JNP: Software Fault Tolerance in Real-Time Systems: Identifying the Future Research Questions

Federico Reghenzani*†, Zhishan Guo*, William Fornaciari†,
*North Carolina State University, †Politecnico di Milano
zguo32@ncsu.edu, {federico.reghenzani,william.fornaciari}@polimi.it

***Abstract*—This "Journal-Never-Presented" paper summarizes our recent survey paper published at ACM Computing Surveys [23] on software fault tolerance in real-time systems.**

I. Motivation and Background

In hard real-time systems, ensuring both temporal and logical requirements is essential. Typically, correctness is verified through scheduling analysis for temporal requirements and formal verification or extensive testing for logical requirements. However, hardware faults can still jeopardize system integrity, even with a flawless design.

To achieve dependability, hardware solutions like redundancy are commonly employed. Alternatively, software techniques, collectively known as Software-Implemented Hardware Fault Tolerance (SIFT) [29], can be used. SIFT solutions are cost-effective, making them valuable for Commercial Off-The-Shelf (COTS) hardware, which may lack strict fault tolerance designs [21].

The trend toward smaller, more powerful hardware components poses challenges. Increased fault rates and complex features like multi-core architectures make it harder to guarantee temporal requirements. The key challenge is computing the Worst-Case Execution Time [6].

II. Relationship with Mixed Criticality

Motivated by the impact of fault tolerance on real-time performance, this paper explores the intersection of fault tolerance algorithms and real-time scheduling. While numerous fault-tolerant solutions and scheduling algorithms exist, the relationship between scheduling decisions and fault tolerance is underexplored. There is significant potential for research in fault-tolerant mixed-criticality (MC) real-time systems [3], an emerging and promising direction.

In our interpretation of MC, `HI`-criticality tasks are allowed to "fail" in timing sense by overrunning their `LO`-criticality WCET, due to an underestimation of it. Such concept of tolerating temporal faults (such as how to apply fault tolerance and graceful degradation, making the system robust and resilient, e.g., in [4]), is orthogonal to the topics in our paper, where we instead focus on hardware faults and SIFT.

III. Major Contributions

The goal of the journal paper is to survey the intersection of fault tolerance approaches and real-time scheduling algorithms and, particularly, on the identification of open problems. In particular, the survey paper made the following contributions:

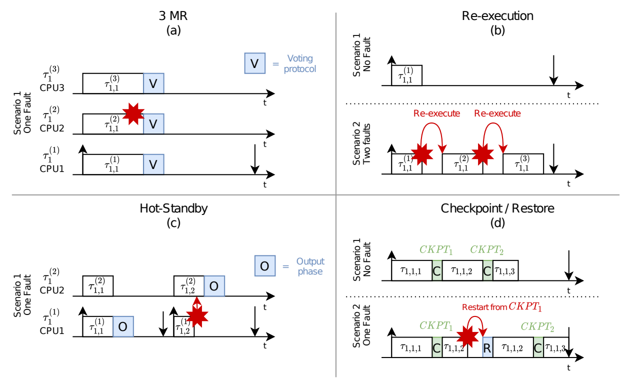

Fig. 1. The four main software fault recovery approaches. In (a), (b), and (c) the symbol $\tau_{i,j}^{(k)}$ represents the j-th job of the i-th task, and k identifies the redundancy or re-execution job. Instead, in (d), $\tau_{i,j,k}$ is the k-th part of the job $\tau_{i,j}$: In the depicted example, the job $\tau_{1,1}$ is composed of three parts.

- We introduced basic related concepts and provided a background of real-time modeling. E.g., Figure 1 depicts software fault recovery approaches.
- We reviewed the current literature on (hard) real-time scheduling analyses and techniques when fault tolerance, in particular SIFT, is considered. E.g., Figure 2 uses Venn diagram to summarize existing work according to the used fault tolerance mechanism(s).
- We proposed a dangerous failure model as well as transient and permanent hardware fault rates (and they are independent of each other). We pointed out the relationship between such failure model and Bernoulli processes, and presented formulas for deriving the probability of observing a fault in a single time unit, as well as the probability of a fault to occur in a given job.
- With the proposed fault model(s), we outlined the current challenges and future possible research directions for fault-tolerant real-time systems in the following seven categories: (1) The impact of scheduling decisions on fault tolerance; (2) Scheduling analysis of fault tolerance approaches; (3) Mixed-criticality and fault tolerance; (4) The effect of power management techniques; and (5) The implementation of the Operating System and scheduler; (6) Exploiting probabilistic information; (7) Other aspects such as (k,n)-failure model, approximate computing, and security issues such as malicious faults.

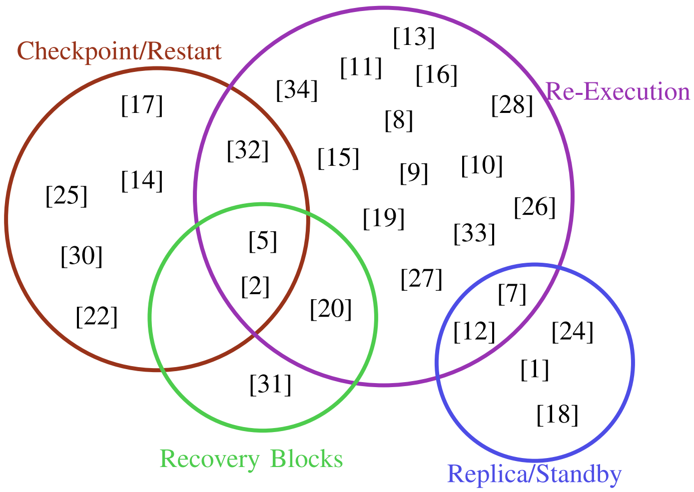

Fig. 2. State-of-the-art papers classified via fault tolerance technique.

References

[1] A. Bhat, S. Samii, and R. R. Rajkumar. Practical task allocation for software fault-tolerance and its implementation in embedded automotive systems. In *Proceedings of the IEEE Real-Time and Embedded Technology and Applications Symposium, RTAS*, pages 87–97, Pittsburgh, PA, USA, 2017. IEEE.

[2] A. Burns and R. Davis. Feasibility Anlaysis of Fault-Tolerant Real-Time Task Sets. In *Proceedings of the Eighth Euromicro Workshop on Real-Time Systems*, pages 29–33, L'Aquila, Italy, 1996. IEEE.

[3] A. Burns and R. Davis. Mixed criticality systems - a review, 2019.

[4] A. Burns, R. I. Davis, S. Baruah, and I. Bate. Robust mixed-criticality systems. *IEEE Transactions on Computers*, 67(10):1478–1491, 2018.

[5] A. Burns, S. Punnekkat, L. Strigini, and D. R. Wright. Probabilistic scheduling guarantees for fault-tolerant real-time systems. In *Dependable Computing for Critical Applications 7*, pages 361–378, San Jose, CA, USA, 1999. IEEE.

[6] R. Canal, C. Hernandez, R. Tornero, A. Cilardo, G. Massari, F. Reghenzani, W. Fornaciari, M. Zapater, D. Atienza, A. Oleksiak, W. Piundefinedtek, and J. Abella. Predictive reliability and fault management in exascale systems: State of the art and perspectives. *ACM Comput. Surv.*, 53(5), Sept. 2020.

[7] K. Chen, G. v. der Brüggen, and J. Chen. Reliability optimization on multi-core systems with multi-tasking and redundant multi-threading. *IEEE Transactions on Computers*, 67(4):484–497, 2018.

[8] S. Ghosh, R. Melhem, and D. Mosse. Enhancing real-time schedules to tolerate transient faults. In *Proceedings 16th IEEE Real-Time Systems Symposium*, pages 120–129, Pisa, Italy, 1995. IEEE.

[9] S. Ghosh, R. Melhem, D. Mossé, and J. Sen Sarma. Fault-Tolerant Rate-Monotonic Scheduling. *Real-Time Systems*, 15(2):149–181, 1998.

[10] M. A. Haque, H. Aydin, and D. Zhu. Real-time scheduling under fault bursts with multiple recovery strategy. In *2014 IEEE 19th Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 63–74, Berlin, Germany, 2014. IEEE.

[11] P. Huang, H. Yang, and L. Thiele. On the scheduling of fault-tolerant mixed-criticality systems. In *Proceedings - Design Automation Conference*, pages 1–6, New York, NY, USA, 2014. IEEE.

[12] S. Kang, Hoeseok Yang, Sungchan Kim, I. Bacivarov, Soonhoi Ha, and L. Thiele. Static mapping of mixed-critical applications for fault-tolerant mpsocs. In *2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC)*, pages 1–6, New York, NY, USA, 2014. IEEE.

[13] A. Kritikakou, P. Nikolaou, I. Rodriguez-Ferrandez, J. Paturel, L. Kosmidis, M. K. Michael, O. Sentieys, and D. Steenari. Functional and timing implications of transient faults in critical systems. In *2022 IEEE 28th International Symposium on On-Line Testing and Robust System Design (IOLTS)*, pages 1–10, Torino, Italy, 2022. IEEE.

[14] S. W. Kwak, B. J. Choi, and B. K. Kim. An optimal checkpointing-strategy for real-time control systems under transient faults. *IEEE Transactions on Reliability*, 50(3):293–301, 2001.

[15] J. D. Lin, A. M. K. Cheng, D. Steel, and M. Y. Wu. Scheduling mixed-criticality real-time tasks with fault tolerance. In *Proc. 2nd Workshop on Mixed Criticality Systems (WMC), RTSS*, pages 39–44, Rome, Italy, 2014. WMC.

[16] F. Many and D. Doose. Scheduling analysis under fault bursts. In *Real-Time Technology and Applications - Proceedings*, pages 113–122, Chicago, IL, USA, 2011. IEEE.

[17] R. Melhem, D. Mossé, and E. Elnozahy. The Interplay of Power Management and Fault Recovery in Real-Time Systems. *IEEE Transactions on Computers*, 53(2):217–231, 2004.

[18] L. Niu, J. Musselwhite, and W. Li. Work-in-Progress: Enhanced Energy-Aware Standby-Sparing Techniques for Fixed-Priority Hard Real-Time Systems. *Proceedings - Real-Time Systems Symposium*, 2018-Decem:165–168, 2018.

[19] M. Pandya and M. Malek. Minimum achievable utilization for fault-tolerant processing of periodic tasks. *IEEE Transactions on Computers*, 47(10):1102–1112, 1998.

[20] R. M. Pathan. Fault-tolerant and real-time scheduling for mixed-criticality systems. *Real-Time Systems*, 50(4):509–547, Jul 2014.

[21] J. Pellish. Commercial off-the-shelf (cots) electronics reliability for space applications. Technical report, NASA / Goddard Space Flight Center, Greenbelt, MD, USA, 2018.

[22] S. Punnekkat, A. Burns, and R. Davis. Analysis of checkpointing for real-time systems. *Real-Time Systems*, 20(1):83–102, Jan 2001.

[23] F. Reghenzani, Z. Guo, and W. Fornaciari. Software fault tolerance in real-time systems: Identifying the future research questions. *ACM Comput. Surv.*, 55(14s), jul 2023.

[24] S. Safari, M. Ansari, G. Ershadi, and S. Hessabi. On the scheduling of energy-aware fault-tolerant mixed-criticality multicore systems with service guarantee exploration. *IEEE Transactions on Parallel and Distributed Systems*, 30(10):2338–2354, 2019.

[25] M. Salehi, M. K. Tavana, S. Rehman, M. Shafique, A. Ejlali, and J. Henkel. Two-State Checkpointing for Energy-Efficient Fault Tolerance in Hard Real-Time Systems. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 24(7):2426–2437, 2016.

[26] A. Thekkilakattil, R. Dobrin, and S. Punnekkat. Fault tolerant scheduling of mixed criticality real-time tasks under error bursts. *Procedia Computer Science*, 46:1148 – 1155, 2015. Proceedings of the International Conference on Information and Communication Technologies, ICICT 2014, 3-5 December 2014 at Bolgatty Palace & Island Resort, Kochi, India.

[27] A. Thekkilakattil, R. Dobrin, S. Punnekkat, and H. Aysan. Resource augmentation for fault-tolerance feasibility of real-time tasks under error bursts. In *Proceedings of the 20th International Conference on Real-Time and Network Systems*, RTNS '12, page 41–50, New York, NY, USA, 2012. Association for Computing Machinery.

[28] G. von der Brüggen, K. H. Chen, W. H. Huang, and J. J. Chen. Systems with dynamic real-time guarantees in uncertain and faulty execution environments. In *2016 IEEE Real-Time Systems Symposium (RTSS)*, pages 303–314, Porto, Portugal, 2016. IEEE.

[29] J. Wensley, L. Lamport, J. Goldberg, M. Green, K. Levitt, P. Melliar-Smith, R. Shostak, and C. Weinstock. Sift: Design and analysis of a fault-tolerant computer for aircraft control. *Proceedings of the IEEE*, 66(10):1240–1255, 1978.

[30] Y. Zhang and K. Chakrabarty. A unified approach for fault tolerance and dynamic power management in fixed-priority real-time embedded systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(1):111–125, 2006.

[31] B. Zhao, H. Aydin, and D. Zhu. Shared recovery for energy efficiency and reliability enhancements in real-time applications with precedence constraints. *ACM Transactions on Design Automation of Electronic Systems*, 18(2), 2013.

[32] Z. Zhengyong, P. Liping, and Y. Fumin. Schedulability analysis for fault tolerance real-time system under fault bursts. In *2014 IEEE 7th Joint International Information Technology and Artificial Intelligence Conference*, pages 20–27, Chongqing, China, 2014. IEEE.

[33] J. Zhou, M. Yin, Z. Li, K. Cao, J. Yan, T. Wei, M. Chen, and X. Fu. Fault-tolerant task scheduling for mixed-criticality real-time systems. *Journal of Circuits, Systems and Computers*, 26(1):1–17, 2017.

[34] D. Zhu and H. Aydin. Reliability-aware energy management for periodic real-time tasks. *IEEE Transactions on Computers*, 58(10):1382–1397, 2009.