International Journal of Research in Advent Technology, Vol.8, No.1, January 2020
E-ISSN: 2321-9637
Available online at www.ijrat.org

Performance Evolution of Efficient Cluster Based Multiprocessor Systems with Analytical Model

Vanitha Kakollu, K Yasudha, K Naga Soujanya

Abstract—Modern society is highly engaged with the High-speed computing, low cost architectures, big data analysis, Internet of Things (IOT), and Cloud computing. All these applications are operated and controlled with multiprocessor systems. Day by day, as the requirements are growing parallelly the complexities increase in system maintenance and data handling. The multiprocessor system alone could not satisfy all the present system complexities. These problems can be minimized with Cluster based Multiprocessor system. There are some effective existing interconnection networks playing vital role in managing nodes in individual cluster and interconnection between clusters. It is observed that some networks are having less reliability in computational recital. In the present work a new network model is proposed with cluster-based system and reliability is evaluated with analytical methods. The proposed work is compared with Ring, Multi Bus System. The proposed network has great advantage in overcoming existing system flaws like overflow, dead locks, and network interconnection failures.

Index Terms—Cluster, Reliability, Network protocols, Network Architectures.

I. INTRODUCTION

In recent decades the demand of high-performance parallel computing increasing day by day. In same proportion the challenges in parallel processing also increasing. High performance represents fast and reliable operations in between processors, clusters and networks. The parallel processing one in which heterogeneous operations are performed simultaneously with high reliability and with minimum number of failures. There are essentially few methods are adopted in implementing multiprocessing such as, software evolution, hardware evolution, and architecture design.

In Software Evolution, operating systems and programming languages plays a vital role in executing multiple tasks simultaneously. There are many efficient algorithms actively involved in control the program execution flow to do the multiple tasks simultaneously. The phenomenon is called multiprogramming. In Multi-program mode several programs execute their sequence of code at

instantaneously and do processor tasks concurrently. Handling multitasking in multiprocessor is a difficult task and need more complex algorithms. In multi programming several programs executes at the same time and multiple tasks are distributed among different programs. In cluster-based multiprocessor system different techniques are required to synchronize the operations between multiprocessors and clusters.

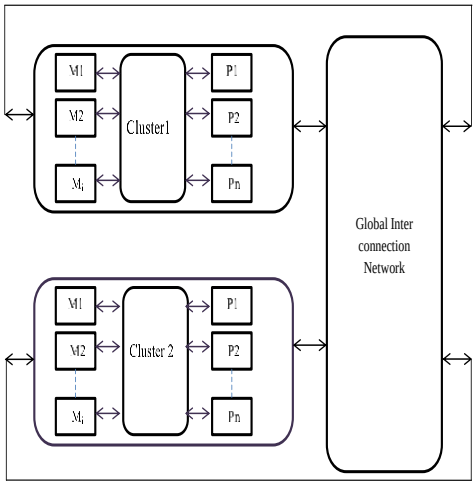

Fig.1. Cluster based Multiprocessor System

The Hardware Evolution is considered in terms of hardware interconnections. Multiple processors are interconnected to share multiple tasks. Different interfaces and topologies are required in between heterogeneous multiprocessors. While programming different processors, care must be taken in terms of source and destination addressing, cluster interconnections. In Multiprocessor systems numerous processors are execute to meet the demands of the system. Multiprocessors are interconnected to meet the demands such as server-client relation, space research, and many more applications of Internet of Things. In cluster-based multiprocessor system several clusters are interconnected with the help of different bus topologies. Each cluster can contain homogeneous and heterogeneous microprocessors. Each cluster contains different Memory modules. In each cluster multiple processors share a common memory module. Sharing a common memory block by multiprocessor system causes a problem called bottle neck. If the memory module is divided into sub modules it is very easy and fast for accessing

Manuscript revised February 5, 2020 and published on February 10, 2020
***Vanitha Kakollu**, Department of CS, GIS, GITAM (Deemed to be University), Visakhapatnam, India, vanithagitam@gmail.com*
***K Yasudha**, Department of CS, GIS, GITAM (Deemed to be University), Visakhapatnam, India, yasudha.p@gmail.com*
K. Naga Soujanya, *Department of CSE, GIT, GITAM (Deemed to be University), Visakhapatnam, India, kn.souji@gmail.com*

doi: 10.32622/ijrat.81202015

sub module by individual processor. This can also minimize bottle neck in clusters. In cluster-based multiprocessor system all memory modules in clusters are interconnected through common interconnection network as shown in figure 1. These connections are established with memory module bus. In real time there may be a chance of failure in these connections. Failure either memory module bus or interconnection network leads total failure in the system. This is one of the reasons for poor reliability in cluster-based multiprocessor system.

The modification of architectural design is another method to control multiple operations in parallel computing. The modification at architectural level such as pipeline enhancement, clock optimization, and on chip device integration gives high performance in parallel computing. In many applications effective clock scheme at pipeline stages plays an important role in improving reliability. The pipeline is one of the important elements integrated in microprocessor for fast and simultaneous data processing. If the pipeline stages enhance the performance and reliability of a single processor in a cluster, the same phenomena can be applied in interconnection network of cluster-based multiprocessor system. When compared with ring and multi bus systems the cluster-based multiprocessor system has low reliability. This can be overcome with modification in cluster-based multiprocessor system.

In the present research work, communication between internal modules, clusters, interconnection networks, and interfacing different hard ware components have been motivated to design highly reliable system.

Then the paper was organized as section by section, the first section introduction, which deals the need of multiprocessor system and cluster, based systems. Section 2 discusses about cluster based related work with different structures. Section 3 assesses the reliability of the present systems such as ring, star, star ring, and multi-bus-based cluster systems. Section 4 describes about the proposed new architecture design operations. Section 5 includes conclusions, which presents comparisons between different systems.

II. RELATED WORK

In the present work the individual cluster is constructed with homogeneous nodes and they are using interconnection network for sharing data. Each node in the cluster contains homogeneous processor, memory module, I/O, and network drivers. When any processor wants to communicate any other processor in the cluster it places a request or response signals on to the system bus. The system bus carries these signals to the respective target processor. In the same way if any cluster wants of sending message to other cluster on the network, the source cluster places request on the interconnection bus. Hence, the nodes which are joined or interconnected together through the interconnection bus will carry the source request to the destination cluster. During the process each node in the individual cluster will carry the destination address and match the processor address in the destination cluster. Once the address of the destination processor matches the node will carry all the requisite data from source cluster to destination cluster. During this process network sharing is very important concept in handling data sharing. In the present work ring, multi bus networks are considered for data sharing between clusters.

A. Ring based Cluster

This network supports high speed data transmission. It is very simple to design and interconnect each node. It provides data reliability in node communication. When a source node wants to send message to destination node, the source node checks the corresponding path. If the consistent path is not busy the processor in source cluster put the information on the path. In this mode the destination node will match with the source node identifier and receives the information from source node. If the path is busy with other destination clusters the source node will wait until the path is released and it is granted to the respective source cluster. During this procedure the intermediate nodes plays very important role in detecting the destination node. The intermediate nodes are busy in tracking the destination node during the search period. Hence, the intermediate nodes creating some delay in tracking of a destination node. Once the destination node is identified the destination node places an acknowledgement on the path for source. This acknowledgement is generated in response to the successful receiving of information at destination node.

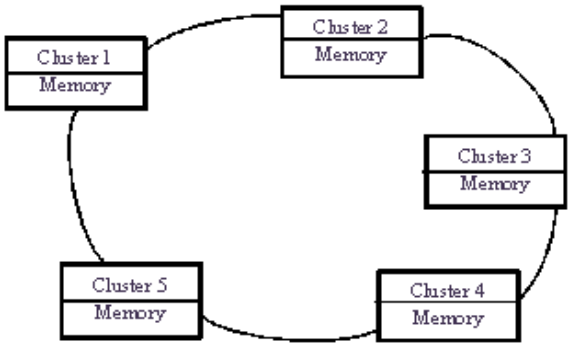

Fig.2. Ring bus cluster topology

B. Star based Cluster

Thisnetwork also supports high speed data communication when compared to other networks as a single cluster is connected with bus controller. Here individual clusters are connected and memory modules are connected to a common bus arbiter. A common Bus arbiter decreases system complexity. The central bus arbiter controls all data communication between clusters. If the bus arbiter failed, the entire network gets stuck and causes cluster failure in the system. All processors in a cluster share a common memory module. Whenever a processor wants to communicate another processor it sends a request to bus arbiter. The bus arbiter grants permission to the source processor to access the destination processor in the cluster or cluster system. If the destination processor is busy the source processor will wait until the busy processor done its task. During this process the bus arbiter plays an important role in establishing communication. So, the failure of the cluster depends on the reliability of the bus arbiter.

doi: 10.32622/ijrat.81202015

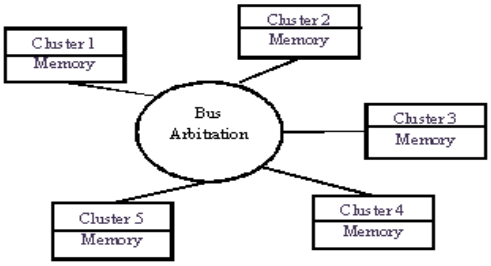

Fig.3. Star Bus cluster topology

C. Star Ring based Cluster

It is the combination of ring and star topologies. It overcomes all the problems present in star and ring topologies. If any node in the ring fails the bus arbiter bypasses the node and it does not interrupt the message passing and network does not collapse. Although the network physically connected and incorporated as star, the data transmitted from one node to another node as in ring network.

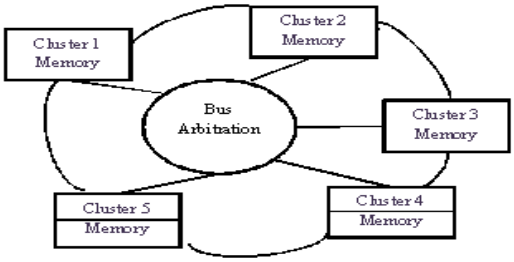

Fig.4. Star Ring bus cluster topology

D. Multi Bus based Cluster

This network presents homogeneous multiple buses to all clusters. This network mainly designed to avoid bus bottle neck on the server network or cluster network. In this network multiple port processors are required to interface multi bus network system. Multiple buses are connected to all cluster resources and they are shared by other cluster resources also. Each resource in the cluster connected with the multiple system bus and these connections are switched between buses based on their availability. Multiprocessor systems with cluster of processors are interconnected with local and global system buses as shown in figure 3. Multi bus system provides much flexibility in interconnections of all system resources. All the bus system control is depends upon central bus controller or bus arbiter connected with the system bus. The failure rate of the clusters depends upon the reliability of the bus arbiter at time *t*, R_a (t).

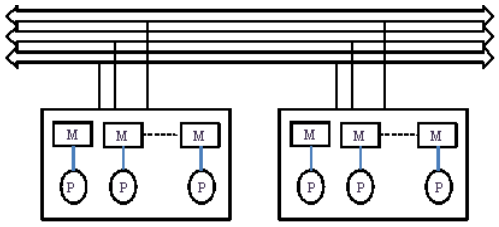

Fig.5. Multi bus Cluster System

In the entire multiprocessor cluster-based system individual cache memory modules are maintained to store a copy of main memory block. This is done for faster accessing of data at processors. Generally, all sub memory modules i.e. cache memory modules share data from a common data memory. The common shared memory processor-based system is done in two techniques like snoopy and directory method. In case of snoopy method every, requested memory block information is accessed by cache memory. This is done for quick response. Snoopy cache protocols do not suit general interconnection networks, mainly because propagation reduces their performance to that of a bus. In directory method unlike snoopy technique the inter cluster pointer information retained in a director. Every local action that effects the global state is maintained by global director.

III. RELIABILITY COMPUTATION

In this paper an analytical model is developed to evaluate reliability of cluster-based systems. The cluster system is designed such that no conflicts are identified in interconnection network and memory management. To compute the reliability of a cluster-based systems first evaluate reliability of the cluster $R_c(t)$ which is shown in equation 2. The cluster is a subsystem of a cluster and consists of *n* processors and *n* memory modules and interconnected by means of bus. The present system contains *k* clusters and it is considered that number of processors and memory modules in the system is, N= *kn.*

The reliability of a system at time *t* is defined as the probability that a system is operational at time between (*0*, *t*). In the present system the basic components are assumed as number of processors, memory modules, and buses. The failure rate in each component is distributed exponentially and represented as λ_P, λ_M, and $\lambda_B$respectively. The reliability of processor, Memory module, and buses are operational at time *t* is $R_P(t)$, $R_M(t)$, and $R_B(t)$ respectively. The reliability with respect to failure rates are derived as shown in equation 1.

$$R_p(t) = e^{-\lambda_p t}$$
$$R_M(t) = e^{-\lambda_M t} \quad (1)$$
$$R_B(t) = e^{-\lambda_B t}$$

Hence the total reliability of the cluster is represented as,

$$R_c = \sum_{p=0}^{n} R_p \sum_{M=0}^{m} R_M \sum_{B=0}^{b} R_B \quad (2)$$

Here each memory module is connected to one bus and hence the failure rate of the bus depends only on one memory module and hence the effective reliability of memory module ($R_{M\_eff}(t)$) will be

$$R_{M\_eff}(t) = e^{-(\lambda_M + \lambda_B)t} \quad (3)$$

doi: 10.32622/ijrat.81202015

Let $P_j(j)$ denote the probability of that *j* memory modules are operative out of *n* memory modules in time *t*, then

$$P_j(t)=\left(\binom{n}{j}C_{M\_eff}^{n-j}\left(R_{M\_eff}(t)\right)^{j}\left(1-R_{M\_eff}(t)\right)^{n-j}\right) \quad (4)$$

Where, $C_{M\_eff}^{n-j}$ is covering factor for memory module, representing that probability that system recovers from memory module failures successfully after time, t.

All the processors in the cluster has connection with all the buses, and hence the failure rate is more in processors when compare with memory module as it is connected with only one bus.

Let P_i (j) denote the probability of that *i* processors are operative out of *n* processors in time *t*, then

$$P_i(t)=\left(\binom{n}{i}C_{M\_eff}^{n-i}\left(R_{M\_eff}(t)\right)^{i}\left(1-R_{M\_eff}(t)\right)^{n-i}\right) \quad (5)$$

Hence the total reliability of the cluster is,

$$R_c(t)=\sum_{i=p}^{n}P_i(t)\sum_{j=m}^{n}P_j(t) \quad (6)$$

A. Ring System

In ring base system the cluster is connected between two other clusters. The failure rate of the cluster depends on any one of the cluster node or both clusters and cluster bus between the clusters. $P_i(t)$ be the probability of the ith operative processors are connected with cluster and, the reliability of the cluster system, $R_c(t)$ is defined as

$$R_r(t)=R_{rp}(t)R_{rn}(t)R_B(t)\sum_{i=1}^{n}P_i(t) \quad (7)$$

Where, $R_{rp}$and R_{rn} are the reliabilities of previous and next clusters connected across the operative processor in a ring network.

B. Star System

In this system each cluster is connected to Bus arbiter. Bus arbitrator is allocating bus control and resources to the processors in the cluster. All processors and memory modules are connected to one bus arbiter. As a single bus arbitrator is connected to all resources the failure rate depends on one bus arbitrator. Here reliability of the bus arbitrator depends on the operative clusters connected to the system. $P_i(t)$ be the probability of the ith operative processors are connected with bus arbiter and, the reliability of the cluster system, $R_s(t)$ is defined as,

$$R_s(t)=R_a(t)\sum_{i=1}^{n}P_i(t) \quad (8)$$

C. Multi-bus System

Each cluster contains *n* processors and *n* memory modules. The probability of that *i* processors are operative in a cluster at time *t* is $P_i(t)$. Similarly, the probability of *j* memory modules is operative in a cluster at time *t* is $P_j(t)$.

$$P_j(t)=\left(\binom{n}{j}C_M^{n-j}\left(R_M(t)\right)^{j}\left(1-R_M(t)\right)^{n-j}\right) \text{---------- (9)}$$

Where, C_M^{n-j} is covering factor for memory module, representing that probability that system recovers from memory module failures successfully after time, t.

It is considered the multi bus system is reliable aslong asthere are at least two clusters connected to the system, each with a minimum number of processors *P***,** and a minimum number of memory modules *M*interconnected by at least one bus [1].

In this case the reliability of the cluster is defined as,

$$R_c(t)=\sum_{i=p}^{n}P_i(t)\sum_{j=m}^{n}P_j(t) \quad (10)$$

The probability the system has *l*such operational clusters, $P_l(t)$ is

$$P_l(t)=\left(C_C^{k-l}\binom{k}{l}\left(R_C(t)\right)^{l}\left(1-R_C(t)\right)^{k-l}\right) \quad (11)$$

Where, C_C^{n-l} is covering factor for cluster, representing that probability that system recovers from cluster failures successfully after time, t.

Let $P_b(t)$ denote the probability that the multi bus system has *b* buses operational at time t,

$$P_b(t)=\left(C_B^{B-b}\binom{B}{b}\left(R_B(t)\right)^{b}\left(1-R_B(t)\right)^{B-b}\right) \quad (12)$$

Where, C_B is covering factor for the bus, representing that probability that system recovers from bus failures successfully after time, t.

Therefore, the reliability fo the multi-bus system, $R_{mb}(t)$ is derived as,

$$R_{mb}(t)=R_a(t)\sum_{b=1}^{B}P_b(t)\sum_{l=2}^{k}P_l(t) \quad (13)$$

D. Star Ring bus System

It is a hybrid network combines the merits of both star and ring network. Because of the star network, the network will not fail even a node or a cluster in the ring fail. Here the failure rate of the cluster depends on reliability of any one cluster that is either previous one or next cluster, and depends on bus arbiter. Here bus arbiter controls the rotation of the clusters in the ring that is either in fixed priority mode or rotating priority mode. The reliability of the network is given by

$$R_{rs}(t)=R_a(t)R_r(t)\sum_{i=1}^{n}P_i(t) \quad (14)$$

IV. PROPOSED SYSTEM

The proposed network topology is similar to distributed bus network, but beyond that it has many features to improve the reliability of the cluster when compared with other topologies. Each cluster contains processors and cache memories to hold the copy of the main memory block. On the other side of the bus a shared memory is connected to the global interconnection bus. All the processors communicated with main memory via cache memory. The status queue bus

doi: 10.32622/ijrat.81202015

arbiter is like director but plays more operations than directory topology [3]. Left bus arbiter is connected to all even clusters group and other right bus arbiter is connected to all odd clusters group to control the memory operations. If the source and destination are having even addressed then left bus arbiter control the processor operations with respect to main memory. On the other hand, if the source and destination are having odd address then right bus arbiter control the processor operations with respect to main memory. The status queue stores all the pointer information regarding request information for a memory block. If at any time any arbiter is failed the other arbiter takes care about remaining group of clusters. If the source and destination belong to different group then central bus controller will take care about left and right arbiter. The central bus controller will send the information of the source and destination to the status queue bus arbiter.

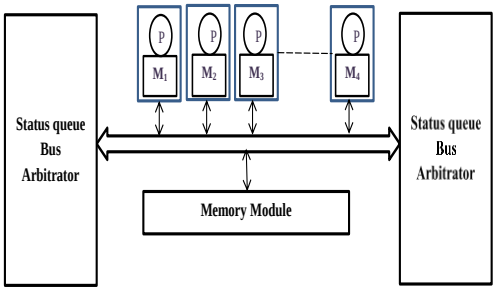

Fig.6. The proposed network topology

Let $P_h(j)$ denote the probability of that $h=k/2$ processors are operative out of n processors in time t, then,

$$P_h(t)=\left(\binom{n}{h} C_M^{n-h}\left(R_M(t)\right)^h\left(1-R_M(t)\right)^{n-h}\right) \quad (15)$$

V. RESULTS AND CONCLUSION

In ring-based cluster the intermediate nodes create interruption during cluster communication. The reliability depends on individual cluster connected in the network. In the star-based cluster the bus arbiter operational complexity is high. In every communication the reliability highly depends on bus arbiter similar to multi-bus system. The star ring bus almost minimized the existing problems in ring and star-based cluster systems. The delay in ring can be minimized with the proposed method. The reliability is high in proposed method when compared with ring, star and multi-bus method. The directory or arbiter-dependent number of processors is reduced to *k/2*. As the number of processors is reduced in the arbiter interconnection network the reliability will become more when compared with other networks and also the propagation delay will be reduced almost to half. Hence the probability of trailing the reliability is nearly reduced to *k/2* times.

REFERENCES

[1] Hu-Jun Wang et al., "Performance Analysis of Cluster based PPMB Multiprocessor Systems", The Computer Journal, Vol 38, No 5, 1995, pp 355-364.
[2] Stenstrom P. A survey of cache coherence schemes for multiprocessors. IEEE Computer 1990; June:12±24.
[3] Mahgoub.L.O, "Reliability of cluster-based multiprocessor systems", IEEE, 1992, DOI: 0-8186-2865-0 192, pp 204-209.
[4] T.S. Barnett et al., "Exploiting defect clustering for yield and reliability prediction", IEE Proc.-Comput. Digit. Tech., Vol. 152, No. 3, May 2005, pp407-413.
[5] Geng Luo-feng et al., "Performance evaluation of cluster-based homogeneous multiprocessor system-on-chip using fpga device", 2009 IEEE.
[6] Antbnio August0 Frohlich et al., "SMP PCs: A Case Study on Cluster Computing", 1998 IEEE, pp953-960.
[7] Stenstrom P. A survey of cache coherence schemes for multiprocessors+. IEEE Computer 1990; June:12+to24.
[8] Yue Luo et al., "Analyzing and Improving Clustering Based Sampling for Microprocessor Simulation", SBAC-PAD'05, 2005 IEEE.
[9] Imad Mahgouba, MazinYousifb, AbuAsaduzzaman, "Evaluation of memory latency in cluster-based cache-coherent amultiprocessor systems with different interconnection topologies", Computers and Electrical Engineering Volume 26 (2000) pp 207-220.
[10] Marek Tudruj, "Embedded Cluster Computing through dynamic Reconfigurability of Inter-processor connections", IWCC2001, LNCS 2326, Springer 2002, pp77-91.
[11] Ashis Mishra et al., "Reliability analysis of Multi-processor system with common cause failure", Journal of rajasthan academy of Physical Science, Vol 12 no 2, June 2013, pp 167-180.

AUTHORS PROFILE

Vanitha Kakollu, is currently working as Assistant Professor in the Department of Computer Science, GIS, GITAM (Deemed to be University). She is pursuing Ph.D. from JNTUH, Department of CSE. Her main areas of research include Image Processing, Data Mining and Machine Learning.

K Yasudha, is currently working as Assistant Professor in the Department of Computer Science, GIS, GITAM (Deemed to be University). Her main areas of research include Machine Learning and Data Mining.

K Naga Soujanya, is currently working as Assistant Professor in the Department of Computer Science and Engineering, GIT, GITAM(Deemed to be University). She is pursuing Ph.D. from Andhra University, Department of CSE, Her main areas of research include Data Mining and Machine learning.

doi: 10.32622/ijrat.81202015