

Hardware Cost Analysis of Master-Slave Star-Ring Super-Hypercube and Master-Slave Super-Super-Hypercube 4-Cube Architectures

M. Amiripour, H. Abachi and K. Dabke
 Monash University
 Department of ECSE
 Wellington Rd, Vic
 AUSTRALIA

Abstract: This paper describes and compares the hardware cost analysis of two newly proposed topologies. The fundamental structure of these architectures is outlined to help define the number of nodes and the number of links in each architecture. This leads to the derivation of the total system costs. Simulation and mathematical modeling of these architectures are presented and the cost comparisons are given in a graphical form which highlights the merits and demerits of each topology.

Key-Words: Hardware Cost, Multiprocessor System, Super-Hypercube, MS^2RSH and MS^3H4 -Cube architectures.

1 Introduction

During the last decade, computing systems have become a necessary engineering tool for scientists to understand and solve complex problems. These include space programs, military applications, nanotechnology, weather forecasting, image and signal processing. These applications are usually characterized by massive and long-running application programs. In addition, the larger a system becomes, the more difficult it is to make it more reliable and efficient. Therefore, reliability, availability, flexibility, compatibility and fault tolerance, cost reduction and scalability of a distributed system will become important aspects in many computing environments. As part of performance evaluation, the authors have tackled a critical aspect of this evaluation, namely cost analysis of two proposed topologies in message passing architectures. The main aim of this research is to identify and highlight the advantages these architectures offer and the ways in which they are most suited to different applications.

2 Master-Slave Star-ring Super-Hypercube Architecture

In this paper the authors have developed two new architectures for cost analysis and comparison purposes. The first topology is called the Master-Slave

Star-Ring Super-Hypercube Architecture which is abbreviated as MS^2RSH and the second scheme, due to the nature of its topology, is named Master-Slave Super-Super-Hypercube 4-Cube architecture which is abbreviated as MS^3H4 -Cube. The former architecture which represents a true multiprocessing method consists of the combination of star and ring topology. In this architecture (Figure 1), the master processor is at the centre 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 2. Super-Hypercube topology is a Hypercube obtained by the addition of a router in the middle of the Hypercube with one connection to each node. One problem with the traditional hypercube is when communication between two indirectly connected nodes is required [1]. This occurs when a message has to travel along one or more hyper-planes which indicates that it must go through intermediate nodes, before reaching its destination. Thus, each processing node is required to compute and handle message routing, which reduces performance. One of the solutions to this problem is to use a Router (R) that routes all indirect messages.

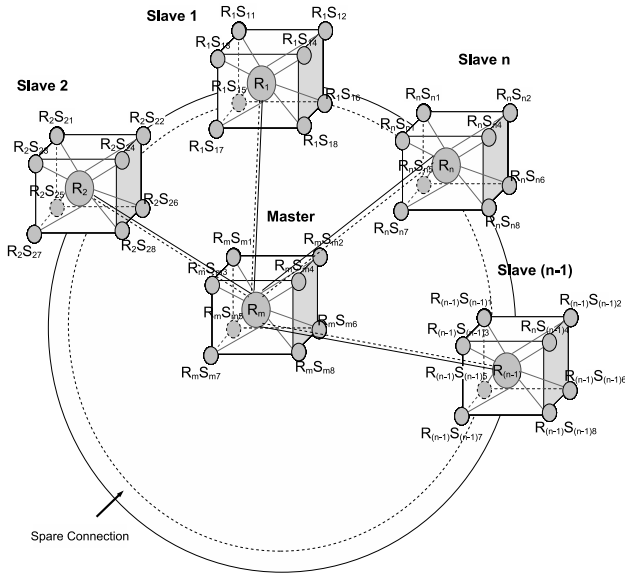


Figure 1: Master-Slave Star-Ring Super-Hypercube Architecture

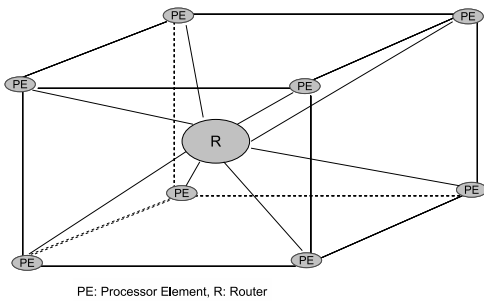


Figure 2: Super-Hypercube Architecture

2.1 MS^2RSH Network Modeling

2.1.1 Number of Links and Number of Nodes Calculations

For the MS^2RSH , there is a star of SHP where the total number of processing nodes (N_N) is given as the product of Super-Hypercube and star nodes:

$$N_{N_{MS^2RSH}} = 2^{h_{shp}} \times n$$

where n is the number of star nodes. To calculate

the number of the communication links (N_L) for the MS^2RSH topology, one has to consider the existence of the Router and its connectivity. To this end one can partition the proposed model into a star, ring and SHPs. In the MS^2RSH architecture as shown in Figure 1, there are n SHPs, and the number of links in the star configuration is $n - 1$. So for the ring connection the number of links is $n - 1$ (bearing in mind that star topology has n nodes). If each SHP has N_{LSHP} links, then the total number of links is:

$$N_{L_{MS^2RSH}} = 2 \times (n - 1) + n \times N_{LSHP}.$$

To calculate the total number of links within a SHP topology, one needs to consider the following cases:

- The number of links connecting processing elements together which is $h_{shp}2^{h_{shp}-1}$ and
- additional number of connections from Router to each processing element which yields $2^{h_{shp}}$. Adding these two combinations results in:

$$N_{LSHP} = h_{shp}2^{h_{shp}-1} + 2^{h_{shp}}.$$

This gives an expression for the number of links for the entire MS^2RSH , which is dependent on the number of star nodes, and SHP dimensionality, This results in

$$N_{L_{MS^2RSH}} = (h_{shp}2^{h_{shp}-1} + 2^{h_{shp}})n + 2(n - 1).$$

2.1.2 MS^2RSH Hardware Cost Analysis

Since in any computer systems' engineering environment, component cost dominantly depends on the economic factors, they play an important role on in the determination of the cost parameter. Therefore, it is difficult to precisely define it. In general, overall total system cost estimate (C_{TSC}) will be dominated by the total node related cost (C_{TN}) and the total communication link cost (C_{TL}) [2]. Therefore,

$$C_{TSC} = C_{TN} + C_{TL}.$$

In this cost analysis, it is assumed that each node consists of a Central Processing Unit (CPU), memory unit, and I/O ports which provide interfacing to the network. The total node cost is the product of the unit node cost (C_N) and the number of nodes (N_N). Hence the total processing node cost is $C_{TN} = C_N N_N$.

The most common assumption in identifying the nature of the link is to consider it as being in the form of an interconnection medium such as parallel wires which join the nodes together. This is one of the most appropriate modes of receiving and transmitting signals in a communication environment. The total communication link cost is the product of the unit link cost (C_L) and the number of links (N_L) connecting nodes, which gives $C_{TL} = C_L N_L$.

Therefore, the total system cost is

$$C_{TSC} = C_N N_N + C_L N_L.$$

However, applicability of a system is normally evaluated in terms of its suitability and the cost effectiveness of the architecture for a given application. This is evaluated in terms of the total system cost compared to the total processing node cost. One can make this assumption because such a figure describes how close a particular network is to the ideal lowest cost network where there are no communication link overhead costs (i.e. $C_L = 0$, giving $C_{TSC} = C_N N_N$). Thus one can normalise the total system cost function C_{TSC} , by $C_N N_N$. Therefore,

$$K_{ST} = \frac{C_{TSC}}{C_N N_N} = \frac{K_L N_L}{N_N}$$
 where $K_L = \frac{C_L}{C_N}$.

The normalised total system cost, K_{ST} , then gives us the total system cost relative to the lowest theoretical system cost. In practice K_L will vary from near zero for a tightly coupled multi-processor system to less than one for a distributed computer network. For tightly and closely coupled multi-processor systems, it has been suggested that $K_L = 0.1$ is a reasonable value [3]. We shall explore normalised system cost for values of $K_L = 0.1$. Based on the result of N_L for MS^2RSH , the K_{ST} can be formulated as follows:

$$K_{ST_{MS^2RSH}} = 1 + \frac{K_L N_L}{2^{h_{shp}n}}$$

therefore,

$$K_{ST_{MS^2RSH}} = 1 + K_L \frac{[(h_{shp}2^{h_{shp}-1} + 2^{h_{shp}})n + 2(n-1)]}{2^{h_{shp}n}}$$

After further simplification, $K_{ST_{MS^2RSH}}$ can be expressed as

$$K_{ST_{MS^2RSH}} = 1 + K_L \left[\frac{h_{shp}}{2} + 1 + \frac{(n-1)}{2^{h_{shp}-1}n} \right].$$

3 Master-Slave Super-Super-hypercube 4-Cube Architecture

The fundamental concept of this new architecture is based on the Super-Hypercube architecture. In this architecture each processing element in each Super-Super-Hypercube which contains the master Router R_{11} , is itself a Super-Hypercube with the Router R_{111} as shown in Figure 3.

The processing elements surrounding Router R_{111} is called satellite slave. In general, the overall control and management of the system is carried out by a master processor labeled "Master". For this reason the overall architecture is called Master-Slave Super-Super-Hypercube 4-cube architecture (MSSSH4-Cube), or further abbreviated as MS^3H4 -Cube.

As can be seen from Figure 3, the addressing label for the processing element of each satellite slave processor starts with the suffix of the main router (R_{11}) which belongs to that Super-Super-Hypercube

and then is followed by the suffix of the router within the satellite slave processor (R_{111}). This is then followed by the slave processor number. For example, a string of labels could be presented as $R_{11}R_{111}S_1, R_{11}R_{111}S_2, R_{11}R_{111}S_3, R_{11}R_{111}S_4, \dots, R_{11}R_{111}S_8$ for the satellite slaves of the first Super-Super-Hypercube configuration in the top left corner in Figure 3. The building block of each processing element and the router proposed for these new architectures are based on the technology developed by Silicon Graphics Inc (SGI). It is believed that the availability of the existing products and the continuous support for the future upgraded products will provide a suitable test-bed for this investigation. Although SGI has many products that are specifically designed to be used in multiprocessor environment, the most recent and suitable architecture (SGI 4700 series) has been recommended for this architecture [4]. The operation of MS^3H4 -Cube architecture 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. The master processor is intended to have the latest version of the processing element and the largest memory capacity in order to have full control of the overall system management. Once the main task is divided into multiple subtasks, then it is placed in the main memory of the master processor.

As the practical option one considers that each Router ($R_{11}, R_{111}, \dots, R_{11}, R_{118}, R_{12}, R_{121}, \dots, R_{12}, R_{128}$) has processing capability and memory facility, so that the subtask could be saved in the memory of each Router (for example first in R_{11} and then in R_{111}), before reaching its final destination in $R_{11}R_{111}S_1$. In reality, one can consider the Routers as co-master processors in this arrangement.

The advantage of this assumption is that if there is hardware or software fault within the master processor, the overall system is not subject to a catastrophic failure. The transfer of information can take place through a direct connection that is provided for this purpose. Implementing this approach facilitates the following task allocation in the proposed architecture.

First the main task is divided into multiple subtasks and then in blocks (normally multiples of eight, since each satellite slave configuration consists of eight processing elements) which are allocated to each Router (co-master processor) within each Super-Super-Hypercube, namely to R_{11}, R_{12} .

Then these co-master processors would transfer a minimum of eight subtasks to each corresponding routers (second level co-master processors) within the

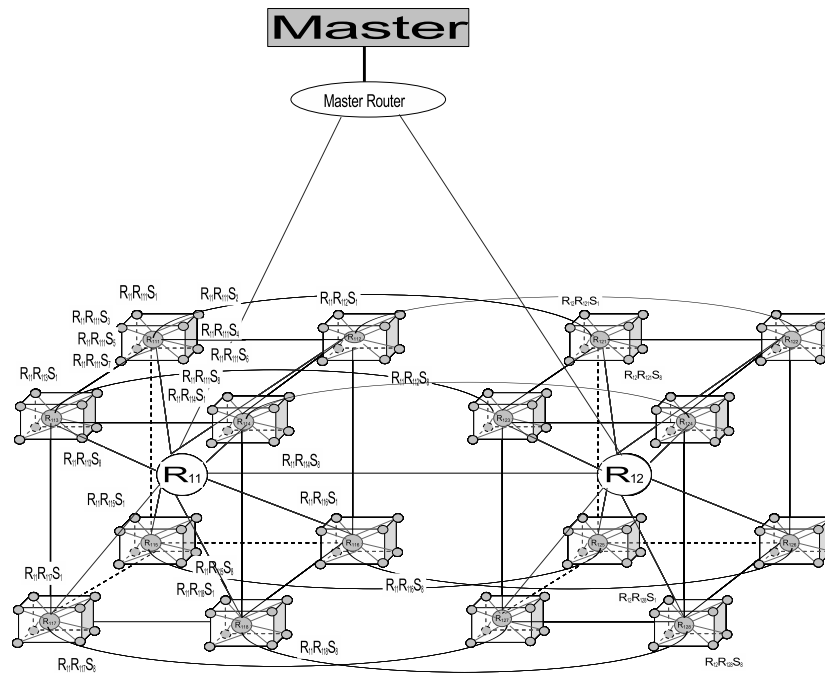


Figure 3: $MS^3 H4$ -Cube Architecture.

satellite slave configuration i.e. $R_{111}, \dots, R_{118}, R_{121}, \dots, R_{128}$. Once these subtasks are allocated to the second level co-master processors then the next phase would be their allocations to the processing elements (namely $R_{11}R_{111}S_1, R_{11}R_{111}S_2, \dots, R_{11}R_{111}S_8, R_{12}R_{121}S_1, R_{12}R_{122}S_1, \dots$ and $R_{12}R_{128}S_8$).

These processing elements $R_{11}R_{111}S_1, R_{11}R_{111}S_2, \dots, R_{12}R_{121}S_1, R_{12}R_{121}S_2, \dots, R_{12}R_{128}S_1, \dots, R_{12}R_{128}S_8$ (for overall configuration) would then start simultaneous execution of these subtasks. However, upon completion of their current subtask, processing elements would send an interrupt request signal through $R_{111}, R_{112}, R_{113}, \dots, R_{118}, R_{121}, R_{122}, \dots, R_{128}$ to R_{11} and R_{12} to request the allocation of the next available subtask. After completion of each subtask the results are saved in second level co-master processors as well as in the co-master processors and also in the main memory of the master processor. This precaution is taken to minimize a catastrophic failure that could result if the master fails. This procedure of sub task allocation would continue, until all the subtasks that are currently allocated to each co-master processor are allocated to and executed by the processing elements. Once co-master processor runs out of the subtasks, then a new set of subtasks would

be allocated by the main master to each co-master processor R_{11} and R_{12} and this procedure would continue until all the subtasks are completed by the processing elements. In the case when the sub-tasks are independent of one another and the execution time is the same, then that could result in simultaneous interrupt request signals arriving at the second level co-master processors and consequently by the co-master processors. This simply indicates request for the allocation of new subtask to each processing element. In this situation, through either hardware or software arrangements, the system designer can allocate interrupt priority arrangement which could reside in the master and co-master processors so that each processing element would receive a new subtask according to the allocated priority scheme. In order to further improve the allocation of sub-tasks and consequently the performance of the overall system, one can incorporate some degree of intelligent decision-making, for example, using an expert system or a neural network within the operating system of the master processor [5]. So this arrangement would provide an automatic allocation of the tasks and therefore other control and management procedures would be significantly improved.

3.1 Mathematical Modeling of MS^3H4 -Cube Architecture

3.1.1 Number of Nodes, Number of Links Calculation for MS^3H4 -Cube

In this model since every Super-Super-Hypercube has a co-master which is connected to the router of each satellite slave configuration, then, each satellite slave has one extra link which is connected to co-master processor. Therefore, number of links for satellite slave is:

$$N_{L_{\text{Satellite-Slave}}} = \text{number of links - SHP} + 1$$

or,

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

Moreover, since in this configuration, the co-master processors of all Super-Super-Hypercubes are connected together in pairs, therefore, the number of links N_L for a MS^3H4 -Cube is:

$$\begin{aligned} N_{L_{MS^3H4-Cube}} &= (\text{number of links - satellite - slave}) \\ &\times (\text{number of nodes}(MS^3H4 - Cube)) \\ &+ (\text{number of links - SH - 4cube}) \\ &+ (1 - \text{link - between - co - masters}). \end{aligned}$$

Therefore,

$$\begin{aligned} N_{L_{MS^3H4-Cube}} &= (h_{shp}(2^{h_{shp}-1}) + 2^{h_{shp}} + 1)2^{h_{hp}} + \\ &(h_{hp}2^{h_{hp}-1} + 2^{h_{hp}} + 1) \text{ (without considering master processor)} \\ &\text{where number of nodes } MS^3H4 - Cube, \\ N_{N_{MS^3H4-cube}} &= 2^{h_{hp}}2^{h_{shp}}. \end{aligned}$$

Since, in our model, there is a connection between the master processor and each co-master processor, hence we have:

$$\begin{aligned} N_{L_{MS^3H4-Cube}} &= (h_{shp}(2^{h_{shp}-1}) + 2^{h_{shp}} + 1)2^{h_{hp}} \\ &+ (h_{hp}2^{h_{hp}-1} + 2^{h_{hp}} + 3). \end{aligned}$$

3.1.2 Total System Cost for MS^3H4 -Cube

Based on the cost metrics formula, the total system cost for MS^3H4 -Cube is as follows:

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

Using the values of N_L and N_N from section 3.1.1 we obtain,

$$\begin{aligned} K_{STN} &= 1 + K_L \frac{(h_{shp}(2^{h_{shp}-1}) + 2^{h_{shp}} + 1)2^{h_{hp}} + (h_{hp}2^{h_{hp}-1} + 2^{h_{hp}} + 3)}{2^{h_{hp}}2^{h_{shp}}} \\ &= 1 + K_L \frac{(h_{shp}(2^{h_{shp}-1}) + 2^{h_{shp}} + 1)(2^{h_{hp}}) + ((h_{hp}2^{h_{hp}-1} + 2^{h_{hp}} + 3))}{2^{h_{hp}}2^{h_{shp}}} \\ &= 1 + K_L \frac{2^{h_{hp}}(h_{shp}(2^{h_{shp}-1}) + 2^{h_{shp}} + 1) + (\frac{h_{hp}}{2} + 1 + \frac{3}{2^{h_{hp}}})}{2^{h_{hp}}2^{h_{shp}}} \\ &= 1 + K_L [(1 + (\frac{h_{shp}}{2})) + (\frac{1}{2^{h_{shp}}})][(\frac{h_{hp}}{2}) + (\frac{3}{2^{h_{hp}}}) + 2]. \end{aligned}$$

Table 1 shows the MS^2RSH and MS^3H4 -Cube networks metrics.

Table 1: MS^2RSH and MS^3H4 -Cube networks metrics.

Architecture	MS^2RSH	$MS^3H4-Cube$
N_N	$2^{h_{shp}}n$	$2^{h_{hp}}2^{h_{shp}}$
N_L	$(h_{shp}2^{h_{shp}-1} + 2^{h_{shp}})n + 2(n-1)$	$(h_{shp}(2^{h_{shp}-1}) + 2^{h_{shp}} + 1)2^{h_{hp}} + (h_{hp}2^{h_{hp}-1} + 2^{h_{hp}} + 3)$
K_{ST}	$1 + K_L [\frac{h_{shp}}{2} + 1 + \frac{(n-1)}{2^{h_{shp}-1}n}]$	$1 + K_L [(1 + (\frac{h_{shp}}{2})) + (\frac{1}{2^{h_{shp}}})][(\frac{h_{hp}}{2}) + (\frac{3}{2^{h_{hp}}}) + 2]$

Figure 4 shows the total system cost for MS^2RSH and MS^3H4 -Cube architectures.

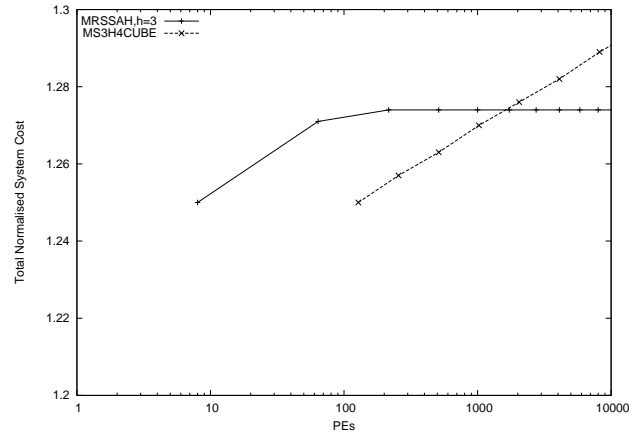


Figure 4: Total System Cost for MS^2RSH and MS^3H4 -Cube with $K_L = 0.1$ as a function of processing elements (PEs).

4 Conclusion

In this paper two newly proposed and developed architectures, namely Master-Slave Star-Ring Super-Hypercube (MS^2RSH) and Master-Slave Super-Super-Hyper 4-Cube (MS^3H4 -Cube) architectures, are presented. The architectures and their operations are described. This is followed by a mathematical model for their cost evaluations. Mathematical simulations for these architectures and the results of the total system cost for the case of $K_L = 0.1$ are depicted in Figure 4. As can be seen from this, the starting number of processing nodes in MS^3H4 -Cube due to the nature of the topology is approximately 100. With this fact in mind the total normalized sys-

tem cost from 100 up to 2000 processing elements (PEs) for MS^3H4 -Cube topology is superior to that of MS^2RSH architecture. On the other hand, in these architectures when the number of PEs exceeds 2000 it can be seen that the normalized system cost rises linearly with system size (for MS^3H4 -Cube topology) which is in contrast to saturating cost for the MS^2RSH topology.

References:

- [1] J. Walker. J, *Performance, Reliability and Cost Analysis of Message Passing Architecture*, Master of Engineering Thesis, Department of Electrical and Computer System Engineering, Monash University, Melbourne 1998.
- [2] H. Abachi, R. Lisner and N. Debnah, Parallel Processing Modelling Methodology in Computer Engineering, *13th International Conference of Computer Architecture*, ISCA, U.S.A. 2000.
- [3] D. A. Reed, *Performance Based Design and Analysis of Multimicrocomputer Networks*, PhD dissertation, 1983, pp. 62-64.
- [4] Sillicon Graphics Inc, *Hardware: End-User, Altix 3700 Bx2*, System Overview, Chapter 3, U.S.A, 2004, PP.1-6.
- [5] F. H. Jordan, G. Alaghband, *Fundamentals of Parallel Processing*, Prentice Hall, 2003.