

Stochastic Modeling and Availability Evaluation of Computer Systems

Wg Cdr Varghese Thattil¹ and Swetha Akula²

¹ CVR College of Engineering/Electronics and Communication, Hyderabad, India
Email: mailthattil@cvr.ac.in

² CVR College of Engineering/Electronics and Communication, Hyderabad, India
Email: swethaakula9@cvr.ac.in

Abstract—There are many applications of the computer systems where in the system availability has to be ensured. The evaluation of the availability is very vital before a computer system is being put into operation in such critical applications. In the case of hardware faults, high degree of reliability can be achieved by hardware redundancy. For microprocessor systems good additional feature is fault tolerance. By the use of dedicated customized hardware, fault tolerance can be achieved which is cost effective. A stochastic modeling of the microprocessor based computer system has been carried out and the lifetime availability is estimated. This evaluation is always being during the entire process of system design. The modeling framework used for the Multiprocessor system is based on an extension of Petri nets called Stochastic Activity Networks (SAN). A major contribution of this paper is that a SAN based comprehensive model for computer system using Mobius simulation tool has been developed which can be extensively used for the lifetime evaluation of systems of various architectures and hardware designs.

Index Terms—Fault tolerance, Stochastic Activity Networks, Multiprocessor Systems, Simulation.

I. INTRODUCTION

In our everyday life computers play a prominent role. There are computers used in mission-critical applications, life support applications, homeland security, battlefield situations, where the availability of the computer system is considered to be vital. A system with 99.99% availability over 30 days is unavailable for 4.32 minutes in 30 days.

The microprocessors are regarded as one of the most paramount contrivance in computer systems. Usually a microprocessor will have a silicon chip having millions of transistors & other devices that will be in processing millions of operations per second [1, 2]. It is a multipurpose, programmable microchip that utilizes digital data as an input. Once it processes the input according to the instructions stored in its memory, it provides results as an output. As microprocessor has

internal memory, it became an example of sequential digital logic. Operation of microprocessors is based on the numbers and symbols represented in the binary numeral system. Main purpose of designing microprocessors is to perform arithmetic and logic operations. Application of microprocessors in PCs is for computation and text editing [3].

Computational control is provided by the microprocessor which functions as a central processing unit (CPU) of a computer. Along with the arithmetic and logic functional units, control logic, instruction processing circuitry, and some part of the memory hierarchy are incorporated in characteristic microprocessors. Sanctioning more frugal overall systems, some parts of the interface logic for the input/output and memory subsystems are saturated. To provide multiple functional units and relatively sizably voluminous caches, microprocessors and single-chip designs depend on a small number of chips.

The system availability is greatly influenced by the way the system architecture is designed, the quality of the components of the system, the design of major data structures, files, and databases and the specifications of the interfaces between components.

Microprocessors are fabricated utilizing techniques homogeneous to those utilized for other integrated circuits, such as memory chips. Microprocessors generally have a more intricate structure than do other chips, and their manufacture requires immensely precise techniques [4].

The working of the microprocessor is done in three steps i.e., fetching, decoding and processing. In the first step it gets an instruction from the computer memory and decodes the instruction in second step. In the last step, the processor carries out the decoded set of instructions. A microprocessor carry out three step process, millions of times per second.

In designing process of computer and communication systems, performance modeling is incorporated. Issues aroused during modeling process is addressed by the new techniques. Better design and manufacturing process ensures high quality components being manufactured however performance of these

components when integrated into a system offers much poorer performance. Therefore it is very essential to estimate the mean life time and reliability of any system before being put into operational use.

Challenging problem faced in many engineering applications is the estimation of mean lifetime and reliability of sophisticated systems. In such cases, obtaining the results for the lifetime and reliability of complex systems; by theoretical development is tedious. Systems have become so complex that the study of their reliability requires extensive modeling and simulation. Modeling is a complex process which can be attained by experience and continuous improvement. The paper discusses method of modeling a disk storage system through stochastic process.

A. Reliability and Availability

Reliability and expected lifetime are very important issues in modern complex systems. Systems have become so complex that the study of their reliability / availability requires extensive modeling and simulation. Reliability can be defined as the conditional probability at a given confidence level that a system will perform its operation properly without failure at specified performance requirements during a given time interval $[0, t]$. Whereas availability is the probability that a system will be operational at any given time taking into consideration the failure time and time to repair. Availability is concerned with repairable systems.

B. Petri Nets

Place/transition Petri nets (p/t-nets, for short); Petri nets is a graphical and mathematical tool which have been used to describe a wide range of systems since their invention by Carl Adam Petri in 1962. High-level Petri Nets were developed to overcome the problem of the explosion of the number of elements of their graphical form for complex systems in Petri nets. As techniques for solving models advanced, formalisms (formal languages for expressing models) were also developed.

C. Stochastic Petri Nets

For the description of Discrete Event Dynamic Systems (DEDS) Stochastic Petri Nets (SPNs) were introduced in 1980 [5],[6]. SPNs represent the dynamic behavior of DEDS with continuous-time homogeneous Markov chains for their performance and reliability evaluation.

D. Models

It's difficult to understand the behavior of real time systems because their organization is complex and the interaction among their components is difficult. The

possibility of computing results from the analysis of a model is the key for closing a loop that starts from the abstraction of the relevant features of the system during modeling construction and that ends with the interpretation of the results provided by the model and reflected on the real system. Designer has to decide on the relevant performance parameters which have to be studied for each model.

It should be noted that a key element in the development of a model is the selection of the level of abstraction (also called level of detail). This amounts to selecting the system features to be included in the model. No precise rule exists for this selection, which rests mainly on the experience and ingenuity of the performance analyst and the knowledge of the system being modeled.

E. Discrete Event Dynamic Systems

Discrete Event Dynamic Systems (DEDS) are systems with a discrete state space and whose evolution is not directly due to the passage of time, but to the occurrence of events. The systems are views as DEDS not because the system itself is discrete in nature but the aspects of their behavior mainly outcomes of a system that we want to study are discrete in nature.

II. PERFORMANCE EVALUATION OF DEDS

Time-related performance indices such as availability, resource consumption, reliability and system response time accounting for system failure are computed using mathematical and simulation models which come under performance evaluation. To solve problem multiple models are evolving and databases are maintained.

A. Stochastic Activity Networks (SAN)

Probabilistic extensions of activity networks are stochastic activity networks. Stochastic Petri Nets are less powerful and inflexible compared to Stochastic activity networks [7]. SAN model is a particular kind of directed graph composed of primitives like places, activities and gates. Timed activities and instantaneous activities are two kinds of activities.

1. Place:

Places are used to represent the "state" of a system. A state of a Petri net is determined by a distribution of tokens (a nonnegative integer) on its places. This mapping is usually called a *marking*. A place is depicted as \bigcirc .

2. Timed Activity:

Marked as \blacksquare timed activities represent activities of the modeled system whose durations impact the system's ability to perform? An activity distribution

function, an enabling rate function, and an n -array computable predicate called the reactivation predicate is associated with timed activity.

3. *Instantaneous Activity:*

Marked as |. Instantaneous activities are completed in a negligible amount of time. An instantaneous activity has m inputs and n outputs. Case probability function is associated with each instantaneous activity. Realization of spatial uncertainty is permitted by the case probabilities.

4. *Input Gate:*

Marked as \leftarrow . To permit greater flexibility in defining, enabling and completion rules gates are introduced. An input gate is confined to one output and finite set of inputs. An n -array computable predicate called the enabling predicate is associated with each input gate called input function.

5. *Output Gate:*

Marked as \rightarrow . An output gate is confined to one input and finite set of outputs. A computable function is associated with each output gate called the output function.

B. *Basic Approach for Building SAN Model*

Modeling complex systems with stochastic processes requires a good knowledge of the system to be modeled along with knowledge of probability theory, advanced calculus, matrix algebra and a general level of mathematical maturity [8]. The process of system modeling using SAN involves building up the individual atomic models and forming the composed model using them. Within the atomic model input / output gates, place markings and activities have to be properly programmed.

The composed model is built precisely interconnecting the atomic models. A Join is used to compose two or more sub models using equivalence sharing. A Replicate is used to construct a model consisting of a number of identical copies of its single child. Each child node of a Replicate or Join node can be a Replicate, a Join, or a single atomic or composed model. The desired performance measures are specified on the composed model. Reward models are built upon atomic and composed models, equipping them with the specification of a performance measure PV [9].

III. APPLICATION: FAULT TOLERANT MULTIPROCESSOR SYSTEM

Fault tolerant is defined as the ability to perform execution correctly in spite of the presence of faults. Much importance has been given to fault tolerance because of the critical applications of multi processor

systems [10]. System failure is modelled based on the failures of the components in the system. The two possible states often allowed for components in the system is either in working state or failed state [11]. The system we consider is highly redundant fault-tolerant multiprocessor as shown in fig 1. This system is modelled in hierarchically, such that the system at highest level consists of multiple computers. Series system is considered not functional, if one component fails total system fails. Whereas, parallel system is considered functional, if at least one component functions. To overcome the disadvantage of series system, for failure recovery purpose a spare unit is also included. Fault tolerance can be achieved by the dedicated hardware [12]. Each computer is composed of 3 CPU units, of which 1 is spare unit; 3 memory modules, of which 1 is a spare unit; 2 I/O ports, of which one is spare port and 2 non-redundant error-handling chips. If we look internally, each unit is made up of chips. Like, memory module consists of 41 RAM chips out of which 2 are spare chips and 2 interface chips. Each CPU unit and I/O port consists of 6 non-redundant chips. We consider the system is operational only when at least one computer is operational. If at least 2 CPU units, at least 1 I/O port, at least 2 memory modules and 2 error-handling chips are functioning then computer is classified as operational. At least 39 of its 41 RAM chips, and its 2 interface chips, are to be working for memory module is to be operational [13].

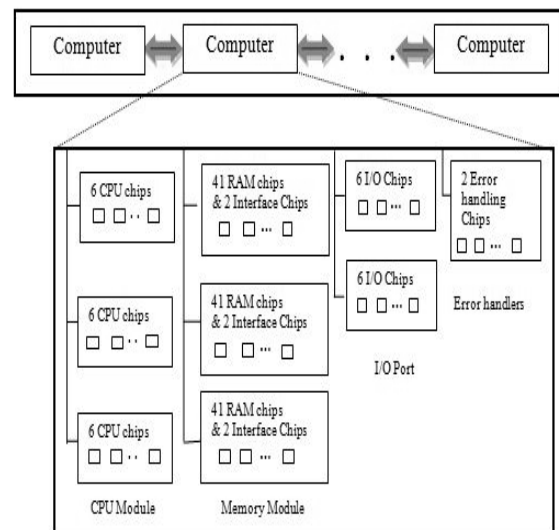


Figure 1. Fault Tolerant Microprocessor System

Redundant units are included in this hierarchy; so that, even when there is a failure of any unit it is replaced with the spare. If the probability of failure of one CPU unit is 0.995 then the failed one is replaced with the spare unit and the computer will be operational. Also, there is a probability of 0.005 (CPU coverage) that fault recovery fails and the computer operation will get ceased. Finally, it is assumed that the failure rate of

every chip is 0.0008766 in the system (1 failure per 1141 years). Coverage probabilities of each unit are given in Table I.

TABLE I
COVERAGE PROBABILITIES

Redundant Component	Fault Coverage Probability
CPU Unit	0.995
I/O Port	0.99
Memory Module	0.95
RAM Chip	0.998
Computer	0.95

IV. MODELING OF A FAULT TOLERANT SYSTEM USING MOBIUS

Mobius tool is used to model the fault tolerant multi processor system. In Mobius atomic models are built for each module. These atomic models are either replicated or joined to form a composed model. Performance variable is retrieved in the reward model. The study model is used to give values of the parameters for simulating various experiments. This composed model is simulated using a solver.

A. SAN Submodels (Atomic Models)

SAN sub model of CPU module is created and then places, input gates, timed activities and output gates are included in the model. Sub model of the CPU module is shown in Fig 2.

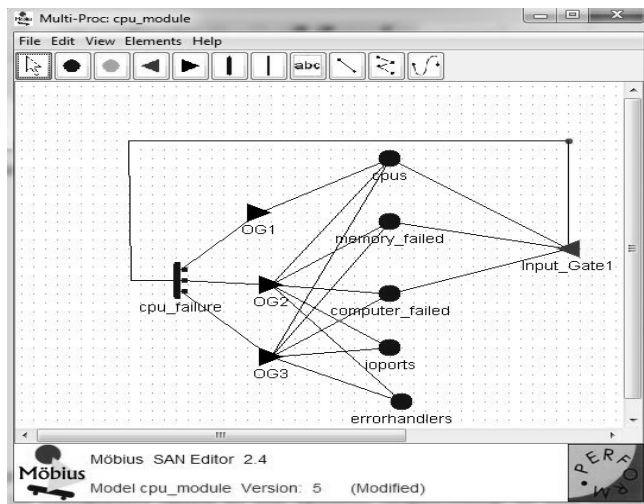


Figure 2. SAN Sub model of cpu_module

Number of tokens in the places cpus and computer_failed represents number of operational CPUs in a computer and the number of computers failed in the system, respectively. In this model the places io ports, errorhandlers and memory_failed are also labelled. Additional lumping of states is done to reduce

complexity because; if a computer fails no need to track the component failure which leads to computer failure. According to the assumption that all internal components are failing for the computer that have failed. CPU unit, a memory module, an I/O port, or an error-handling chip is combined to form a single state which represents the failure of a computer. Combined state marking is achieved by setting the number of tokens in each of the places CPUs, I/O ports, and error handlers to zero, setting the number of tokens in memory failed to 2, and incrementing the number of tokens in computer failed. cpu_failure is the timed activity and it takes place only when the condition in input gate is satisfied. If the timed activity CPU_failure is completed then it corresponds to cpu failure. If a spare CPU unit is available (i.e., CPUs->Mark() == 3), three cases are associated with the activity completion, as designated in the Case quantity field. First case represents that spare CPU is replaced with the failed CPU and the computer operates normally. Second case represents the situation in which the spare unit CPU is not available to replace failed CPU such that, spare computer is replaced with the failed computer. Third case arises when there is no spare CPU and no spare computer is available leading to total system failure. CPU module activity distribution is given in (1).

cpu failure = exp(0.0052596*cpus->Mark()) (1)
We consider 3 cases in timed activity of cpu_module. Probabilities of those 3 cases are mentioned in Table II.

TABLE II
CPU_MODULE CASE PROBABILITIES OF ACTIVITY

Case	Probability
	cpu_failure
1	if (cpus->Mark() == 3) return(0.995); else return(0.0);
2	if (cpus->Mark() == 3) return(0.00475); else return(0.95);
3	if (cpus->Mark() == 3) return(0.00025); else return(0.05);

Functions of Input Gate and Output Gates are given in Tables III & IV.

TABLE III
CPU_MODULE INPUT GATE FUNCTION

Gate	Function
Input_Gate1	(cpus->Mark(>1))&&(memory failed->Mark(<2))&&(computerfailed->Mark(<num comp))

TABLE IV
CPU_MODULE OUTPUT GATE FUNCTIONS

Gate	Function
OG1	if (cpus->Mark() == 3) cpus->Mark(--);
OG2	cpus->Mark() = 0; ioports->Mark() = 0; errorhandlers->Mark() = 0; memory failed->Mark() = 2; computer failed->Mark(++);
OG3	cpus->Mark() = 0; ioports->Mark() = 0; errorhandlers->Mark() = 0; memory failed->Mark() = 2; computer failed->Mark() = num comp;

Modelling the other sub models is similar to the cpu_module sub model.

B. Composed Model

To form a composed model from the atomic models replicate and join operations are used. Figure 3 shows the multi_proc composed model for the fault tolerant multiprocessor system. Atomic models are represented as leaf nodes. Variables included in the respective atomic models can be shared in all sub models of the system; this sharing is done in the composed model.

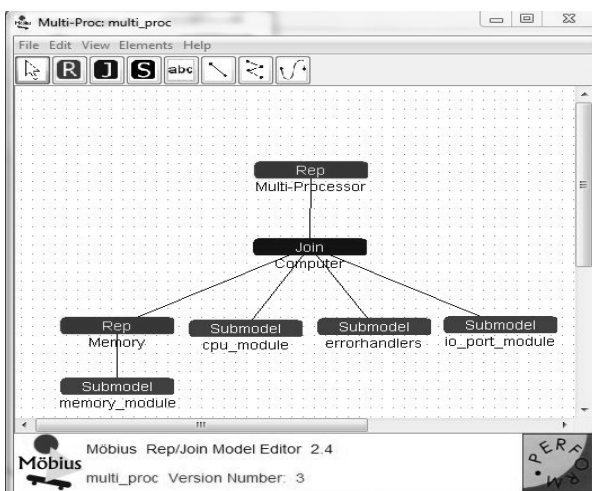


Figure 3. Multi_proc composed model

C. Reward Model

The reward model specifies the performance variable which has to be evaluated. The reward model can be used to measure many performance measures. Reward model is as shown in Fig 4.

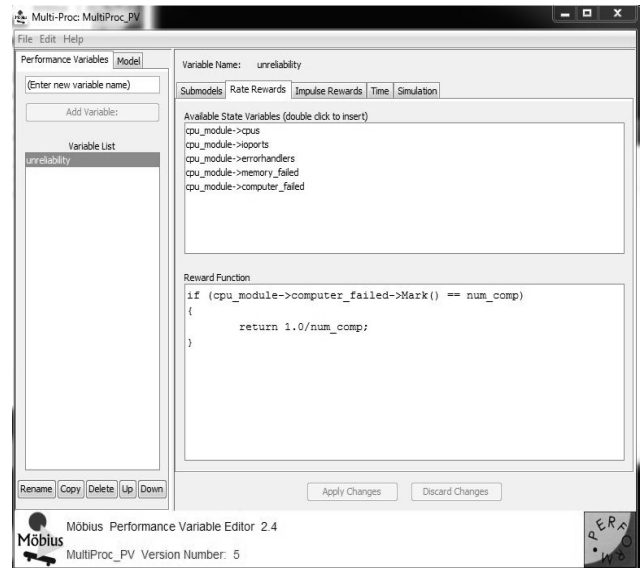


Figure 4. Multiproc_pv reward model

D. Study Model

Global variables of each of the atomic and composed models define the experimental parameters. Different experiments on the system model are defined by giving suitable values in the study model. Study model is shown in Fig 5.

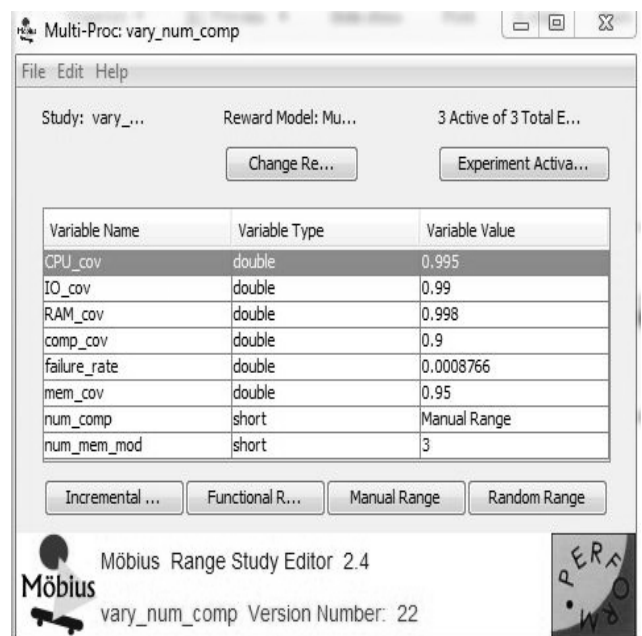


Figure 5. Multi_proc study model

Case I: Number of computers vary in study model. If the number of spare computers is more than 1, then the system reliability lasts longer compared to the system with a single computer. System reliability with varying number of computers is shown in Fig 6.

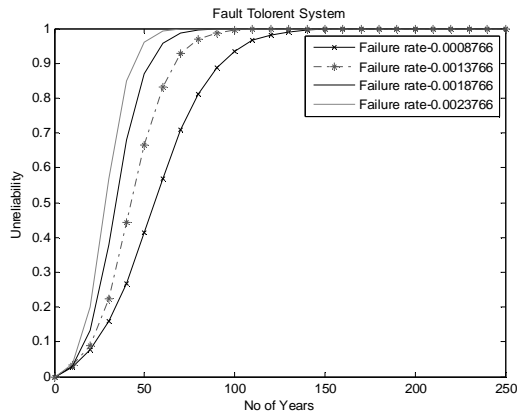


Figure 6. Probability of achieving system lifetime for Variation No of computers

Case II. To represent the operating characteristics of a unit that tends to frequency as it ages, failure rate is a good measure [14]. Failure rates vary in study model. As the failure rate falls, the lifetime of the system increases. The result obtained after varying failure rates is shown in Fig7.

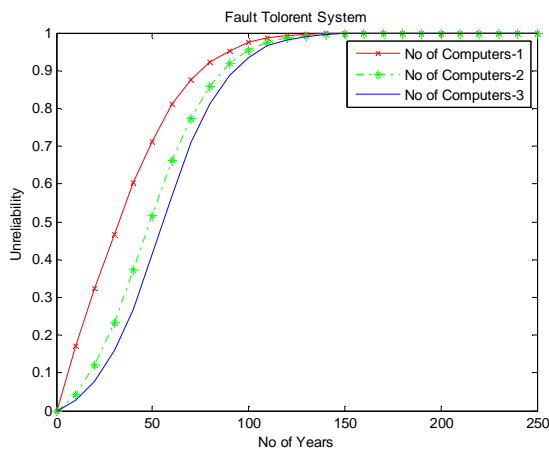


Figure 7. System lifetime for varying failure rates

CONCLUSION

As reliability is a major issue in real time applications, the reliability of a fault tolerant microprocessor system is obtained by modeling it using Mobius. The probability of achieving system lifetime for various spare computers and failure rates is shown in Fig 6 & 7. The proposed system gives the lifetime of the reliable system per number of years.

REFERENCES

- [1] Basset Ross (2003). "When is a Microprocessor not a Microprocessor? The Industrial Construction of Semiconductor Innovation". In Finn, Bernard. Exposing Electronics. Michigan State University Press. p. 121. ISBN 0-87013-658-5.
- [2] Borkar; Shekhar; Chien; Andrew A(2011); The Future of Microprocessor, Communications of the ACM, Vol. 54 Issue 5, May 2011, pp67-77, 11p
- [3] The Uses of Microprocessors (2012); an online material downloadable at http://web.engr.oregonstate.edu/~qassim/index_files/FinalECE570_ASP_2012_Project_Report.pdf Aldosari: "Microprocessors".
- [4] F. J. W. Symons. "Introduction to numerical Petri nets, a general graphical model of concurrent processing systems". Australian Telecommunications Research, 14(1):28– 33, January 1980.
- [5] Gianfranco Balbo "Introduction to Stochastic Petri Nets" Universit`a di Torino, Torino, Italy, Dipartimento di Informatica.
- [6] A. Movaghar, "Stochastic Activity Networks: A New Definition," Proc. of the IASTED Int. Conf. on Modeling and Simulation (MS'97), Pittsburgh, PA, May 1997, pp. 27-30.M.
- [7] M. Abdollahi Azgomi And A. Movaghar "A Modeling Tool For A New Definition Of stochastic Activity Networks" Dept. of Computer Engineering, Sharif University of Technology, Tehran, I. R. of Iran Iranian Journal of Science & Technology, Transaction B, Engineering, Vol. 29, No. B1 Printed in the Islamic Republic of Iran, 2005 © Shiraz University
- [8] W. H. Sanders and J. F. Meyer. A unified approach for specifying measures of performance, dependability, and performability. In A. Avizienis, J. Kopetz, and J. Laprie, editors, Dependable Computing for Critical Applications, volume 4 of Dependable Computing and Fault-Tolerant Systems, pages 215– 237. Heidelberg: Springer-Verlag, 1991.
- [9] "Fault tolerance in Multiprocessor systems".sadhana, vol 11,parts 1&2, October 1987,pp.93-110.
- [10] Rong-Tsong Wang. "Reliability Evaluation Techniques".
- [11] Hussain Al-Asaad and Alireza Sarvi. "Fault tolerance for multi processor systems via time redundant task scheduling".
- [12] "MobiusManual"(<http://www.mobius.illinois.edu/docs/MobiusManual.pdf>).
- [13] T. Nakagawa, Maintenance Theory of Reliability (Springer, London, 2005)