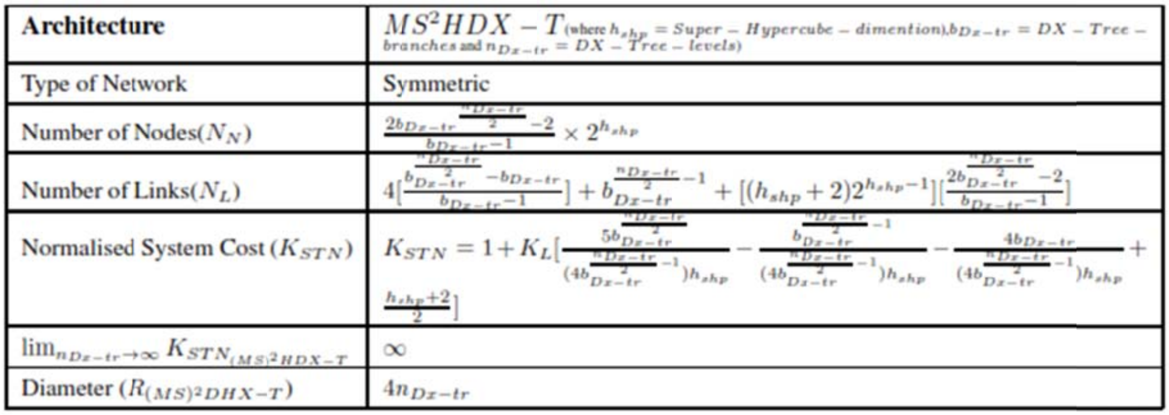
This satisfies ones expectations from the topology point of view

Acknowledgement

This work was supported by the Research Center of Computer and Information Sciences, King Saud University. The authors are grateful for this support.

References

1. H. El-Rewini and A. El-Barr, *Advanced Computer Architecture and Parallel Processing*, John Wiley 2005.
2. M. Amiripour and H. Abachi, *Average Routing Distance Analysis and Comparison of MasterSlave Super-Super-Hypercube 4-Cube Topology with different Message Passing Architectures* 6th IEEE/ ACIS International Conference on Computer and Information Science (ICIS 2007), Australia, pp. 622- 628 July 2007.
3. G. Alaghband and H. Jordan, *Fundamental of Parallel Processing*, Prentice Hall 2003.
4. M. Zargham, *Computer Architecture Single and Parallel System*. Prentice Hall,1996.
5. D.A. Reed et al, *Cost-Performance Bounds for Multi-Computer Network*, IEEE Trans.Comp, Vol. 32, No.1, 1998.
6. D.A. Reed, *Performance Based Design of MultiMicro Computer Networks*, Ph.D Dessertation, 1983.
7. J. Walker, *Performance and Cost Analysis of Message Passing Architecture*, Master of Engineering Thesis, Department of Electrical and Computer System Engineering, Monash University, Australia, Feb 1998.



Architecture	MS^2HDX-T (where $h_{shp} = Super - Hypercube - dimention$), $b_{Dx-tr} = DX - Tree - branches$ and $n_{Dx-tr} = DX - Tree - levels$)
Type of Network	Symmetric
Number of Nodes(N_N)	$\frac{2b_{Dx-tr}^{\frac{n_{Dx-tr}}{2}}-2}{b_{Dx-tr}-1} \times 2^{h_{shp}}$
Number of Links(N_L)	$4[\frac{b_{Dx-tr}^{\frac{n_{Dx-tr}}{2}}-b_{Dx-tr}}{b_{Dx-tr}-1}] + b_{Dx-tr}^{\frac{n_{Dx-tr}}{2}-1} + [(h_{shp}+2)2^{h_{shp}-1}][\frac{2b_{Dx-tr}^{\frac{n_{Dx-tr}}{2}}-2}{b_{Dx-tr}-1}]$
Normalised System Cost (K_{STN})	$K_{STN} = 1+K_L[\frac{5b_{Dx-tr}^{\frac{n_{Dx-tr}}{2}}}{(4b_{Dx-tr}^{\frac{n_{Dx-tr}}{2}-1})h_{shp}} - \frac{b_{Dx-tr}^{\frac{n_{Dx-tr}}{2}-1}}{(4b_{Dx-tr}^{\frac{n_{Dx-tr}}{2}-1})h_{shp}} - \frac{4b_{Dx-tr}}{(4b_{Dx-tr}^{\frac{n_{Dx-tr}}{2}-1})h_{shp}} + \frac{h_{shp}+2}{2}]$
$\lim_{n_{Dx-tr}\to\infty} K_{STN_{(MS)^2HDX-T}}$	∞
Diameter ($R_{(MS)^2DHX-T}$)	$4n_{Dx-tr}$

Table 4: (*MS*)^2*HDX-T* network metrics

8. M. Amiripour, H. Abachi and R. Lee, *Total System Cost and Average Routing Distance Analysis and Comparison of Master-Slave SuperHypercube 4-Cube Architecture*, International Journal of Computer and Information Science, Vol. 8, No.2, June 2007.
9. M. Amiripour, H. Abachi and K. Dabke, *Hardware Cost Analysis of Master-Slave Star-Ring and Master-Slave Super-Super-Hypercube 4Cube Architecture*, The 6th International Conference of Software Engineering, Parallel and Distributed Systems (SEPAD’07), Greece, Feb 2007.
10. H. Abachi, *Analysis of Network Metrics of Master-Slave Multi-Super-Hypercube X-Tree Architecture*, The 16th International WSEAS Conference on Computers (ICCOMP’12), Greece, July 2012.
11. R. Bajaj and D.J. Agrawall, *Improving Scheduling of Tasks in Hetergenous Environment*, IEEE Trans on Parallel and Distributed Systems, Vol. 15, No.2, 2004.
12. H. Abachi, *Cost Analysis of Master-Slave Multi-Super-hypercube DX-Tree Architecture*, The 14th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SPND 2013), USA, July 2013