Digital Twin for Modelling and Optimizing Industrial Multi- and Many-Core Systems

Xiaotian Dai, Shuai Zhao, Iain Bate, Alan Burns
xiaotian.dai@york.ac.uk

Real-Time Systems Research Group
University of York

Nov 2021

Contents of this talk

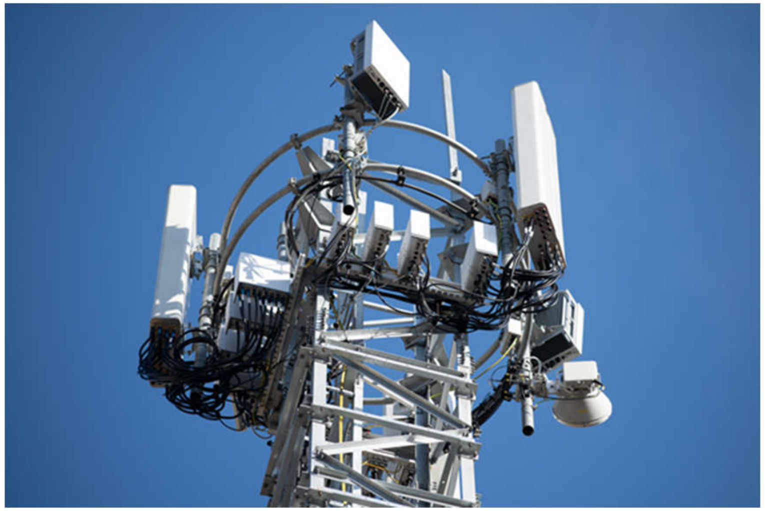
- Background of MOCHA project
- Challenges in many-core real-time systems
- Digital twin to achieve adaptation in E/RTS
- Current research and key problems to solve
- Research plan

MOCHA

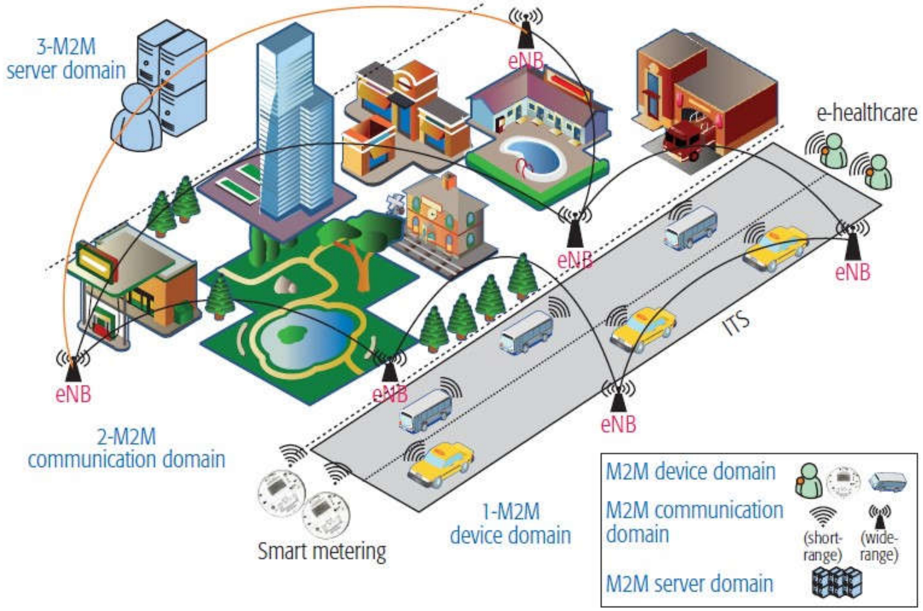
Modelling and Optimizing Complex Heterogenous Architectures

- Collaboration between UoY and Huawei
- Working on 5G base stations.
- Funding size: £985,927
- Dec 2019 – Dec 2022

https://www.cs.york.ac.uk/rts/mocha/

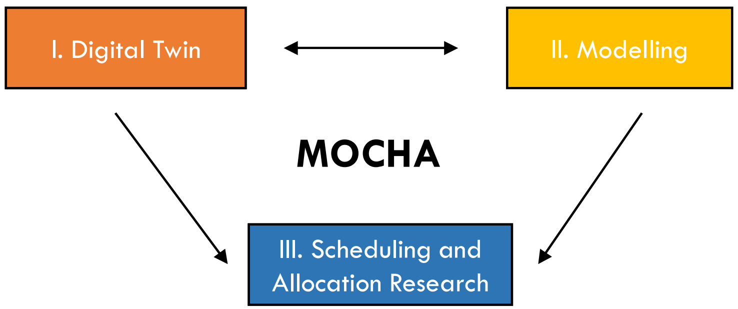


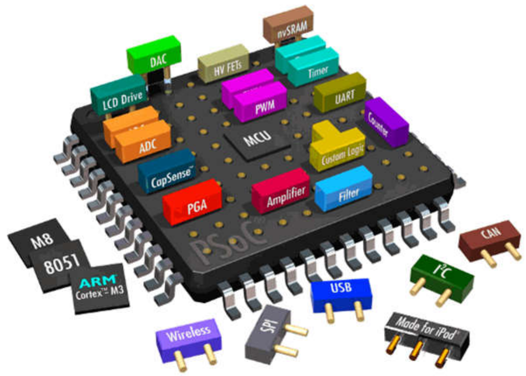
Background - 5G Communication



Background - Research Goals

- Working on the next-generation of 5G base stations to meet the increasing computing demands and timing requirements.
- General methods and tools that can be applied to similar systems.
- Towards trustworthy many-core systems (# of cores is more than 16).



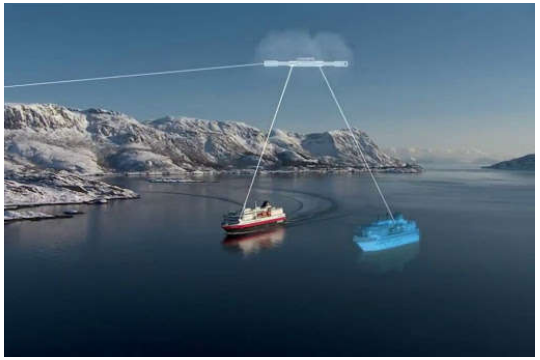


Challenges in many-core real-time systems

- Systems are designed and verified with models, but these models could be inaccurate/ invalidated.
  - The number of cores keeps increasing for industrial embedded devices or “edge” computing, with flexible yet complex architecture – parallelism and interference need to be further explored.
  - Working environment is subjected to more uncertainties, which is hard to be predicted at design-time. However, design decisions have to be made before hardware is being made available.
- Demand on enhancing predictability, adaptiveness and performance (e.g. by better use of cache.).
- Assuring that the system has meet the timing requirements. For example, URLLC (ultra-reliable low latency communications) has a deadline requirement of 1 *ms*.

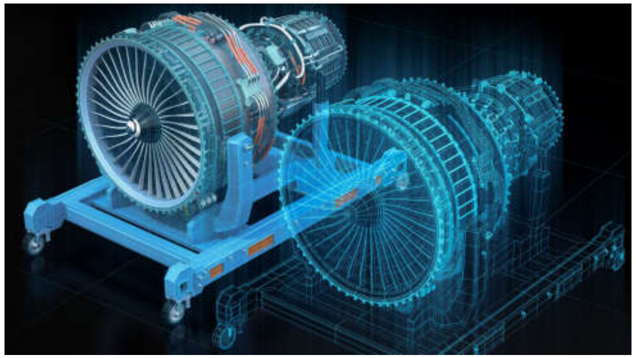
Digital Twin (DT) in Industrial Systems

- "A digital twin is a *virtual representation* that serves as the *real-time* digital counterpart of a physical object or process."
- Attempts have been made in space and avionics, transportation, industrial automation, medical systems and autonomous driving.
- DT is an ideal way of verifying a system before it is deployed, and through its life-cycle (more than a simulator).



Digital Twin for Maritime Transportation

Digital Twin for Process Control



Digital Twin for Avionic Engine Design and Control

DT for Embedded/Real-Time System

However, DT is not considered as much in E/RT systems.

In our vision, DT can be helpful from two perspectives:

1. **Design-time**: fast evaluation and exploration with approximated models;
2. **Run-time**: refinement through observations from the real system.

When applying the DT, we have these principles in mind:

- **Efficiency** --- a satisfactory model without full coverage profiling.
- **Acceptability** --- following engineering practice and reduce efforts.

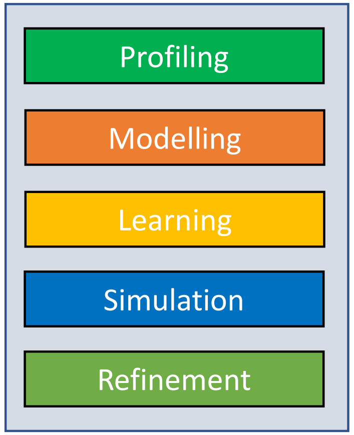
DT for Embedded/Real-Time System

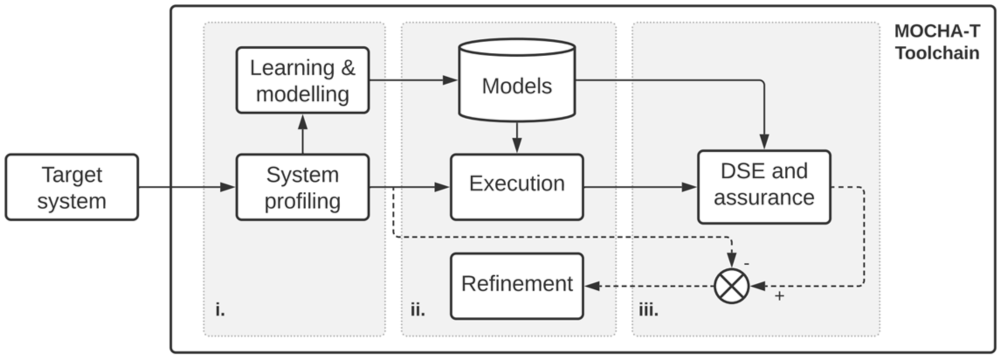
- In our previous RTAS'21 paper, we introduced five key challenges to adopt Digital Twin in real-time multi-core systems:
  - C1 - Determining the Key Parameters for the Range of Operational Usage
  - C2 - Achieving Sufficient Coverage in an Efficient Manner
  - C3 - Creating Representative Models Supporting Reliability Assessment
  - C4 - Managing Uncertainties as Part of Establishing Confidence
  - C5 - Robust Decision Making in the Presence of Inaccuracies
- The success of a Digital Twin largely depends on the solutions to these challenges.

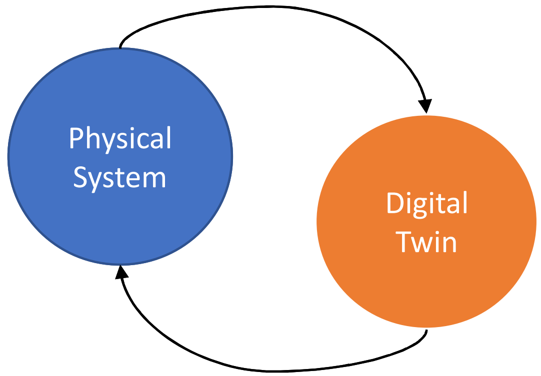
[1] X. Dai, et al. "Brief Industry Paper: Digital Twin for Dependable Multi-Core Real-Time Systems---Requirements and Open Challenges". RTAS. 2021.

Current Research and Key Problems

- Aiming at using Digital Twin for modelling high performance/ reliability multi-core embedded systems, as a mitigation solution to modelling errors, and to achieve adaptation of scheduling.



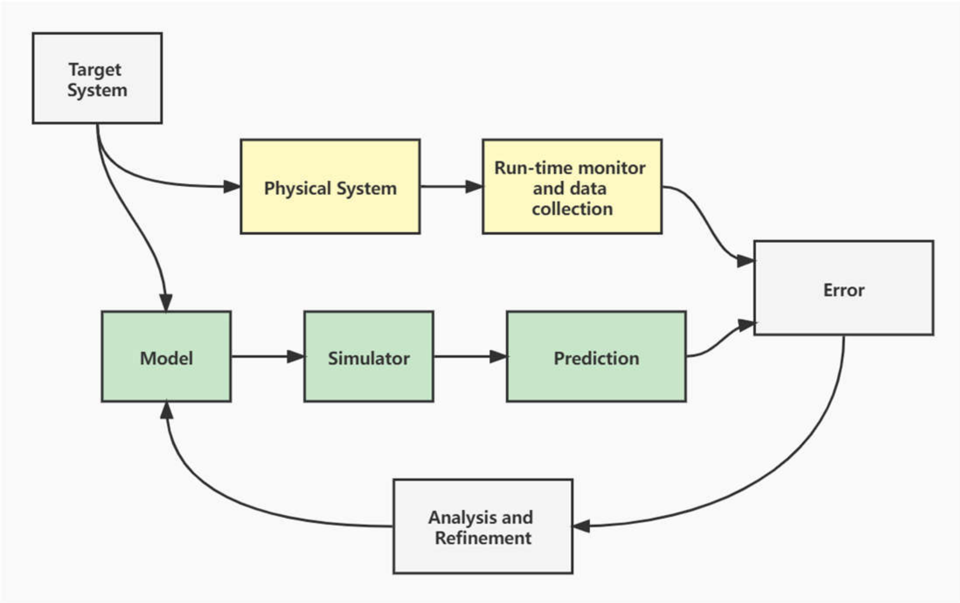




Current Research and Key Problems

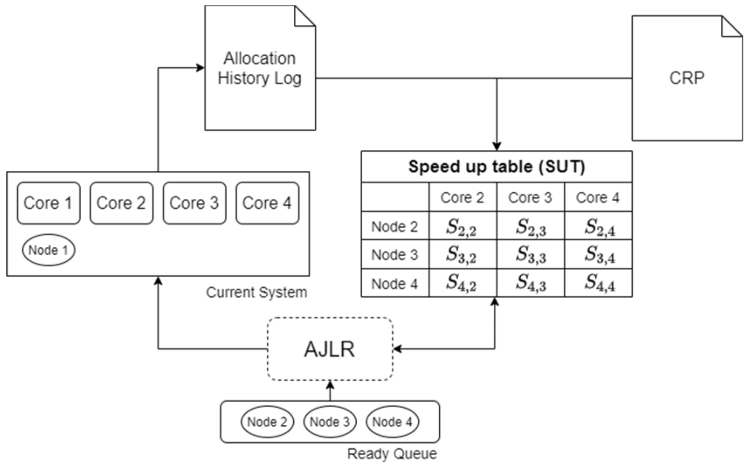
Execution Time Model (ETM)

- The system initially has a relatively simple model, i.e., without considering multi-core and OS interference.
- Once the system is up and running, operational data can be collected. The comparison of the real data and the output from the model will produce some error.
  - The output from the model is produced by some prediction-based method, e.g., from a simulator in our case
- The error would be collected, analyzed and modelled, by an run-time learning process.
- The error model can then be used to improve the digital twin.



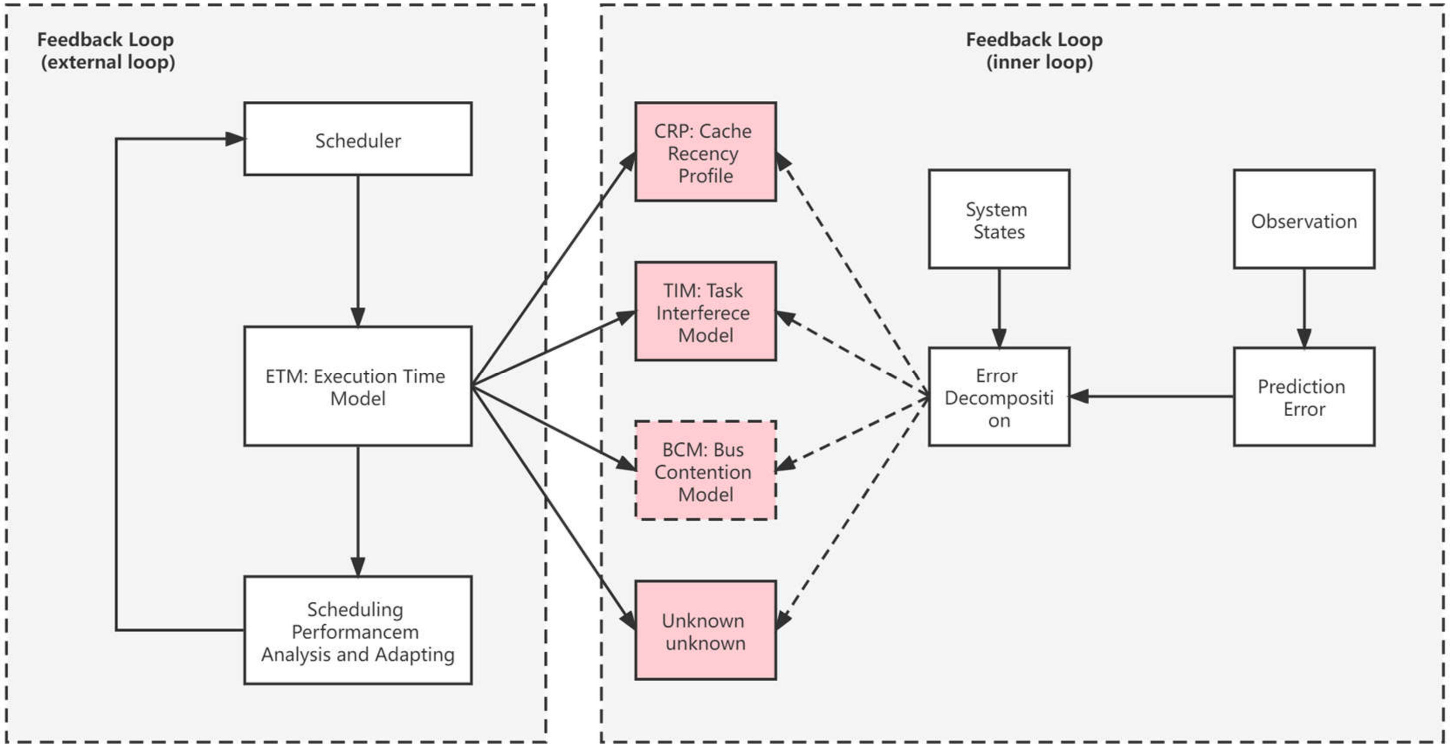
Current Research and Key Problems

- Decide the error is anomaly or not (C.1/C.2).
- Decide from which source the error is coming from (C.3).
- Evaluate the impact of an error (C.4).
- Partial information / execution scenarios / confidence / level of tolerance (C.5).
- Engineers would be in the loop when *critical* decisions have to be made to mitigate, e.g. changing level of abstraction (C.5).



Current Research and Key Problems

- Explanation-based learning: Error decomposition and reasoning.



Research Plan

- Short-term plan
  - Make a workable prototype digital twin that can adapt errors, and improve the system model by changing, for example, the parameters of the underlying models.
  - Use the DT to improve scheduling and allocation decisions.
- Long-term plan
  - How we can adapt this method, not just for modelling and scheduling, but also for other purposes to answer “what if” questions. For example, design space exploration (DSE).

Questions and Discussions

Thank you!

Dr. Xiaotian Dai (xiaotian.dai@york.ac.uk)