On the suitability of Time-Randomized Processors for Secure and Reliable High-Performance Computing

David Trilla[†,‡], Carles Hernandez[†], Jaume Abella[†], Francisco J. Cazorla[†,⋆]
[†] Barcelona Supercomputing Center (BSC). Barcelona, Spain
[‡] Universitat Politècnica de Catalunya (UPC), Barcelona, Spain
[⋆] Spanish National Research Council (IIIA-CSIC). Barcelona, Spain.

***Abstract*—Time-randomized processor (TRP) architectures have been shown as one of the most promising approaches to deal with the overwhelming complexity of the timing analysis of high complex processor architectures for safety-related real-time systems. With TRPs the timing analysis step mainly relies on collecting measurements of the task under analysis rather than on complex timing models of the processor. Additionally, randomization techniques applied in TRPs provide increased reliability and security features. In this thesis, we elaborate on the reliability and security properties of TRPs and the suitability of extending this processor architecture design paradigm to the high-performance computing domain.**

I. Introduction

Several years ago probabilistic timing analysis (PTA) [1] arised as a new timing analysis paradigm with the aim to facilitate the timing-verification of complex processors running safety-critical applications. Amongst other approaches, PTA proposes the utilization of time-randomized processors (TRPs) as a way to enable the derivation of timing bounds using probabilistic methods. Typically, complex commercial off-the-shelf (COTS) processor architectures are not amenable for timing-critical applications since they include hardware features such as caches, branch predictors, and multicore technology that severely complicates the derivation of execution time-bounds for the programs running on top of these processors. PTA applied on top of TRPs allows to reduce the complexity of the timing analysis process. To do so, TRPs employ hardware randomization techniques to make the latency of jittery resources to exhibit a probabilistic nature and thus, to allow the application of extreme value theory (EVT) [13] to bound program's execution time. Basically TRPs simplify the timing verification process.

Typically processors in each domain (i.e. personal devices, supercomputers, etc...) have been subject to different requirements in terms of area, power consumption, temperature, performance, reliability and security. For instance, in the high-performance domain reliability has been a second order concern for many years in comparison with performance first and power/temperature later. Conversely, in other domains like safety-critical systems, reliability and security have been primary concerns due to their potential threads affecting human lives.

However, the need to push technology scalability limits further every generation in the high-performance domain has made faults to be more frequent and thus, reliability has become also a primary concern in this domain. Additionally, high-performance processors used in data-centers and servers have to meet strong security requirements to avoid malicious attacks.

The properties of TRPs to enable the utilization of high-performance processors in time-critical applications at a reasonable cost have been deeply analyzed [12]. Our focus is to highlight the reliability and security properties TRPs can offer to both high-performance and safety-critical domains.

II. Achieving predictability with non-reproducible timing behavior

The main difference between TRPs and conventional (time-deterministic) processor designs resides on the way hardware resources exhibiting jitter, i.e. what factors trigger different latencies for those resources that do not exhibit a constant latency.

TRPs provide hardware support so that execution time measurements collected during analysis match or upperbound tightly those during operation. To upper-bound the jitter probabilistically, randomization (RND) techniques are required to make the jitter have a probabilistic nature during both analysis and operation phases.

RND techniques have been applied satisfactorily to caches and shared resource arbitration. Regarding caches, random replacement and random placement policies have been proposed to match TRP requirements [8]. Figure 1 shows and example of random modulo architecture. For the arbitration of shared resources the lottery bus [14] has been shown suitable for TRPs, while other new approaches like random permutations [10] have been shown also suitable and offer improved performance.

When all jittery resources are properly handled, EVT can be applied to the measurements collected for the programs executed in TRPs. The timing analysis methodology for which TRPs were proposed is called Measurement-Based Probabilistic Timing Analysis (MBPTA) [3], [2]. Time-measurements collected on top TRPs follow a probabilistic nature, are independent and identically distributed, and represent an upper-bound of the worst events due to the interactions of the different jittery resources that can occur in the processor. Once appropriate tests are passed, those measurements are used

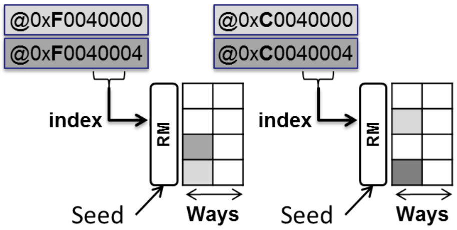

Fig. 1. Random modulo architecture.

as input for EVT, which is a powerful statistical method to approximate the tail of a distribution that represents the high execution times. Then, the probabilistic WCET (pWCET) is the execution time value of the obtained distribution whose risk of being exceeded is upper-bounded by an arbitrarily low probability to be neglected (i.e. residual risk [9]). Figure 2 shows a pWCET curve with a cutoff probability of 10^{-14}.

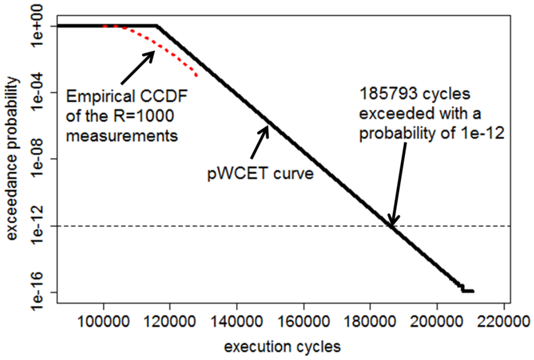

Fig. 2. Example pWCET obtained with a TRP.

III. Reliability Properties of Time-Randomized Processors

The majority of current processors, regardless of the application domain, are provided with some form of fault tolerance. We analyze how TRPs help maximizing the effectiveness of existing protection mechanisms and also provide enhanced robustness capabilities.

A. Improving graceful performance degradation

Fault-tolerant mechanisms are able to keep system functioning despite the presence of faults but sometimes this comes at the expense of a reduction in performance. For example, when one or several cache lines are permanently damaged, protection mechanisms disabling faulty lines allow the processor to operate correctly enlarging its lifetime. However, depending on the particular location of those faulty lines in cache, the performance provided by the processor can vary significantly [4]. Random cache policies included in TRPs make this degradation to occur more gracefully [16]. Figure 3 shows how the execution time for two benchmarks behaves as we decrease the cache space.

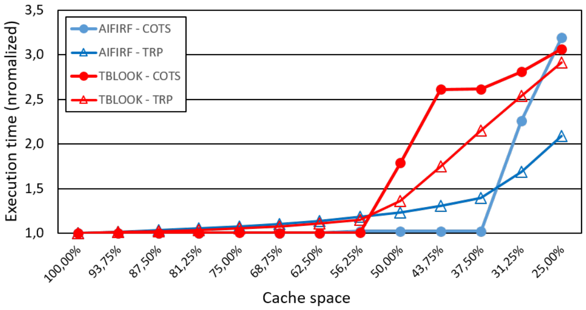

Fig. 3. Execution time (normalized) as cache size decreases for COTS and TRP designs in DL1 caches.

B. Reducing the bias in resource utilization

With conventional placement algorithms, cache sets access distribution is completely program dependent. On the contrary, with random placement algorithms [8], [11] accesses to the cache sets are randomly distributed since random placement algorithms employ a combination of address tags with random bits from a random seed to generate the cache index so, for every run, a different set is accessed for any given address. Having a highly biased cache set utilization is expected to lead to higher degradation since the most used sets are more exposed to hot carrier injection (HCI) [6] among other sources of transistor degradation. In this context, having set access distributions as uniform as possible is very convenient to mitigate those aging effects [7], [17]. Figure 4 shows that almost perfect set access distributions can be achieved with TRPs.

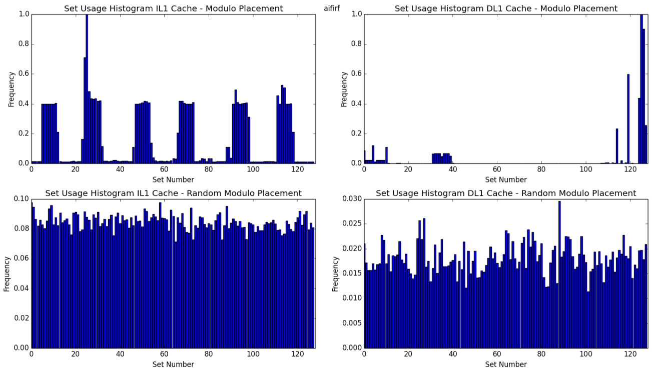

Fig. 4. Cache utilization across sets with and without random placement.

C. Power stability and voltage noise resilience

Randomization removes systematic pathological scenarios that can lead to corner situations with significantly bad performance. High demanding power events that occur at a given frequency and align with the resonance frequency of the power network distribution will significantly amplify voltage noise fluctuations [5]. Pathological cases occur, for example, due to systematic cache conflicts.

TRPs break systematic alignments of the events. Thus, TRPs can diminish the impact of voltage noise effects. Figure 5 shows the fast fourier transform of the power consumption resulting from the execution of a LRU pathological case in a time-deterministic processor (a) and in a TRP (b).

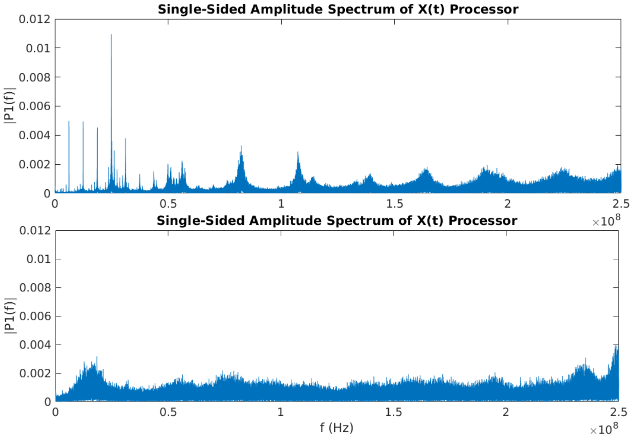

Fig. 5. Fast fourier transform of the power consumption of pathological patterns in a TRP (a) and in an time-deterministic one (b).

IV. Security Properties of Time-Randomized Processors

Side-channel attacks extract secret key data by exploiting the information leakage resulting from the physical implementation of the system. TRPs can mitigate this effects.

A. Cache side-channel attacks

Cache side-channel attacks are characterized by the abuse of it's controllable and known behavior. Cache misses and hits are reflected on the execution time, thus they can be measured. This hits and misses are dependent on the placement of the data used, a pattern of execution times may indicate that certain data is being used and through this knowledge, cryptographic keys can be inferred. Layout randomization has been shown to be an effective mechanism to protect against contention-based attacks [18]. Since TRPs are based on the utilization of random cache designs [8], these processor designs are inherently protected.

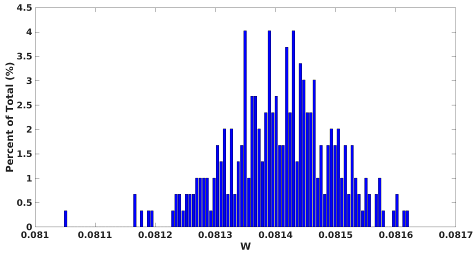

Fig. 6. Power dissipation variability in a TRP.

B. Power analysis side channels

Power dissipation can also leak cryptographic information. When instructions are executed with fixed-time repetitive executions, they provide similar power profiles. Therefore, since cryptographic algorithms use multiple iterations for a given secret key, attackers can match the similar power profiles obtained to infer the cryptographic data. Randomizing the execution time delay to achieve protection against power analysis attacks has been proved useful. TRPs provide time randomization by default by randomizing the existing processor jitter and thus, avoid detection of the execution of cryptographic algorithms. Figure 6 shows the power variability resulting from running 1000 times an encryption algorithm in a TRP.

V. Towards Secure and Reliable High Performance Time Randomised Processors: Conclusions and Future Work

A. Further Hardware Randomization and other EVT applications

In order for TRPs to be fully embraced some research is needed on unexplored features of EVT and hardware randomization. Some are listed below.

Hardware Randomization. Two important improvements to processors that due to being too complex to time analyze are not suitable for critical real-time systems are Data prefetchers and Out-of-Order execution. Both are able to partially hide latency of the program execution. At the same time they require randomization so systematic behaviors are avoided and probabilities attached to events. Furthermore randomizing prefetching policies can also enhance TRPs protection against side-channel attacks [15].

Extended EVT applications. EVT is an important tool that can be applied in other important aspects different from execution time bounding. Voltage Noise and Maximum Power Density are important factors when designing systems since reaching unsuspected limits on those domains can severely affect the behavior of the processor. Applying EVT on top of TRPs might provide safe bounds to power dissipation and can help in mitigating overdimensioning the safety margins.

B. Conclusions

As shown, TRPs provide unique properties that make them an ideal baseline to combat some of the reliability and security challenges that high-performance processors have to face these days. However, future work is needed to enhance these features. Despite this TRPs seem promisingly suitable to be used in markets other than safety-critical systems.

References

[1] F. Cazorla et al. PROARTIS: Probabilistically analysable real-time systems. *ACM TECS*, 2012.

[2] F. Cazorla et al. Upper-bounding program execution time with extreme value theory. In *WCET Workshop*, 2013.

[3] L. Cucu-Grosjean et al. Measurement-based probabilistic timing analysis for multi-path programs. In *ECRTS*, 2012.

[4] J. A. et. al. Low vccmin fault-tolerant cache with highly predictable performance. MICRO 42, pages 111–121, New York, NY, USA, 2009. ACM.

[5] R. B. et. al. Voltage noise in multi-core processors: Empirical characterization and optimization opportunities. In *MICRO '14*, pages 368–380, Cambridge, UK, December 2014. IEEE Computer Society.

[6] V. H. et al. Managing sram reliability from bitcell to library level. In *Reliability Physics Symposium (IRPS), 2010 IEEE International*, pages 655–664, May 2010.

[7] E. Gunadi, A. A. Sinkar, N. S. Kim, and M. H. Lipasti. Combating aging with the colt duty cycle equalizer. In *Microarchitecture (MICRO), 2010 43rd Annual IEEE/ACM International Symposium on*, pages 103–114, Dec 2010.

[8] C. Hernandez et al. Random modulo: a new processor cache design for real-time critical systems. In *DAC*, 2016.

[9] International Organization for Standardization. *ISO/DIS 26262. Road Vehicles – Functional Safety*, 2009.

[10] J. Jalle, L. Kosmidis, J. Abella, E. Quinones, and F. Cazorla. Bus designs for time-probabilistic multicore processors. In *DATE*, 2014.

[11] L. Kosmidis et al. A cache design for probabilistically analysable real-time systems. In *DATE*, 2013.

[12] L. Kosmidis et al. Probabilistic timing analysis and its impact on processor architecture. In *DSD*, 2014.

[13] S. Kotz and S. Nadarajah. *Extreme value distributions: theory and applications*. World Scientific, 2000.

[14] K. Lahiri, A. Raghunathan, and G. Lakshminarayana. LOTTERYBUS: a new high-performance communication architecture for system-on-chip designs. DAC '01, pages 15–20, 2001.

[15] F. Liu and R. B. Lee. Random fill cache architecture. In *Proceedings of the 47th Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO-47, pages 203–215, Washington, DC, USA, 2014. IEEE Computer Society.

[16] M. Slijepcevic et al. Timing verification of fault-tolerant chips for safety-critical applications in harsh environments. *IEEE Micro - Special Series on Harsh Chips*, 34(6), 2014.

[17] D. Trilla, C. Hernandez, J. Abella, and F. Cazorla. Resilient random modulo cache memories for probabilistically-analyzable real-time systems. In *IOLTS*, 2016.

[18] Z. Wang and R. B. Lee. A novel cache architecture with enhanced performance and security. In *2008 41st IEEE/ACM International Symposium on Microarchitecture*, pages 83–93, Nov 2008.

David Trilla is a PhD. Student for the CAOS group at BSC. He obtained his M.S. degree in 2016 and graduated in Informatics Engineering in 2014, both titles obtained from the Universitat Politecnica de Catalunya. He enrolled BSC in 2014 and has participated in the European project ESA-HAIR. He has been working on timing prediction models for real-time software for multicore during early design stages and his current research focuses on the effects on energy consumption behavior and security on randomized architectures.