Modelling and Optimizing Complex Heterogeneous Architecture

Iain Bate
Real-Time Systems Group, University of York

Outline

- Overview of the MOCHA project
- Modelling of the system – a digital-twining approach
- Key research questions we plan to address

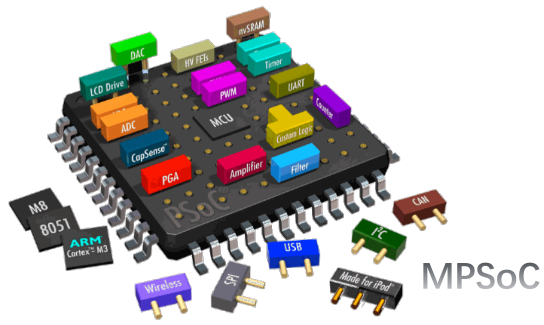
MOCHA – Background

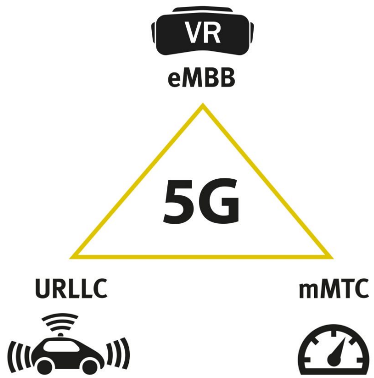
- The applications, resources and architectures of real-time embedded systems are becoming ever more complex to understand, control and maintain.

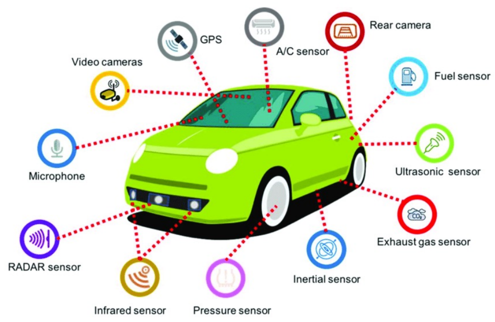
- Traditional (static) methods are difficult to apply in heterogeneous MPSoC as:
  - Increased complexity in software design.
  - Difficulties in modelling, scheduling and analyzing the system.

- Systems are hard to optimise
  - To understand how the software executes on the platform.
  - To design the MPSoC and then schedule / allocate the software.

- This has led the research into:
  - ➢ building high-level models of systems.
  - ➢ developing dynamic policies based on statistical models.



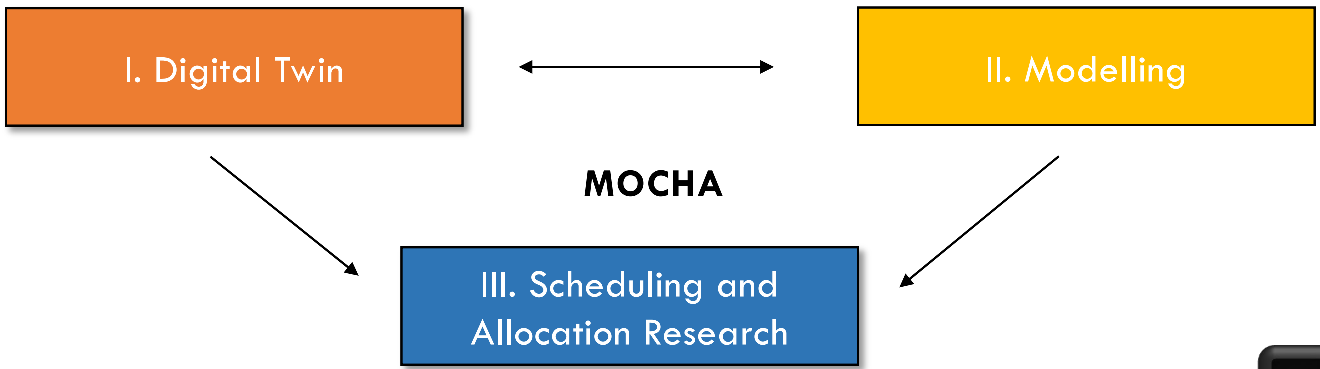




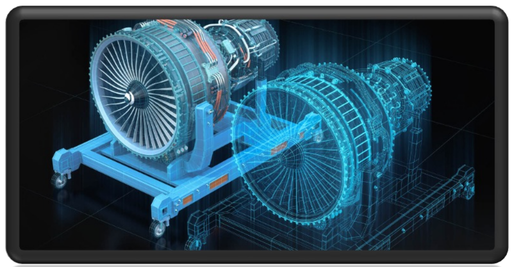
System Modelling - a digital-twinning approach

- Realistic and configurable simulation.
- Supports varies of hardware/scheduler configurations.
- Supports large-scale evaluation and design space exploration.

- Profiling and data analysis for real systems.
- Allows the digital twin to be validated and refined.
- Delivers appropriate understanding of the systems so large-scale evaluations are representative



- Reduced number of cache misses on the platform
- Reduced and more predictable latencies
- Improved system throughput

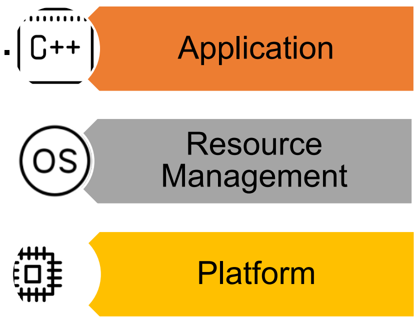


(*) Picture from: https://job-wizards.com/en/digital-twins-doubling-the-potential-for-innovation

MOCHA – Research vision

Our vision is to develop a set of statistical models and adaptive methods that can be applied to all layers (from application to platform) of a MPSoC system.

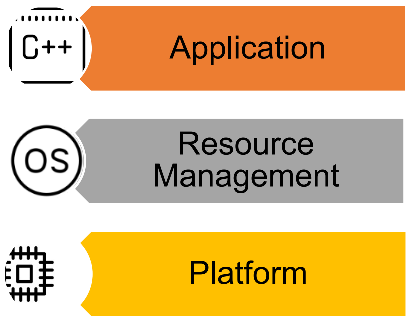
1. *Application* - A high-validity **digital twin** for the timing aspects of a complex MPSoC systems built upon
   - Systematically profile the real system based on the significant factors.
   - Establish valid statistical models of the system.
   - A high-speed and configurable simulator for generic heterogeneous architectures.



MOCHA – Research vision

Our vision is to develop a set of statistical models and adaptive methods that can be applied to all layers (from application to platform) of a MPSoC system.

2. *Resource management* - Better control of systems execution and improved performance
   - Feedback-based task scheduling and allocation methods.
   - Configurable system policies, including the management of hardware resources, e.g., cache locking and scratch bad allocation.
   - Controlling back pressure of work coming into the main processors.



3. *Platform* – Later use the models for design space exploration of the platform itself

MOCHA – Challenges

- Building realistic execution models with partial/limited information.
  - The scale of the studied system can be very large (millions lines of codes) with numerous execution scenarios.
  - Some information of the system from industrial partners is strictly confidential due to protection of intellectual properties and thus cannot be fully accessed.
- Refining the predicted models and the techniques to reflect all execution scenarios.
  - The predicted models are constructed based on limited information and cannot reflect the general case without further refinement.
- Producing error tolerant systems that can deal with inevitable differences between the models and the real systems
  - The predicted models may still behave differently from the actual system in corner cases and special situations.

MOCHA – Challenges

- Building realistic execution models with limited information.
  - ✓ Choosing an appropriate abstraction level.
  - ✓ Identifying the most influential factors from the observations.

- Refining the predicted models and the techniques to reflect all execution scenarios.
  - ✓ Identifying the key differences in the influential factors under different execution scenarios.

- Producing *error* tolerant systems that can deal with inevitable differences between the model and the real system
  - ✓ Understanding the trends in behaviour.
  - ✓ Identifying problematic/anomaly cases that might occur.
  - ✓ Determining the mitigation strategies to avoid faulty cases.

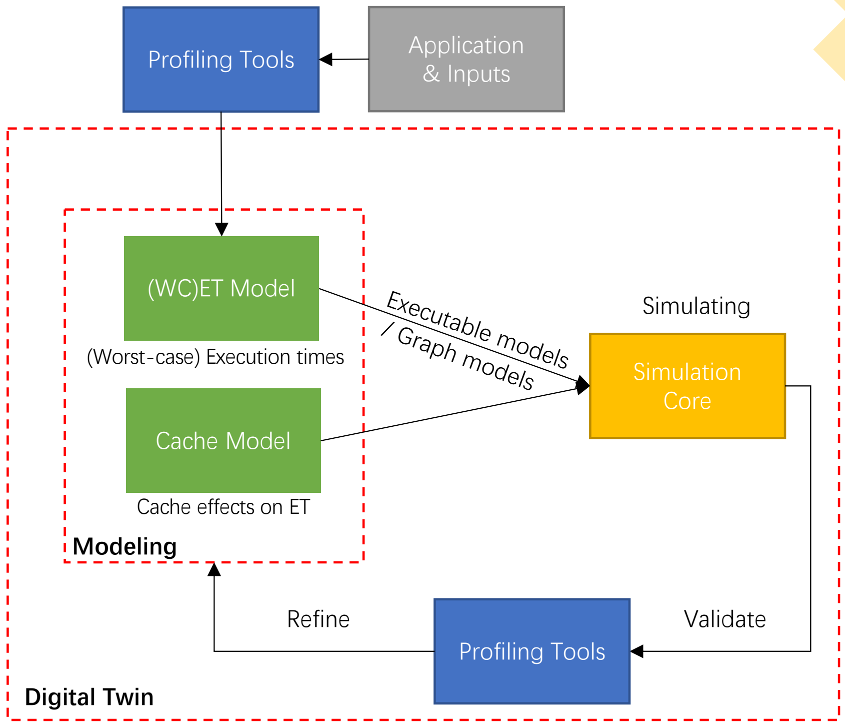
System Modelling - a digital-twinning approach

1. Simulate and model actual application using profiling tools, and then create the digital twin.

2. Ensure the digital twin is valid with changing input workloads.
   - (Worst-Case) Execution Time models.
   - Cache (and memory) models.

3. Extend the digital twin to support multi-core scheduling with shared cache and memory.
   - Upgraded models with inter-thread/core effects.

4. Validate the digital twin against a real target.
   - Reproduce the scheduling and timing behavior.



Open Research Questions

- Scheduling and allocation of software task to caches / scratchpads and cores again well established.

- Design space exploration as part of MPSoC design are also well understood
  - *Robustness* – How to assess the effect of inaccuracies and uncertain operational contexts on the models?
  - *Sensitivities* – How to understand the sensitivities so the design can be made robust and the impact of errors understood?
  - *Predictability* – How to assess the predictability of systems?
  - *Scalability* – How to perform design space exploration for large numbers of configurations and scenarios?

Open Research Questions

- Digital Twins (DT) is a well-established practice but what are the challenges around timing.
  - *Acceptability* – What information can we realistically be expected to extract from a real system?
  - *Accuracy* - What does it mean for a simulator to be accurate?
    - This very much depends on the questions to be answered with DT.
  - *Efficiency* – What is the right level of abstraction for the model and the right type of feedback?
  - Success – Is a successful solution one where the model has similar accuracy with less data (types and quantity) at a higher level of abstraction?

- Note
  - Some colleagues would not define the presented work as a DT as the changes to resource management are offline.
  - Do we have a shared definition of what a DT is?


Thank you for your attention!
The MOCHA Research Group