

Development of Cluster-based Multiprocessor Systems with Analytical Model: A Approach

Mallikharjuna Rao K

Sr. Assistant Professor (G1), School of Computer Science and Engineering, VIT-AP University, Amaravati, Andhra Pradesh, India

Email: rao.mkrao@gmail.com

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—Network protocols, cluster, reliability, multiprocessors, network architectures

I. INTRODUCTION

In recent days 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. The existence of more failures in any system may cause low reliability [1]. There are essentially few methods are adopted in implementing multiprocessing such as, software evolution, hardware evolution, and architecture design. Reliability is the one for evaluating the quality of any software system [2].

In Software Evolution, operating systems and programming languages plays an important 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 multiprocessing. In Multi-program mode several programs executes 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 [3]-[5].

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 cluster 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 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 Fig. 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 [6]-[8] and any software based application [9], [10].

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 [11], [12]. 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.

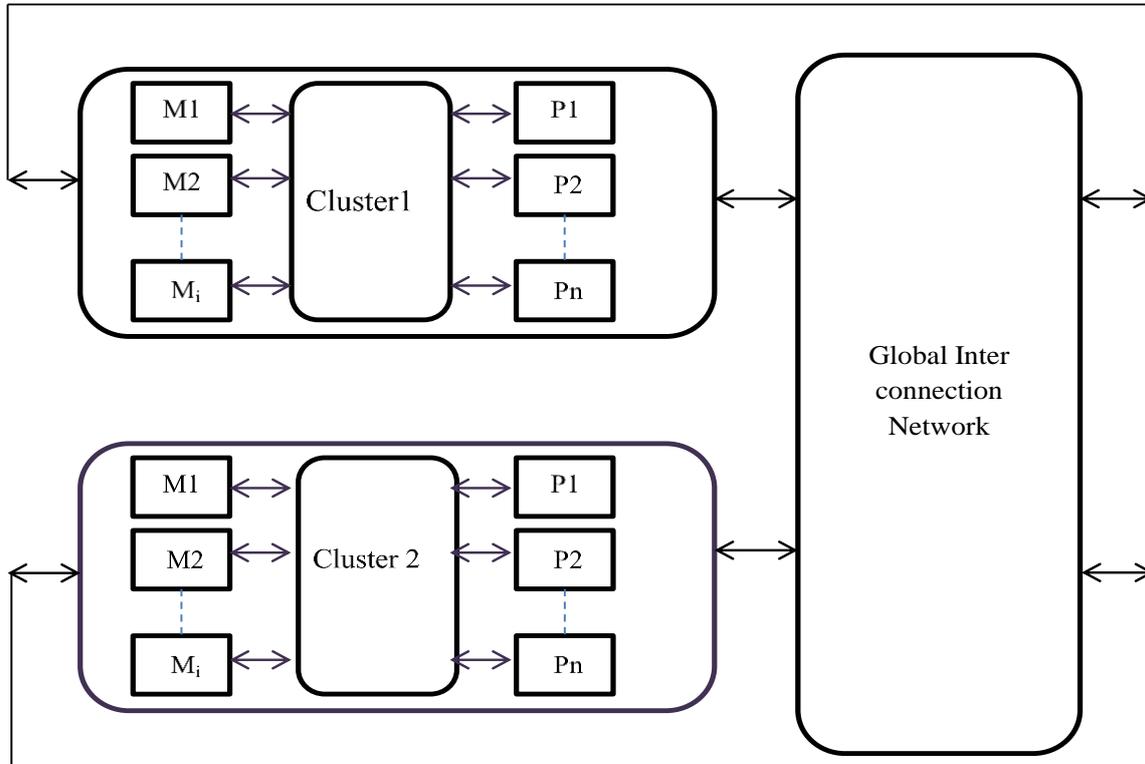


Fig. 1. Cluster based multiprocessor system

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

The rest of the paper is organized as follows. 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 [13]-[15]. 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 [16].

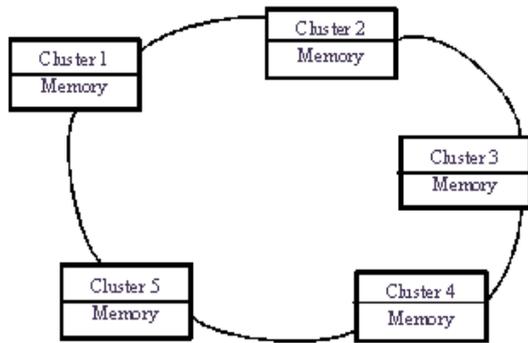


Fig. 2 Ring bus cluster topology.

B. Star based Cluster

This network 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 want to communicate other 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.

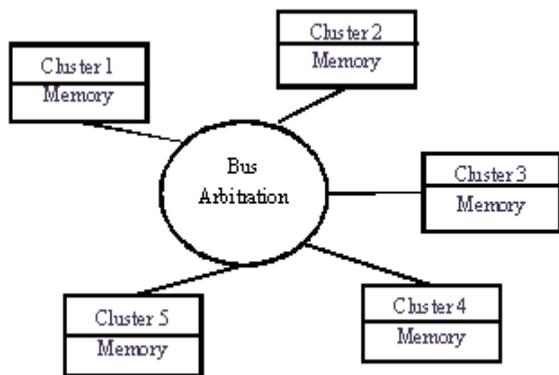


Fig. 3. Star bus cluster topology

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 interrupted the message passing and network does not collapse. Although the network physically connected and incorporated as star, the data transmitted from one node to other node as in ring network.

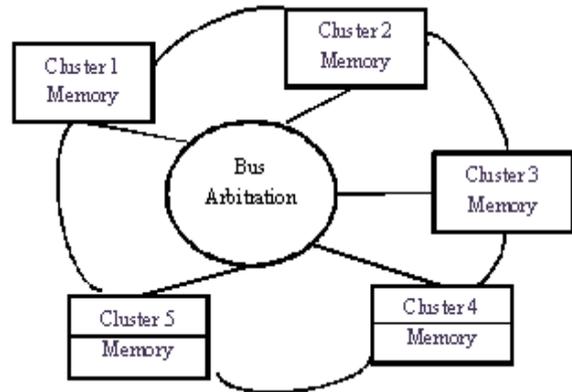


Fig. 4. Star Ring bus cluster topology

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 Fig. 3. Multi bus system provides much flexibility in interconnections of all system resources [14]. 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 , $R_a(t)$.

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 ie 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

[6], [17], [18] (see Fig. 4-5).

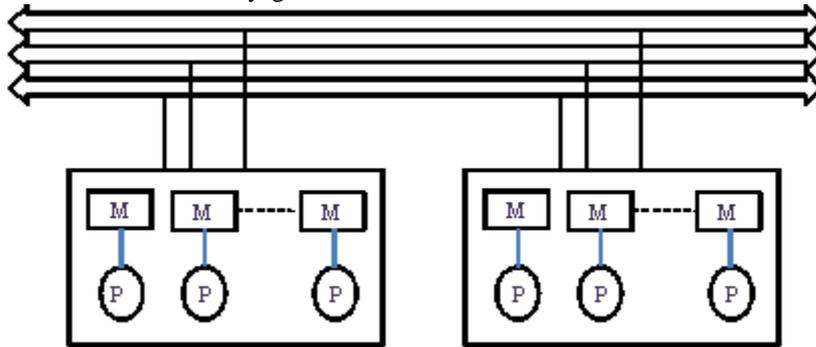


Fig. 5. Multi bus cluster system

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. The importance of reliability computation and performance evaluation are explained in [1], [2], [19] for any computer based system. 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 λ_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.

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

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

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

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

$$P_j(t) = \binom{n}{j} C_{M_eff}^{n-j} (R_{M_eff}(t))^j (1 - R_{M_eff}(t))^{n-j} \quad (4)$$

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

$$P_i(t) = \binom{n}{i} C_{M_eff}^{n-i} (R_{M_eff}(t))^i (1 - R_{M_eff}(t))^{n-i} \quad (5)$$

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 i th operative processors are connected with cluster and, the reliability of the cluster system, $R_c(t)$ is defined as,

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

where, R_{rp} and R_m 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 i th 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 are operative in a cluster at time t is $P_j(t)$.

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

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 as long as there 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

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

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 ,

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

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 for 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 METHOD

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 arbiter is like director but plays more operations than directory topology [4]. 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 belongs 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).

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

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

V. 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.

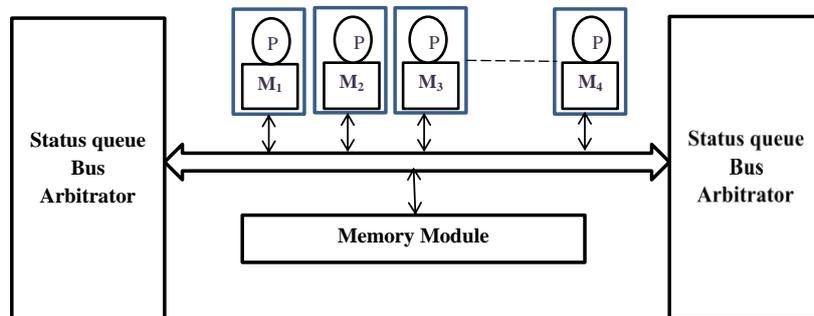


Fig. 6. The proposed network topology

REFERENCES

- [1] K. M. Rao and A. Kodali, "An efficient method for parameter estimation of software reliability growth model using artificial bee colony optimization," *Springer-Lecture Notes in Computer Science*, vol. 8947, pp. 765-776, August 2015.
- [2] K. M. Rao and K. Anuradha, "An efficient method for software reliability growth model selection using modified particle swarm optimization technique," in *International Review on Computers and Software*, Praise Worthy Prize Publishers, Italy, December 2015, pp. 1169-1178.
- [3] W. Hu-Jun, *et al.*, "Performance analysis of cluster based PPMB multiprocessor systems," *The Computer Journal*, vol. 38, no. 5, pp. 355-364, 1995.
- [4] P. Stenstrom, "A survey of cache coherence schemes for multiprocessors," *IEEE Computer*, June 1990.
- [5] D. Lenoski, J. Laudon, T. Joe, D. Nakahira, L. Stevens, A. Gupta, and J. Hennessy, "The DASH prototype: Logic overhead and performance," *IEEE Transactions on Parallel and Distributed Systems*, vol. 4, no. 1, pp. 1-31, 1993.
- [6] L. O. Mahgoub, "Reliability of cluster-based multiprocessor systems," *IEEE*, 1992, pp. 204-209.
- [7] A. Kumar, *et al.*, "Impact of network failures on the performance degradation of a class of cluster-based multiprocessors," *IEEE Transactions on Reliability*, vol. 40, no. 1, April 1991.
- [8] T. S. Barnett, *et al.*, "Exploiting defect clustering for yield and reliability prediction," *IEE Proc.-Comput. Digit. Tech.*, vol. 152, no. 3, pp. 407-413, May 2005.
- [9] K. M. Rao and K. Anuradha, "An efficient method for enhancing reliability and selection of software reliability growth model through optimization techniques," *Journal of Software*, IAP Publishers, Finland, vol. 12, no. 1, January 2017.
- [10] K. M. Rao and K. Anuradha, "A new method to optimize the reliability of software reliability growth models using modified genetic swarm optimization," *International Journal of Computer Applications, Foundation of Computer Science*, NY, USA, July 2016, pp. 1-8.
- [11] G. Luo-feng, *et al.*, "Performance evaluation of cluster-based homogeneous multiprocessor system-on-chip using fpga device," *IEEE*, 2009.
- [12] K. M. Rao and K. Anuradha, "A hybrid method for parameter estimation of software reliability growth model using modified genetic swarm optimization with the aid of logistic exponential testing effort function," in *Proc. IEEE Proceedings of International Conference on Research Advances in Integrated Navigation Systems*, 2016, pp. 82-89.
- [13] A. A. Frohlich, *et al.*, "SMP PCs: A case study on cluster computing," *IEEE*, 1998, pp. 953-960.
- [14] P. Stenstrom, "A survey of cache coherence schemes for multiprocessors," *IEEE Computer*, June 1990, pp. 12-24.
- [15] Y. Luo, *et al.*, "Analyzing and improving clustering based sampling for microprocessor simulation," *SBAC-PAD'05*, 2005.
- [16] I. Mahgouba, M. Yousifb, and A. Asaduzzaman, "Evaluation of memory latency in cluster-based cache-coherent amultiprocessor systems with different interconnection topologies," *Computers and Electrical Engineering*, vol. 26, pp. 207-220, 2000.
- [17] M. Tudruj, "Embedded cluster computing through dynamic reconfigurability of inter-processor connections," *IWCC2001, LNCS 2326*, 2002, pp. 77-91.
- [18] A. Mishra, *et al.*, "Reliability analysis of multi-processor system with common cause failure," *Journal of Rajasthan academy of Physical Science*, vol. 12, no. 2, pp 167-180, June 2013.
- [19] K. M. Rao and K. Anuradha, "Performance evaluation of software reliability growth models using evolutionary algorithms," *International Journal of Computer Science and Information Security*, vol. 14, no. 11, pp. 355-362, November 2016.



Dr. Mallikharjuna Rao K currently working as Sr.Assistant Professor(G1) in School of Computer Science and Engineering, VIT-AP University, Amaravati Andhra Pradesh, India. He obtained Bachelor's degree in Information Science and Technology Engineering from Acharya Nagarjuna University, Guntur, Andhra Pradesh, India in 2004. Then obtained Master's degree in Computer Science from Jawaharlal Nehru Technological University Hyderabad, India in 2010, and obtained Ph.D. in Computer Science and Engineering from

Jawaharlal Nehru Technological University Hyderabad, India in 2018. He has more than 15+ years of teaching and 5+ years of research experience in Engineering Institutions. He is a member of MIAENG, AMIE, India.

E-Mail: rao.mkrao@gmail.com