Brief Industry Paper: Digital Twin for Dependable Multi-Core Real-Time Systems — Requirements and Open Challenges

Xiaotian Dai*, Shuai Zhao*, Iain Bate*, Alan Burns*, Xing Guo†, Wanli Chang*
*Department of Computer Science, University of York, UK
†AISN Auto R&D Co., Ltd., China
*{xiaotian.dai, shuai.zhao, iain.bate, alan.burns, wanli.chang}@york.ac.uk
†guoxing_aisn@163.com

***Abstract*—Development of dependable multi-/many-core systems requires assurance that the system is operable in a range of conditions, subjected to both functional and non-functional requirements. To achieve this, tools need to be implemented that can enable exploration of design options and be able to detect deficiencies earlier to avoid costly system re-design. In this work we discuss the challenges of design of multi-core real-time systems with timing assurance and discuss what are the requirements for modelling, testing and analysis tools. Digital Twin-based predictive modelling and fast design space evaluation are studied that work toward addressing these challenges.**

***Keywords*—Real-Time Systems, Digital Twin, Design Space Exploration, Predictive Cache Model, Multi-core Scheduling, Test Coverage, Validation and Verification.**

I. Introduction

In traditional software and hardware co-design of real-time systems, understanding the performance of software cannot be fully achieved until the hardware is available and even then a good understanding may only be available when all the software is ready. For example, the cache behaviour is dependent on the processing device, the software and the operational context including the data being processed. Problems may only be realised later in the development cycle when the hardware cannot be changed and any software optimisation takes time and is expensive.

In this work, we motivate our use of predictive models based on currently available systems to better assess how different hardware configurations may result in a better architecture, and how the system may cope with future changes and operational usage scenarios. We note that *better* and *future* may not be well defined at the time the assessment is performed which means that any model and Design Space Exploration (DSE) needs to be *robust* to uncertainties (epistemic and aleatoric) and further system changes (design and operational use).

For example in DSE, making predictions of the performance of a defined configuration provides usefulness only if the underlying model, the data being used to generated that model, and the data used to stimulate that model are all *representative* (see its definition below). These bring difficulties not only due to the increased complexity of software, hardware and the interactions between them, but also the context and the mode that the system is experiencing.

As part of the *Modelling and Optimising Complex Heterogeneous Architecture* (MOCHA) research project, we propose a Digital Twin-based simulation and DSE toolchain (namely *MOCHA-T*; See Figure 1), which aims to design and optimise high performance heterogeneous many-core real-time systems and provide the evidence needed as part of assurance. The research features the following components: a *profiler* allowing the current software to be executed on either a host or target; a *learning module* that takes the profile data and predicts what the performance characteristics will be in different configurations and situations; an *executable model* that evaluates a specific configuration in a particular operational context; a *design space searching and assurance* that can propose and evaluate different design options across a wide range of operational context; and finally a *Digital Twin* that co-exists with the deployed system to help understand if the system is performing as expected, whether it needs optimising and whether the executable models need fine tuning.

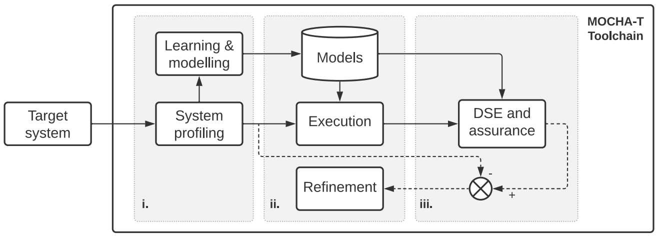

Fig. 1. The MOCHA-T Toolchain for real-time systems design and analysis. The sub-systems in the diagram are: (i) Profiling and Learning; (ii) the Digital Twin; (iii) DSE and assurance.

Success relies on components using models at the right level of abstraction and being able to capture the dependencies of functional software components, for example, by modelling parallel tasks as Directed Acyclic Graphs (DAGs) [1]. The models would then enable the evaluation of feasibility and schedulability.

The structure of the paper is as follows: the open challenges and opportunities for real-time digital twins are discussed in Section II. Based on the proposed questions, the potential solution with respect to modelling is further discussed in Section III, followed by a conclusion in Section IV. The main contribution of this paper is the definition of some open challenges and opportunities for future research.

II. Open Challenges for Real-Time Digital Twins

In this section some of the challenges faced in this work from the perspective of assurance are introduced. The challenges are heavily based on previous work on the certification of critical real-time systems [2], assurance of wider dependable systems [3], and a previous open challenges paper written in the context of probabilistic Worst-Case Execution Time (pWCET) analysis [4]. The challenges are considered across some of the components and properties outlined in the introduction. For reasons of space, a full consideration cannot be presented. The challenges (Cx) then lead to specific research questions ($Rx.y$).

A. C1 - Determining the Key Parameters for the Range of Operational Usage

Most systems have a wide operational range, especially considering abnormal situations such as overload and failure scenarios. It is crucially important that a system operates as expected across the whole range and importantly that any degradation is graceful [5]. At the same time a system has many layers both horizontally (e.g. a number of subsystems and components) and vertically (e.g. application, run-time support including the operating system, and then the hardware). Across the range of components involved in our research relies on appropriate *key* parameters across the whole operational range and the different parts of the system. For example when considering the profiler as part of modelling the multi-core timing effects, there is a need to understand which variables in which (hardware and software) components lead to sufficient operational scenarios and the different behaviours of the multi-core processors. This leads to the following research questions:

- *R1.1* - How to balance sufficient coverage of operational usage with avoiding pessimism by considering implausible scenarios?
- *R1.2* - How to identify the key parameters across the layers, range of components and operational scenarios?

In the context of multi-mode system, e.g. mixed-criticality or fault adaptive system, each mode and the transition between modes need to be examined. In the work of [6], the multi-core interference is examined using deep learning based on data from performance measurement counters (PMCs). In [7], scenario-based analysis combined with heuristic search are applied to study the changeability of a system.

The profiler, when generating test vectors, should be able to simulate the running environment by using models that can represent system states and being able to reproduce failures, for example by using fault injection. The profiler should also be able to choose the right inputs that can reflect the operational range while reducing the amount of data that is generated and collected.

B. C2 - Achieving Sufficient Coverage in an Efficient Manner

Given the set of key parameters to be manipulated as part of testing, it is important to understand what is meant by sufficient testing. In the area of real-time systems, the only work that considers coverage is [8] which targeted at the Worst-Case Execution Times (WCET) for avionics systems. In [8] coverage metrics were proposed, search-based testing approaches employed to efficiently meet associated coverage targets, and it was demonstrated the approach reliably outperformed the previous state of the art techniques. The work in [8] was only applicable to a small part of the overall *MOCHA-T* system and the type of system was much more constrained and predictable. Here are the associated research questions:

- *R2.1* - How to establish coverage metrics for Digital Twins of Complex Systems?
- *R2.2* - How to efficiently process the big data associated with the Digital Twin?
- *R2.3* - How to reduce the test cases needed?

A common practice in coverage is to use extensive testing. The test coverage can be achieved by test automation. An example is given in [8] which uses simulated annealing (SA) to create test vectors that are applied to software under test (SUT) using data from a Rolls-Royce control system. Another example is given in [9] that uses coverage techniques to analysis SUT. The coverage test should reproduce contentions on shared resource and the interference that is caused by this.

For emerging systems that have increased internal and external interactions, the coverage should not be limited to the more-traditional software coverage, but also to a wider scope including the system context that the software program is to be executed within. An example from 5G base stations, the transmission workload is based on the number of cells and users that are sharing the cell simultaneously. Another example is in space systems where the system is exposed to extreme environment such as radiation and high temperature.

Any test vector generator should be able to cover the operational scenarios defined in *C1* as well as the traditional software coverage metrics, e.g. the branch and local path coverage metrics proposed in [8].

C. C3 - Creating Representative Models Supporting Reliability Assessment

A challenge with search-based techniques targeting coverage is representativity. As part of assurance, an overarching aim is to determine a realistic reliability target, however a technique such as [8] has inherent bias. For example, rarely executed scenarios will have been executed more often than should happen in practice. A balance is therefore sought between sufficient test cases especially for rarely occurring situations which are fundamentally important, e.g. to understand how a system gracefully degrades, and achieving a realistic model and determining an accurate reliability estimate. Related research questions are listed below:

- *R3.1* - How to choose the right abstractions to profile, model and analyse systems?
- *R3.2* - How to ensure the models are representative for the wide range of operational scenarios?
- *R3.3* - How to estimate the reliability of the system from the Digital Twin?

Probabilistic modelling gives a full spectrum of probability of reliability. One of the pioneering work in this area is to use Extreme Value Theory (EVT) with pWCET to overcome the limitation of static WCET [10], [11]. The pWCET can produce probability distribution of execution times and EVT is then applied to find extreme values. The estimations can then be translated into response times by using measurement-based probabilistic timing analysis [11] to evaluate the schedulability.

Machine learning is also used in the area of exploring execution times influenced by computer architectures [12]. The authors in [6] use machine learning to explore inter-core cache interference, known as Forecast-Based Interference (FBI) analysis. It is also shown that machine learning can be used for dimensionality reduction, for example, using Principal Component Analysis (PCA) to identify the main inputs that influence the desired behaviour(s). In [13], a comparison study is made to evaluate the performance of WCET prediction with various of machine learning methods. It is claimed in the work that non-parametric methods perform better than parametric method, and non-linearity should also be considered.

Our work will recognise that even though the modelling of time is the core part of DSE, there are other considerations. Two examples are: (1) The design space can only be partially sampled so the prediction has to be made based on interpolation. This requires the model to be generalised and not be over-fitted to a certain range of operational conditions; and (2) As hardware and software are developed simultaneously, i.e. the hardware may not be ready when the software is being developed, this brings new challenge of how to make prediction on the host machine with little or no data that can be collected from the target. A solution could be to stress the search to enhance robustness [5].

D. C4 - Managing Uncertainties as Part of Establishing Confidence

It is inevitable that uncertainties would occur, e.g. from a deficiency in modelling or a systematic bias. To establish confidence, it is vital that the uncertainties either from modelling or from simulation should be handled explicitly. While some of the uncertainties can be eliminated by simply running the experiment multiple times, the others need a more elaborated approach to deal with. The current practice towards verification of multi-core systems has suggested that static analysis would not be sufficient as the contention is too complicated to be accurately modelled. As an alternative, measurement-based worst-case execution times and pWCET (probabilistic WCET) are prevailing as: (1) they explicitly consider uncertainties as probabilities in the modelling; (2) they explicitly include the uncertainties by capturing from the real system what would otherwise not be considered from static modelling. The following research questions are identified:

- *R4.1* - How to assess the uncertainties associated with the Digital Twin?
- *R4.2* - How to assess the confidence associated with the Digital Twin?
- *R4.3* - How to refine the test cases and models to give appropriate confidence?

The uncertainties need to be categorised according to Johari's window [14] into known-unknown and unknown-unknown. For known-unknowns, i.e. uncertainties we know we do not know, the uncertainties can be modelled and considered. However, for unknown-unknowns, i.e. uncertainties that we do not know we do not know, we need to make sure their presence will not jeopardise the system.

The uncertainties could be justified through empirical evaluations comparing the model output with the prediction using a Digital Twin approach. Through statistical testing of difference evaluation, a mismatch would drive the system, for example, to generate more test cases around the region or increase the level of abstraction in the region where low accuracy is presented.

E. C5 - Robust Decision Making in the Presence of Inaccuracies

With the DSE system in place, it is doubtful that whether the evidence could provide sufficient confidence for decision making. This introduces the argument of differentiate what is 'belief' and what is 'reality'. When the decision is made based on the belief of the model, it is possible that the decision can have unexpected results even with a strong belief. For example, if an underlying assumption on modelling is violated when the system executes; or the system is beyond the desired operational boundary.

For safety-critical systems, sufficient evidence is required to support an argument of safety [2]. A common practice is then constructing a safety argument (safety case) using *GSN* (Goal Structured Notation) [15] or *SACM* (Structured Assurance Case Metamodel) [16] to analysis the safety objectives, safety goals/sub-goals and what evidence should be provided.

In general, the presence of inaccuracies suggests that all the processes leading to decision making, for example timing analysis, cannot be fully automated and engineers need to be included in the loop to provide insightful interpretation of the result. On the other hand, the tool should be able to collect evidence and be able to provide confidence in the evidence to support better decision making. Based on the discussion, we list the research questions as below:

- *R5.1* - How to make robust decisions given the inaccuracies and confidences in the models?
- *R5.2* - How to explore the design space in a scalable fashion?
- *R5.3* - How to present a convincing assurance argument?

These questions are more difficult to answer than the others. First, we understand the complication in the system would sometimes make it intractable to produce a fully accurate twin as a duplicate. It is also understood that a high confidence in the modelling does not (and should not) lead to a high confidence in decision making, as the former is often dependant on assumptions that are not always true. It is thus considered by the authors how to reduce this problem by introducing feedback based on the difference between the collected data trace from the real system and the model output (following *C4*), as earlier explored in [17].

As emerging systems (e.g. autonomous driving) and new architectures (e.g. many-core and heterogeneous systems) occur, the challenges mentioned in this section are becoming ever more significant and cannot be ignored in the design process of tool implementation. However, it is notable the challenges introduced in this section are *far from completed*. This work is to provide insights of the position of where the current practice is as well as to encourage contributions to be made in related real-time systems research.

III. Considerations of Modelling in Digital Twin

To address the uncertainty and representativity issues, we propose *Predictive Analysis of Cache Models with AbstractioN* (PACMAN) as part of this work. PACMAN uses an executable model, with intra- and inter-task (including inter-core interference) modelling. We use block-level abstraction and cache correlation model in the modelling level. In the scheduling layer, we introduce the probabilistic execution of these cache models to obtain the distribution of response times.

To enable a Digital Twin with fast and indicative feedback, the target system is abstracted to a higher level and focus on the high-level system behaviours of interest. By doing so, we hide the irrelevant implementation details so that it effectively highlights the high-level system behaviours (e.g., cache misses of a function) of interest. In addition, although working at a low abstraction level would reveal more details of the system, it is not always true that this can lead to a higher modelling accuracy. This is due to the possibility of introducing irrelevant data into the training, which can cause significant noise and increases the search space, leading to reductions in model accuracy given limited searching time. Therefore, a lower level of analysis should only be performed when necessary, e.g. for a small and important part of the system in which the current abstraction level is insufficient for a full understanding.

The objective of cache modelling is to fit the function $\hat{y} = f(X)$, where $\hat{y}$ is the predicted cache miss rate and $X = \{x_1, x_2, ..., x_n\}$ is a set of system state variables. The function $f(\cdot)$ can be represented by statistical models including *Linear Regression* model or Neural Networks that can allow for temporal dependencies, e.g. *Long Short-Term Memory*. When determining X, *Principal Component Analysis* is used to identify the major factors that influence cache within/between cores.

Inevitability, the process of modelling and prediction is imprecise. Our way of evaluating this is to compare the prediction against the actual output under the same conditions. The modelling precision can be quantified by the mean prediction error over multiple trials. Through feedback, the precision can be improved by, for example, adjusting the block size (i.e. granularity), or collecting more evidence (i.e. data). It is notable that functional and non-functional properties have different requirements on precision and should be treated differently. Also in case of a conflict, resolution should consider the dependability requirements of the system.

IV. Conclusions

In this work, we explore the assurance challenges and state-of-the-arts of designing dependable multi-core real-time systems. Based on the understanding of the requirements of representativity, coverage and confidence, we developed a Digital Twin-based method targeted for multi-/many-core real-time system design and analysis. We discuss the usability and requirements of profiling, modelling and feedback. The method benefits the design exploration, modelling and timing assurance of high-reliable multi-core computing systems. Future work includes formulation of an assurance argument to address these challenges in the domain of avionics, automotive and aerospace.

References

[1] S. Zhao, X. Dai, I. Bate, A. Burns, and W. Chang, "DAG scheduling and analysis on multiprocessor systems: Exploitation of parallelism and dependency," in *Real-Time Systems Symposium*. IEEE, 2020.

[2] P. Graydon and I. Bate, "Realistic safety cases for the timing of systems," *The Computer Journal*, vol. 57, no. 5, pp. 759–774, 2014.

[3] M. L. Fairbairn, I. Bate, and J. A. Stankovic, "Improving the dependability of sensornets," in *International Conference on Distributed Computing in Sensor Systems*. IEEE, 2013, pp. 274–282.

[4] S. J. Gil, I. Bate, G. Lima, L. Santinelli, A. Gogonel, and L. Cucu-Grosjean, "Open challenges for probabilistic measurement-based worst-case execution time," *Embedded Systems Letters*, vol. 9, no. 3, pp. 69–72, 2017.

[5] P. Emberson and I. Bate, "Extending a task allocation algorithm for graceful degradation of real-time distributed embedded systems," in *Real-Time Systems Symposium*. IEEE, 2008, pp. 270–279.

[6] D. Griffin, B. Lesage, I. Bate, F. Soboczenski, and R. I. Davis, "Forecast-based interference: Modelling multicore interference from observable factors," in *International Conference on Real-Time Networks and Systems*, 2017, pp. 198–207.

[7] I. Bate and P. Emberson, "Incorporating scenarios and heuristics to improve flexibility in real-time embedded systems," in *Real-Time and Embedded Technology and Applications Symposium*. IEEE, 2006, pp. 221–230.

[8] S. Law and I. Bate, "Achieving appropriate test coverage for reliable measurement-based timing analysis," in *Euromicro Conference on Real-Time Systems*. IEEE, 2016, pp. 189–199.

[9] B. Lesage, S. Law, and I. Bate, "TACO: An industrial case study of test automation for coverage," in *International Conference on Real-Time Networks and Systems*, 2018, pp. 114–124.

[10] D. Griffin, B. Lesage, A. Burns, and R. I. Davis, "Static probabilistic timing analysis of random replacement caches using lossy compression," in *Proceedings of the 22nd International Conference on Real-Time Networks and Systems*, 2014, pp. 289–298.

[11] L. Cucu-Grosjean, L. Santinelli, M. Houston, C. Lo, T. Vardanega, L. Kosmidis, J. Abella, E. Mezzetti, E. Quiñones, and F. J. Cazorla, "Measurement-based probabilistic timing analysis for multi-path programs," in *Euromicro Conference on Real-Time Systems*. IEEE, 2012, pp. 91–101.

[12] D. D. Penney and L. Chen, "A survey of machine learning applied to computer architecture design," *arXiv preprint arXiv:1909.12373*, 2019.

[13] X. Dai and A. Burns, "Predicting worst-case execution time trends in long-lived real-time systems," in *Ada-Europe International Conference on Reliable Software Technologies*. Springer, 2017, pp. 87–101.

[14] J. Luft and H. Ingham, "The Johari window: a graphic model of awareness in interpersonal relations," *Human relations training news*, vol. 5, no. 9, pp. 6–7, 1961.

[15] T. P. Kelly, "Arguing safety: a systematic approach to managing safety cases," Ph.D. dissertation, University of York York, UK, 1999.

[16] R. Wei, T. P. Kelly, X. Dai, S. Zhao, and R. Hawkins, "Model based system assurance using the structured assurance case metamodel," *Journal of Systems and Software*, vol. 154, pp. 211–233, 2019.

[17] X. Dai and A. Burns, "Period adaptation of real-time control tasks with fixed-priority scheduling in cyber-physical systems," *Journal of Systems Architecture*, vol. 103, p. 101691, 2020.