POLITECNICO DI MILANO

September 7-8, 2017

Università La Sapienza - Roma

IWSES - Italian Workshop on Embedded Systems (2° Edition)

Speech





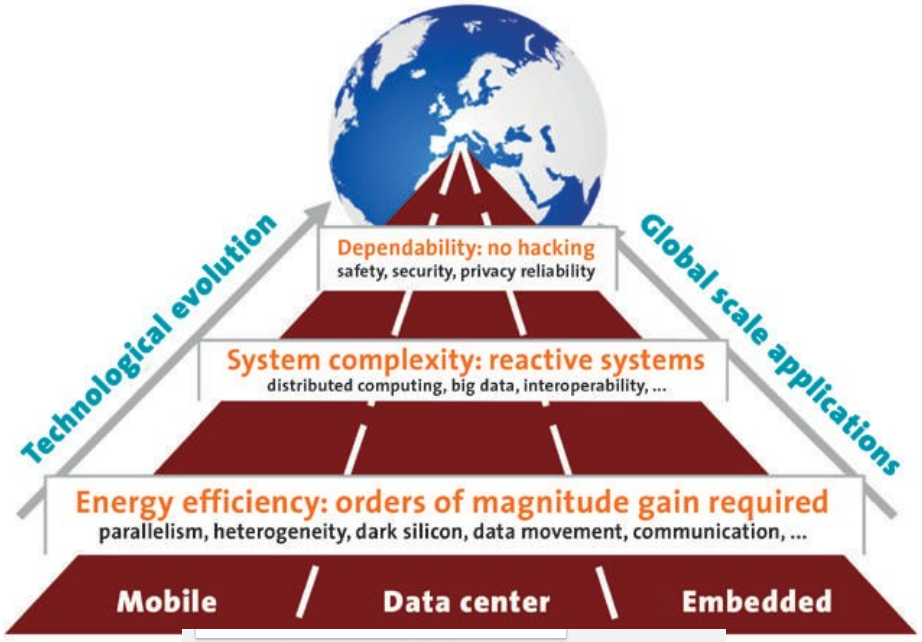
Thermal Analysis and Management of Multi-core Systems

Prof. William Fornaciari
Politecnico di Milano, DEIB
Dipartimento di Elettronica, Informazione e Bioingegneria
Via Ponzio 34/5, 20133, Milano, ITALY
william.fornaciari@polimi.it

Outline

- **Introduction**
  - Application needs, multi-core trend and design showstoppers
  - HEAP Lab
- **Thermal-Performance analysis**
  - Thermal analysis and DVFS policy development
  - Run-Time Resource Manager
- **Conclusions**
  - Ongoing work
  - Exploitable results and projects

Main topics from the HiPEAC 2020 vision

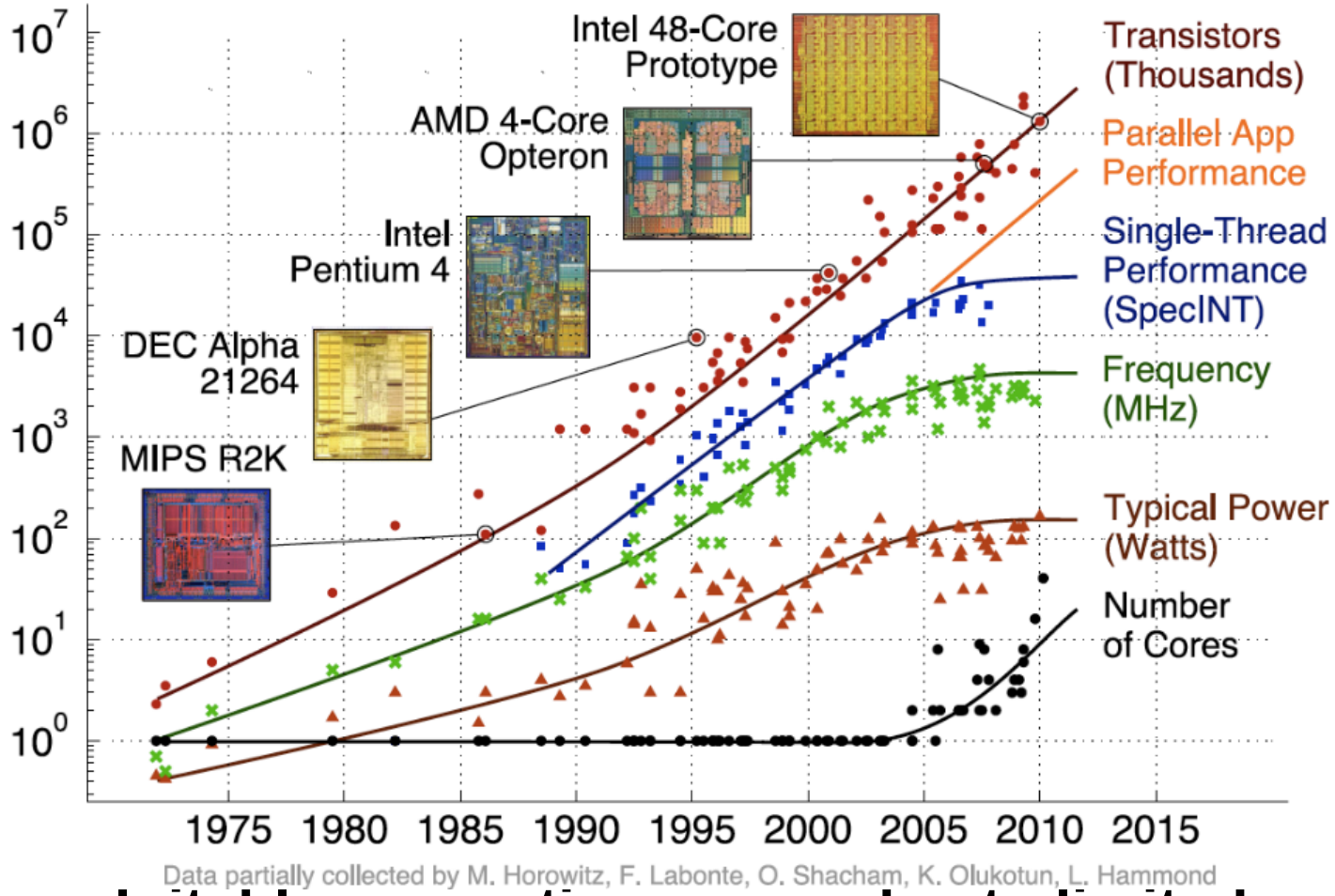


Energy and power dissipation: the newest technology nodes made things even worse

Dependability, which affects security, safety and privacy, is a major concern

Complexity is reaching a level where it is nearly unmanageable, and yet still grows due to applications that build on systems of systems

Towards the dark silicon



- **Not-exploitable computing power due to limited power dissipation**
  - Part of the silicon area is …*dark silicon*

Industry Changes in Requirements

Functionality

Up to 1980s

Supercomputers & mainframes

Functionality

$

1990s

The personal computer

Functionality

Power × $

2000s

Notebooks

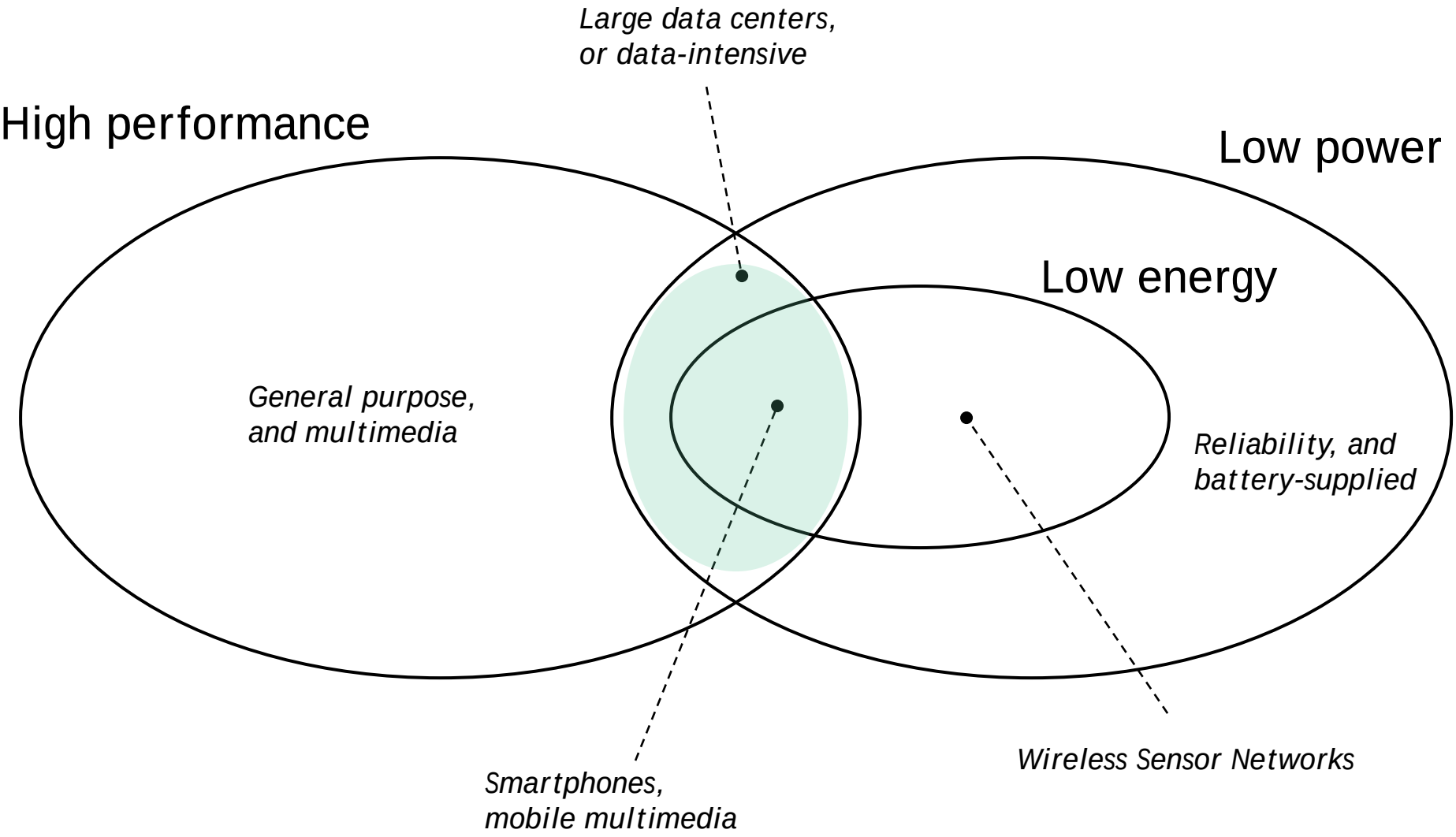
Functionality

Energy × $

2010s

Mobiles & mobility

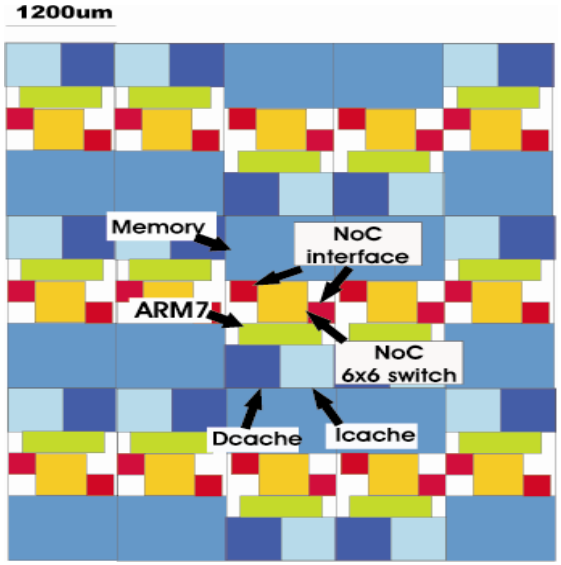
Application scenarios



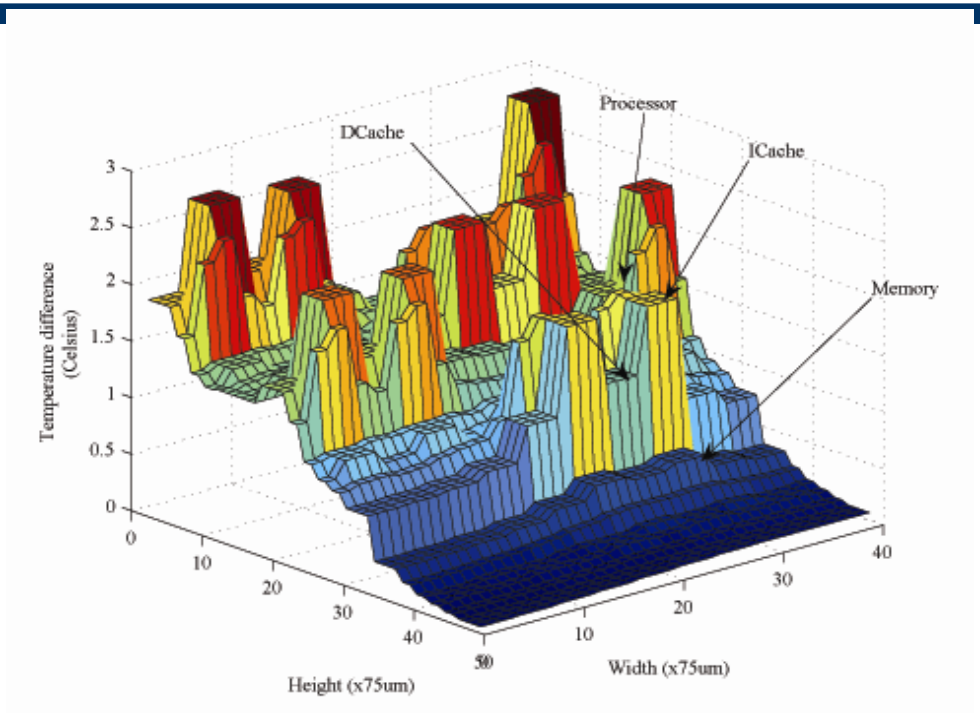
HEAP Lab @POLIMI

Layers Apps	Problems & Solutions	Outputs & Tools
Many cores, HPC	Thermal control for ageing and reliability Run/time load balancing Optimization of non functional aspects Application mapping Power/energy coarse grain monitoring and control	Tip/Top patent filed in 2016 for thermal control (rack level) BarbequeRTRM HPC extension (open source + commercial customizations) OpenCL backend, OpenMP, MPI, … Compilers, DSE tools
Multi-cores, Heterog. Computing High-End ES	Load distribution on heterogeneous cores power/energy fine grain control Design of accelerators Reliability issues	Tip-Top thermal control (firmware) BarbequeRTRM for several commercial boards (Odroid, x86, Zynq, Panda, …) NoC, Simulation toolchain (HANDS), Memory interface optimization DVFS exploitation Compilers, DSE tools
Low-end embedded systems	Energy optimization Size, cost, multi-sensor bords, small footprint OSs DVFS exploitation	Low level run-time optimization of energy and performance Application specific design of software and firmware Development of analysis toolsuite Power attack - countermeasures
Wearable CPS, IoT	Design of ultra-low power boards with sensors, feature extraction, security and privacy WSN clock synchronization	Methodology for clock synch in WSNs Development of platforms for wearable apps Use of georef sources of information and GPRS Miosix open source OS Privacy and security protocols
Chip	Thermal modeling NoC design and optimization Sensor & Knobs	Tip-Top hw for thermal control NoC power aware design Simulation toolchain (HANDS)

Hot spots and Thermal problems



Chip floorplan

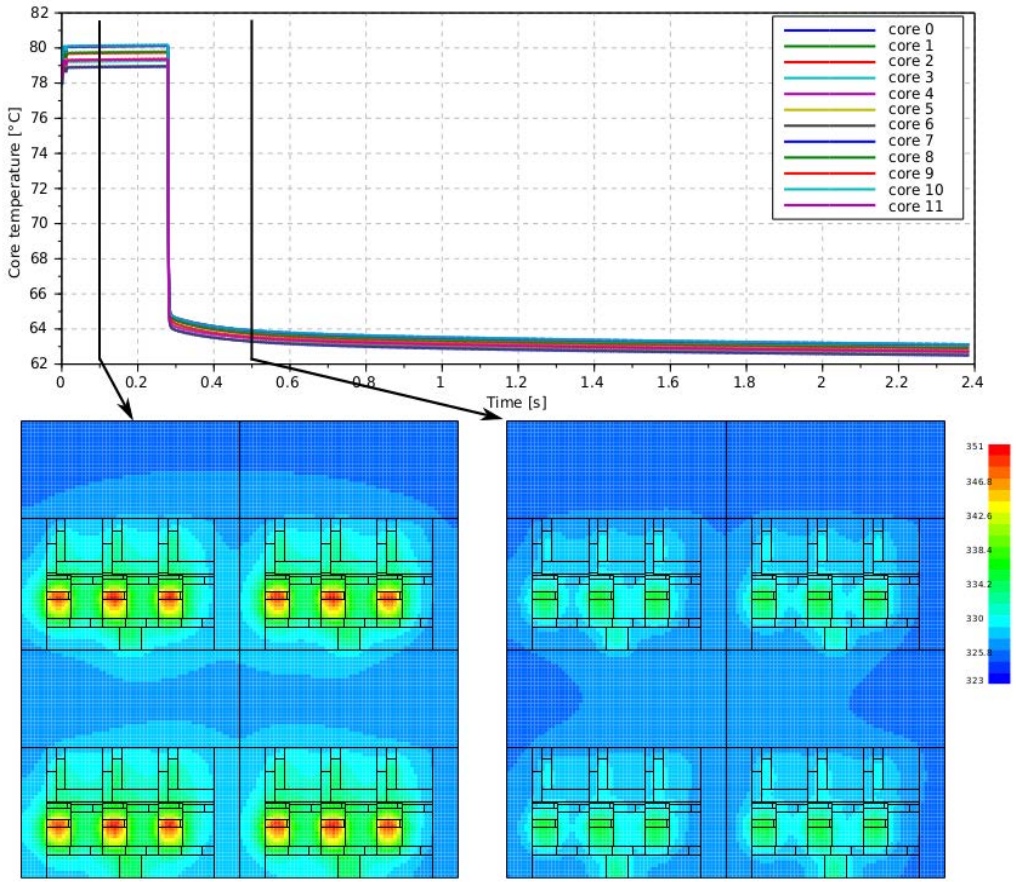


Steady state temperature

Some hot spots in steady state:

- Silicon is a good thermal conductor (only 4x worse than Cu) and temperature gradients are likely to occur on large dies
- Lower power density than on a high performance CPU (lower frequency and less complex HW)

The importance of the thermal transient state



- Thermal transient behavior of a 12-core multi-core considering a frequency step-down from 2GHz to 1GHz at 0.3s of simulation
- Two thermal snapshots are reported to highlight the flexibility of the our flow to compute transient temperature analysis

Thermal dynamic is in the order of 10°C / ms

Steady state analysis is not enough

Dynamic Thermal Management

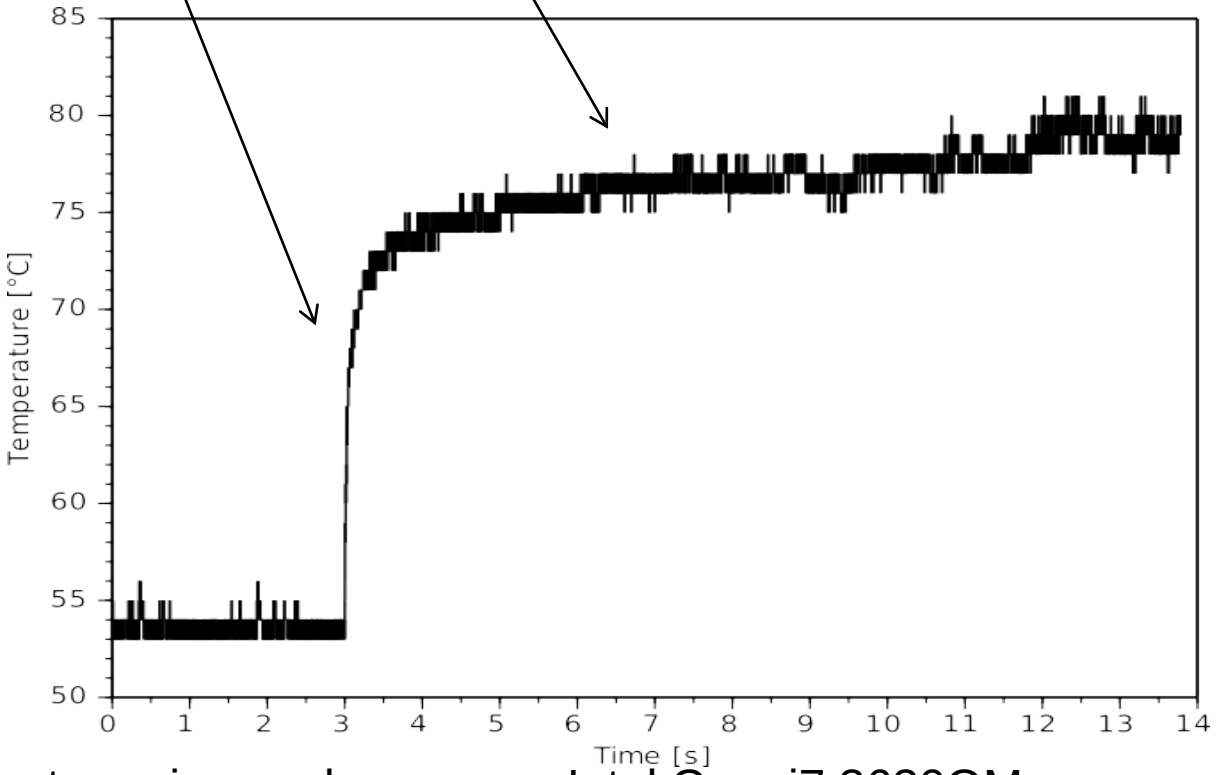
- Design and simulation of an event-based thermal control policy
- Comparison with fixed rate control
- Experiments on Intel-i7

Dynamic Thermal Management (DTM)

- MPSoC power density keeps increasing
  - 3D die stacking will further exacerbate thermal issues
- Temperature needs to be controlled
  - To prevent immediate failures (e.g: thermal runaway)
  - To increase reliability (e.g: electromigration, NBTI, thermal cycles)
- Solution
  - Employ novel dynamic thermal management to maximize performance under temperature constraints

Facing the monster(s)

- **Temperature variation on a chip occur at two timescales**
  - A **fast** one whose time constant (3..30ms) is dictated by the silicon bulk thermal capacity
  - A **slow** one whose time constant (seconds, minutes) depends on the heatsink



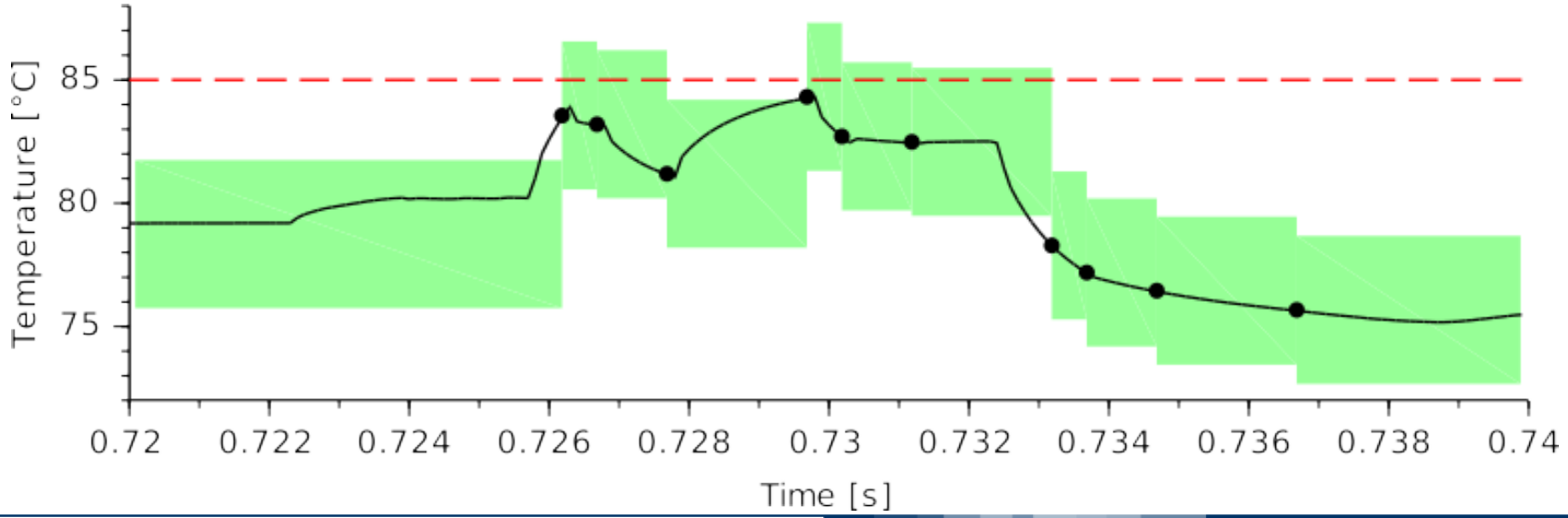
Data: thermal transient running cpuburn on an Intel Core i7 3630QM

Event-based thermal control: rationale

- DTM policies need to be **lightweight** (low overhead)
- **Problem**
  - timescale at which sensing, control and actuation loop needs to be operated has to be **faster than the timescale of temperature** changes
- This timescale is expected to shrink (e.g: 3D chips) requiring **sub-millisecond** control
- Conventional DTM policies operate at a fixed rate, by periodically monitoring the temperature
  - This is inflexible, as the rate needs to be set considering worst-case conditions
- **Solution**
  - Dynamic thermal management using **event-based** control theory

Event-based T control: principle of operation

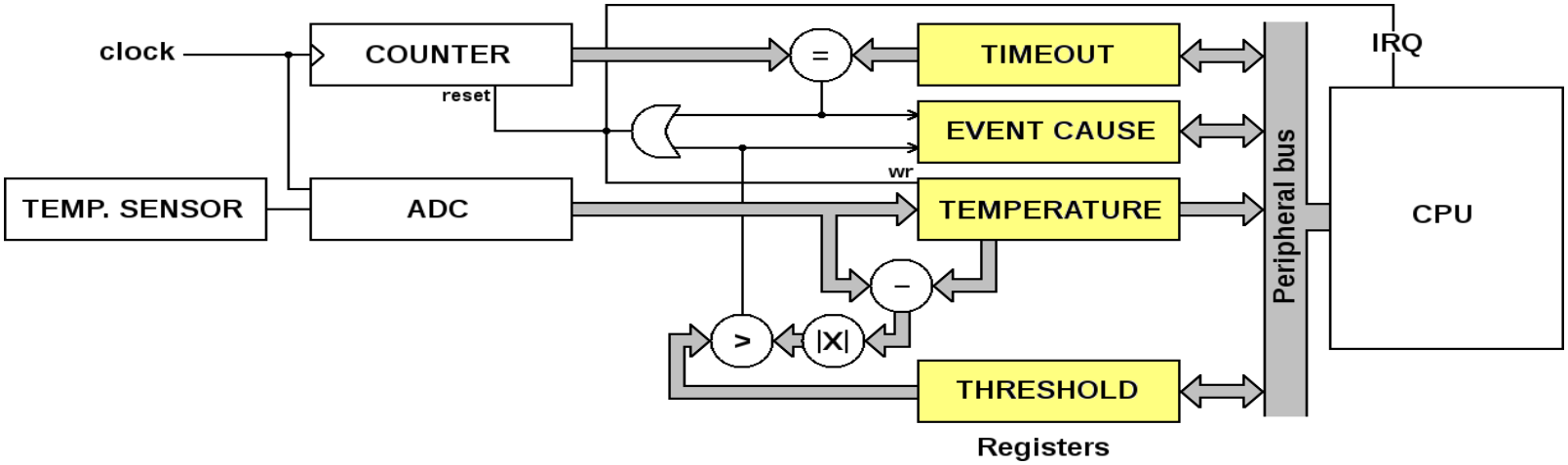
- The software controller configures the event generation state machine to generate an event if:
  - Temperature changes by more than a given threshold from the last time the controller is run (green band)
  - A timeout occurs. Timeouts are progressively increased if temperature changes slowly
- Goal of the controller is keep temperature below a given limit (red line)



Event-based thermal control: architecture

- The proposed solution is based on a hardware-software split
- A hardware state machine monitors the temperature and generates events upon threshold exceeding or timeout
- A software interrupt routine runs the controller, preserving the flexibility of a software DTM policy

$$\begin{cases} x_R(k) = x_R(k-1) + \frac{\mu}{\Gamma} e_T(k-1) \\ u_R(k) = x_R(k) + \frac{\mu}{\Gamma\left(1 - e^{-T_s/\tau}\right)} e_T(k) \end{cases}$$



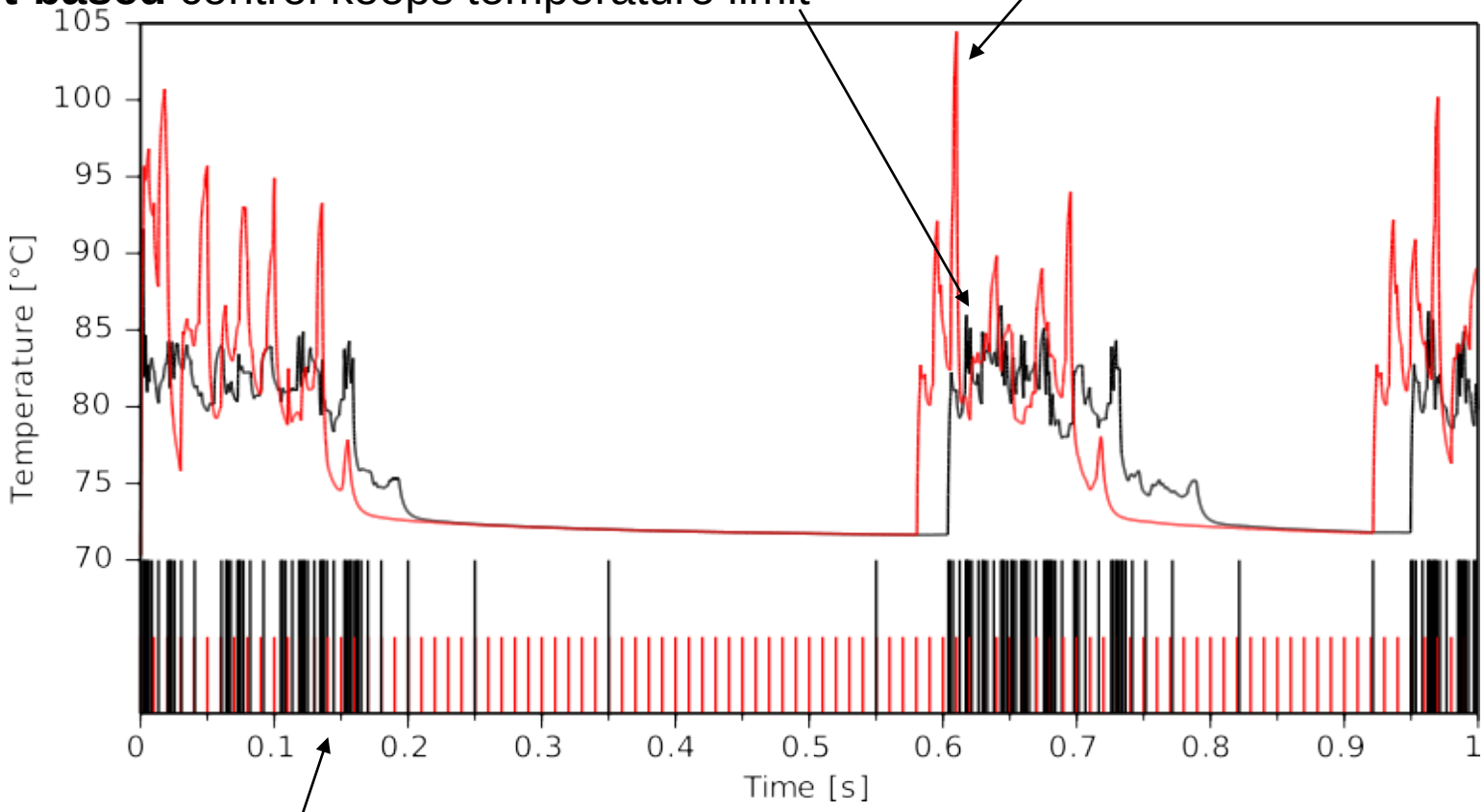
Design and validation using the framework

- The proposed DTM policy was **designed and simulated using the HANDS framework**
- The simulated architecture is a 24-core 3D chip with two layers (12 cores per layer)
- Cores were running the bitcount benchmark from MiBench, with idle times between executions
- The temperature limit was set to 85°C
- Two policies were simulated
  - The proposed event-based thermal controller
  - A fixed rate PID policy at 10ms

Fixed rate vs event-based control

Fixed-rate control cannot prevent fast temperature transients despite running every 10ms

Event-based control keeps temperature limit



Event-based controller generated many events when temperature changes rapidly, and few events when temperature is nearly constant

Experimental validation: setup

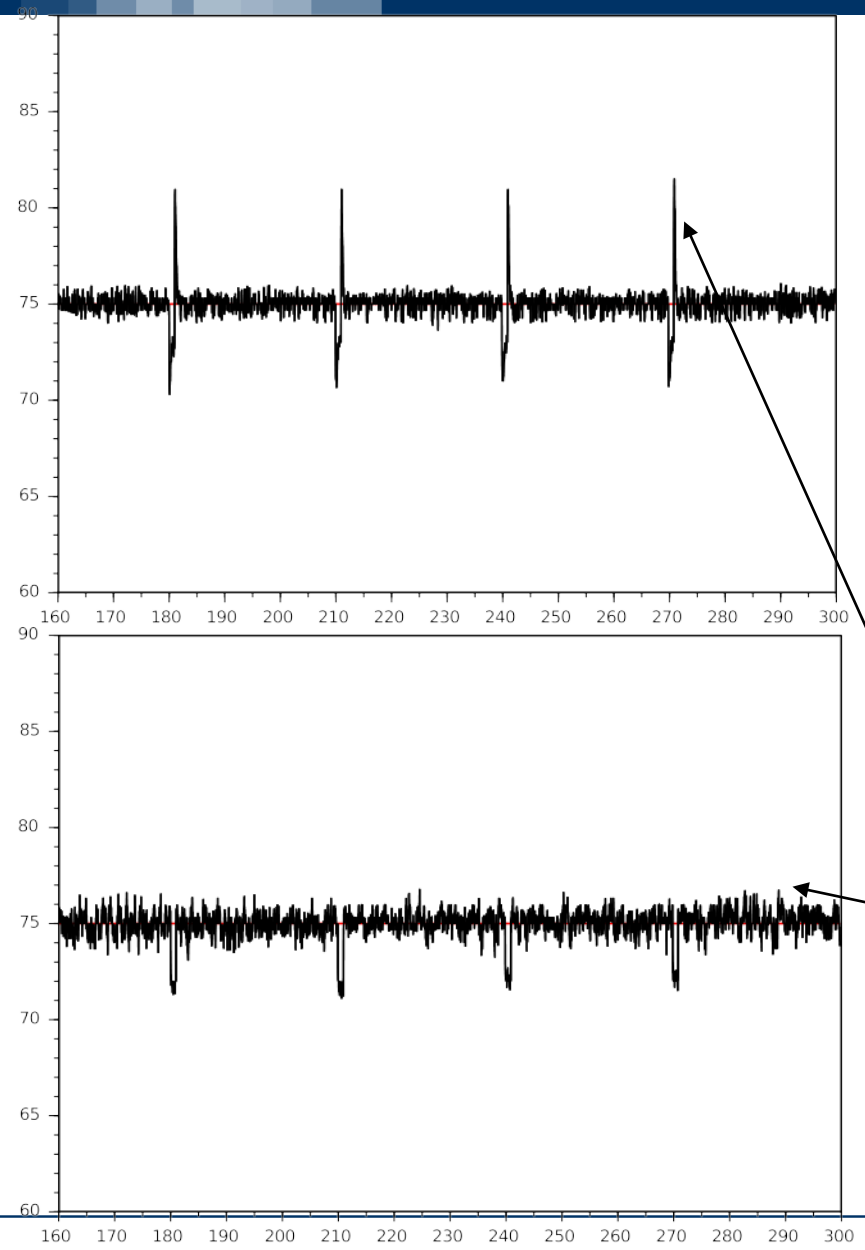
- Implementation on an Intel Core i7 2640 with ubuntu Linux
- FSM of the event-based controller is **software** emulated (implemented)
  - The goal is to show the feasibility
  - Lower overhead is expected with Hw/Sw realization (FSM generating events implemented in hw)
- All kernel modules implementing DVFS policies for power-performance and power capping are disabled
- A daemon in user space implementing the controller uses the msr kernel module of Linux is to read temperature and to drive the DVFS
- Synthetic benchmark alternating intense computing phases with high cache miss phases
- Temperature limit 75°C (not to break the Laptop, just demo!)

Experimental validation and comparisons

- To quantify overhead also for the software controller, the obtained code was benchmarked using RDTSCP [17] instructions
  - It takes on average 39 clock cycles on a Core i7 3630QM processor
  - Considering that the processor operates at 2.4GHz, the time required to run the controller code is 16ns (fully sw implementation)

- Note that the frequency is set in accordance to the actual CPU temperature, thus implicitly accounting for mutual thermal influences between CPUs

Experimental validation



- Tested policies
  - Event-based controller
  - Fixed-rate PID
- Alternating intense computation and high cache misses phase produce a variable power consumption for the CPU
- The fixed rate controller is too slow to counteract the fast thermal transients
- The event based controller keeps temperature below the limit

Exploitable results – PCT application

TIPTOP
Tightly Integration of Power and Temperature for Optimal Performance

Priority date: 15/02/2016

Int. Application number: PCT/IT2016/000037

Assignee: Politecnico di Milano, Milano, Italy

Inventors: Alberto Leva, William Fornaciari, Federico Terraneo

Status: Available

Looking for commercial partners and industrial exploitation

Userspace Run-time Resource Management

The BarbequeRTRM

William Fornaciari, Giuseppe Massari,
Simone Libutti, Federico Reghenzani

BarbequeRTRM Open Source Prj (BOSP)

Overview

- The core of the project is a Run-Time Resource Manager for
- Multi/Many-core Systems (the BarbequeRTRM)
  - *Scheduling*, *resource allocation*, *power management*

What the BarbequeRTRM can do?

- Bound the assignment of CPU quota and/or cores
- Bound the assignment of MEMORY
- Bound the assignment of NETWORK bandwidth
- Power/Energy and Thermal management
- QoS monitoring and resource allocation tuning
- Profiling of the application execution

- What BBQRTRM cannot do?
  - React at the time of ms, best performance around 100ms
  - Proactive vs “a bit” reactive

The BarbequeRTRM

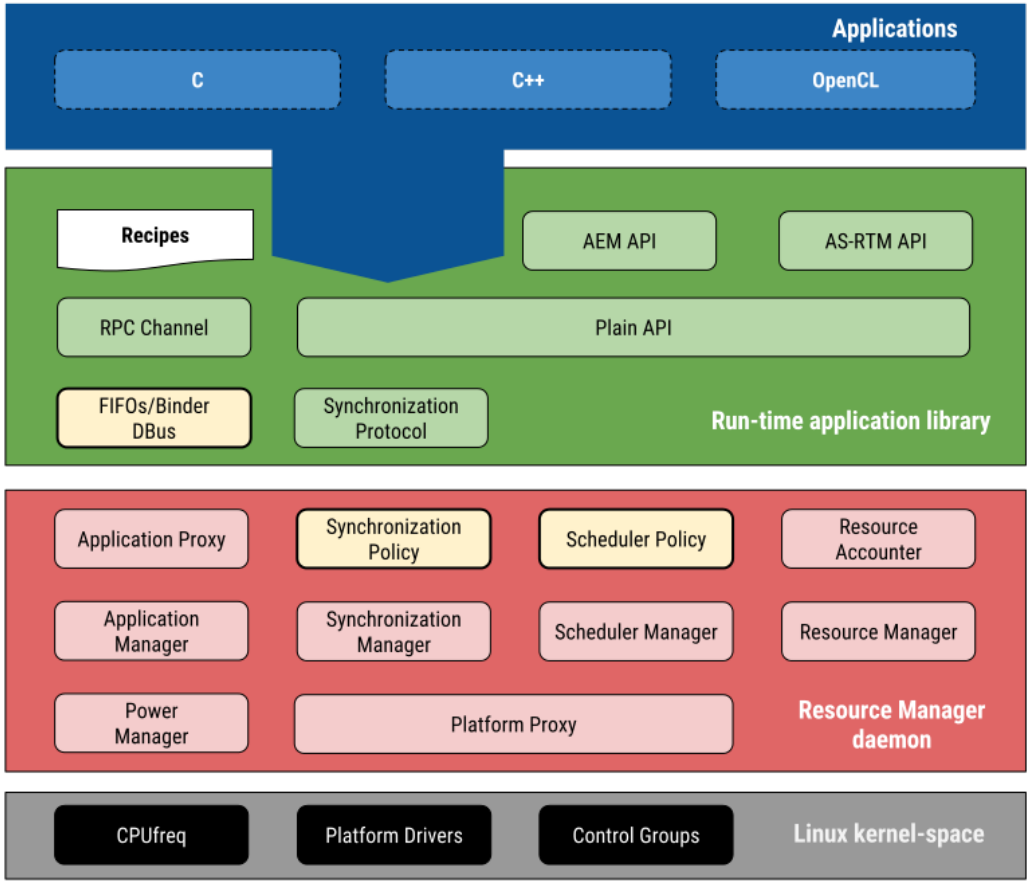
Layer view

C/C++, OpenMP, OpenCL supported

Application library (RTLib) for synchronized execution

The RTRM includes core components and plug-in modules

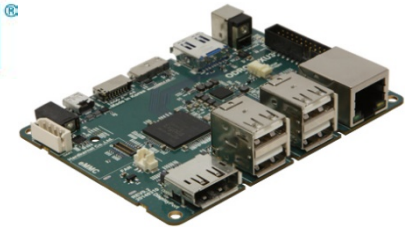
HW support exploiting Linux frameworks or custom drivers and libraries



The BarbequeRTRM

Currently supported hardware systems

- Intel/AMD x86 single multi-core processor systems
  - + Multiple GPUs (AMD) through OpenCL runtime
- Intel/AMD x86 multi-processor NUMA systems
- ARM Cortex A9 multi-core CPU based SoC
  - PandaBoard
  - Freescale iMX6 Quad SABRE
- ARM big.LITTLE 8-core (Cortex A7 + Cortex A15)
- (Samsung Exynos 54xx)
  - Insight Arndale OctaCore
  - ODROID XU-E
  - ODROID XU-3
- *MANGO heterogeneous system*
  - *CPU + Different custom processors*

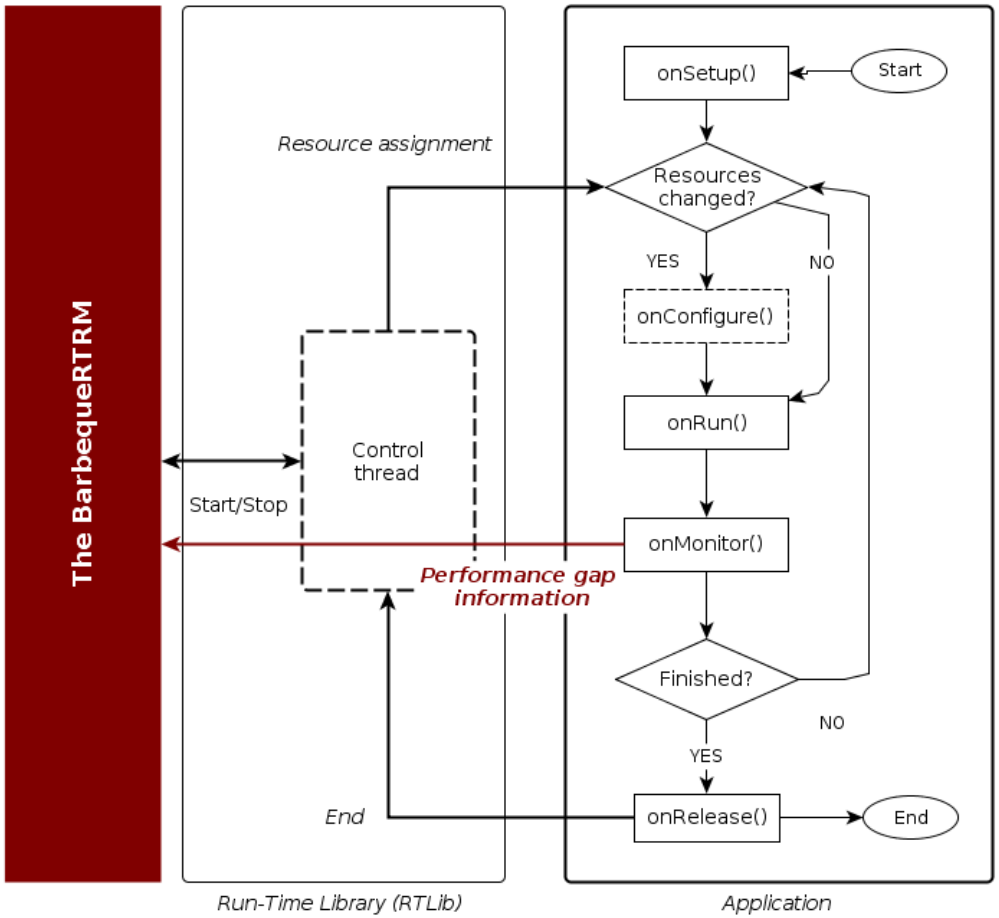


The BarbequeRTRM

Application Execution Model

The application is aware of the amount of resources assigned (#cores, type of processors, etc...)

BarbequeRTRM receives feedback about the curren application performance



Use case from HARPA project

Beesper system by CAMLIN Italy

- Landslide detection and prediction system

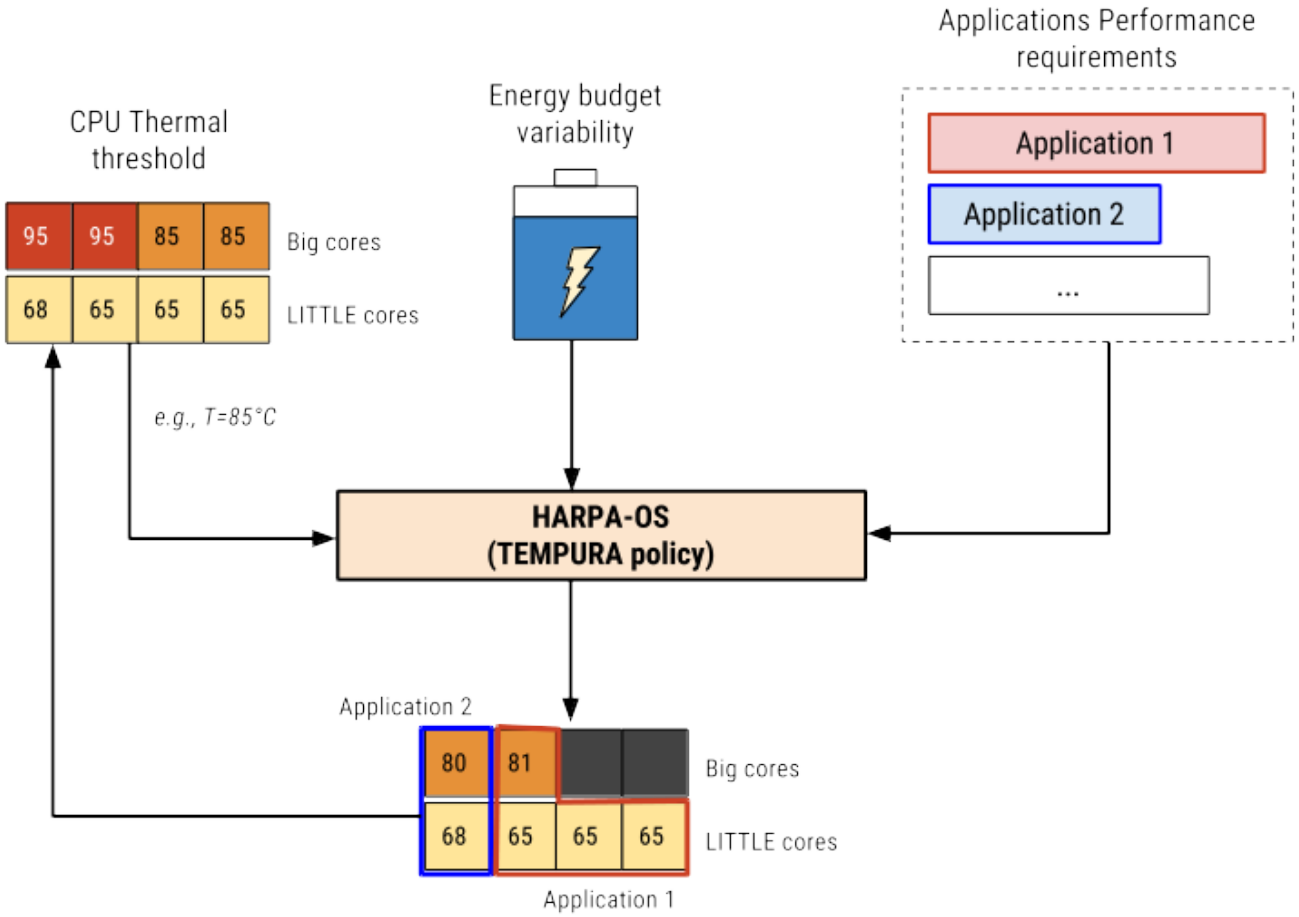
Goal

- The system (solar panel / battery – powered) must remain online for the entire daylight

Use case from HARPA project

Beesper system by CAMLIN Italy

- BarbequeRTRM using *TEMPURA* policy



Use case from HARPA project

Beesper system by CAMLIN Italy

- BarbequeRTRM using *TEMPURA* policy

Experienced results

- Tests performed on the field
- System uptime guaranteed over both summertime and wintertime
- CPU temperature kept under the safety value
- QoS of the machine learning and computer vision applications guaranteed

Use case from HARPA project

System for disaster management support by IT4Innovations (Ostrava, Czech Republic)

- Rainfall-Runoff modeling with flood prediction
  - Several instances in execution
  - Monitoring of water levels in different areas

Use case from HARPA project

System for disaster management support by IT4Innovations (Ostrava, Czech Republic)

- Rainfall-Runoff modeling with flood prediction

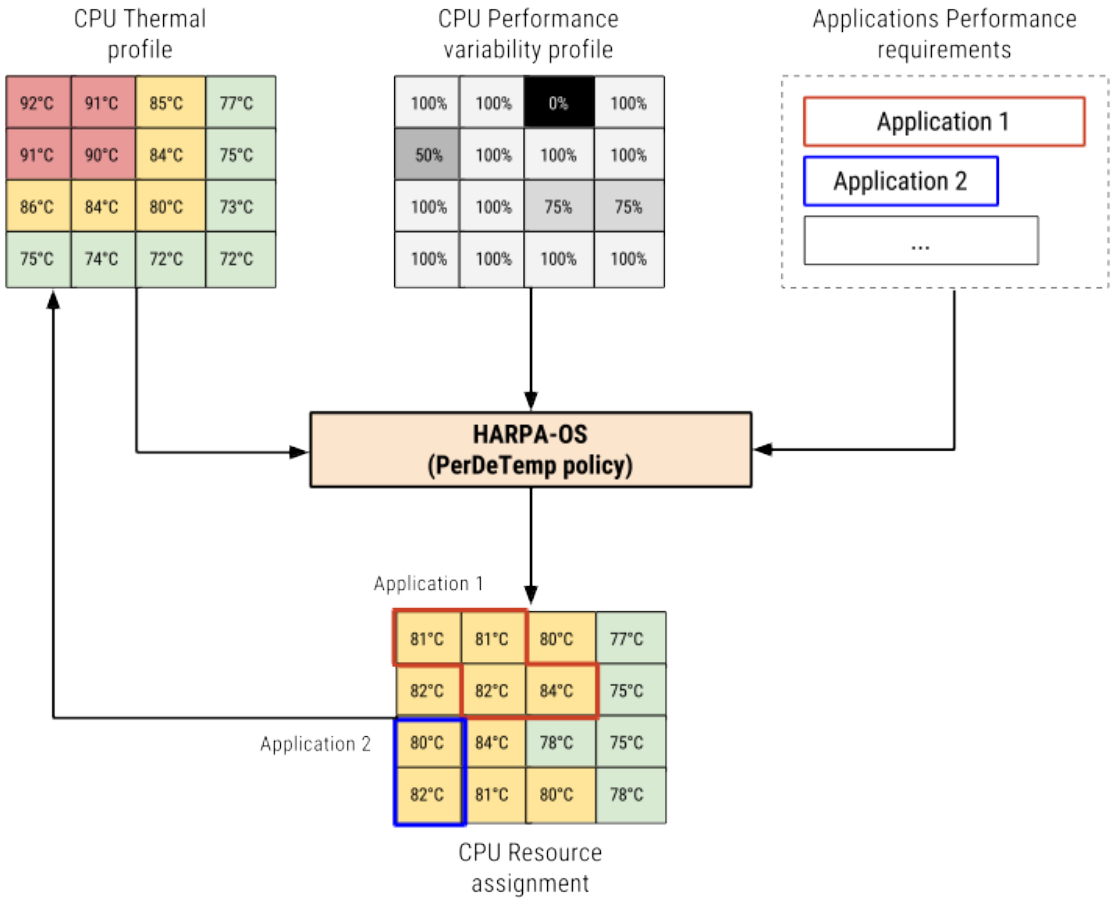
Goals

- Adaptive QoS management
  - According to different alert/warning levels
- Resource consolidation on HPC systems
- Awareness of performance variability due to HW degradation

Use cases

System for disaster management support by IT4Innovations (Ostrava, Czech Republic)

- BarbequeRTRM using *PerDeTemp* policy



Use cases

System for disaster management support by IT4Innovations (Ostrava, Czech Republic)

- BarbequeRTRM using *PerDeTemp* policy

Experienced results

- QoS guarantee (output prediction carried out according to the deadline)
- Peak power consumption reduction (20-45% saving)
- Cooling costs reduction (10-15% estimated saving)
- Improved HW reliability
  - Expected processors lifetime increased (11-47%)

BarbequeRTRM Open Source Prj (BOSP)

EU funded project involving BOSP

- [2010 – 2012] **2PARMA**: PARallel PAradigms and Run-time MAnagement techniques for Many-core Architecture
- [2014 – 2016] **CONTREX**: Design of embedded mixed-criticality CONTRol systems under consideration of EXtra-functional properties (https://contrex.offis.de/home/)
- [2014 – 2016] **HARPA**: Harnessing Performance Variability (http://www.harpa-project.eu/)

- [2015 – 2018] **MANGO**: MANGO: exploring Manycore Architectures for Next-GeneratiOn HPC systems (http://www.mango-project.eu/)

BarbequeRTRM Open Source Prj (BOSP)

On going developments

- Extend the BarbqueRTRM with further resource allocation and power management policies
- Programming and resource management support for extremely heterogeneous systems (MANGO) to finalize
- Android support (already available) to develop for *distributed mobile systems*
- Development for *mixed-criticality* systems

Links and contacts

- Website: http://bosp.dei.polimi.it/
- Mailing lists:
  - User / News: https://groups.google.com/d/forum/bosp
  - Developers : https://groups.google.com/d/forum/bosp-devel

Exploitable results

Run on the market?

- Open discussion with our Technology Transfer Office
  - How far can arrive a University?
  - Product vs consultancy
  - Legal issues and liability
- Perspective application to the Launchpad program (H2020)
- Expected Business Model
  - Open source (free): still existing, with mechanisms and standard policies
  - Customizations (€): for specific platforms, better tuned ad-hoc policies

Concluding remarks

Power (resources) and thermal management

- cross linked
- mix of proactive and reactive solutions
- different timing scale and level of abstraction

Results are promising

- Cannot make only research forever
- How to find a reasonable technology transfer path?

How to achieve a valuable commercial exploitation of a research output can be a session topic for the next IWES