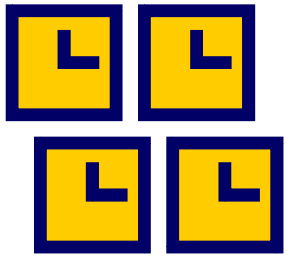


P-SOCRATES

Parallel Software Framework for Time-Critical many-core Systems

Time-Critical Parallelised Execution

Luís Miguel Pinho

(CISTER Research Centre, ISEP, Portugal)

CISTER - Research Center in Real-Time & Embedded Computing Systems

Barcelona Supercomputing Center

Centro Nacional de Supercomputación

UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

EVIDENCE EMBEDDING TECHNOLOGY

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement nº 611016

Project in a glance

- 3-year FP7 STREP project (Oct-13, Oct-16)

CISTER - Research Center in Real-Time & Embedded Computing Systems

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

ETH zürich

EVIDENCE
EMBEDDING TECHNOLOGY

AT

Atos

Industrial Advisory Board

RAPITA SYSTEMS LTD

AIRBUS DEFENCE & SPACE

Honeywell

AIRBUS

IBM

EXPERT SYSTEM
SEMANTIC INTELLIGENCE

KALRAY

SAAB

City of Bratislava

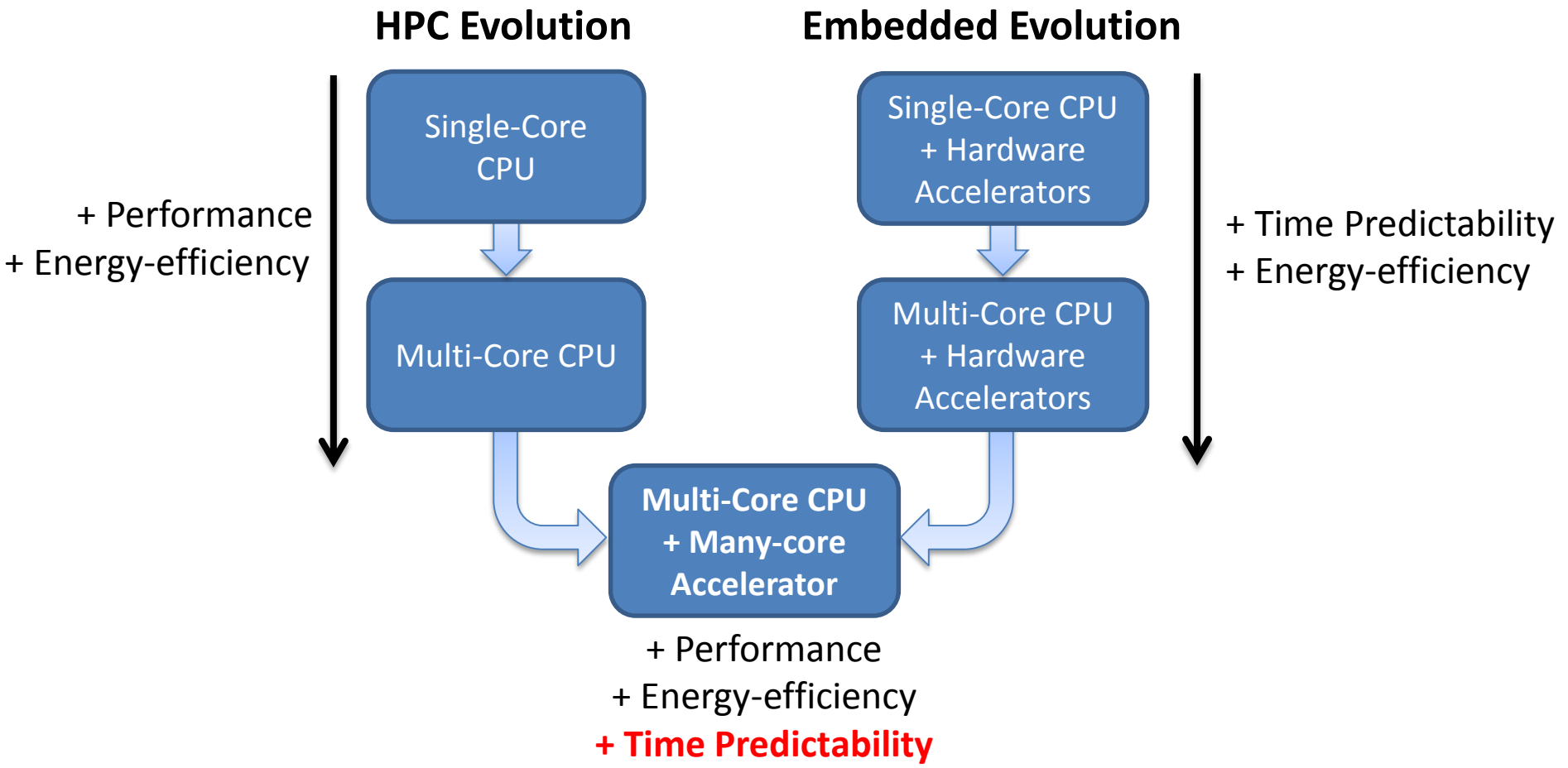
BOSCH

MBDA

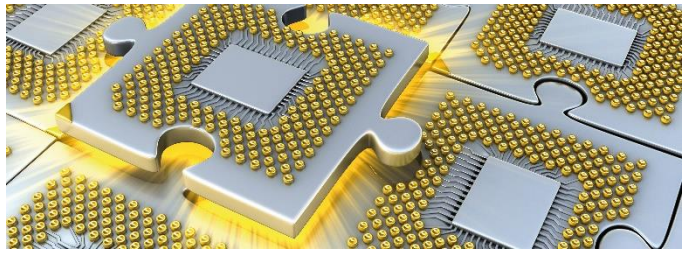
Motivation

- New systems require a combination of time predictability and high performance

Motivation


HPC Evolution
Embedded Evolution
+ Performance
+ Energy-efficiency
Single-Core CPU
Multi-Core CPU
Single-Core CPU + Hardware Accelerators
Multi-Core CPU + Hardware Accelerators
+ Time Predictability
+ Energy-efficiency
Multi-Core CPU + Many-core Accelerator
+ Performance
+ Energy-efficiency
+ Time Predictability

Vision



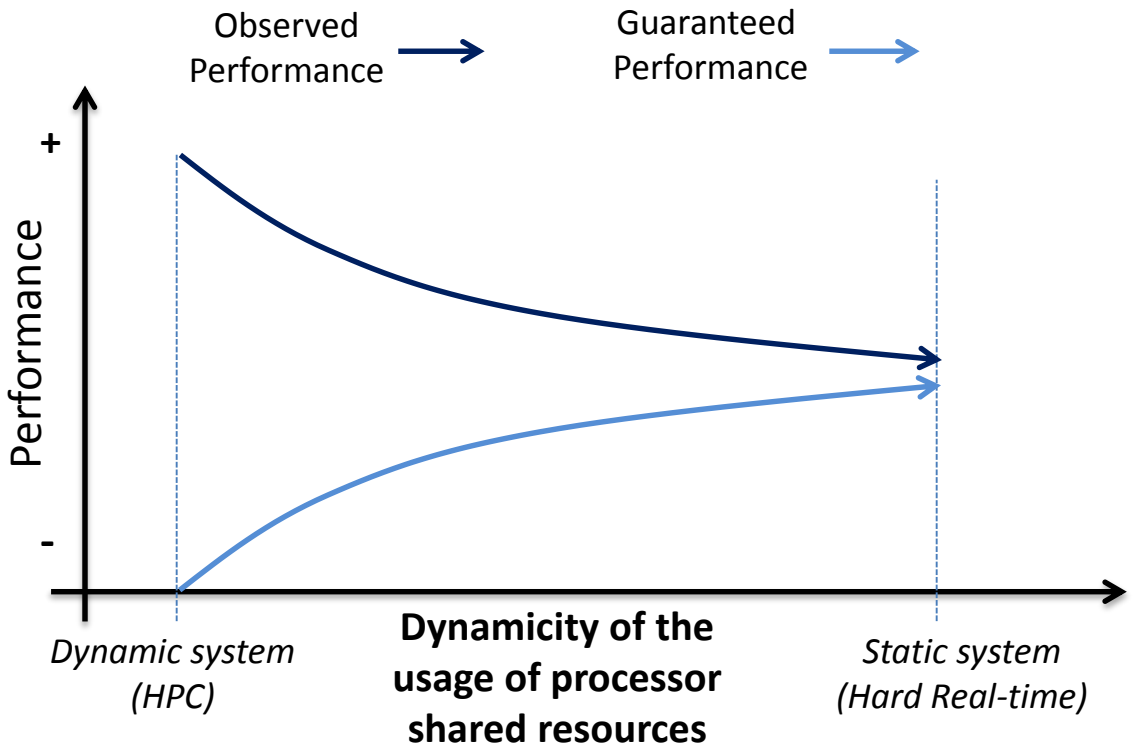
next-generation embedded **many-core accelerators**

real-time methodologies to provide **time predictability**

programmability of many-core accelerators from HPC

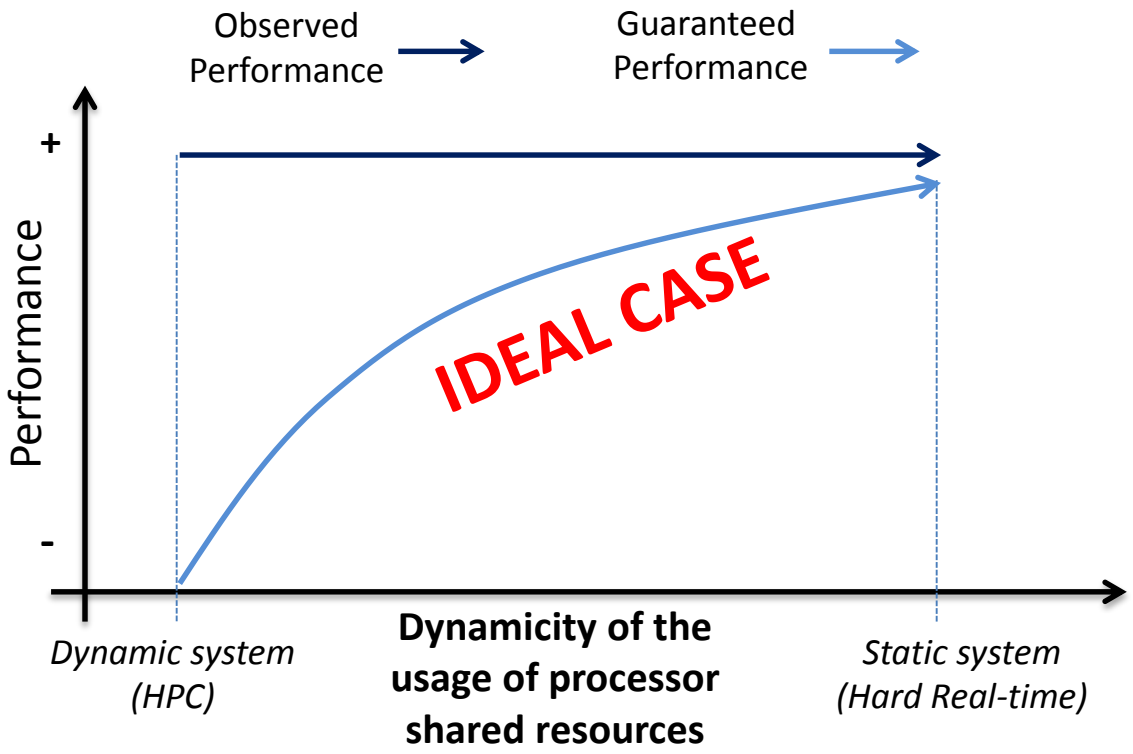
Vision

- Minimizing performance lost, maximizing guaranteed performance



Vision

- Minimizing performance lost, maximizing guaranteed performance

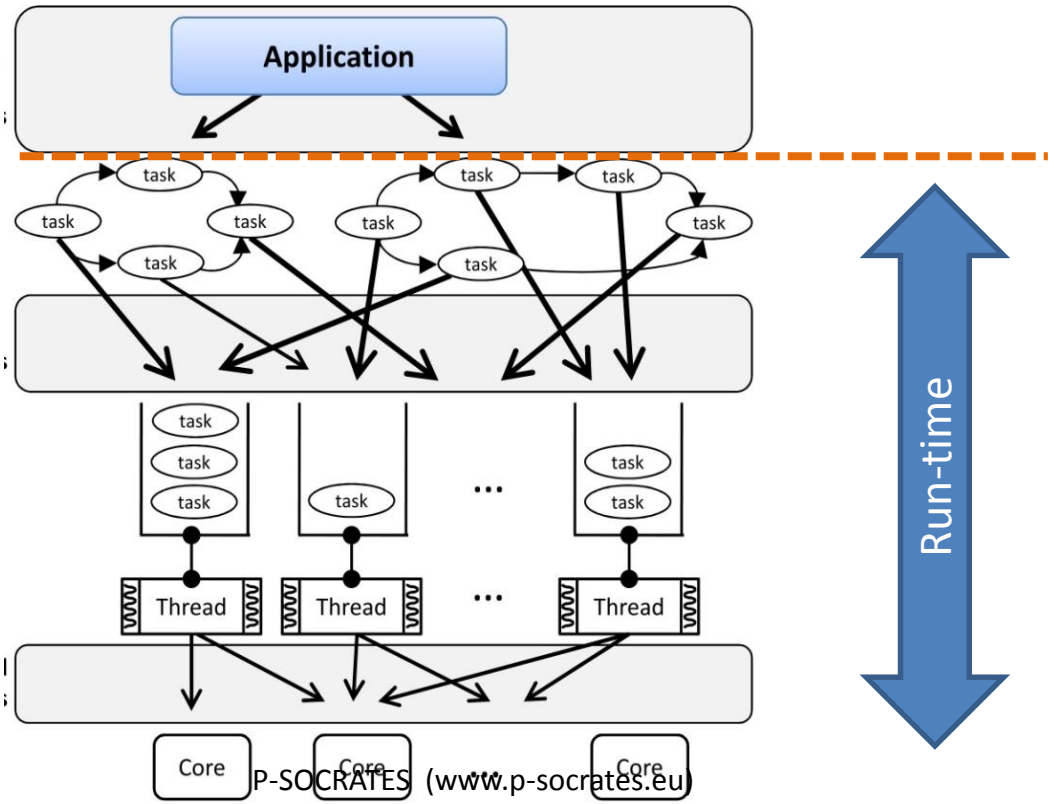


Objectives

- A **generic framework**, integrating models, tools and system software, to parallelize applications with **high performance** and **real-time** requirements
  - Fine-grain **parallelization** of **real-time high-performance** systems
  - Automatic **extraction of parallelism** with **timing and data-flow** annotations
  - Dynamic **allocation of tasks** to, and appropriate **scheduling** of, processor resources **minimizing interference** and guaranteeing **bounded execution time**
  - New methods to **upper-bound** the **run-time interference** between the tasks and between these and the hardware architecture

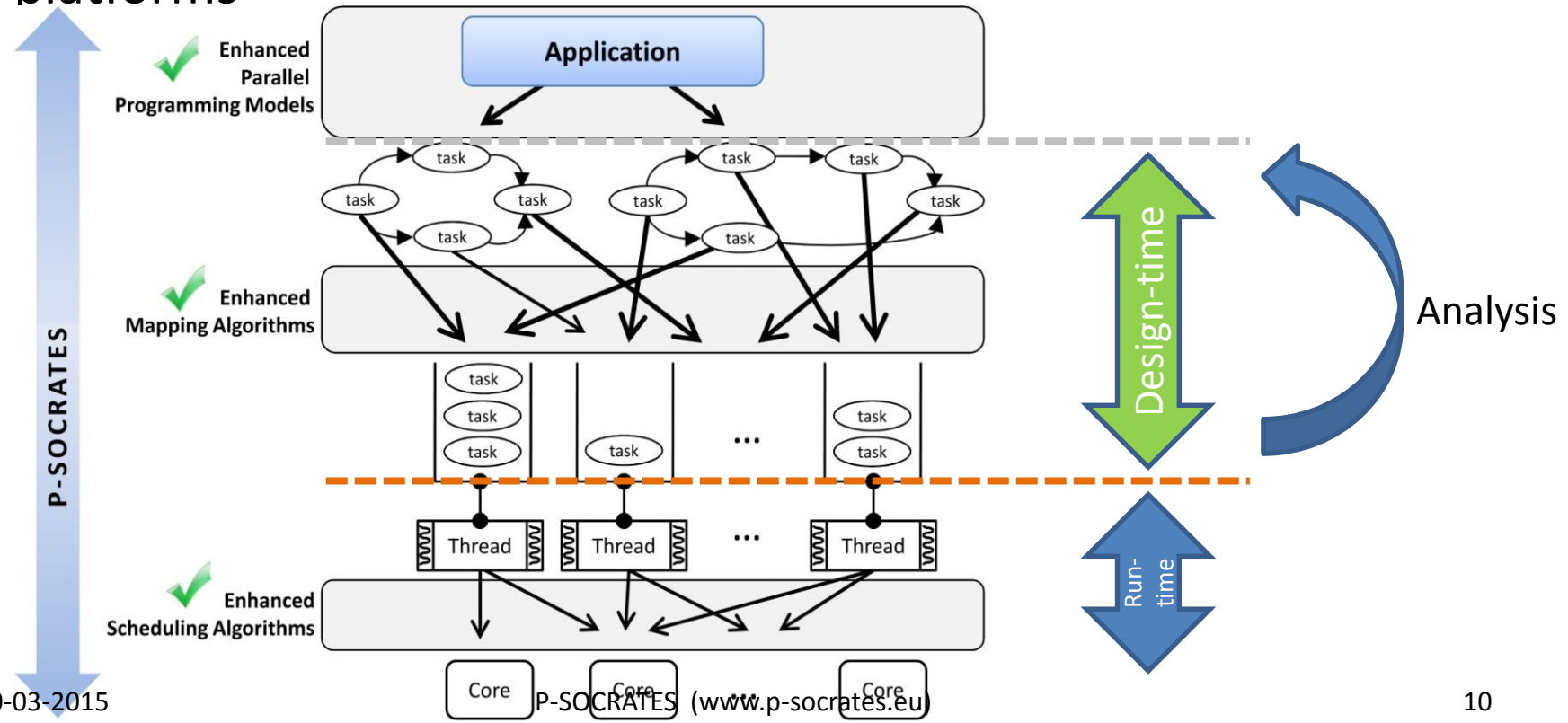
Approach

- Enhanced **parallel programming models** with automatic extraction of parallelism, and **mapping** and **scheduling algorithms** guided by timing analysis methods on many-core platforms



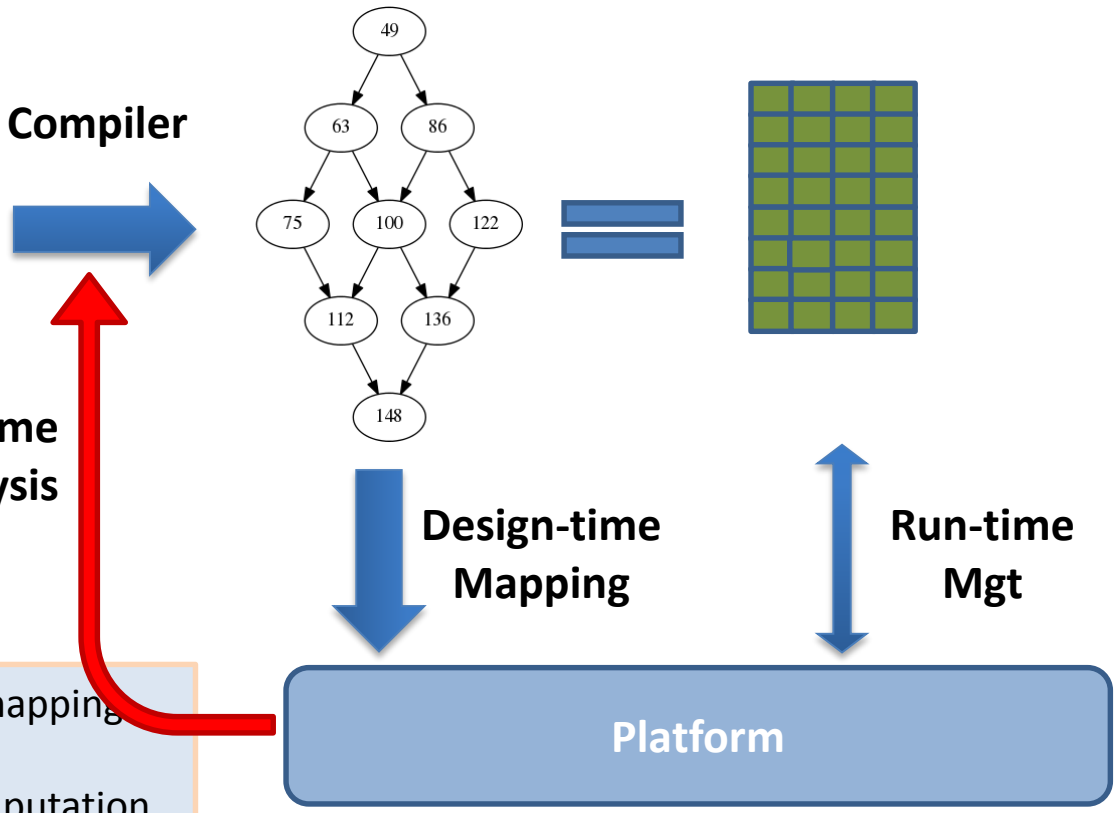
Approach

- Enhanced **parallel programming models** with automatic extraction of parallelism, and **mapping** and **scheduling algorithms** guided by timing analysis methods on many-core platforms



Approach

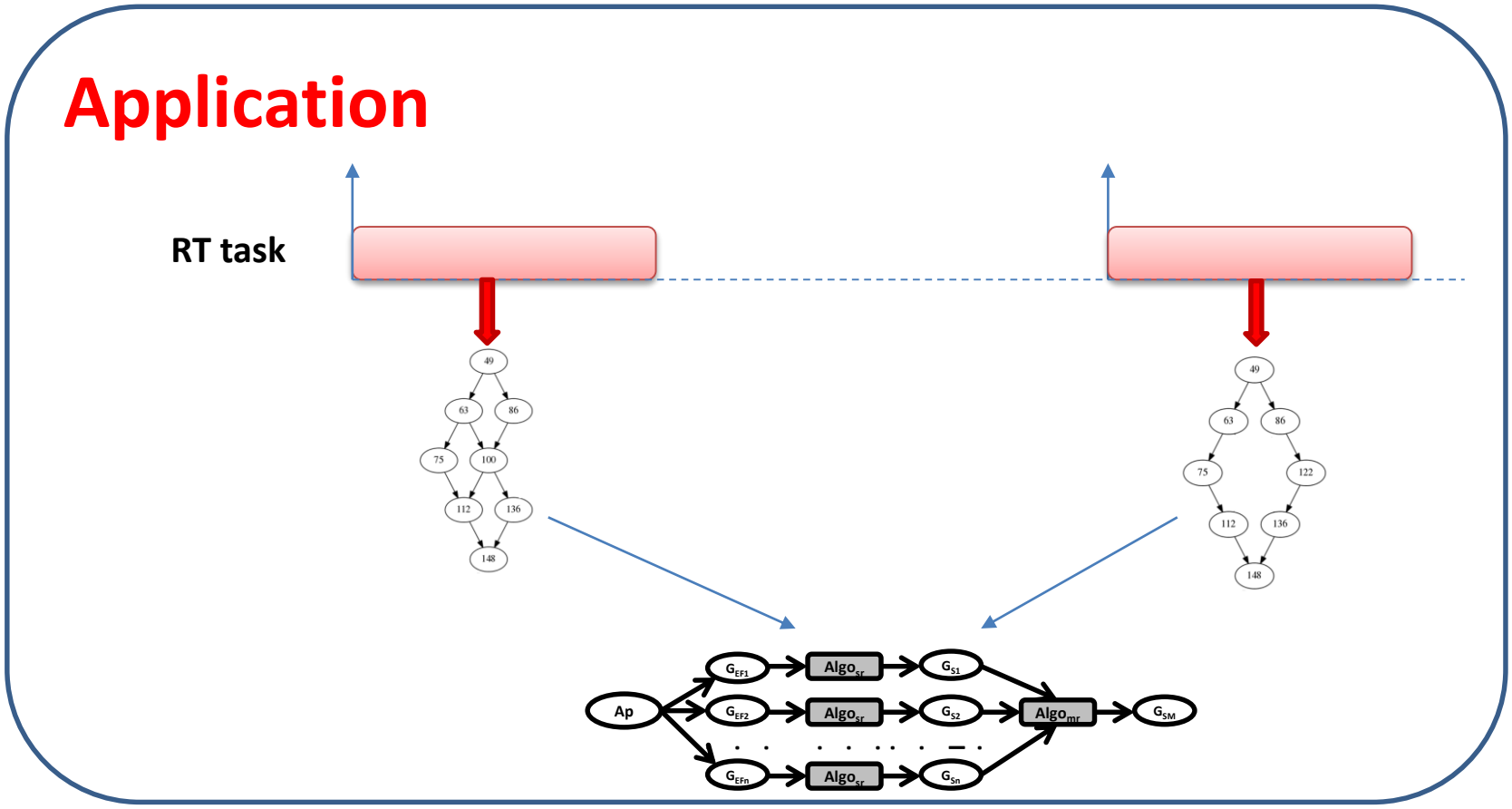
```
for(int i=0; i<3; i++) {
  for(int j=0; j<3; j++) {
    if(i==0 && j==0) { // Task T1
      #pragma omp task
depend(inout:m[i][j])
        compute_block(i, j);
    } else if (i == 0) { // Task T2
      #pragma omp task depend(in:m[i][j-1],
inout:m[i][j])
        compute_block(i, j);
    } else if (j == 0) { // Task T3
      #pragma omp task depend(in:m[i-1][j],
inout:m[i][j])
        compute_block(i, j);
    } else { // Task T4
      #pragma omp task depend(in:m[i-
1][j],m[i][j-1],m[i-1][j-1],

inout:m[i][j])
        compute_block(i, j);
```



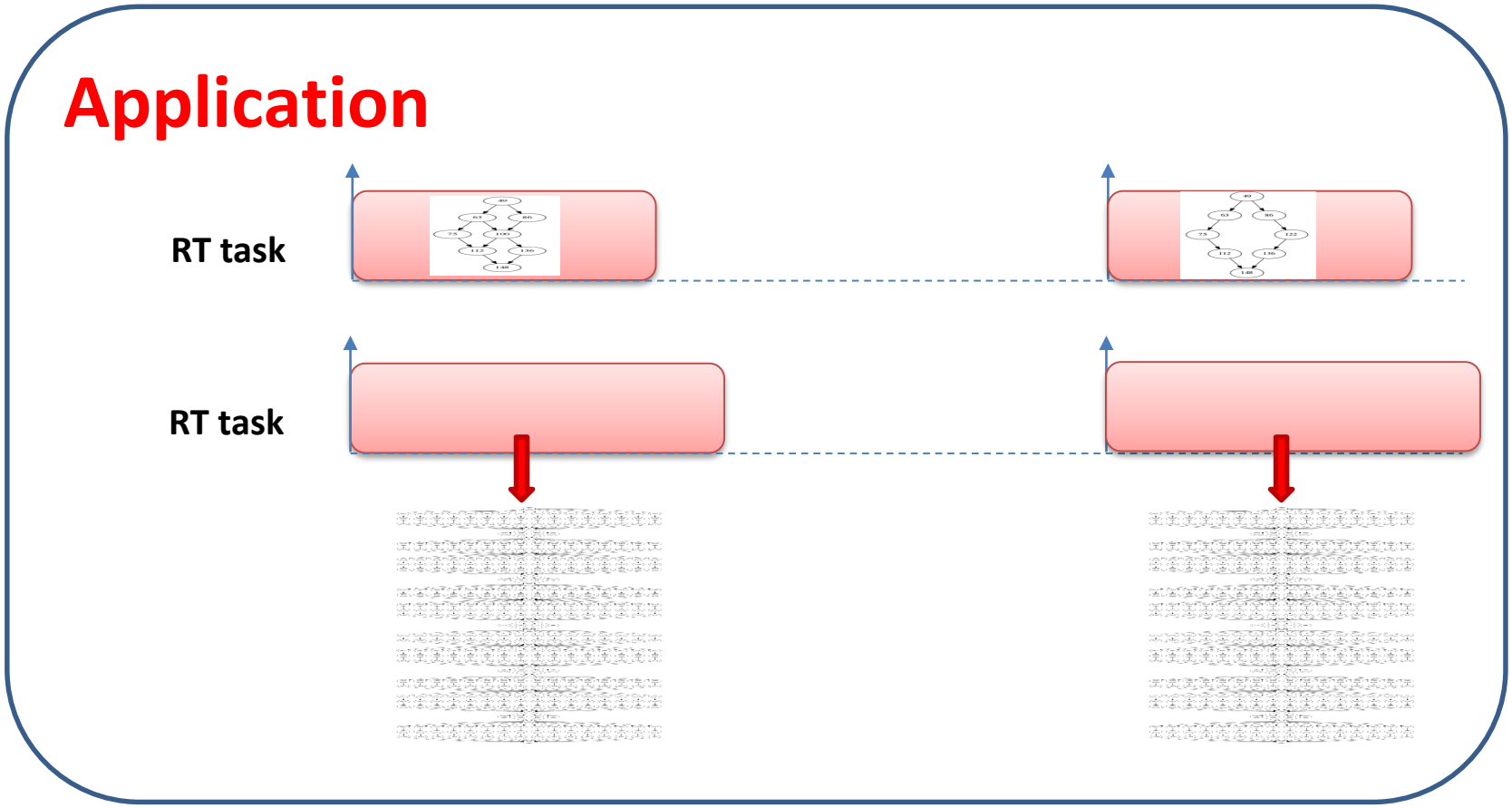
Exploring Hardware model to guide mapping
- Reducing complexity, grouping
- Scheduling communication/computation
- Explore measurement-based approaches

Clustered architecture to provide composability

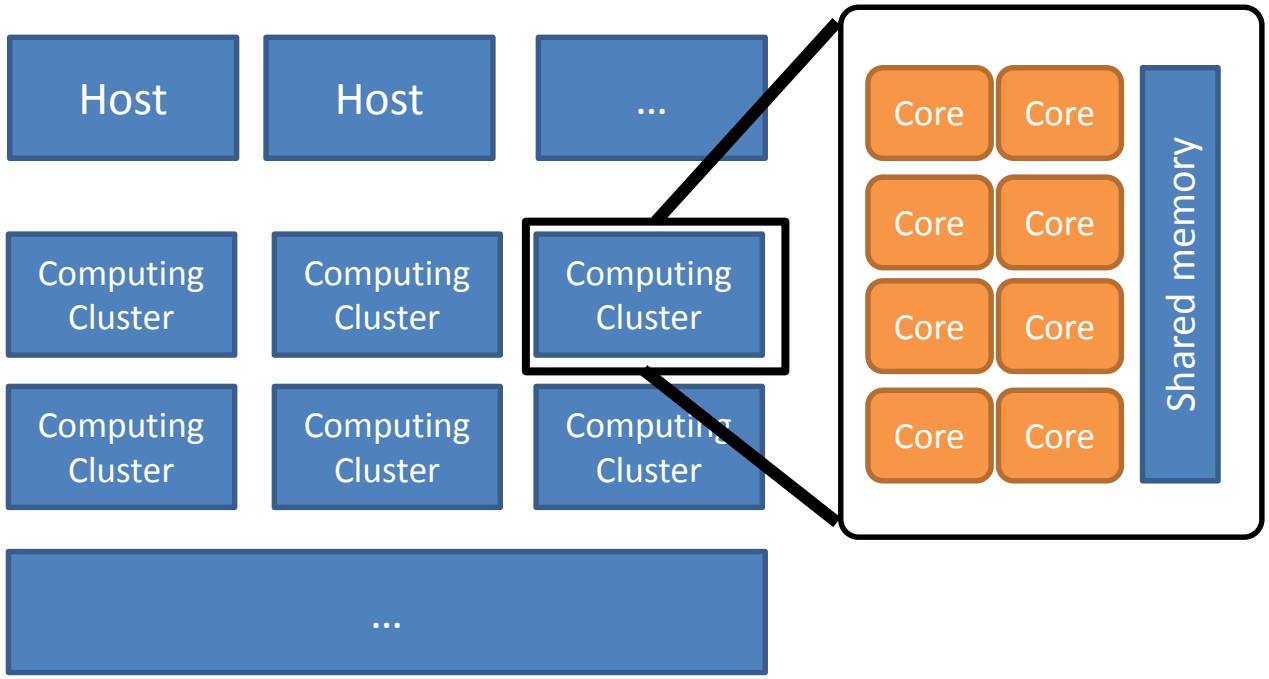
Application model



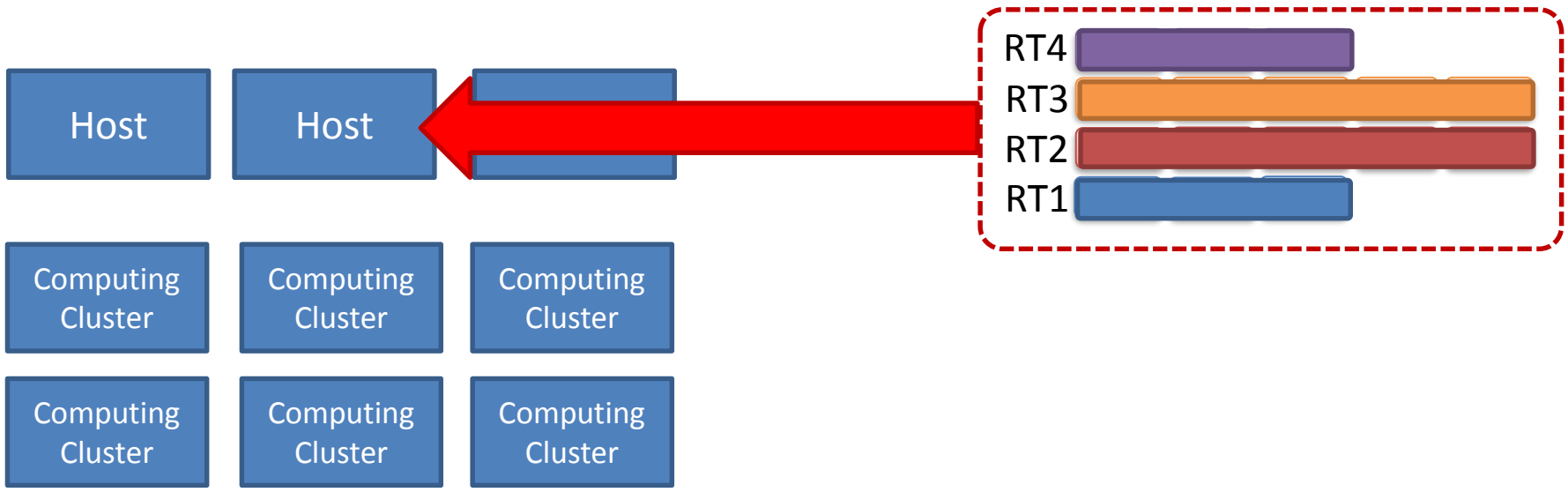
Application model



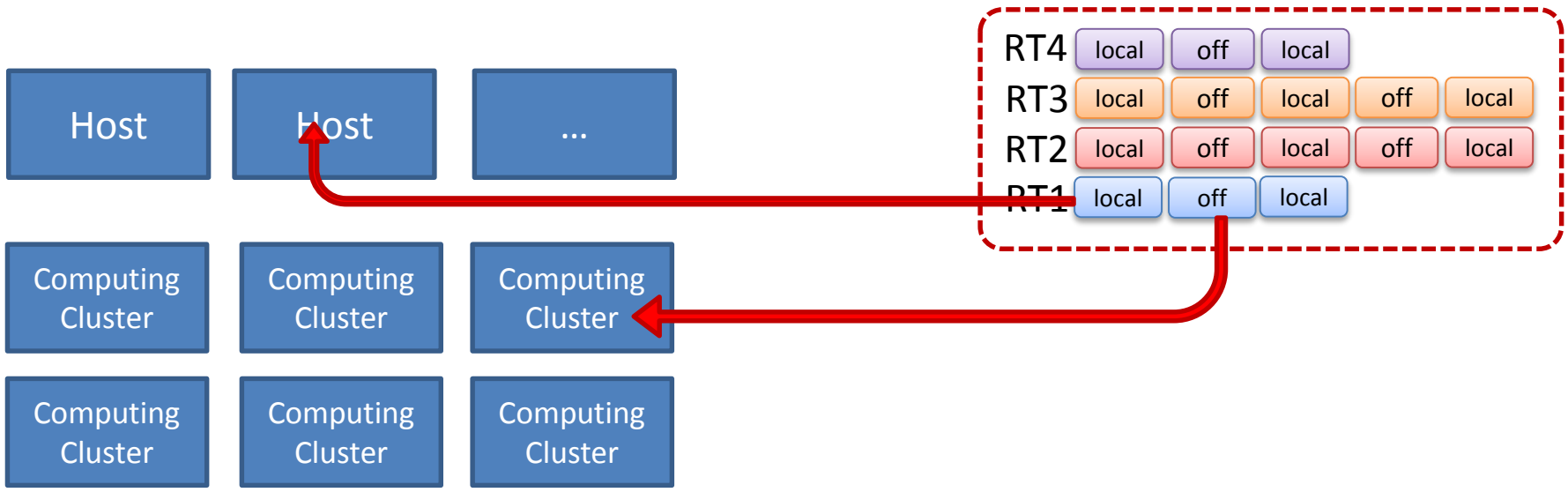
Platform model



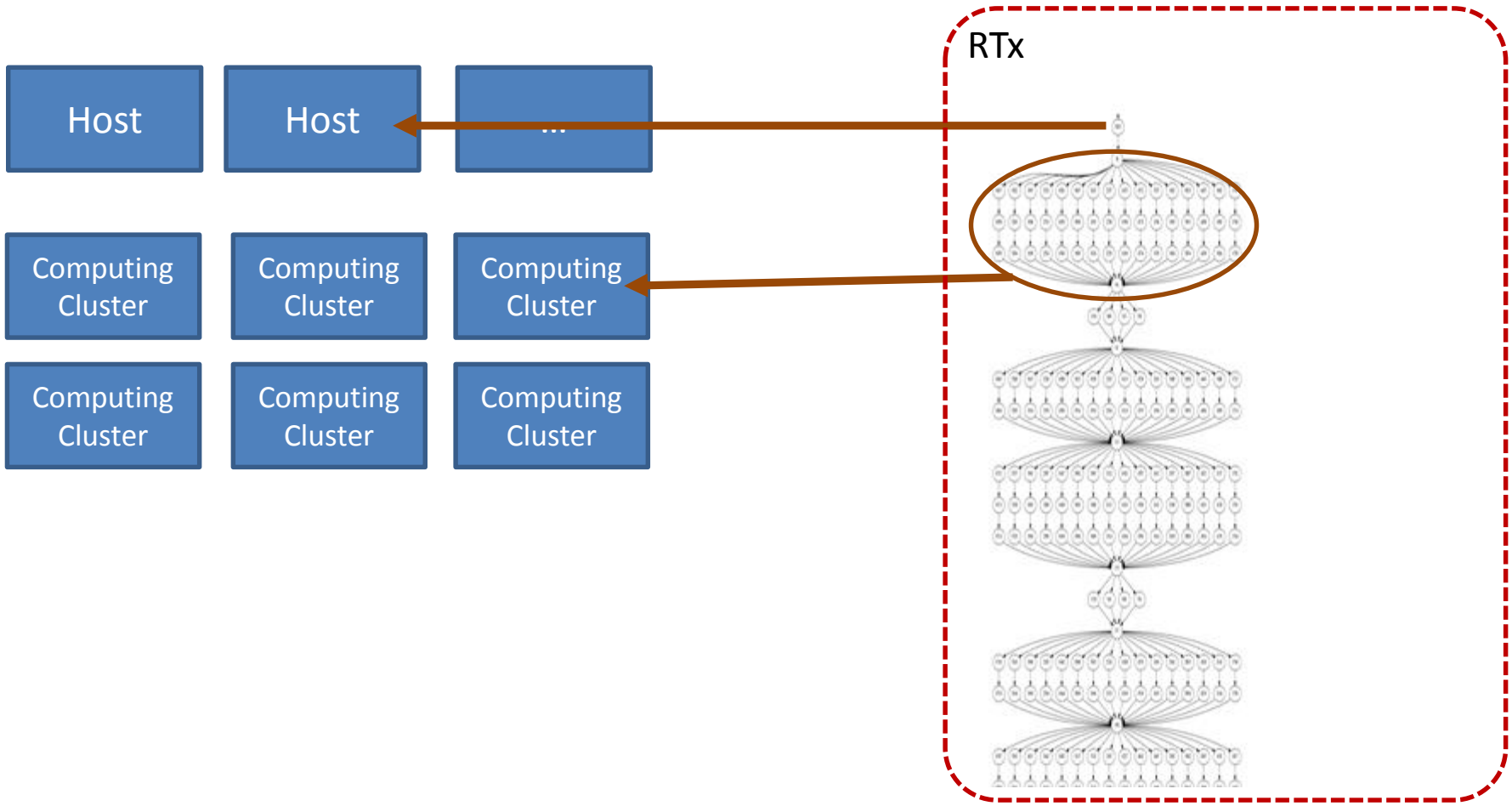
System model

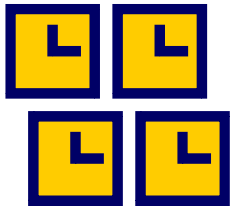


System model



System model





Summary

- P-SOCRATES intends to reconcile parallelism with predictability
  - Explore the performance opportunities of next-generation many-core embedded processor architectures
  - Maintaining the predictability associated with real-time systems
    - Static extraction of task dependency graphs from high-performance parallel programming models
    - Design-time mapping of TDG to a clusterised many-core platform
    - Real-time timing and schedulability analysis feedback

Discussions – Questions & Answers

Thank you very much for your attention!