Designing and Valuating System on Dependability Analysis of Cluster-based Multiprocessor System

Dr. Sudarson Jena

Received: 15 December 2019 Accepted: 31 December 2019 Published: 15 January 2020

Abstract

Analysis of dependability is a significant stage in structuring and examining the safety of protection systems and computer systems. The introduction of virtual machines and multiprocessors leads to increasing the faults of the system, particularly for the failures that are software- induced, affecting the overall dependability. Also, it is different for the successful operation of the safety system at any dynamic stage, since there is a tremendous distinction in the rate of failure among the failures that are induced by the software and the hardware. Thus this paper presents a review or different dependability analysis techniques employed in multiprocessor systems

---

Index terms— multiprocessor, fault-tolerant, task scheduling, dependability, reliability

1 Introduction

istributed Systems comprise of inexactly coupled processors that communicate with each other just by transferring messages and lack of common memory among them, consequently the name multiprocessors. In the course of the most recent years, they have gained popularity as an extremely alluring alternative for quick processing of information in enormous scales data frameworks, for example, factory networking, transportation management, and defense must fulfill the standard requirements bound by service interruption, maintainability, and reliability **??**Singhal,2011). Enhancement of the multiprocessor reliability with no equipment cost is the most conventional aim. The framework can contain a few programming upgrades with realistic deadlines and various periods where every upgrade in the software is apportioned into a lot of procedures-related by dependencies on the data. Multiprocessors frameworks have developed a ground-breaking means of computation for ongoing applications, for example, those found in atomic plants and procedure control in light of their ability for superior performance and dependability (Sutar et al., 2006).

The quick progression in innovation over the past decade has empowered us to create many advanced frameworks that range from omnipresent handheld gadgets (like tablets and cellphones) to top of the line processing hardware utilized in health care devices, nuclear plants and power systems. Guaranteeing the dependable working of these modern frameworks is a significant concern to significant engineers (Ahmad et al., 2016). Errors in the external factors and the designs, for example, issues in production, disturbances, and damage caused by external factors, cause changes which are not desired in the physical tent of the framework. These issues are particularly hard to display since it is difficult to anticipate their events and impacts. Hence the fault tolerance is critical for improving the dependability by empowering the PC to carry out its functions within the number of specified flaws **??**Zhao et al., 2013) Dependability is fundamentally characterized as the capacity of a framework for delivering the services that could be trusted reasonably. Dependability is a concept that has developed from considerations of availability and reliability. Various authors depict the system dependability as the set of traits, for example, integrity, confidentiality, availability, safety, maintainability and reliability. A portion of these traits, for example, availability and reliability are qualitative, while some are quantitative. Reliability investigation is a significant pat of the design and evaluation of a computing model that is fault-tolerant to faults. The primary objective of analyzing the system dependability is the development model representing the time which the entire system fails and the policies about maintaining the assemblies, subassemblies, and components from which the framework is formed (Distefano & Puliafito, 2009). One of the important tasks of analyzing dependability of the fault -tolerant systems is predicting the system reliability for the mission time that were mentioned, which is evaluated by probabilistic measures. A general way to deal with handle this is by combining the models of fault trees and Markov chain, for example, parametric fault-tree (PFT) dynamic (DFT), fault-tree (FT).

2 II.

3 Review of Existing Studies a) Fault-tolerant multiprocessor scheduling

There are many challenges in carrying out scheduling in the multiprocessor environment. The scholars are on the edge of determining answers for these difficulties. Scheduling is the specialty of assigning constrained assets to tasks that are competing over time. A practical schedule fulfills the requirements that are related to a specific sort of resources and tasks to improve the multiprocessor performance (Nirmala and Girijamma, 2014). Numerous methods like genetic and fuzzy algorithms were applied, while a few researchers have also developed probabilistic techniques (Berten et al., 2006) for task scheduling in fault-tolerant systems. Zarinzad et al., (2008) introduced an algorithm for fault-tolerant scheduling dependent on GA. This algorithm was intended for the planning of non-preemptive autonomous tasks for a real-time multiprocessor framework. All the tasks are periodic and expected to have backup and primary copies that are allocated to various processors since the backup is executed just if the primary fails as a result of faults. But this algorithm is appropriate for the planning of the periodic tasks.

Bagheri and Jervan, (2014) tended to the underusage issue by proposing a mixed critical scheduling technique with the end goal that the performance of the whole system is increased along with meeting the SC task deadlines in the existence of faults. This methodology handles mixed-criticality in assignments as well as in messages between the tasks. The experiments demonstrated the performance enhancement in various run-time execution conditions and with various MC benchmark applications, including a practical robot control framework. Irrespective of the redundancy level, NoC size, and size of the task graph, an improvement in the performance is achieved. Venkataraman et al., (2013) developed a technique for enduring permanent faults in MPSoC framework. The tasks were mapped to MPSoC that they limit the correspondence and overhead mitigation energy. Also, this study also proposed a module for migrating hardware tasks independent of a processor with predictable delay for speeding up the migration of tasks in case of permanent faults in any processor. In comparison to the existing techniques, the proposed technique was quicker when compared to that of migrating the tasks. Further, utilizing a solitary TMM for different processors can be studied, prompting a further reduction of area overhead. Peng, and Yang, (2015) proposed an algorithm RRFTGS for scheduling in faulttolerant multiprocessor frameworks and also for tests of schedule ability, RRFTGS utilizes both passive and active backups and decides without anyone else which backup should be passive or active. Compared to the approaches that just passive, RRFTGS performed better when the bound was under 0.5. RRFTGS can also manage tasks with usage bound greater than 0.5, while the technique that is just passive cannot. This paper assumed that there is no synchronization between the primary and the backup copies. However, considering synchronization that is present, the processor requesting tolerance to faults will decrease since interference overestimation is decreased. Chatterjee et al. (2017) developed an improvised dynamic solution for the combined issue of scheduling and application mapping for the multicore platforms based on Noc. The introduced algorithm gives a unified technique for scheduling and mapping for realtime frameworks concentrating on fulfilling application time constraints and reducing the energy for communication; To determine the cores within the system, A prescient model is utilized that is prone to failure for which the allocation of fault-tolerant assets along with task redundancy has to be carried out. By specifically utilizing the policy of the replication, the application reliability, executed on the provided NoC stage, is enhanced. Samal et al. (2019) developed a hybrid GA for PBFTS for multiprocessor conditions. Simulation results show that the methodology has been fruitful in acquiring good outcomes for scheduling the tasks in the range of 10 to 100 and also achieved enormous processor utilization for the fault conditions being simulated. The uniqueness lays on the methodology of hybridizing customary GA utilizing the key thought and the information taken from the scheduling of RT tasks and fault tolerance. This has influenced the execution of the PBFTS by improving the genetic operator design and the complete algorithm by the adoption of chromosomal portrayal. Since there is an increase in the size of the multiprocessor system, odds of processors, turning out to be faulty rises, making it a significant issue for diagnosing the faulty nodes within the framework. Various models have been proposed and considered. Lv et al., (2019) stretched out the threshold for applying the probabilistic algorithm for multiprocessor framework based on the hypercube and analyzed the effectiveness of the algorithm. The investigation shows an exceptionally high pace of accurate diagnosis, for separate as well as a group of nodes. In spite of the examination that is accomplished for a specific standard system (the hypercube), the result can fill in as a valuable reference and can add insights into the viability of the probabilistic diagnosis for a group of multiprocessor frameworks that are free of triangles.

4 b) Dependability analysis of fault-tolerant multiprocessor

Evaluating the dependability is significant, often an indispensable stage in the process of designing and analyzing the systems ??Ditstefano & puliafito, 2007). It is broadly perceived that the assessment of the features of dependability in PC frameworks is a complicated task. Conventional procedures which are dependent on simulation and analytical models must be supplemented with experimental techniques that are dependent on estimations, obtained from the models from real frameworks. The above methods including field measurements, robustness testing, injection of faults, have been broadly used to access explicit mechanism of fault tolerance, validating the robustness of the components of the software, or to evaluate the general effect of the framework **??**Pinter et al., 2005).

Bertolino et al., (2011) introduced approaches for monitoring and analyzing connected framework performance and dependability along with their combined utilization. These methodologies need to represent the evolvability and dynamicity of a connected framework. This investigation covered the quantitative evaluation of the properties of performance and dependability via a technique based on the stochastic model. The study initially provided an overview of measurements related to dependability and the approaches based on the stochastic model to give an understanding of the topic. At that point, the proposition in the view of dependability analysis structure for systems that are connected dynamically was portrayed. This structure can be utilized offline for designing the systems (explicitly, in Connect, for the synthesis of Connecter), and online, to consistently to evaluate the behavior of the system and distinguish potential issues emerging at run-time.

Pournaghdali et al., (2013) developed a simulation tool based on the injection of multi-bit faults referred to as VHDLSFI for the model of VHDL. The main objective of this tool development was to infuse multi-bit and single bit fault in the VHDL model of computerized circuits to examine parameters of dependability, for example, error propagation and latency of error propagation. There is an increase in the number of synchronous faults because of higher frequency, lower voltage and decreased code capacitance. Along these lines, studying the impact of multi-bit faults, particularly MB, is challengeable. Grinschgl et al., (2013) presented a case study on various evaluations of security and dependability. An exceptionally modularized controller for fault injection systems is utilized. The campaigns of fault injection can be executed effectively through a summed up interface with an elevated level abstraction of physical sources of fault has been demonstrated. The structure is versatile to permit both completely computerized campaigns with a larger memory of fault patterns and more clientcontrolled campaigns utilizing a little silicon impression. Such full-scale examinations require completely robotized saboteur injection procedures that are presently being developed. Considering such attacks will be important to structure, secure and effective smart card frameworks are additionally valid for a profoundly coordinated framework or framework under high stress from the environment.

Nguyen et al,. (2014), utilizing Petri Nets, introduced a methodology for demonstrating the CBTC framework, where the data is communicated using the LTE network. This model permits the consideration of the transmission methodology and to consider the errors or failures of communication framework. It is likewise incredible for analyzing the dependability of the CBTC framework. The outcomes featured the accessibility of the LTE-based on DCS in applications controlling the trains. Suyama and Sebe, (2014) presented another concept of "available state" and utilizing Markov models, the author described the framework for analyzing the dependability for faulttolerance, and hence it prevails with regards to improving the system practically using constrained integrators. Also, the thought can assimilate fundamental job in setting up the participation with dependability or reliability engineering. The outcomes highlighted the availability of the LTE-based DCS in train control application.

5 III.

6 Comparative Analysis

IV.

7 Conclusion

Analyzing dependability in the process of designing and evaluating the systems is crucial since the introduction of computing systems for automating the processes results in increasing the system complexity thus, affecting the dependability. This paper reviewed different dependability analysis techniques for fault-tolerant multiprocessor. It was observed that the techniques like Petri nets, Markov chains, DFT, Fuzzy theory conveniently supported the dependability analysis for fault-tolerant systems. Finally, it, ought to be focused on the hypothetical action that is identified with the advancement of models of dependability. It is crucial for adapting to the increased complexity of the faulttolerant frameworks but, for this to be fruitful and applicable, practically one must continue together with actually implementing the frameworks that are tolerant to faults and with the information on experimental data with respect to the features of performance and rate of failure of the integrated modules in the process of FT realization. [1] [2]

Miele et al., (2014)	Error analysis and fault injection	Anti-lock braking framework, edge detector	Error propagation	The experimental outcomes demonstrated the effectiveness of the technique in producing a precise dependability report focusing on the criticalities of the system application as well as architecture.
Nguyen et al., (2014)	Petrinet modelling	Train control	Mean values of Down and Up Times, Mean Time to First Failure	
Author	Technique	Applications	Parameters studied	Results obtained
Ter et al., (2010)	Self-testing techniques	Satellite Navigation	Reliability, Maintainability, Unavailability	The developed approach enabled quick diagnosis or detection of electronic fault and repair, thus increasing the MPSoc availability.
Masci et ., (2011)	Automatic dependability analysis	Dynamic and heterogeneous environments	Sensitivity, message retransmission	Defined the automated process for supporting the dependable connector synthesis.
Gultai et al., (2012)	Fuzzy theory	Distributed system	Failure rate, CPU Time, reliability	Increasing the no of processors.

Figure 1:

[Pinter et al.] , G Pinter , H Madeira , M Vieira , I Majzik .

[Patricza (2005)] ‘A datamining approach to identify key factors in dependability experiments’. A Patricza . *European Dependable Computing Conference*, (Berlin, Heidelberg) 2005. April. Springer. p. .

[Micle ()] ‘A fault-injection methodology for the system-level dependability analysis of Multiprocessor embedded systems’. A Micle . *Microprocessors and Microsystems* 2014. 38 (6) p. .

[Berten et al. (2006)] ‘A probabilistic approach for fault tolerant multiprocessor real-time scheduling’. V Berten , J Goossens , E Jeannot . *Proceedings 20th IEEE International Parallel & Distributed Processing Symposium*, (20th IEEE International Parallel & Distributed Processing Symposium) 2006. April. IEEE. p. 8.

[Gulati and Bhatia ()] ‘A reliability model for the task scheduling in Distributed System based on Fuzzy Theory’. S Gulati , Yadav P K Bhatia , K . *CIIT International Journal of Networking and Communication and Engineering* 2012. 4 (11) p. .

[Grinschgl et al. ()] *Case study on multiple fault dependability and security valuations Microprocessors and Microsystems*, J Grinschgl , A Krieg , C Steyer , R Weiss , H Bock , J Haid , C ?& Ulbricht . 2013. 37 p. .

[Zhu et al. ()] ‘Dependability analysis for fault tolerant computer systems using dynamic and graphs’. F Zhu , H Jin , D Zou , P Qin . *China Communications* 2014. 11 (9) p. .

[Suyama and Sebe (2014)] ‘Dependability analysis of fault-tolerant servo systems using limited integrators’. K Suyama , N Sebe . *European Control Conference* 2014. June. 2014. IEEE. p. .

[Bertolino et al. (2011)] ‘Dependability and performance assessment of dynamic connected systems’. A Bertolino , A Calabro , F Di Giandomenico , N Nostro . *International school on formal methods for the design of computer, communication and software systems*, (berlin, Heidelberg) 2011. June. Springer. p. .

[Distefano and Puliafito ()] ‘Dependability evaluation with dynamic reliability block diagrams and dynamic fault trees’. S Distefano , A Puliafito . *IEEE Transactions on Dependable and Secure Computing* 2009. p. .

[Distefano and Puliafito (2007)] ‘Dependability modeling and analysis in dynamic system’. S Distefano , A Puliafito . *2007 IEEE International Parallel and Distributed Processing Symposium*, 2007 March. IEEE. p. .

[Singhal ()] ‘Distributed systems computing upgradation & reliability tecnia’. V Singhal . *Journal of Management Studies* 2011. 6 (2) p. 61.

[Peeg (2015)] ‘Fault Tolerant Global Scheduling for Multiprocessor Hard Real Time Systems’. H &yang F Peeg . *First International Conference on Information Sciences, Machinery, Materials and Energy*, 2015. July. Atlantis Press.

[Samal et al. ()] ‘Fault tolerant scheduling of hard real-time tasks on multiprocessor system using a hybrid genetic algorithm’. A K Samal , R Mall , C Tripathy . *Swarm and Evolutionary Computation* 2014. 14 p. .

[Chatterjee et al. ()] ‘Fault-tolerant dynamic task mapping and scheduling for network-on-chip-based multicore platform’. N Chatterjee , S Paul , S Chattopadhyay . *ACM Transactions on Embedded Computing Systems (TECS)* 2017. 16 (4) p. 108.

[Bagheri and Jervan (2014)] ‘Faulttolerant scheduling of mixed-critical application on multi-processor platforms’. M Bagheri , G Jervan . *12th IEEE International Conference on Embedded and Ubiquitous Computing*, 2014 August. 2014. IEEE. p. .

[Ahmad and Hasan tahar (2016)] ‘Formal dependability modeling and analysis .A Survey’. W Ahmad , S Hasan &tahar . *International Conference on Intelligent Computer Mathematics*, (Cham) 2016. July. Springer. p. .

[Nirmala and Girijamma ()] ‘Fuzzy scheduling algorithm for real time multiprocessor system’. H Nirmala , H A Girijamma . *Int. J. Sc Eng Res* 2014. (7) p. 5.

[Venkataraman et al. (2015)] ‘Hardware task migration module for improved fault tolerance and predictability’. S Venkataraman , R Santos , A Kumar , J Kuijsten . *2015 International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS)*, 2015. July. IEEE. p. .

[Masci et al. ()] ‘March) Towards automated dependability analysis of dynamically connected systems’. P Masci , M Martinucci , F Di Giandomenico . *2011 Tenth International Symposium on Autonomous Decentralized Systems*, 2011. IEEE. p. .

[Nguyen et al. ()] ‘Modelling communication based train control system for dependability analysis of the LTE communication network in train control application’. Nguyen , J Beugin , M Berbineau , M Kassab . *European modeling Symposium* 2014. 2014. IEEE. p. . (October))

[Ter Braak et al. (2010)] ‘On-line dependability enhancement of multiprocessor SoCs by resource management’. T D Ter Braak , S T Burgess , H Hurskainen , H G Kerkhoff , B Vermeulen , X Zhang . *2010 International Symposium on System on Chip*, 2010. September. IEEE. p. .

[Zhou et al. ()] 'Probabilistic diagnosis of clustered faults for hypercube-based multiprocessor system'. L , M Zhou , S Sun , X Lian , G Liu , J D Wang . *Theoretical Computer Science* 2019. 793 p. .

[Sutar et al. ()] 'Task scheduling for multiprocessor systems using memetic algorithms'. S Sutar , J Sawant , J Jadhav . *4th International Working Conference Performance Modeling and Evaluation of Heterogeneous Network (HETNET '06)*, 2006.

[Pournaghdali et al. (2013)] *VHDLSFI: A simulation-based multi-bit fault injection for dependability analysis In ICCKE*, F Pournaghdali , A Rajabzadeh , M Ahmadi . 2013. October. 2013. IEEE. p. .

[Zarinzad et al. (2008)] G Zarinzad , A Masoud Rahmani , N Dayhim . *A novel intelligent algorithm for fault-tolerant task scheduling in real-time multiprocessor system In2008 Third International Conference on Convergence and Hybrid Information Technology*, 2008. November. IEEE. 2 p. .