A Virtual Platform Environment for Exploring Power, Thermal and Reliability Management Control Strategies in High-performance Multicores

Andrea Bartolini†,

Matteo Cacciari †, Andrea Tilli †, Matthias Gries‡, Luca Benini †

University of Bologna, DEIS †
Italy

Intel Labs ‡
Germany Microprocessor Lab

a.bartolini, matteo.cacciari, andrea.tilli, luca.benini@unibo.it, matthias.gries@intel.com

Outline

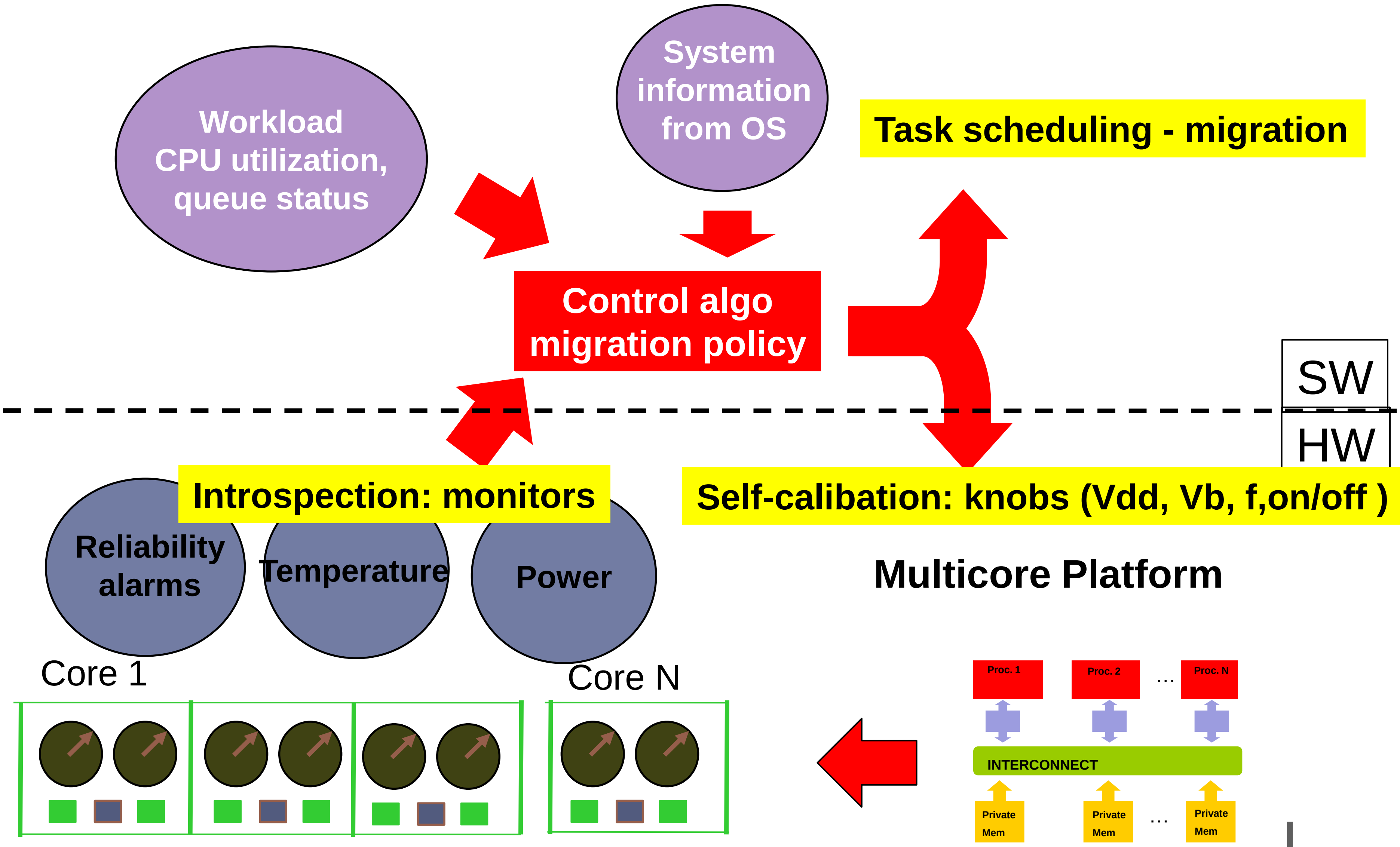
- Introduction
- Simulation Strategy
- Virtual Platform
- Platform Characterization
- Case Study
- Conclusion

Today and Future Multicores

Resource Management solution:

on-line tuning of system performance and temperature through closed-loop control management policies

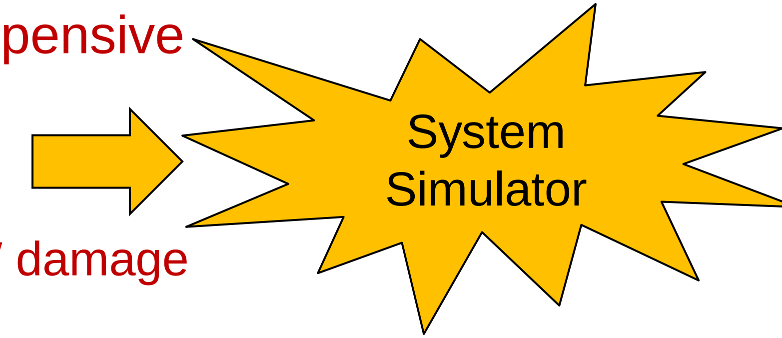
Management Loop: Holistic view



Evaluation Strategies

HW prototype and test-chips: accurate but expensive

- Long and complex development stage
- Limited introspection
- Incorrect control policy can lead to permanent HW damage



Simulators *key features* :

- must **accurately simulate** the **entire system evolution** (real workload, program flow, data coherency conflicts, complex memory latency)
  - to avoid unrealistic simulation artifacts
  - to allow **evaluation of control solution** that **fits in the system**
- must **emulate** the **same performance knobs** and introspective **sensors** of real HW
- must **co-simulate** the **physical effects** we want the controller to be developed for
  - Power model, Thermal model, Reliability model
- allows **high-level control strategies co-simulation**
  - acting on virtual performance knobs / reacting on virtual introspective sensors

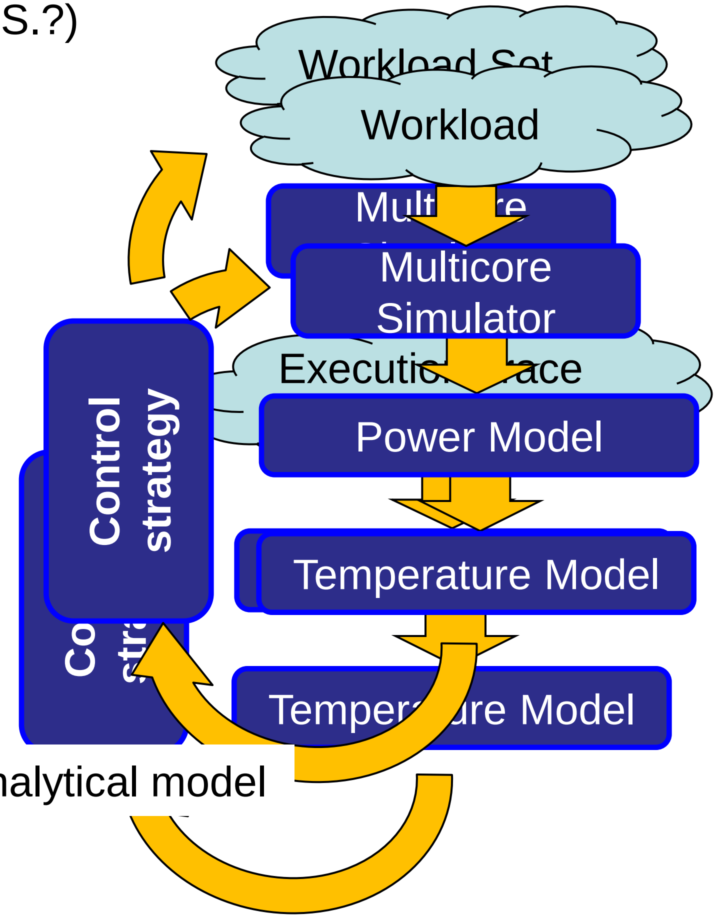
Simulator Strategy

Trace driven Simulator [1]:

- Not suitable for full system simulation (How to simulate O.S.?)
- looses information on cross-dependencies
  - → How to account **data, memory contention?**
  - → resulting in degraded simulation accuracy

Close loop Simulator:

- Cycle accurate simulators [2] :
  - **High** modeling **accuracy**
  - support **well-established power** and **temperature** co-simulation based on **analytical models** and system **micro-architectural knowledge**
  - Low simulation speed
  - Not suitable for full-system simulation
- Functional and instruction set simulators:
  - allow full system simulation
  - less internal precision → no micro-architectural analytical model
  - less detailed input data
  - introduces the **challenge of having accurate power and temperature physical models**



[1] *P Chaparro et al.* Understanding the thermal implications of multi-core architectures. *2007*
[2] *Benini L. et al*. MPARM: Exploring the multi-processor SoC design space with SystemC 2005

Contributions

Novel Virtual Platform - Goals:

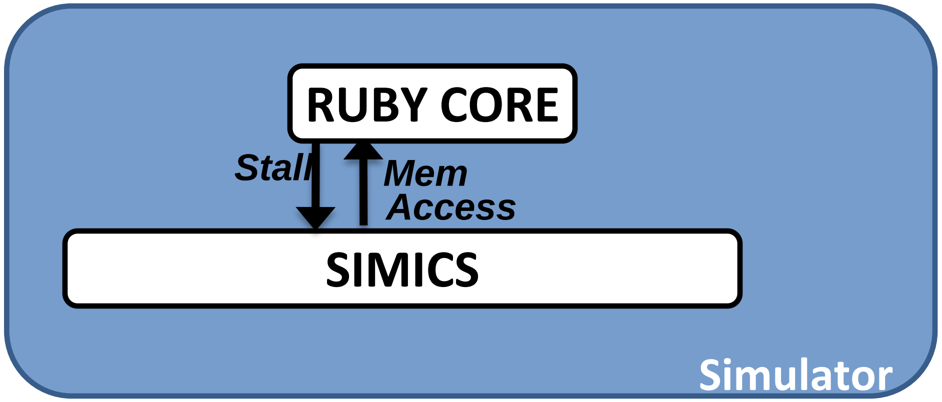
- Allows *fast but physical accurate* multicores simulator
  - Based on a full-system multicore high-level functional simulator (Virtutech Simics [1])
  - Integrates Power and Thermal model derived by real HW characterization
- Enables *fast control-algorithms prototyping*, *design* and *testing*:
  - Allowing co-simulation of multicores SoCs with high-level description of control-algorithms (Mathworks Matlab/Simulink [2])

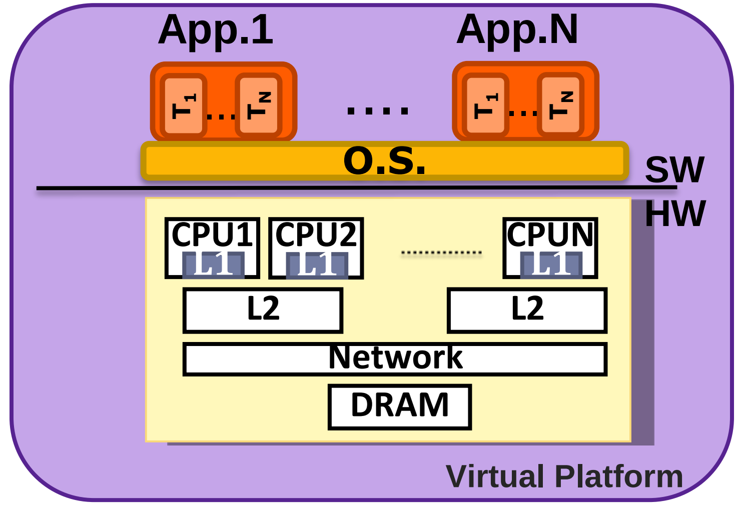
Challenges:

- Identification, from experiments on a real general-purpose multicore platform, of accurate, but component-oriented, power and thermal models
- Interfacing physical model with functional simulator engine
- Development of a co-simulation bridge between the multicores simulator and Mathworks Matlab/Simulink.

[1] http://www.virtutech.com/ *[2]* http://www.mathworks.com/

Virtual Platform





Simics by Virtutech:

- full system functional simulator
- models the entire system: peripherals, BIOS, network interfaces, cores, memories
- allows booting full OS, such as Linux SMP
- supports different target CPU (arm, sparc, x86)
- x86 model:
  - in-order
  - all instruction are retired in 1 cycle
  - does not account for memory latency

Memory timing model

- RUBY – GEMS (University of Wisconsin)[1]
  - Public cycle-accurate memory timing model
  - Different target memory architectures
  - fully integrated with Virtutech Simics
  - written in C++
  - we use it as skeleton to apply our add-ons (as C++ object)

[1] Martin Milo M. K. et al. Multifacet’s general execution-driven multiprocessor simulator (GEMS) toolset 2005

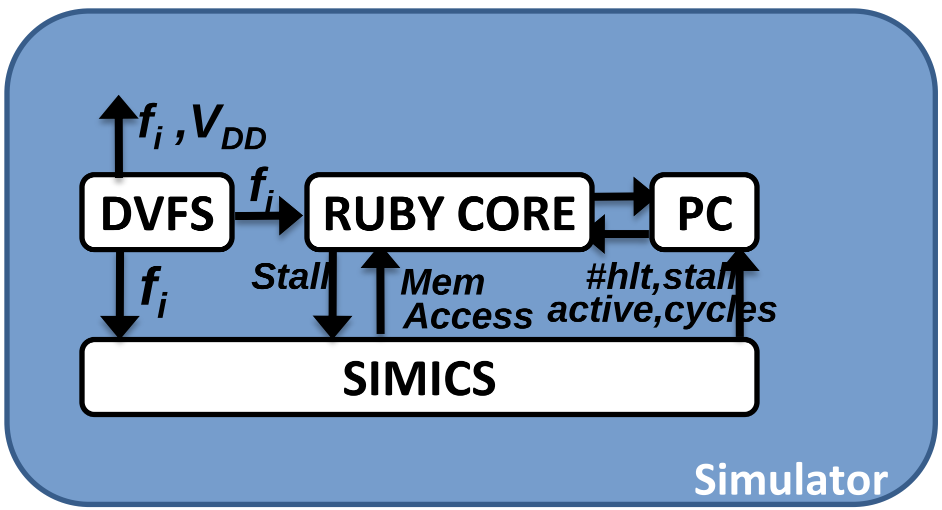
Virtual Platform

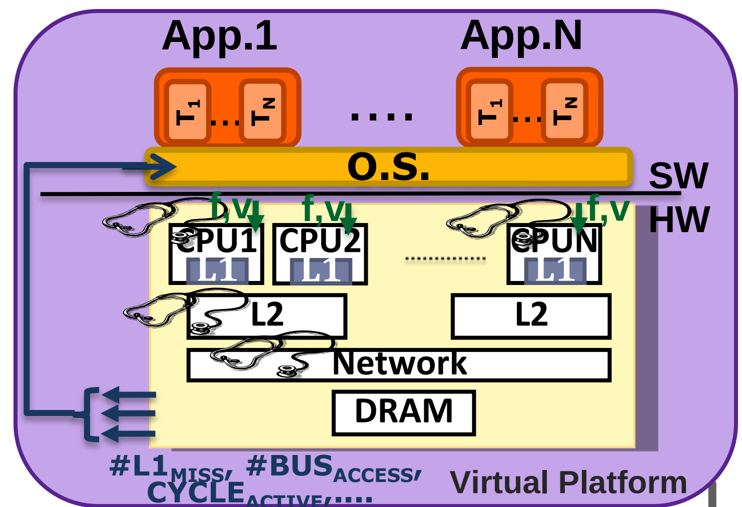
Performance knobs (DVFS) module:

- Virtutech Simics support frequency change at run-time
- RUBY does not support it
  - does not have internal knowledge of frequency
- We add a new DVFS module to support it,
  - ensures L2 cache and DRAM to have a constant clock frequency
  - L1 latency scale with Simics processor clock frequency

Performance counters module:

- Needed by performance control policy
- We add a new Performance Counter module to support it
  - expose to O.S. and application different quantities:
    - the number of instruction retired, clock cycles and stall cycles expired,
    - halt instruction, ...

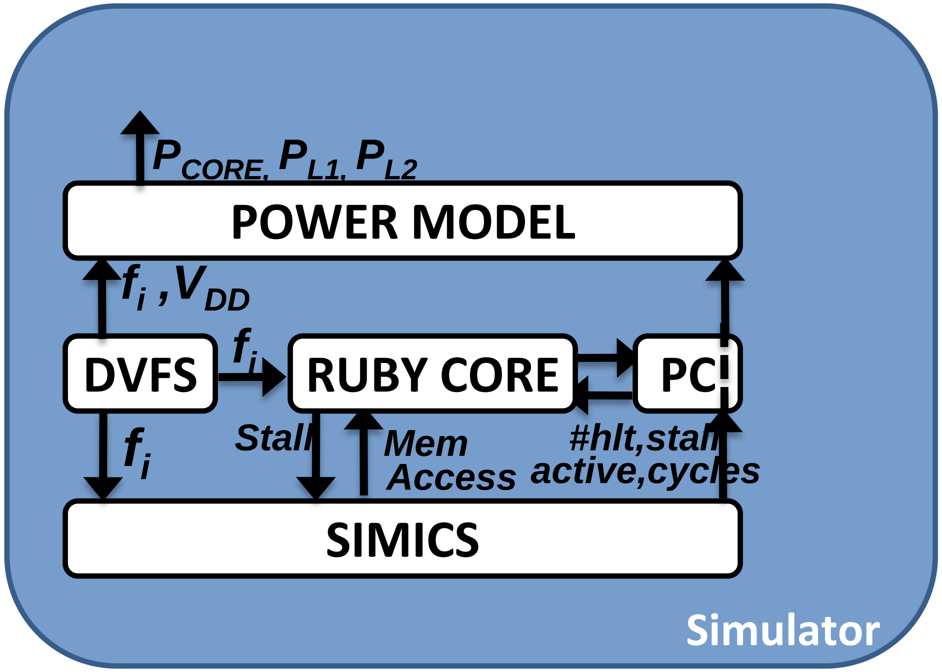


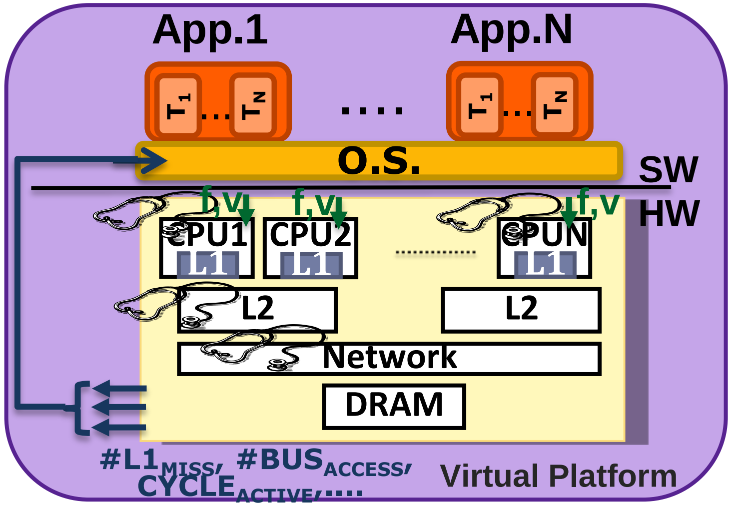


Virtual Platform

Power model module:

- At run-time estimate the power consumption of the target architecture
- Core model $P_T = P_D(f,CPI) + P_S(T, VDD)$
- P_D experimentally calibrated analytical power model
- Cache and memory power – access cost estimated with CACTI [1]

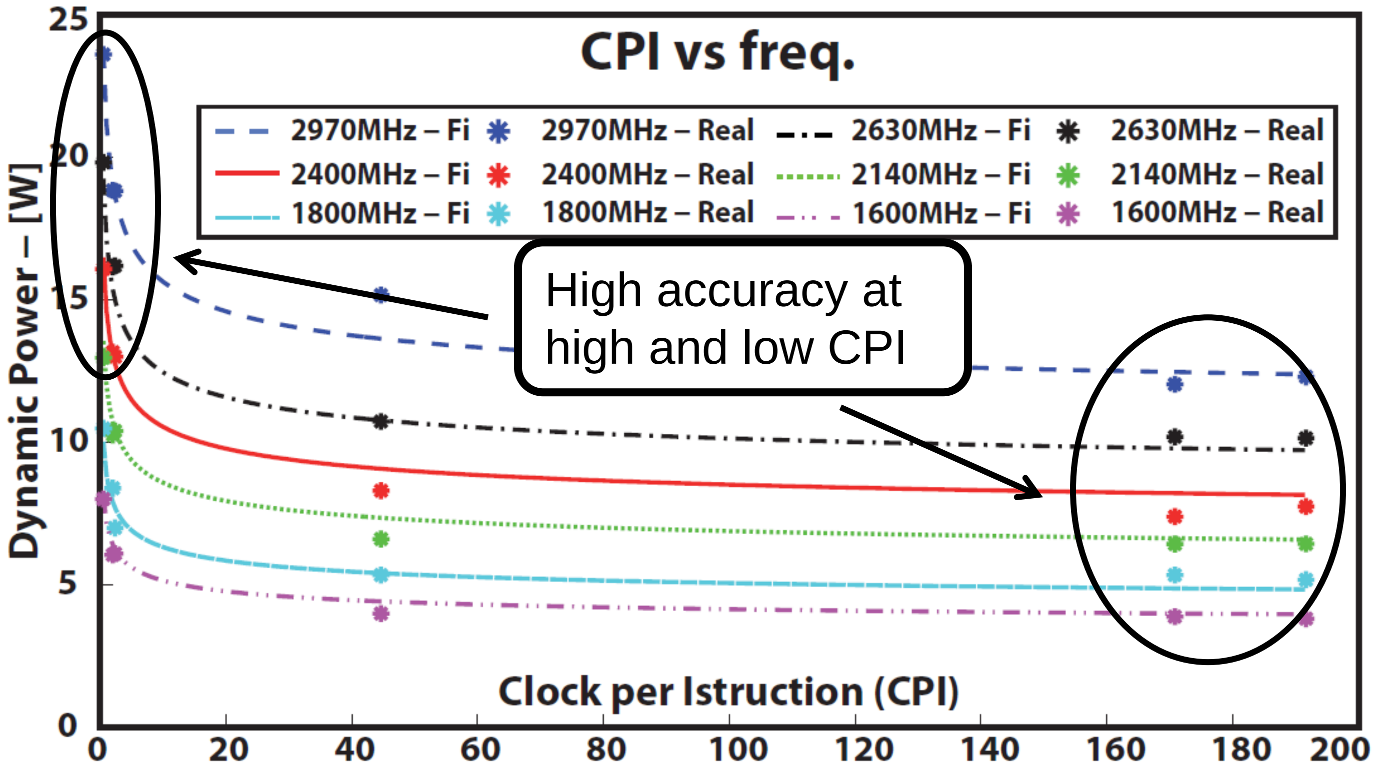




[1] Thoziyoor Shyamkumar et al. A comprehensive memory modeling tool and its application to the design and analysis of future memory hierarchies. 2008

Modeling Real Platform – Power

Real Power Meas
- *Intel server system*
  - 16 cores - 4 qu
  - 16GB FBDIMM
  - Intel® Core™
- *At the wall Power c*
  - test:
    - set of synt
    - forcing all
    - for each b
    correlate it w



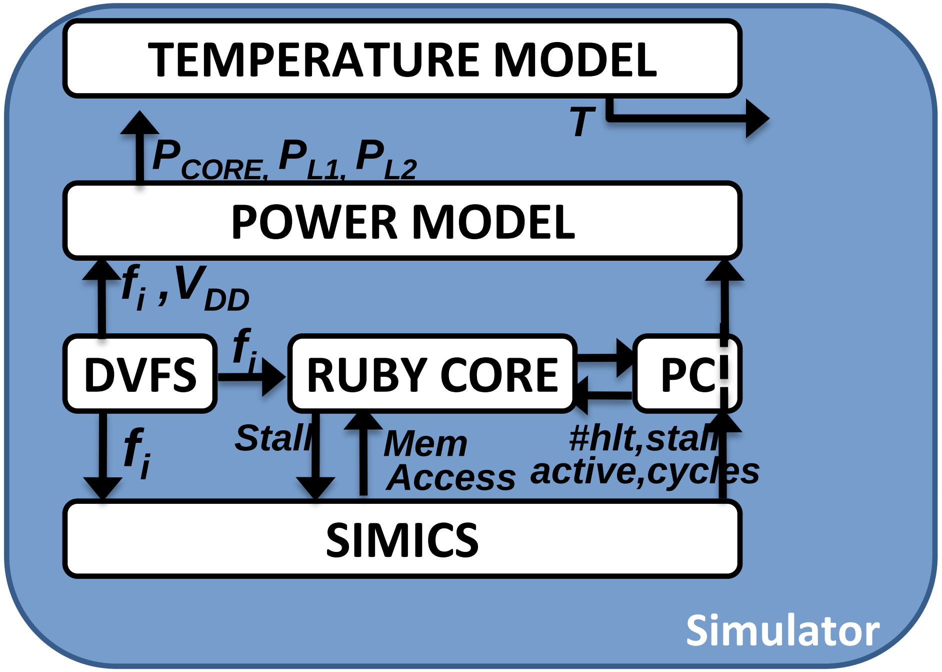
$$P_D = k_A \cdot V_{DD}^2 \cdot f_{CK} + k_B + (k_C + k_D \cdot f_{CK}) \cdot CPI^{k_E}$$

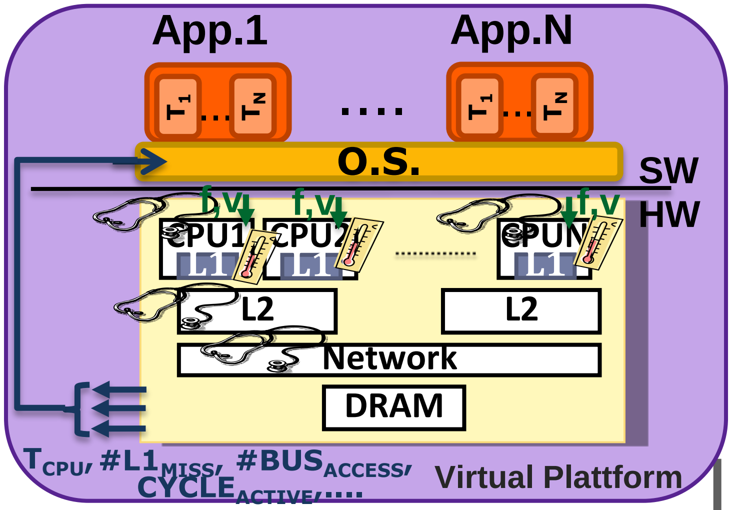
- We relate the static power with the operating point by using an analytical model

Virtual Platform

Temperature model module:

- we integrate our virtual platform with a thermal simulator [1]
- Input: power dissipated by the main functional units composing the target platform
- Output: Provides the temperature distribution along the simulated multicore die area as output

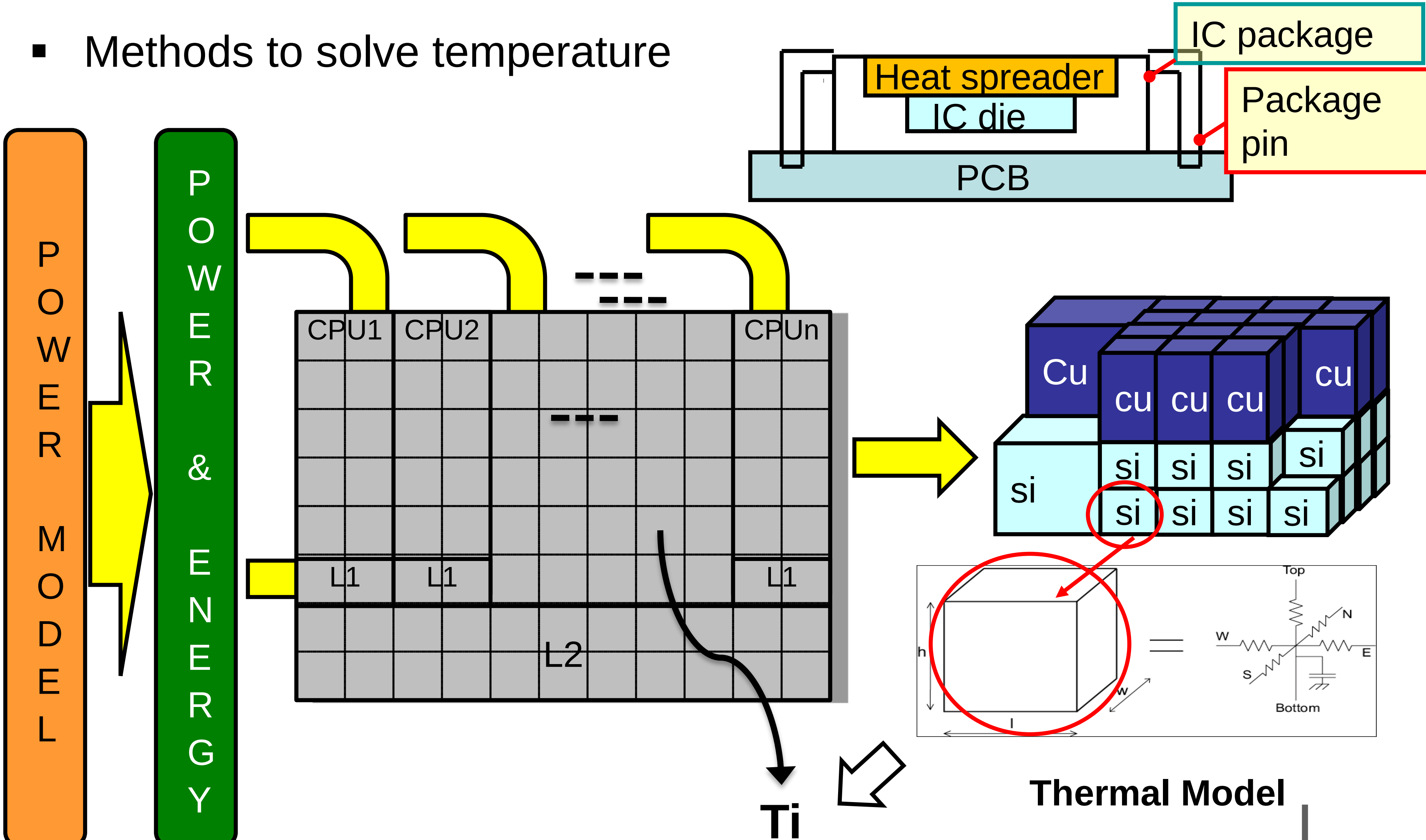




[1] Paci G. et al. Exploring "temperature-aware" design in low-power MPSoCs

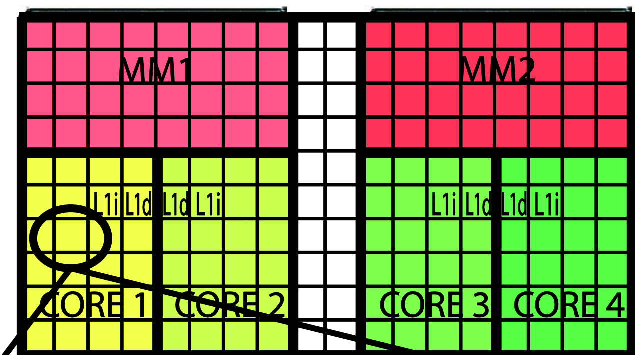
Thermal Model

- Methods to solve temperature

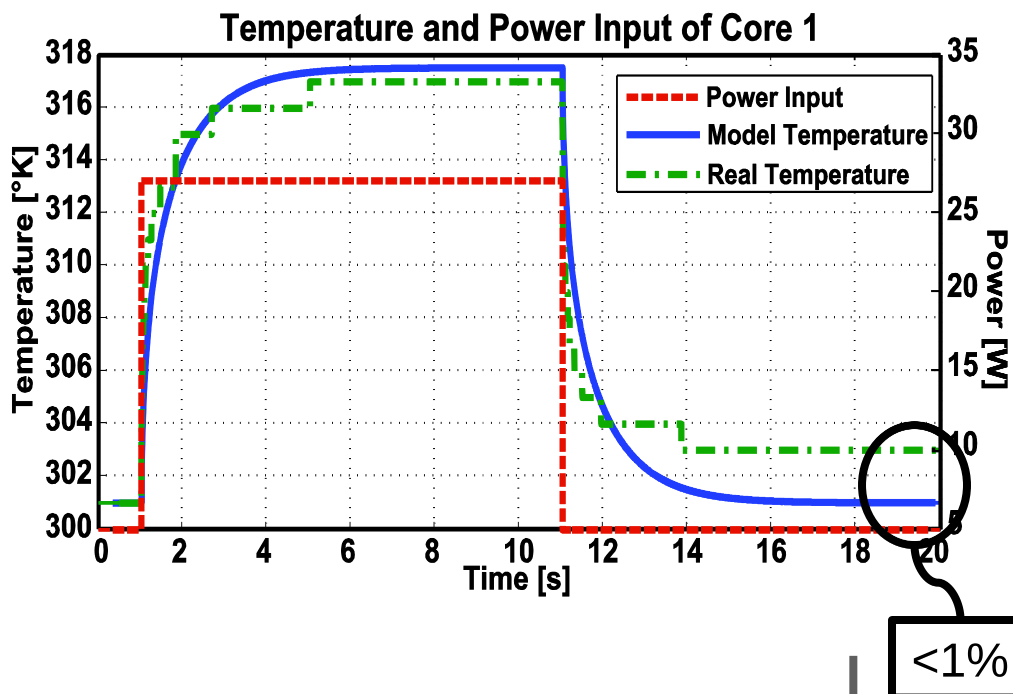


Modeling Real Platform– Thermal

- Thermal Model Calibration :
  - Derived from Intel® Core™ 2 Duo layout
  - We calibrate the model parameter to simulate real HW transient
  - High accuracy (error < 1%) and same transient behavior



silicon thermal conductivity	$150 \cdot (\frac{300}{T})^{4/3}$ W/mK
silicon specific heat	$1.628e^{-12}$ J/um
silicon thickness	350um
copper thermal conductivity	400W/mK
copper specific heat	$3.55e^{-12}$ J/um 3K
copper thickness	2057um
elementary cell length	1312um
package-to-air conductivity	0.4K/W



Virtual Platform Performance

- Target:
  - 4 core Pentium® 4
  - 2GB RAM
  - 32 KB private L1 cache
  - 4 MB shared L2 cache
  - Linux OS
- Host:
  - Intel® Core™ 2 Duo
  - 2.4 Ghz
  - 2GB RAM

1 Billion instruction

Simics + Ruby:

Tsim = 1040 s

Simics + Ruby + DVFS:

Tsim = 1045 s

Simics + Ruby + DVFS + Power:

+ 7%

Tsim = 1110 s

Compute every 13us

Simics + Ruby + DVFS + Power + Thermal interface:

Tsim = 1160 s

Simics + Ruby + DVFS + Power + Thermal Model:

+ 19.2%

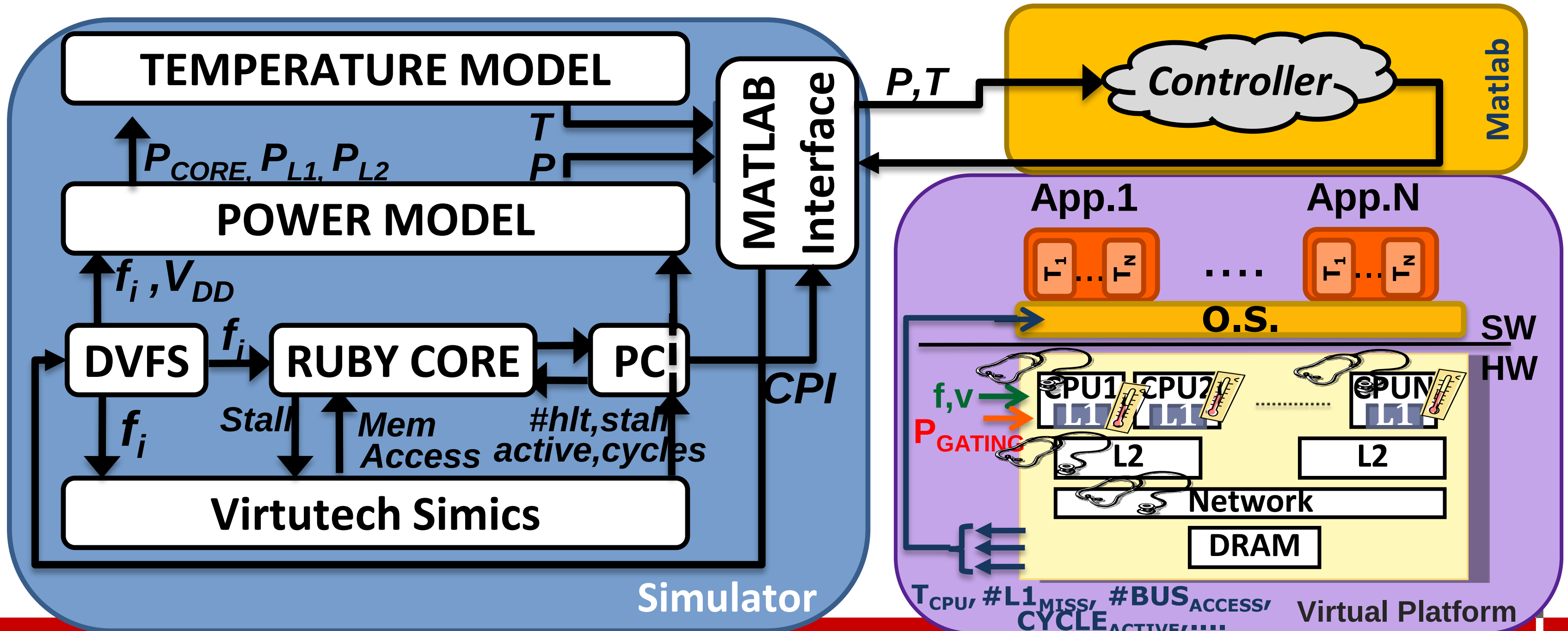
Tsim = 1240 s

68 cells
T = 100ns

Virtual Platform

Mathworks Matlab interface:

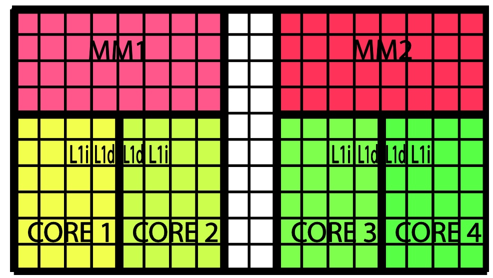
- New module named Controller in RUBY
- Initialization: starts the Mathworks Matlab engine concurrent process,
- Every N cycle - wake-up:
  - send the current performance monitor output to the Mathworks Simulink model
  - execute one step of the controller Mathworks Simulink model
  - propagate the Mathworks Simulink controller decision to the DVFS module



Case Study

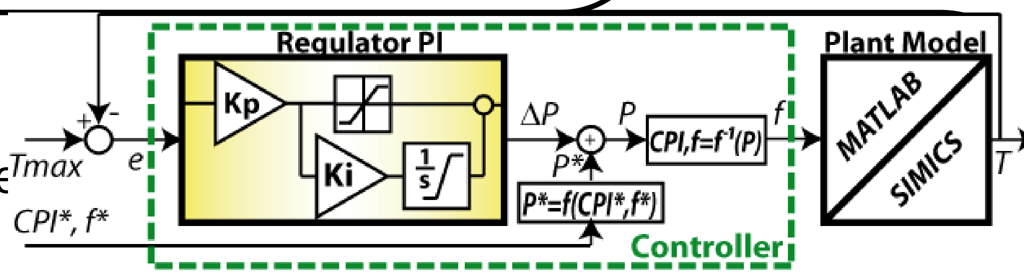
System

- 4 core Pentium ® 4
- 2GB RAM
- 32 KB private L1 cache
- 4 MB shared L2 cache
- Linux OS



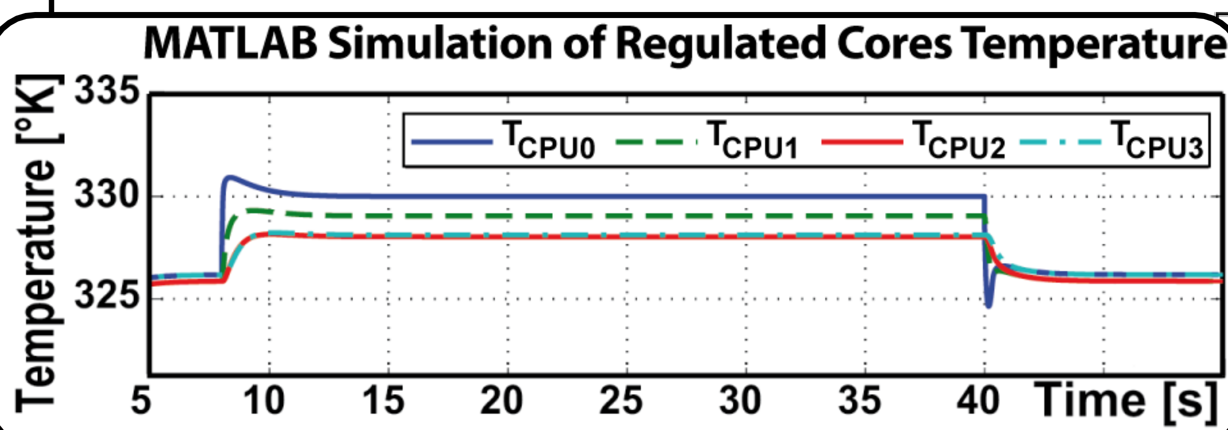
- $T_{THRESHOLD}$ = 330°K
- $T_{ENVIRONMENT}$ = 300°K
- 4 Temperature sensors

1. Simulink model
   - Low accuracy thermal mode
     - one cell per core

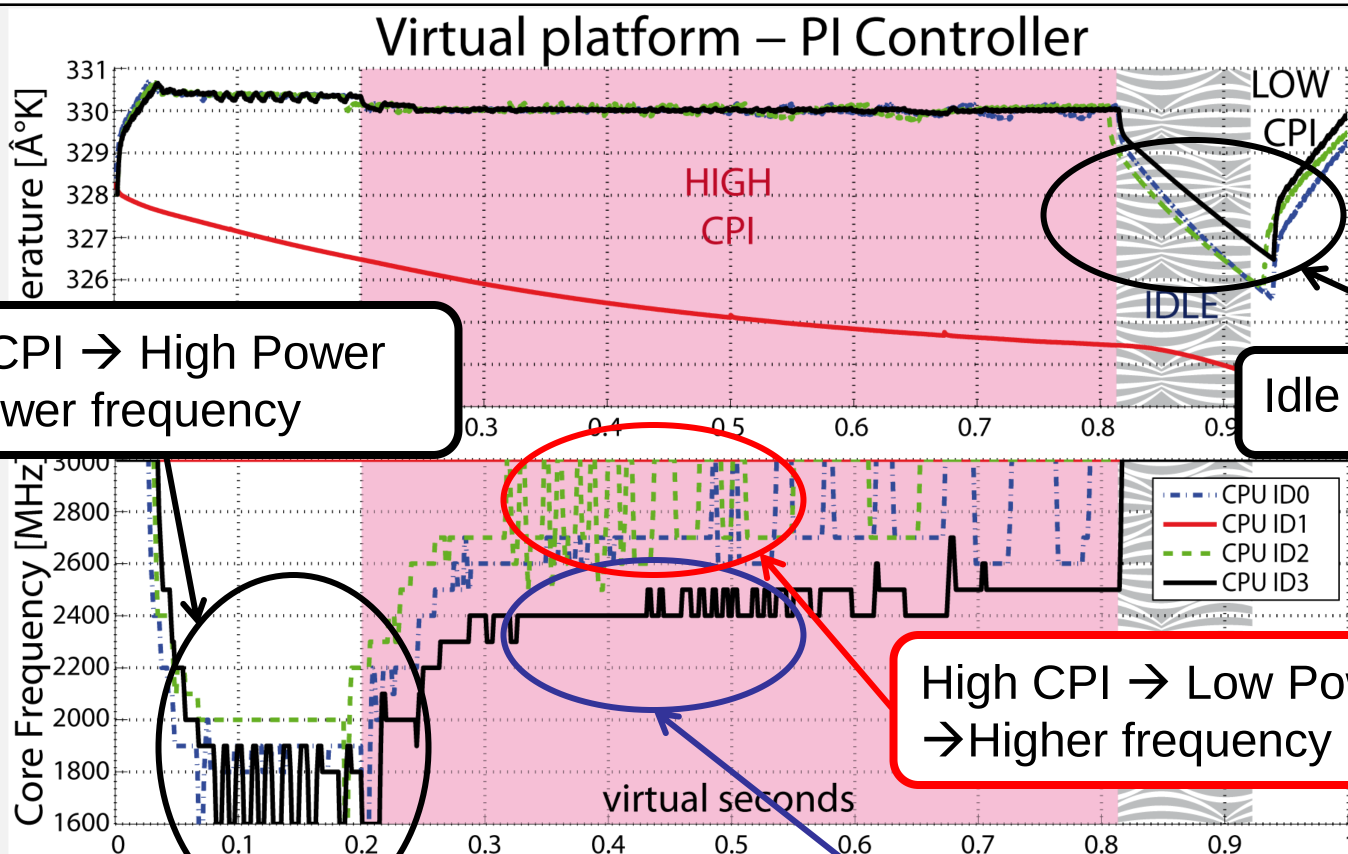


Cntr 0	Cntr 1	Cntr 2	Cntr 3
3.9439	13.7875	13.7875	13.9439
.9840	0.9840	0.9840	0.9840

se margin and a crossing



Case Study



- Each task made up by three section:
  - Low CPI phase (CPU bou
  - High CPI phase (Memory bounded)
  - Idle (sleep)

Conclusion

- We have presented a novel virtual platform for efficiently designing, evaluating and testing power, thermal and reliability closed-loop resource management solutions
- We create a modular high-level simulation platform for multicore systems
- Models for power dissipation and thermal dynamics, compatible with high-level instruction set emulation, derived from real hardware characterization
- We enhance the performance controller design and the simulation capability by allowing a Mathworks Matlab/Simulink description of the controller to execute natively as a new component of the virtual platform
- The controller directly drives the performance knobs of the emulated system
- This brings to an new controller development cycle that helps the developer in converging to an optimum "in the field" performance control solution

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA