PERFORMANCE SIMULATION MODEL FOR A SHARED MEMORY MULTI-CORE COMPUTER SYSTEM USING TIME COLORED PETRINETS

Elizabeth A. Amusan[1], Undie A. Gabriel[2]

[1] Ladoke Akintola University of Technology - Ogbomoso, Nigeria, Department of Cyber Security Science
[2] Ladoke Akintola University of Technology - Ogbomoso, Nigeria, Department of Computer Science

Corresponding author: Elizabeth A. Amusan, eaadewusi@lautech.edu.ng

ABSTRACT: *A shared memory multi-core computer system is a computing paradigm in which processors have more than one core to process requests and also have access to a common memory. Most existing works are limited to modeling of a shared memory single-core computer system and thus the models are not flexible enough to study the operations of multi-core computer systems. Hence, in this paper, a high-level Petri Nets formalism (Timed Coloured Petri Nets) was used to develop a simulation model for a shared memory multi-core computer system. Intel HP core i5 was used as a case study in developing the TCPN model for a shared memory multi-core computer system. The developed TCPN model was simulated using Coloured Petri Net (CPN) tools. One hundred and fifty simulation runs were carried out in order to obtain average utilization rate of the shared memory and average waiting time of the processor's cores in accessing the shared memory. The developed TCPN model was validated based on both real and simulated average memory utilization of the shared memory multi-core computer system. The validation result of the developed TCPN model showed that there was no significant difference between the simulated and real average memory utilization of the shared memory multi-core computer system.*

KEYWORDS**:** *modelling, multi-core, petri net, shared memory, simulation*

1. INTRODUCTION

In the branch of computer architecture, shared memory computer system architecture is a form of computing in which a computing system with more than one processor core (multi-core) share a common memory known as a global memory. In this context, communication between tasks running on different core of the processor is performed through writing to and reading from the global memory. All inter-core coordination and synchronization is also accomplished via the global memory. A shared memory computer system can be simultaneously accessed by multiple programs to provide communication among them and to avoid redundant copies. It is a large block of memory that is accessed by several processors' cores [10].

Lately, multi-core processors have been applied in areas ranging from embedded systems to large-scale data centres. However, simulation technology for multi-core systems, lags behind and does not provide the simulation speed required to effectively support design space exploration and parallel software development [1].

Memory access contention can be a cause of performance problems and should be assessed at early stages of development. In the hierarchical modeling approach, the designer needs to provide a model for estimating the degree of resource contention for each shared resource such as central processing units (CPUs) or main memory [9]. Modeling and simulation have been assigned crucial roles in the design, development, analysis and evaluation of computer system architectures [3]. Nearly all computer system architecture research and design, quantitative evaluation of future architectures is possible only by using simulation models. Simulation models reduce the cost and time of a project by allowing the architect to quickly evaluate the performance of a wide range of architectures [8].

In order to study and improve performance of a shared memory system in terms of global memory utilization rate and its waiting time before it can be accessed by core(s) of a processor, simulation models are necessary. Petri Net (PN) is a graphical and mathematical modeling language that can be used to develop a simulation model for a system characterized as being synchronous, asynchronous, deterministic, stochastic, concurrent, distributed and parallel in nature [6]. A Timed Coloured PetriNet (TCPN) model is the introduction of time concepts into a Coloured Petri Net model; thereby making it possible to calculate performance measures, such as the speed by which a system operates, mean waiting time and throughput [5].

[2] developed a methodology for high-speed multi-core instruction set simulation using Just-in-Time Dynamic Binary Translation. Also, Asaduzzaman proposed in [4] a methodology of developing conceptual models of multi-core networked systems

for the purpose of analyzing their performances. However, the goal of this paper is to use a high-level Petri Nets formalism (Timed Coloured Petri Nets) to develop a simulation model for a shared memory multi-core computer system and validate the model based on memory utilization rate.

2. RESEARCH METHODOLOGY

2.1 Overview of the Modeling Approach

In this paper, Colored Petri Nets (CPN) and Time Colored Petri Nets (TCPN) formalisms represented by (1) and (2) respectively were used to model the shared memory HP core i5 computer system which is the case study being considered here.

Both CPN and TCPN are tuples defined by Jensen [7] as:

$$CPN = (\Sigma, P, T, A, N, C, G, E, I) \quad (1)$$

where:

a. Σ is a finite set of non-empty types, also called colour sets.

b. P is a finite set of places.

c. T is a finite set of transitions.

d. A is a finite set of arcs such that:

$$P \cap T = P \cap A = T \cap A = \emptyset.$$

e. N is a node function. It is defined from A into $P \times T \cup T \times P$.

f. C is a colour function. It is defined from P into Σ.

g. G is a guard function. It is defined from T into expressions such that:

$$\forall t \in T : [Type(G(t)) = B \wedge Type(Var(G(t))) \subseteq \Sigma].$$

h. E is an arc expression function. It is defined from A into expressions such that: $\forall a \in A :$ $[Type(E(a)) = C(p)MS \wedge Type(Var(E(a))) \subseteq \Sigma]$ where p is the place of N(a).

i. I is an initialization function. It is defined from P into closed expressions such that:

$$\forall p \in P : [Type(I(p)) = C(p)MS].$$

A Time Colored Petri Net is a tuple;

$$TCPN = (CPN, R, r_o) \quad (2)$$

where:

a. CPN satisfying the definition of CPN.

b. R is a set of time values, also called *time stamps*.

c. r_o is an element of R called the *start time*.

In the development of a simulation model for a shared memory multi-core computer system using Timed Coloured Petri Nets formalism, the following steps constitute the approach being employed:

a. Description of the shared memory multi-core computer system

b. Development of the considered shared memory multi-core computer system using TCPN formalism described earlier.

c. Simulation of the developed TCPN model of the shared memory multi-core computer system using TCPN supported construction and simulation tool known as CPN tools.

d. The validation of the developed TCPN model based on memory utilization rate

2.2 Description of the Shared Memory Computer System (Case Study)

The operation of the shared memory multi-core computer system considered in this project is depicted in Figure 1. The system consists of five processor's cores that want to access a common shared memory. The five processor's cores have the same capability. They work locally for some time, then request access to the shared memory and finally access it. Precisely, the operation of the shared memory multi-core computer system as depicted in Figure 1 is such that when one of the five processor's cores accesses the shared memory, the other processor's cores have to wait till the processor's core that first accesses the shared memory completes its operation with the memory.

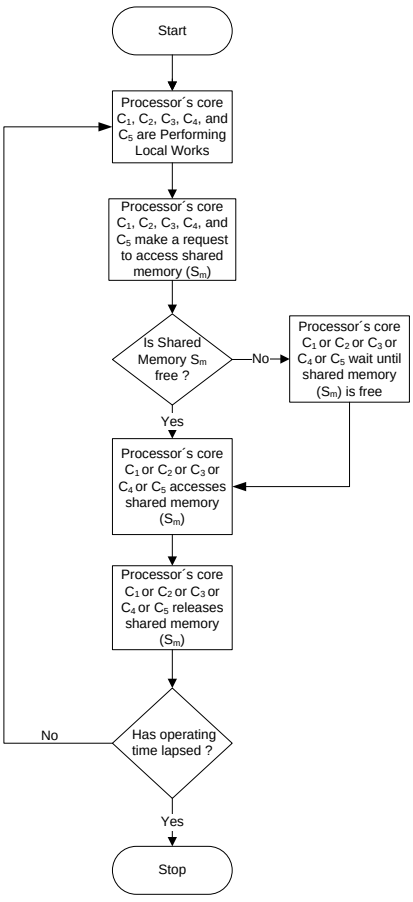

Figure 1: Flowchart of Shared Memory Multi-Core Computer System

2.3 Development of the TCPN Model of the Shared Memory Multi-Core System

CPN Tools (versions 4.0.0) was used in constructing a Timed Colored Petri Net (TCPN) model for the shared memory multi-core computer system under consideration. In the TCPN model, four (4) places and six (6) transitions were utilized. The places were drawn as ovals while transitions were drawn as rectangles. Places and transitions were connected with directed arcs which modeled the relations among the individual elements of the developed model. The arcs with their arc expressions defined the flows of tokens in the net. The descriptions of the places and transitions that were used in the TCPN model are enumerated in Tables 1 and 2, respectively. The color sets, variables and initial parameters employed in the developed TCPN model are also depicted in Figure 2. In developing the model, the following assumptions were made:

a. One time stamp unit represents one minute in the TCPN model.
b. The system operating time is 8 hours that is 480 minutes.
c. Multi-core system characterized by single shared memory architecture was considered.

2.4 Simulation and Validation of the Developed TCPN Model

The developed TCPN model of the shared memory multi-core computer system was simulated on Intel(R) Core(TM) i5-3230M CPU @ 2.60GHz 64-bit Operating System using CPN tools version 4.0.0. In simulating the model, the number of processor's core and shared memory, the local working time of the processor's core and access time of the processor's cores were used as simulation input parameters. Table 3 shows details of these simulation input parameters. The idea of simulation is to take an executable model and let it run several times. One hundred simulations were run on the developed TCPN model to obtain average utilization rate of the shared memory and average waiting time of processor's cores accessing the shared memory. The developed model was validated based on the real and simulated average memory utilization of the shared memory multi-core computer system.

Table 1: Description of Places in the TCPN Model

Place	Description
Processor@local_operation	This place was used to model activity of processor's cores while performing local operation
Wait	This place was used to model waiting state of the other processor's cores until the processor's core that first accessed the shared memory has finished its operation with the memory
Access	This place was used to model state in which a processor's core is busy using the shared memory
Shared_memory_Idle	This place was used to model a state in which the shared memory is free

Table 2: Description of Transitions in the TCPN Model

Transition	Description
C1RequestSm	Model process of requesting for shared memory Sm by processor's core C1
C2RequestSm	Model process of requesting for shared memory Sm by processor's core C2
C3RequestSm	Model process of requesting for shared memory Sm by processor's core C3
C4RequestSm	Model process of requesting for shared memory Sm by processor's core C4
C5RequestSm	Model process of requesting for shared memory Sm by processor's core C5
C1Access_memory	Model process of accessing the shared memory Sm by processor's core C1
C2Access_memory	Model process of accessing the shared memory Sm by processor's core C2
C3Access_memory	Model process of accessing the shared memory Sm by processor's core C3
C4Access_memory	Model process of accessing the shared memory Sm by processor's core C4
C5Access_memory	Model process of accessing the shared memory Sm by processor's core C5
C1ReleaseSm	Model process of releasing the shared memory Sm by processor's core C1
C2ReleaseSm	Model process of releasing the shared memory Sm by processor's core C2
C3ReleaseSm	Model process of releasing the shared memory Sm by processor's core C3
C4ReleaseSm	Model process of releasing the shared memory Sm by processor's core C4
C5ReleaseSm	Model process of releasing the shared memory Sm by processor's core C5

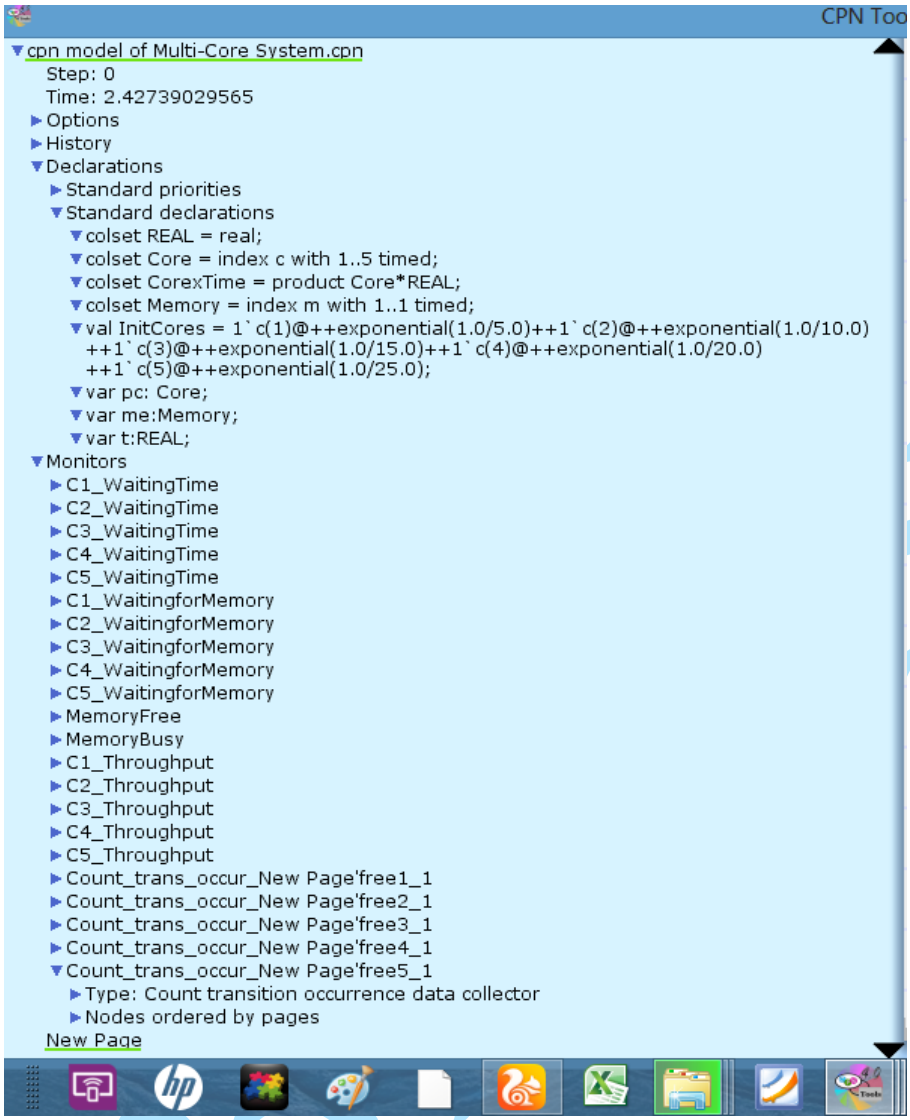

Figure 2: The Declaration of Color sets, Variables and Initial Parameters used in the Developed TCPN Model

Table 3: Simulation Input Parameters

Parameter	**Value/Expression**
Number of processor's core	5
Number of shared memory	1
Local operation time of processor's core 1 (C1)	Exponential distribution with mean of 5 minutes
Local operation time of processor's core 2 (C2)	Exponential distribution with mean of 10 minutes
Local operation time of processor's core 3 (C3)	Exponential distribution with mean of 15 minutes
Local operation time of processor's core 4 (C4)	Exponential distribution with mean of 20 minutes
Local operation time of processor's core 5 (C5)	Exponential distribution with mean of 25 minutes
Access time of processor 's core 1 (C1)	Exponential distribution with mean of 5.0 minutes
Access time of processor 's core 2 (C2)	Exponential distribution with mean of 4.0 minutes
Access time of processor 's core 3 (C3)	Exponential distribution with mean of 3.0 minutes
Access time of processor 's core 4 (C4)	Exponential distribution with mean of 2.5 minutes
Access time of processor 's core 5 (C5)	Exponential distribution with mean of 2.0 minutes

3. RESULTS AND DISCUSSION

The developed TCPN model of the considered shared memory multi-core computer system is depicted in Figure

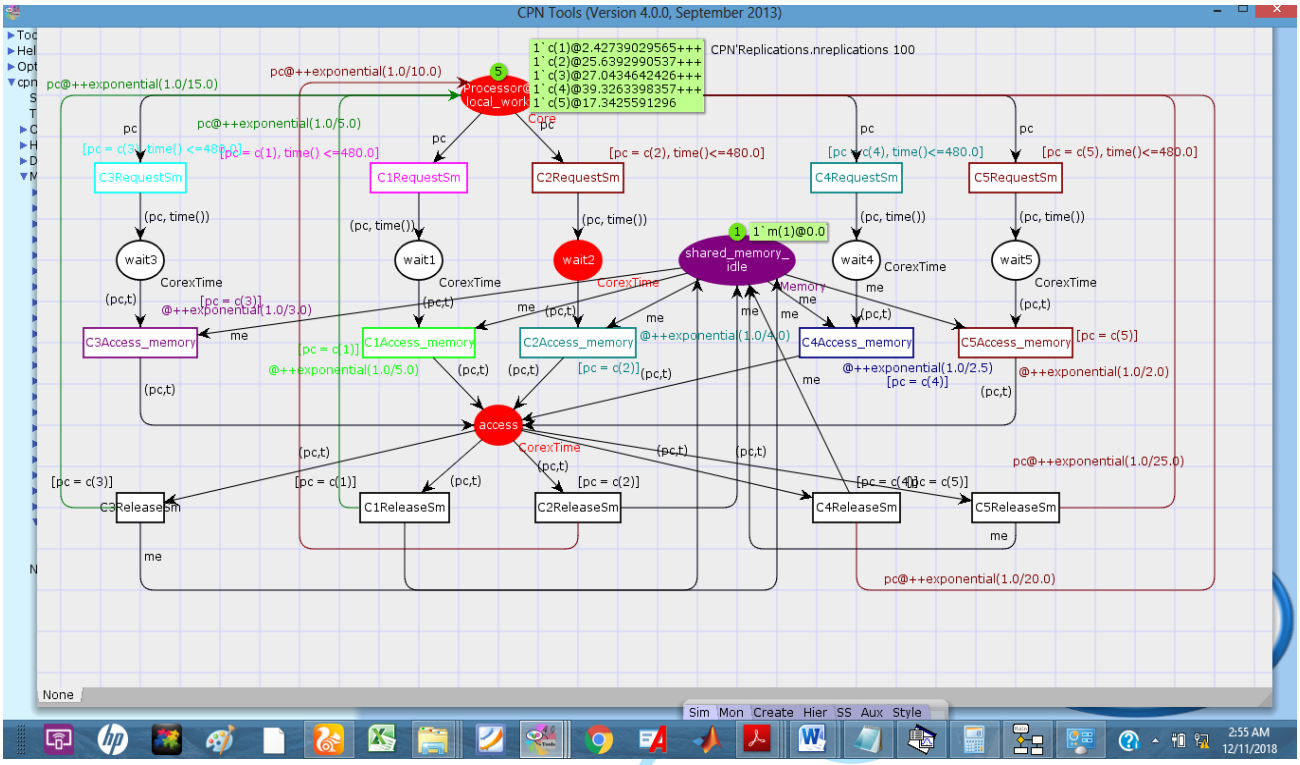

Figure 3: The Developed TCPN Model of the Shared Memory Multi-Core Computer System

In the developed TCPN model, the processor's core either C(1) or C(2) or C(3) or C(4) or C(5) performing local processes (place *Processor@local_operation*) request access (transition:
C1RequestSm for processor's core C(1),
C2RequestSm for processor's C(2),
C3RequestSm for processor's core C(3),
C4RequestSm for processor's core C(4) and
C5RequestSm for processor's core C(5))
to the shared memory (m(1)).
The processor's core requesting access to the shared memory wait (place *wait*) if the any of the other processor's core is using the shared memory (place *Access*). Otherwise, the processor's core accesses (transition:
C1Access_memory for processor's core C(1),
C2Access_memory for processor's core C(2),
C3Access_memory for processor's core C(3),
C4Access_memory for processor's core C(4) and
C5Access_memory for processor's core C(5))
the shared memory (place *Shared_memory_Idle*) and work (place *Access*) with the shared memory. After finishing work with the shared memory, the processor's core releases (transitions: *C1ReleaseSm* for processor's core C(1), *C2ReleaseSm* for processor's core C(2), *C3ReleaseSm* for processor's core C(3), *C4ReleaseSm* for processor's core C(4) and *C5ReleaseSm* for processor's core C(5)) the shared memory (place *Shared_memory_Idle*) and return to perform its local operation again (place *Processor@local_operation*).
The simulation of the developed TCPN model was done using CPN tools and the results are as depicted by Figure 4. The results revealed that the average utilization rate of the shared memory and the average waiting time of a processor's core accessing the shared memory were 42% and 5.73 seconds respectively.

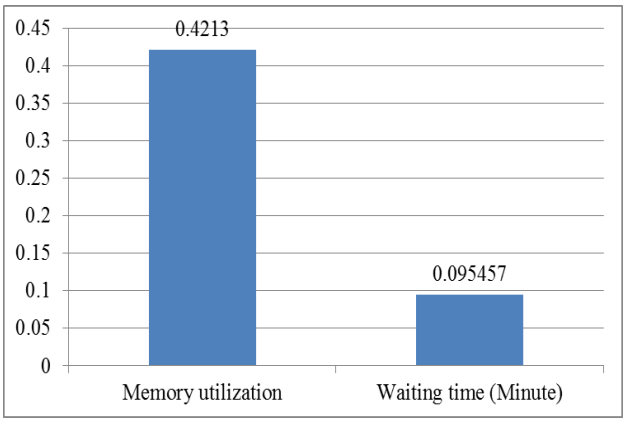

Figure 4: Simulation Results of the Developed TCPN

Figure 5 represents the validation result of the developed TCPN model. The results compare the

simulation output data (average memory utilization of the shared memory computer system) with the average memory utilization of the shared memory multi-core computer system as observed from the real system under consideration. The developed TCPN model is a valid model and accurately represents operation of the shared memory multi-core computer system under study.

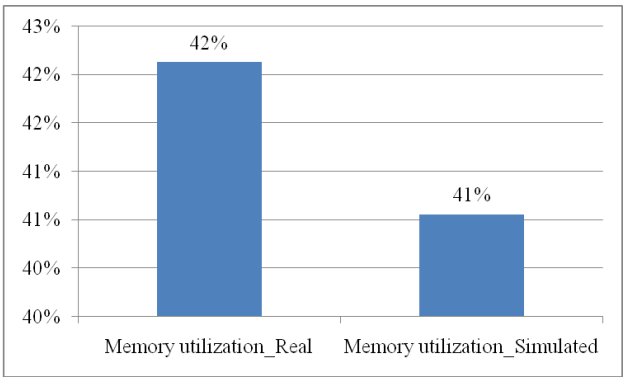

Figure 5: Validation Results of the Developed TCPN Model

CONCLUSION

In this paper, Timed Colored Petri Net (TCPN) formalism has been used to develop a model for shared memory multi-core computer system using HP core i5 computer system as a case study.

The simulation results of the developed TCPN model revealed that the shared memory is not always overloaded with tasks. Also, the developed TCPN model showed valid representation of the considered shared memory multi-core computer system. This is evident from the result of validation of the model which shows that there is no much significance difference between the simulated and the average utilization rate of the shared memory.

Thus, the developed TCPN model could accurately describe the operation of the considered shared memory multi-core computer system or any other related multi-core computer systems.

REFERENCES

[1] **O. Almer, I. Böhm, Edler T. von Koch**. *et al.*- *A Parallel Dynamic Binary Translator for Efficient Multi-Core Simulation, Int J Parallel Prog* **41,** pp.212–235, 2013.
https://doi.org/10.1007/s10766-012-0222-9

[2] **O. Almer et al.** - *Scalable Multi-Core Simulation Using Parallel Dynamic Binary Translation,* International conference on Embedded Computer Systems, DOI: 10.1109/SAMOS.2011.6045461, IEEE Xplore, 2011.

[3] **P. Andreas, K. T. Georgios, S. Michael and L. Dimitris** - *Efficient Modeling and Simulation of a Virtually Shared Memory Architecture*, 3rd International Eurosim Congress, Helsinki, Finland, 1998.

[4] **A. Asaduzzaman** - *Performance Modelling of Multi core And Many core Networked Systems,* International Journal of Computer Networks & Communications (IJCNC) Vol.4, No.2, pp. 53 - 67, 2012.

[5] **R. A. Ganiyu, E. O. Omidiora, S. O. Olabiyisi, O. T. Arulogun, O. O. Okediran** - *The Underlying Concepts of Coloured Petri Nets and Timed Coloured Petri NetsModels through Illustrative Example*, Proceedings of the Second International Conference on Engineering and Technology Research, Ladoke Akintola University of Technology, Ogbomoso, 2013.

[6] **R. A.Ganiyu, S. O. Olabiyisi, T. A. Badmus, O. Y. Akingbade** - *Development of a Timed Coloured Petri Nets Model for Health Centre Patient Care Flow Processes*, International Journal Of Engineering And Computer Science ISSN:2319-7242Volume 4 Issue 1, Page No. 9954-9961, 2015.

[7] **K. Jensen and L. M. Kristensen** - *Coloured Petri Nets: Modelling and Validation of Concurrent Systems*, Springer, 2009.

[8] **J. Y. Joshua and J. L. David** - *Simulation of Computer Architectures: Simulators, Benchmarks, Methodologies, and Recommendations*, IEEE Transactions on Computers, Vol. 55, No. 3, pp. 268 – 280, 2006.

[9] **K. Ryo, O. Kouichi and N. Takeo** - *MVA-Based Probabilistic Model of Shared Memory with a Round Robin Arbiter for Predicting Performance with Heterogeneous Workload*, International Conference on Multicore Software Engineering, Performance, and Tools, Multicore Software Engineering, Performance, and Tools, pp. 13 – 24, 2013.

[10] **M. Tripathy and C.R. Tripathy** - *A Comparative Analysis of Performance of Shared Memory Cluster Computing Interconnection Systems*, Journal of Computer Networks and Communications, Hindawi Publishing Corporation, vol. 2014, pp. 1-9, 2014.