Total System Cost Analysis of Different Classes of Master-Slave Super-Hypercube Message Passing Architectures

Hamid Abachi

Abstract- The need for high processing power, as well as advances in the semiconductor technology has resulted in the rapid development of High-Performance Message Passing Architectures (HPMPAs). The former claim is supported through excessive data processing requirements of today's advanced research topics which are contineously on the rise. These areas include global warming, weather forecasting and simulation of the performance, safety and reliability of nuclear weapons to name a few. This is one of the main reasons that many researchers attempt to develop new architectures and through modeling and simulations try to justify improvements in the areas such as, performance, speed, fault tolerance and cost which are determining factors and would identify the suitability or otherwise of a system for a given application. With this motivation in mind the author has introduced a new architecture as part of the HPMPAs, which is coined as Master -Slave Multi-Super Hypercube X-Tree architecture $(MS)^2HX-Tree$. For this architecture the total system cost through mathematical modeling and simulation are compared with the similar parameters of the existing High Performance Computing (HPC) systems. The result highlights any merits or demerits of the proposed architecture from scientific research point of view.

Keywords- Master slave architecture, Message Passing Architecture, Total System Cost, Super-Hypercube, X-Tree.

I. INTRODUCTION

THE evolution of semiconductor technology (from vacuum tube technology in 1940s to multi-core technology, to present) has resulted in advancement of the memory technology. This achievement together with software development and their applications have significantly contributed towards the advancement of computer industry. Evidently this has resulted in an increase in computer's capabilities including speed of operation and cost reduction. To this effect and in order to satisfy the market expectations and to further improve the speed and reliability of modern computers, one can implement different classes of parallel processing systems. It is evident that such implementation could be one of the technical solutions to many engineering problems that suffer from deficiencies in the areas of speed, performance, reliability and scalability [1]. One of the integral and critical aspects of the performance evaluation in the HPC is the cost analysis. Therefore the author has proposed a novel architecture based on the derivitives of the Tree architecture namely X-Tree topology.

II. MIMD ARCHITECTURE

Multiple- instruction multiple- data 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 [2]. 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 [3]. 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 [4] 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 complement the above parameters and fulfill the performance evaluation requirements. These include hardware system, architecture schemes, operating system, language, program and algorithm which can be considered as further work in this area [5]. To this effect, one of the main research areas of message passing architecture could be related to the simulation, modeling of parallel systems as well as analysis and compariosn of improved

Hamid Abachi is with the Department of Computer Engineering, King Saud University, Riyadh 11543 Kingdom of Saudi Arabia. Phone: 9664696302. Email:habachi@ksu.edu.sa

task scheduling as reported in [6], [7], [8].

III. 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.

A. COST METRICS

As reported in [3], the overall total system cost $(O-T-S)_C$ is dominated by the total node-related cost $(T-N-R)_C$and the total communication-link cost $(T-C-L)_C$ which leads to:

$$(O-T-S)_C = (T-N-R)_C + (T-C-L) - C \qquad (1)$$

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

$$(T-N-R)_C = (U-N)_C \times N_N \qquad (2)$$

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 $(T-C-L)_C$ is the product of the unit link cost $(U-L)_C$ and the number of links (N_L),

$$(T-C-L)_C = (U-L)_C \times N_L \qquad (3)$$

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 $(O-T-S)_C$ as being:

$$(O-T-S)_C = (U-N)_C \times N_N + (U-L)_C \times N_L \qquad (4)$$

However, a far more difficult task for the multiprocessor 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 $(O-T-S)_C$. We take $(U-N)_C$ as the base cost, because it is a constant from a system designer's point of view, since $(U-L)_C$ is likely to be a fraction of $(U-N)_C$. This is accomplished by scaling $(O-T-S)_C$ down by $(U-N)_C$.

$$K_{S-T} = \frac{(O-T-S)_C}{(U-N)_C} = N_N + K_L N_L \qquad (5)$$

where $K_L = \frac{(U-L)_C}{(U-N)_C}$. Therefore,$K_{S-T}$ gives the total system cost in units of $(U-N)_C$. 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_{S-T} 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_{S-T}which is coined as K_{S-T-N} for a better comparison between different network architectures. To this end we can express K_{S-T-N} as being:

$$K_{S-T-N} = \frac{K_{S-T}}{N_N} = 1 + \frac{K_L N_L}{N_N} \qquad (6)$$

The normalized cost function for message-passing architectures is summarized together with the rest of the parameters that are used for the system evaluation. For comparison purpose, the graphical presentation for different values of K_L, are included for completeness for different architectures.

IV. MESSAGE PASSING ARCHITECTURES

A. BINARY TREE ARCHITECTURE

Binary Trees are hierarchical structures that have some similarities to natural trees. For the definition purpose and ease of mathematical calculations, one can assume that the binary trees start with a node at the top called the root, this node is connected to other nodes by edges. These nodes may spawn further nodes forming a multilayered structure. Nodes at one level can only connect to nodes in adjacent levels. Furthermore, a node may only stem from one other node (it may only have one parent), even though it may give rise to several nodes (children). The connections are such that the branches are disjoint and are no loops in the structure [9]. Fig. 1 illustrates a binary tree structure.

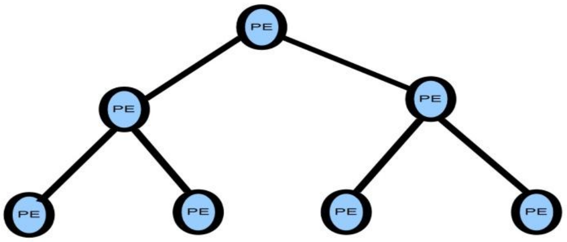


Fig. 1 Binary Tree Architecture

B. 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 structure in addition to providing alternative routing in the topology which results in the reduction of the traffic congestion, it also improves performance and diversity of the system [1], [9]. This scheme is called an X-Tree as shown in Fig. 2.

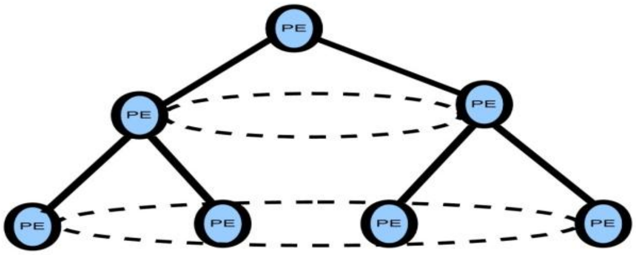
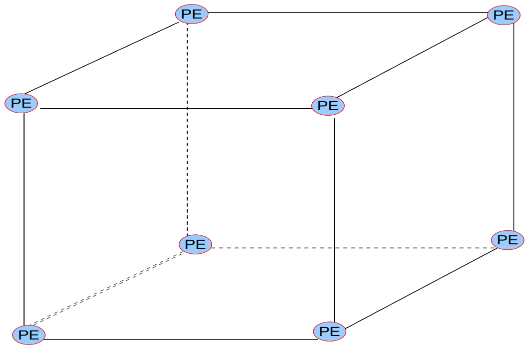


Fig.2 X-Tree Architecture

C. HYPERCUBE ARCHITECTURE

From Mathematical point of view, simple cubes have three dimensions, while Hypercubes are produced by increasing the number of dimensions in a cube. A four- dimensional cube can be thought of as two cubes whose corresponding vertices have been connected together [9]. The h- dimensional Hypercube is a special case of the torus, which has a fixed width of two (w = 2), and no edge connection folds. Hence the hypercube is expanded by increasing its dimensionality (h_{hp}) and has one less processor port than the corresponding Torus [4]. Table III shows the hypercube network metrics based on the above assumptions. A typical Hypercube topology is depicted in Fig. 3.

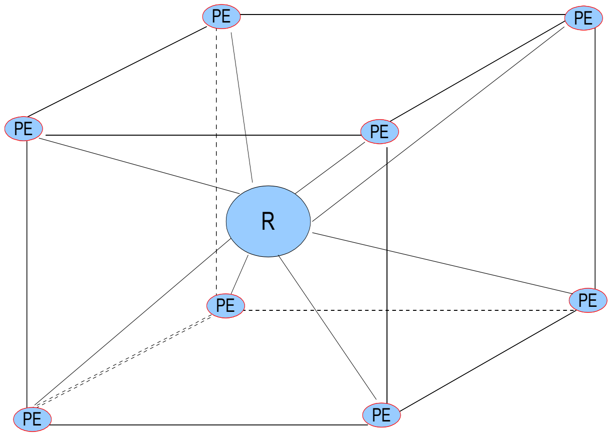


PE: Processor Element

Fig. 3 Hypercube Architecture

D. 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 [4]. 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. Fig. 4 shows the basic principle of this architecture.



PE: Processor Element, R: Router

Fig. 4 Super-Hypercube Architecture.

E. SUPER – HYPERCUBE ARRAY (SHA) ARCHITECTURE

As discussed earlier the Super Hypercube topology is originated by inclusion of a Router in the middle of the Hypercube topology that routes all indirect messages.

Further to this extended Hypercube is inclusion of Torus linkage of SHs to form Super- Hypercube Array (SHA) architecture as shown in Fig. 5, [10].

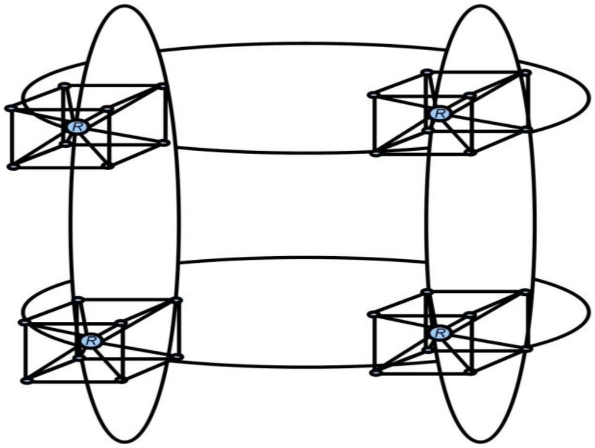


Fig. 5 Super Hypercube Array Architecture

F. TORUS ARCHITECTURE

The two dimensional torus is in fact structured as a regular mesh of processing nodes. In this architecture, Processing Elements (PEs) are linked in the two orthogonal dimensions and wrap around at the edges join the opposite sides. One derivate of the torus is the hexagonal grid. It provides higher reliability and performance at the expense of more communication links. Alternatively higher-dimensional Toruses are possible such as 3D torus. A Torus is combination of a ring and a mesh. The diameter is shorter than the mesh diameter [1]. A typical Torus architecture is shown in Fig. 6 .

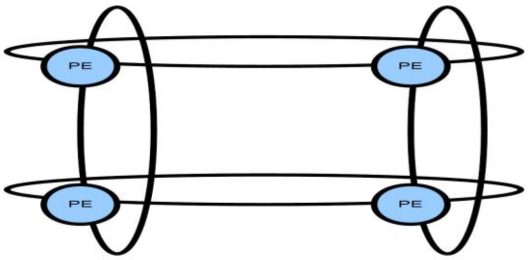


Fig. 6 Torus Architecture

V. NETWORK METRICS FOR EXISTING MESSAGE PASSING ARCHITECTURES

As reported in [10], 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 Tables I, II, III, IV, V and VI respectively.

Table I . Tree network metrics

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})	$2(n_{tr} - 1)$

Table II . X-Tree network metrics

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})	$2(n_{x-tr} - 1)$

Table III . Hypercube network metrics

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}

Table IV. Super-Hypercube network metrics

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

Table V . Super-Hypercube-Array network metrics

Architecture	**Super-Hypercube-Array (SHA)**
Type of Network	Symmetric
Number of Nodes(N_N)	$2^{h_{shp}} w_{trs}^{t_{trs}}$
Number of Links(N_L)	$(h_{shp}2^{h_{shp}-1} + 2^{h_{shp}})w_{trs}^{t_{trs}} + t_{trs}w_{trs}^{t_{trs}}$
Normalised System Cost ($K_{STN-SHA}$)	$1 + K_L(2^{-h_{shp}}t_{trs} + \frac{h_{shp}}{2} + 1)$
$\lim_{w\to\infty} K_{STN-SHA}$	$1 + K_L(2^{-h_{shp}}h + \frac{h_{shp}}{2} + 1)$
Diameter (R_{SHA})	$\frac{t_{trs}w_{trs}}{2} + 2$

Table VI . Torus network metrics

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})	$1 + K_L \frac{t_{trs}(w_{trs}^2)}{w_{trs}^2} = 1 + K_L t_{trs}$
$\lim_{w_{trs}\to\infty} K_{STN-Torus}$	$1 + K_L t_{trs}$
Diameter(R_{trs})	$t_{trs}\lfloor\frac{w_{trs}}{2}\rfloor$

VI. MASTER-SLAVE MULTI SUPER-HYPERCUBE X-TREE ARCHITECTURE

The $(MS)^2HX - T$ architecture which is depicted in Fig. 7 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. 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 direct alternative paths) between all the routers at all levels (not shown in the diagram).

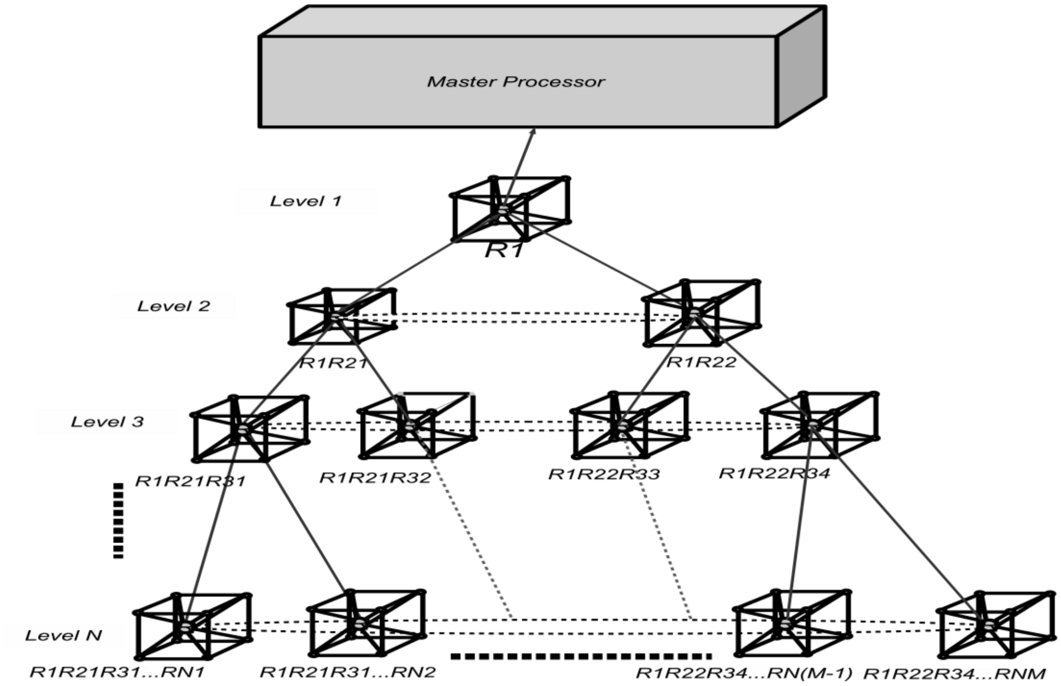


Fig. 7 Master- Slave Super-Hypercube X-Tree

VII. NODE ADDRESS FORMAT

The fundamental concept of this proposed architecture is based on the Super-Hypercube architecture . In this architecture, each node of the Tree architecture is presented by Super-Hypercube architecture and the Routers are connected according to the following pattern. This means R_1 at level 1 is connected to R_1R_{21} and R_1R_{22} at level 2. Then, the second level routers are connected to third level routers that include $R_1R_{21}R_{31}$ and $R_1R_{21}R_{32}$ and $R_1R_{22}R_{33}$ and $R_1R_{22}R_{34}$. This arrangement continues until all the routers are connected in the same manner to those at level N. The processing element within each SH (to the total of 8) is called satellite slave processors. The overall arrangement of this architecture is shown in Fig. 7 . As can be seen in this configuration, the addressing format for each node within any satellite processor starts with the suffix of the router of the respective SH. This is then followed by the slave processor's number. For example a string of labels could be presented as $R_1S_1, R_1S_2, R_1S_3, \cdots, R_1S_8$ that belongs to the SH at level 1. Following this pattern, the labeling arrangement for the first Super-Super-Hypercube at level N could be summerised as $R_1R_{21}R_{31}R_{41}\cdots R_{N1}S_1, R_1R_{21}R_{31}\cdots R_{N1}S_2, \cdots, R_1R_{21}\cdots R_{N1}S_8$. Finally, the last SH will have the addressing format such as $R_1R_{21}\cdots R_{NM}S_1, R_1R_{21}\cdots R_{NM}S_2,, \cdots, R_1R_{21}R_{34}R_{41}\cdots R_{NM}S_8$. The above node address format illustrates the general configuration for NM matrix in Multi Super-Hypercube Tree architecture

VIII. 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 [11], which facilitates the comparison between the model parameters of this topology and the remaining message-passing architectures including a developed Master-Slave Super-Super-Hypercube 4-Cube ($MS^3H4-Cube$). The Master- Slave Super-Hypercube X-Tree is a X-Tree with b branches and n- level in which nodes have been replaced by a SHP. In this case, the number of nodes in Master-Slave Super-Hypercube X-Tree ($N_{N_{(MS)^2SHP-X-T}}$) would be the number of nodes in X-Tree multiply by the number of nodes in SHP which simply results in having:

$$N_{N_{(MS)^2SHP-X-T}} = \frac{b_{xtr}^{n_{x-tr}}-1}{b_{x-tr}-1} \times 2^{h_{shp}}$$

where $b_{x-tr} = X - Tree - branches$, $n_{x-tr} = X - Tree - levels$ and $h_{shp} = Super - Hypercube - dimension.$

The number of links of Master-Slave Multi Super-Hypercube X-Tree would be calculated as follows:

$$N_{L_{(MS)^2HX-T}} = (number - of - linksX - Tree)$$

$+(number - of - links - SHP) \times (number - of - nodesX - Tree)$, Therefore,

$$N_{L_{(MS)^2HX-T}} = (2 \times \frac{b_{x-tr}^{n_{x-tr}} - b_{x-tr}}{b_{x-tr}-1} - 1)$$

$$+[(h_{shp} + 2)2^{h_{shp}-1}][\frac{b_{x-tr}^{n_{x-tr}}-1}{b_{x-tr}-1}].$$

Now,we proceed to compute the total system cost for $(MS)^2HX-T$. As reported in [4]: $K_{STN} = 1 + \frac{K_LN_L}{N_N}$ 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:

$$K_{STN} = 1 + K_L[\frac{2b_{x-tr}^{n_{x-tr}}-3b_{x-tr}+1}{2^{h_{shp}}(b_{x-tr}^{n_{x-tr}}-1)} + \frac{h_{shp}+2}{2}]$$

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

and the diameter is:
$R_{(MS)^2HX-T} = 2(n^{x-tr} - 1) + 2.$

Table VII illustrates the networks metrics for the $(MS)^2HX - T$ architecture.

Table VII . $(MS)^2HX - T$ network metrics

Architecture	$\boldsymbol{MS^2HX - T}$ (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})	$2(n_{x-tr} - 1)$

IX. MASTER-SLAVE SUPER-SUPER-HYPERCUBE 4-CUBE

A proposed architecture entitled Master-Slave Super-Super-Hypercube 4 Cube ($MS^3H4 - Cube$) architecture as reported in [10] is chosen for comparison purpose with the proposed model. The basic building blocks of this architecture is based on the Super-Hypercube architecture. To this effect, each processing element in each Super-Super-hypercube which contains the Router R_{11}, is itself a Super-Hypercube with the Router R_{111}.

The processing element with the Router R_{111} is called satellite slave. The management and control of the system is devoted to the master processor. These management tasks include but not limited to the allocation of subtasks, synchronization between satellite slave processors as well as collection of the results of operation plus responding to any interrupt result generated by the slave processors. Fig. 8 illustrates the interconnection of two Super-Super-Hypercube which results in construction of $MS^3H4 - Cube$ topology.

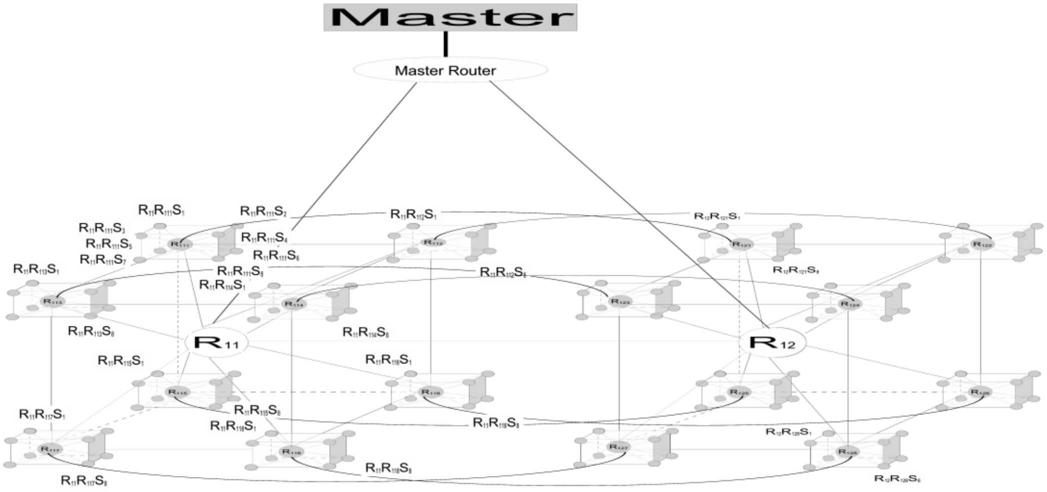


Fig. 8. The Master-Slave Super-Super-Hypercube 4-cube Architecture.

The addressing label for the processing element is the same as what was described for $(MS)^2HX-T$ architecture.

X. MATHEMATICAL MODELING OF $MS^3H4-Cube$ ARCHITECTURE

In $MS^3H4-Cube$ architecture, one could consider two Super-Super-Hypercube where its co-master processors, R_{11} and R_{12} are not connected together. This is followed by deriving the number of nodes (N_N) and number of links (N_L) for the above special case of $MS^3H4-Cube$, which is called Super-Hypercube 4-Cube, abbreviated as SH-4Cube (with central co-master processors not connected).

The SH-4cube is a Hypercube with dimension four ($h_{hp}=4$) in which every node has been replaced by a Super-Hypercube (SHP). So the number of nodes of SH-4cube ($N_{N_{SH-4cube}}$) would be calculated by multiplying the number of nodes in Hypercube ($h_{hp}=4$) by number of nodes in Super-Hypercube which simply results in;

$$N_{N_{SH-4cube}} = 2^{h_{hp}}.2^{h_{shp}} = 2^{h_{hp}+h_{shp}}$$

where $h_{hp} = hypercube-dimension$ and $h_{shp} = Super-Hypercube-dimension.$

The number of links of SH-4Cube would be calculated as follows:

$$N_{L_{SH-4Cube}} = \Big(number-of-links-Hypercube\big(h_{hp}=4\big)\Big) + (number-of-links-SHP) \times (number-of-nodes-Hypercube(h_{hp}=4)).$$

Therefore,

$$N_{L_{SH-4Cube}} = h_{hp}2^{h_{hp}-1} + [(h_{shp}+2)2^{h_{shp}-1}]2^{h_{hp}}$$
$$= 2^{h_{hp}-1}\big[h_{hp} + 2\big(h_{shp}+2\big)2^{h_{shp}-1}\big]$$
$$= 2^{h_{hp}-1}[h_{hp} + (h_{shp}+2)2^{h_{shp}}].$$

With these derived formulas for special case, one can extend them to cover metrics for $MS^3H4-Cube.$

A. CALCULATION FOR NUMBER OF NODES, NUMBER OF LINKS AND DIAMETER FOR $MS^3H4-Cube$ ARCHITECTURE

In $MS^3H4-Cube$ architecture, it is evident that since there is an extra link connecting the co-master level one to the co-master level two, therefore, [12]:

Number of links for satellite slave is:

$N_{L_{SatteliteSlave}}$ = number of links SHP + 1, or

$$N_{L_{Sattelite-Slave}} = h_{shp}(2^{h_{shp}-1}) + 2^{h_{shp}} + 1.$$

By careful consideration of the topology and interconnection scheme of $MS^3H4-Cube$, it is apparent that co-master units are connected in pairs. This assumption necessitates that the number of links (N_L) for a $MS^3H4-Cube$ to be:

$N_{L_{MS^3H-4cube}}$ = (number of links satellite slave).
×(number of nodes $MS^3H-4cube$)
+ (number of links SH-4cube)
+ (one link between each pair of co-masters)

Therefore,

$$N_{L_{MS^3H4-Cube}} = \big(h_{shp}\big(2^{h_{shp}-1}\big) + 2^{h_{shp}} + 1\big)2^{h_{hp}} + (h_{hp}2^{h_{hp}-1} + 2^{h_{hp}} + 1)$$

where number of nodes in $MS^3H4-Cube, N_{N_{MS^3H4-cube}} = 2^{h_{hp}}2^{h_{shp}}$

However, due to the fact that in $MS^3H-4cube$ architecture, there is a link between the master processor and each co-master processors, hence, this results in having:

$$N_{L_{MS^3H4-Cube}} = \big(h_{shp}\big(2^{h_{shp}-1}\big) + 2^{h_{shp}} + 1\big)2^{h_{hp}} + (h_{hp}2^{h_{hp}-1} + 2^{h_{hp}} + 3)$$

One of the most important advantages of this architecture is in relation to having a constant value of five for its Diameter.

For example, if node $R_{11}R_{111}S_1$ from satellite slave containing R_{111} needs to send a message to $R_{12}R_{128}S_8$ which belongs to satellite slave containing R_{128}, then the sequence of path segments used for this communication is $R_{11}R_{111}S_1 \rightarrow R_{111} \rightarrow R_{11} \rightarrow R_{12} \rightarrow R_{12}R_{128}S_8$ which simply means five segments. Therefore,

Diameter $(R_{MS^3H4-Cube}) = 5$

B. TOTAL SYSTEM COST FOR $MS^3H4-Cube$

Based on the cost metrics explained previously, we proceed to compute the total system cost for $MS^3H4-Cube$ as follows [12],

$$K_{STN} = 1 + \frac{K_L N_L}{N_N}$$

Using the values of N_L and N_N calculated earlier, we obtain,

$$K_{STN} = 1 + K_L\frac{\big(h_{shp}\big(2^{h_{shp}-1}\big) + 2^{h_{shp}} + 1\big)2^{h_{hp}} + \big(h_{hp}2^{h_{hp}-1} + 2^{h_{hp}} + 3\big)}{2^{h_{hp}}2^{h_{shp}}}$$
$$= 1 + K_L\frac{\big(h_{shp}\big(2^{h_{shp}-1}\big) + 2^{h_{shp}} + 1\big)\big(2^{h_{hp}}\big) + \Big(\big(h_{hp}2^{h_{hp}-1} + 2^{h_{hp}} + 3\big)\Big)}{2^{h_{hp}}2^{h_{shp}}}$$
$$= 1 + K_L\frac{2^{h_{hp}}\big(h_{shp}\big(2^{h_{shp}-1}\big) + 2^{h_{shp}} + 1\big) + \left(\frac{h_{hp}}{2} + 1 + \frac{3}{2^{h_{hp}}}\right)}{2^{h_{hp}}2^{h_{shp}}}$$
$$= 1 + K_L[(1 + (\frac{h_{shp}}{2})) + (\frac{1}{2^{h_{shp}}})[(\frac{h_{hp}}{2}) + (\frac{3}{2^{h_{hp}}}) + 2]]$$

Therefore, $\lim_{h_{hp}\to\infty} K_{STN_{MS^3H4-Cube}} = \infty$.

Table VIII illustrates the network metrics for $MS^3H4-Cube$ architecture.

Table VIII. $MS^3H4 - Cube$ network metrics

Architecture	$\boldsymbol{MS^3H4 - Cube}$
Type of Network	Symmetric
Number of Nodes (N_N)	$2^{h_{hp}}2^{h_{shp}}$
Number of Links(N_L)	$(h_{shp}(2^{h_{shp}-1}) + 2^{h_{shp}} + 1)2^{h_{hp}} + (h_{hp}2^{h_{hp}-1} + 2^{h_{hp}} + 3)$
Normalised System Cost (K_{STN})	$1 + K_L[(1 + (\frac{h_{shp}}{2})) + (\frac{1}{2^{h_{shp}}})[(\frac{h_{hp}}{2}) + (\frac{3}{2^{h_{hp}}}) + 2]]$
$\lim_{h_{hp}\to\infty} K_{STN-MS^3H4-Cube}$	∞
Diameter($R_{MS^3H4-Cube}$)	5

Fig. 9 illustrates the Normalised System Cost (K_{STN}) for all described network architectures.

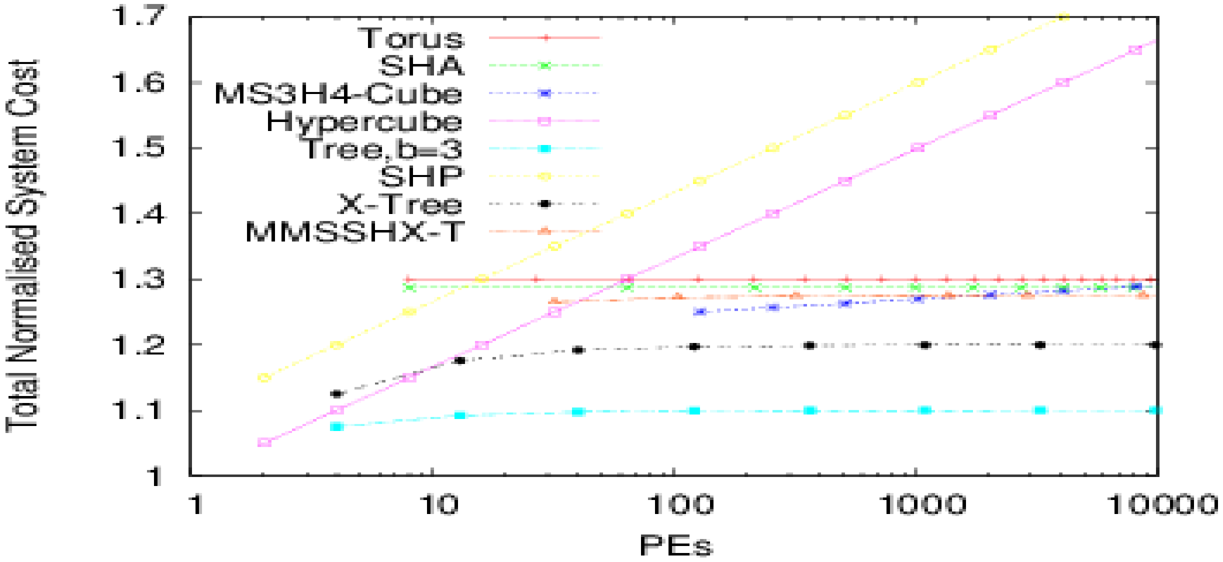


Fig. 9. Normalised System Cost for different Networks

XI. CONCLUSION

This paper addresses the total system cost of different message passing architectures. To achieve this, a number of the common massage passing architectures such as Torus, Tree, Hypercube and super Hypercube and Super Hypercube Array plus two advanced and contemporary architectures are included for comparison purposes. These contemporary architectures include master-slave super Hypercube X-Tree and master-slave super -super Hyper Cube 4 cube. Fig. 9 illustrates the graphical presentation of the normalized system cost of all message passing architectures discussed in this paper. A careful analysis of this graphical presentation discloses that the normalized system cost of Tree and its derivatives plus those two contemporary architectures which are coined as $(MS)^2HX - T$ and $MS^3H4 - Cube$ are more favorable when are compared with that of Hypercube and Super Hypercube. This claim is due to the fact that all the architectures except HP and SHP enjoy having a relative constant K_{STN} for a large range of processing elements. The demerit that HP and SHP experience is mainly related to an increasing proportion of the system cost that is devoted to the architectural topology and the lack of scalability feature that hypercube and superhypercube architectures in general suffer from. This finding from the practical point of view justifies direct reflection of the scalability on the communication overhead.

REFERENCES

[1] H.Abachi, J.Walker " Reliability Analysis of Tree, Torus and Hypercube Architectures"Proceeding of the 9th IEEE South-Eastern Symposium on System Theroy (SSST), Tennesse, U.S.A, pp. 44-48 March 1997.

[2] H.El-Rewini and A.El-Barr, "*Advanced Computer Architecture and Parallel Processing"* John Wiley 2005.

[3] M.Amiripour and H.Abachi, "Average Routing Distance Analysis and Comparison of Master-Slave 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.

[4] 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.

[5] G. Alaghband , F.Jordan,"*Fundamentals of Parallel Processing*" Prentice Hall,2003.

[6] M.Amiripour, H.Abachi, "Simulation, Modeling and Analysis of Task scheduling on Message Passing Architectures" *WSEAS International Journal of Mathematical and Computers in Simulation,* Issue 2, Volume 2, pp. 118-124, 2008

[7] M.Amiripour, H.Abachi, "Time Complexity of a Matrix Product on Message Passing Architectures" *The 7th WSEAS International Conference on applied computer and applied computational science* (acacos '08), Hangzhou, China, April 2008.

[8] Jasbir Singh and Gurvinder Singh "Improved Task Scheduling on parallel System Using Generic Algorithm*" International Journal of Computer Applications"* volume 39- No. 17 February 2012.

[9] Hwang K, Briggs F, "Computer Architecture and Parallel Processing " Mc Graw Hill London 1984.

[10] M.Amiripour, H.Abachi and R.Lee, "Total System Cost and Average Routing Distance Analysis and Comparison of Master-Slave Super-Hypercube 4-Cube Architecture" *International Journal of Computer and Information Science,* Vol. 8, No.2, June 2007.

[11] H.Abachi, "Analysis of Network Metrics of Master-Slave Multi-Super-Hypercube X-Tree Architecture" *The 16th WSEAS International Conference on Computers (ICCOMP12),* Greece, July 2012.

[12] M.Amiripour, H.Abachi and K.Dabke, "Hardware Cost Analysis of Master-Slave Star-Ring Super-Hypercube and Master-Slave Super-Super-Hypercube 4-Cube Architectures" *The 6th Wseas International Conference on Software Engineering, Parallel and Distributed Systems* (SEPAD'07), Greece, Feb 2007.

Hamid Abachi received his Ph.D. degree in Electrical and Computer Systems Engineering from University of Wales in Britain in 1981. From 1981 to 1982 he worked as a Postdoctoral fellow at University of Wales. He has been in academic life for more than 30 years as well as working and gaining practical experiences in heavy to light industries. For more than 18 years he has held a faculty position in the Department of Electrical and Computer Systems Engineering, at Monash University in Australia where for more than eight years he served as the Director of the International Program, and Director of Postgraduate (Coursework) studies as well as Director of Professional Development Programs.

Since 2008 to present he is in the department of Computer Engineering at King Saud University in Saudi Arabia contributing towards the accreditation program. He was a member of the Editorial Board of the IEEE Systems Journal in the USA and WSEAS Transactions on Computer Research. He has been a keynote speaker at many international conferences. In addition, He also has been a member of Technical Program Committees and a reviewer to more than 75 international conferences where in a number of occasions he has been invited to serve as the conference chair. He has been a Fellow of IET (formally IEE, UK) and a Fellow of IEAust (Engineers Australia) and also a Senior Member of the IEEE, USA.

His research areas include the modelling and simulation of Parallel Processing Systems, Design of Advanced Computer Architectures, and Engineering Education. He is author and co author of many journal and international conference papers in these areas.