On Dynamic Thermal Conditions in Mixed-Criticality Systems

Seyedmehdi Hosseinimotlagh, Ali Ghahremannezhad and Hyoseung Kim

University of California, Riverside

{shoss007, aghah001, hyoseung}@ucr.edu

UNIVERSITY OF CALIFORNIA, RIVERSIDE

Motivation

- Powerful multi-core embedded systems come at the cost of power consumption, resulting in <u>heat dissipation problems</u>
  - Power consumption increase
  - System reliability reduction
  - Unwanted performance degradation
- **Motivation:** Dynamic ambient temperature affects the operating temperature in multi-core systems significantly
  - In auto-mobiles, cabin air temperature can reach up to 82 °C in Phoenix, Arizona.
  - The computing system of fire-containment drones starts warning when the ambient temperature reaches 35°C and becomes non-operational at 40 °C.
  - For iPhones, maximum fully-operational ambient temperature is 35 °C and partially-operational ambient temperature is 45 °C.
- **Questions:**

1. Up to what ambient temperature is the system fully or partially operational?
2. If the system moves from a hot to a cold region, how long will it take for the system to operate fully operationally?
3. How to capture heat conduction on a modern multi- core processor?

Prior Approach

- Dynamic ambient temperature for non-real-time systems[1,2,3,4]
- Fixed ambient temperature for real-time systems
  - *Hot* and *cold* tasks [5,6]
  - DVFS [5,6,7]
  - Enforcing idle time [8,9]
  - Multi-core processors [10,11,12,13,14]

Uni-processors

No prior work on dynamic ambient temperature in multi-core real-time systems

1- Y. Fu, et al. "Feedback thermal control for real-time systems." *RTAS'10.*
2- Y. Lee, et al. "Thermal-aware resource management for embedded real-time systems." *CADICS'*18.
3- F. Paterna, et al. "Modeling and Mitigation of Extra-SoC Thermal Coupling Effects and Heat Transfer Variations … ." *ICCAD'*15.
4- S. Park, et al. "Dynamic thermal management for networked embedded systems under harsh ambient temperature variation." *ISLPED'10*.
5- S. Zhang, et al. "Thermal aware task sequencing on embedded processors." *DAC'*10.
6- H. Huang, et al. "Throughput maximization for periodic real-time systems under the maximal temperature constraint." *TECS'*14.
7- T. Chantem, el al. "Online work maximization under a peak temperature constraint." *ISLPED'*09.
8- P. Kumar, et al. "Cool shapers: shaping real-time tasks for improved thermal guarantees." *DAC'11.*
9- P. Kumar, et al. "System-level power and timing variability characterization to compute thermal guarantees." *CODES+ ISSS'11.*
10- S. D'souza, et al. "Thermal implications of energy-saving schedulers." *ECRTS'17.*
11- D. Rai, et al. "Worst-case temperature analysis for real-time systems." *DATE'11*
12- L. Schor, et al. "Worst-case temperature guarantees for real-time applications on multi-core systems." *RTAS'*12.
13- S. Pagani, et al. "MatEx: Efficient transient and peak temperature computation for compact thermal models." *DATE'*15.
14- R. Ahmed, et al. "On the design and application of thermal isolation servers." *TECS'*17.

State-of-the-art

- Thermal-aware server approach on multi-core embedded systems.†
  - All sporadic tasks execute within servers
  - Determine the budget and period of servers to bound the peak maximum temperature

  - Benefits:
    - Thermal-aware servers provide **thermal isolation** by construction
      - Thermal budgets of servers are time and space composable
  - Limitations:
    - Fixed ambient temperature
      - Does not consider CPU budget availability as ambient temperature changes
    - Does not support various server budget replenishment policies
    - Assumes fixed server schedule (no preemption among servers)
      - Does not comply with recent server use cases (e.g., RT-Xen)

†R. Ahmed, et al. "On the design and application of thermal isolation servers." *TECS*'17

Challenge

- What if CPU frequency scaling, enforced fixed CPU idling or active/passive cooling packages are not enough under harsh ambient temperature?

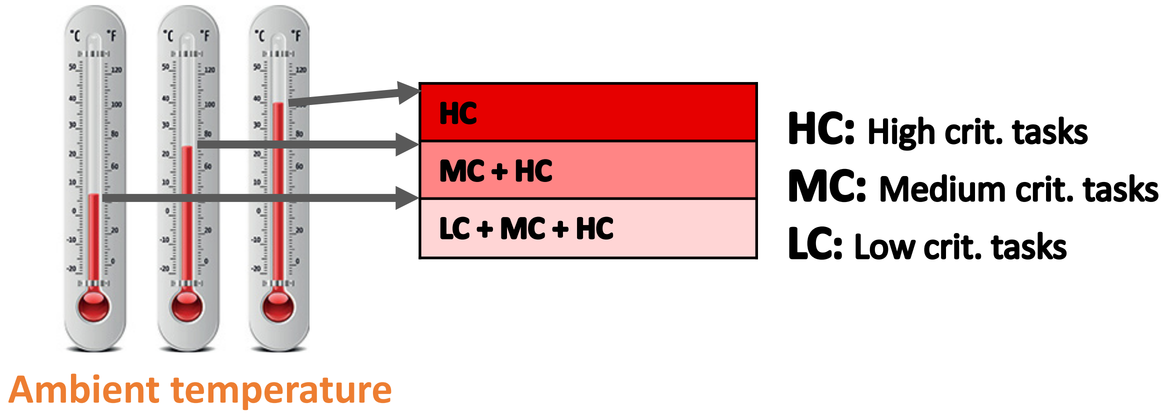
- **Answer:** suspending less critical tasks to secure cooling time in high ambient temperature

- **Thermally mixed-criticality systems** are the systems that assure ambient temperature changes and heat dissipation from lower-criticality task execution do not adversely affect the real-time schedulability of higher-criticality tasks.
  - Different from the well-known Vestal model† (varying assurance of execution time)

†Vestal, Steve. "Preemptive scheduling of multi-criticality systems with varying degrees of execution time assurance." *RTSS'07*

Contributions

- **Ambient temperature-aware framework**
  - **Criticality mode change is triggered by ambient temperature**



- Analyze the thermal safety of a multi-core system and bound the maximum operating temperature
  - Determine the **minimum time** for the system to transition from one criticality level to a **lower level**
  - Introduce the notion of **idle servers** that allow bounding the maximum operating temperature caused by multiple preemptive servers
- Perform a case study on mixed-criticality applications running on an ODROID-XU4 embedded platform

Outline

- Introduction and motivation
- On dynamic thermal conditions in mixed-criticality systems
  - System model
  - Framework design
  - Thermal model
  - Multiple server analysis
- Evaluation
- Conclusion

System Model

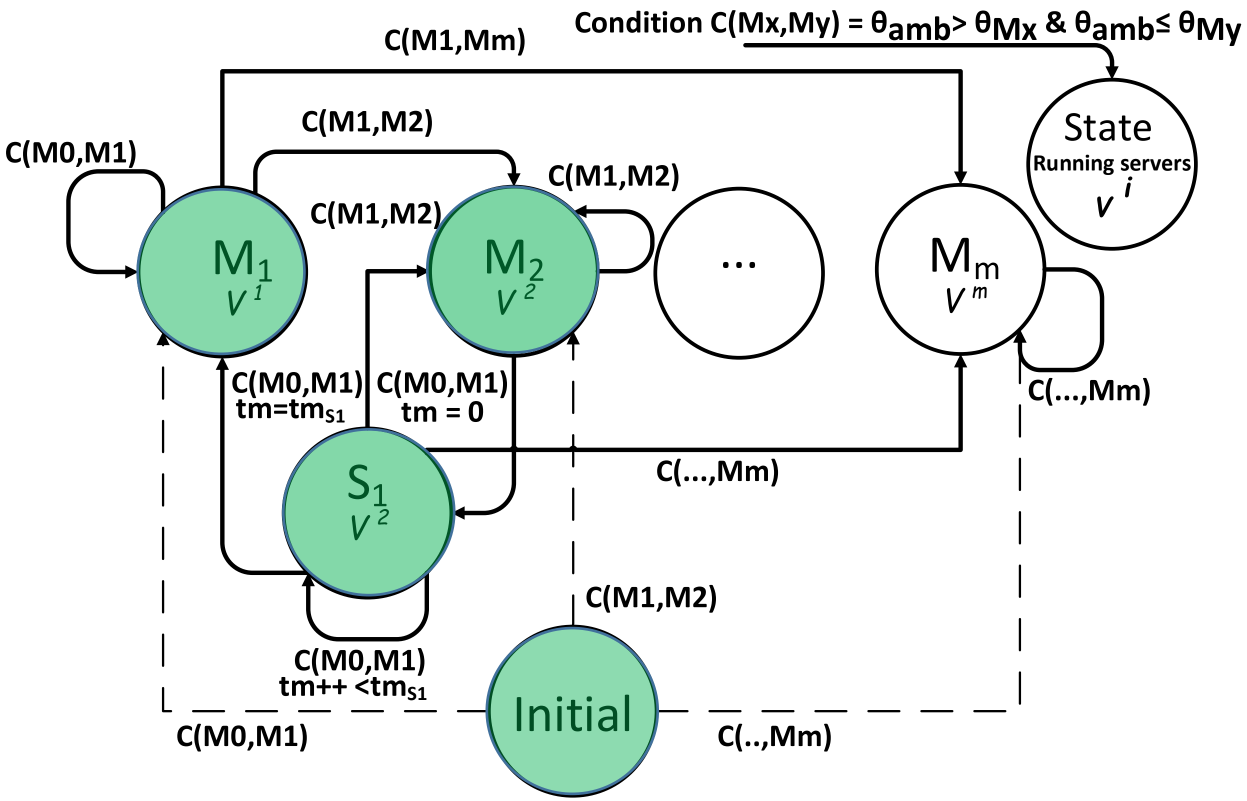
- **Servers:**
  - Multiple preemptive thermal-aware servers for each CPU core
  - Each server has its **criticality level**

- **Tasks:**
  - **Statically** allocated to one thermal-aware server
  - Non-real-time tasks running with **the lowest priority level** in the server
    - Execute only if there is no real-time task ready to execute and the server has remaining budget

- **Criticality Model:**
  - m criticality levels
  - At criticality mode l, only the servers with criticality level higher than or equal to l are eligible to execute.

Framework Design (I)

- The **critical ambient temperature** of the criticality level l (*i.e.,* θ_{M_l}): the maximum ambient temperature that the system can execute eligible servers without violating the system's maximum temperature constraint.



Condition:	**Action:**
$\theta_{amb} \leq \theta_{M_1}$	$Initial \Rightarrow M_1$
$\theta_{M_1} < \theta_{amb} \leq \theta_{M_2}$	$M_1 \Rightarrow M_2$
$\theta_{amb} \leq \theta_{M_1}$	$M_2 \Rightarrow S_1$
$\theta_{amb} \leq \theta_{M_1}$	$S_1 \Rightarrow M_1$
$\theta_{M_1} < \theta_{amb} \leq \theta_{M_2}$	$S_1 \Rightarrow M_2$

Framework Design (II)

Design-time analysis

- Check the **timing schedulability** of tasks
- Find the parameters of thermal-aware servers that ensure **thermal schedulability** for each criticality
- Compute the corresponding **critical ambient temperature** for each criticality
- Compute the **shifting time** from each criticality level to its immediate lower one

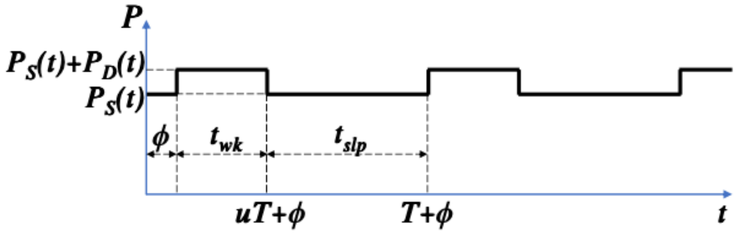
Run-time support

- Monitor ambient temperature
- Criticality mode change (state diagram)

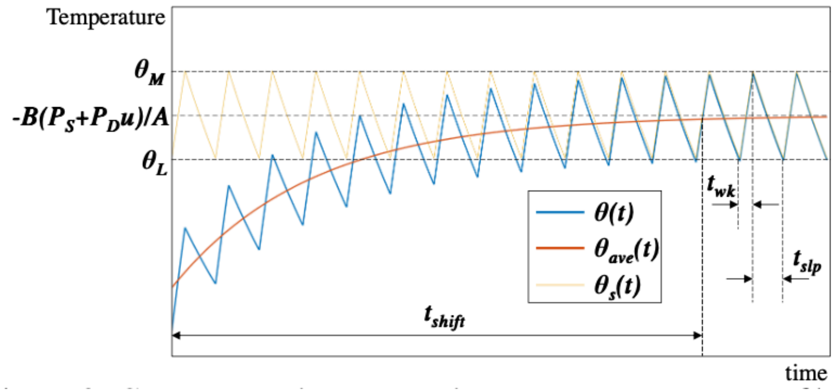
Timing schedulability of real-time tasks refers to the ability to complete their execution by the deadline
Thermal schedulability is to guarantee that under any task execution patterns, the CPU does not exceed the maximum temperature constraint.

Thermal Model

- Power signal
  - P_D: dynamic power, P_S: static power



- Periodic workload
  - T: period, t_{wk}: waking time, t_{slp}: sleeping time, Φ: delay, u: utilization



- Temperature model

Thermal parameters

$$\frac{d\theta(t)}{dt} = A\theta(t) + BP(t)$$

$$\theta(t) = \theta_0 e^{A(t-t_0)} - \frac{B}{A}(P_S + P_D)\left(1 - e^{A(t-t_0)}\right)$$

- Question1: Up to what ambient temperature is the system fully or partially operational?
  - Answer: Maximum ambient temperature support

Thermal constraint

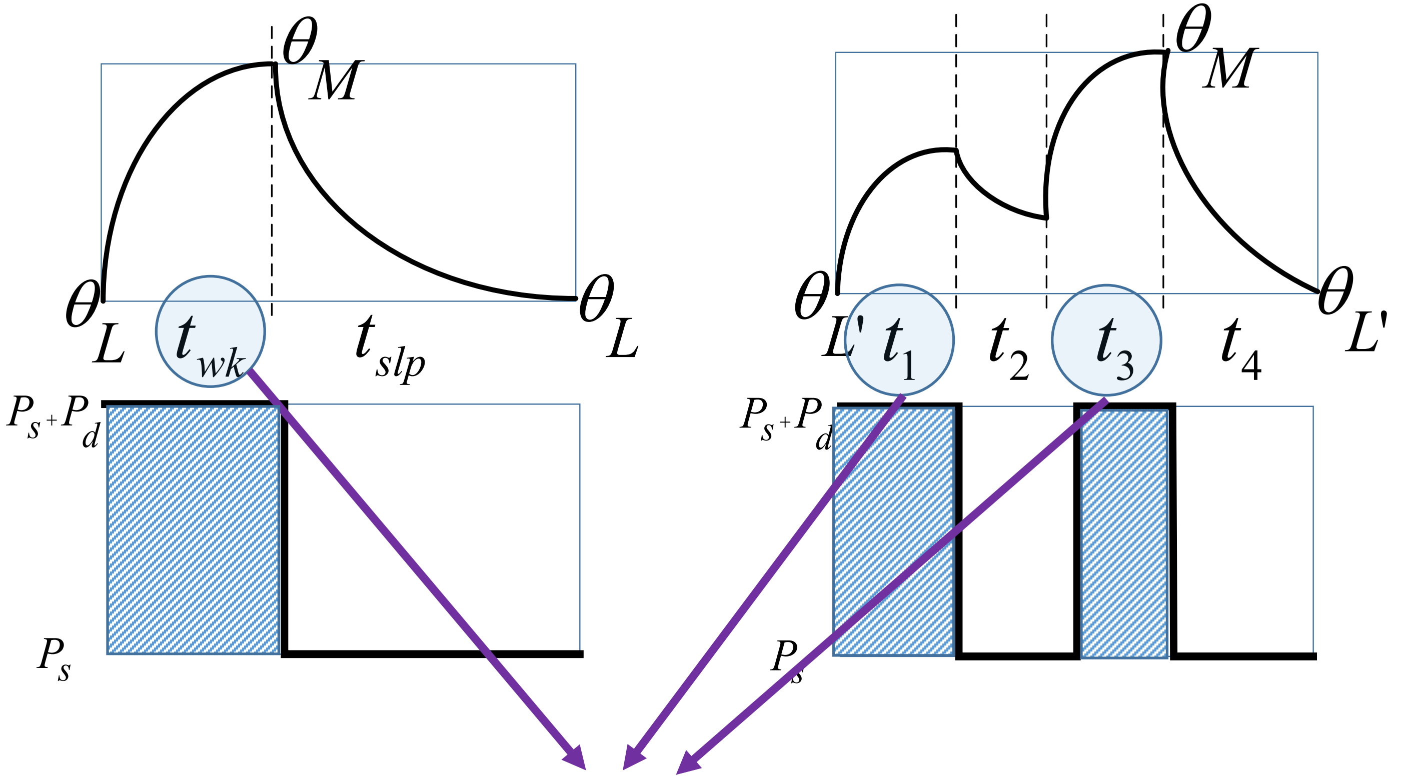
$$\Theta_{amb} = \Theta_M + \frac{B}{A}\left(P_S + P_D \frac{1 - e^{AuT}}{1 - e^{AT}}\right)$$

- Question2: If the system moves from a hot to a cold region, how long will it take for the system to operate fully operationally?
  - Answer: Shifting time

$$t_{shift} = \frac{1}{A}\ln\left(\left|\frac{0.01\frac{B}{A}(P_S + P_D u)}{\theta_0 + \frac{B}{A}(P_S + P_D u) + BS_k}\right|\right)$$

Worst Task Execution Pattern (Uni-Processor)

- **Theorem 1 :** The amount of waking time to reach the maximum temperature constraint is minimized when all workloads execute consecutively.



$t_w < t_1 + t_3$

Server's replenishment budget

$$t_{wk} <= \frac{1}{\beta} \ln \frac{\theta_M(e^{\beta T} - 1) + \alpha}{\alpha}$$

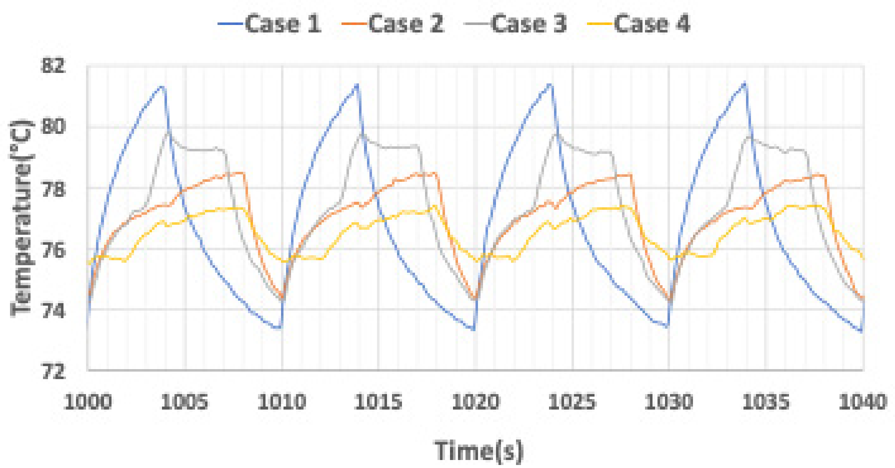
Thermal parameters

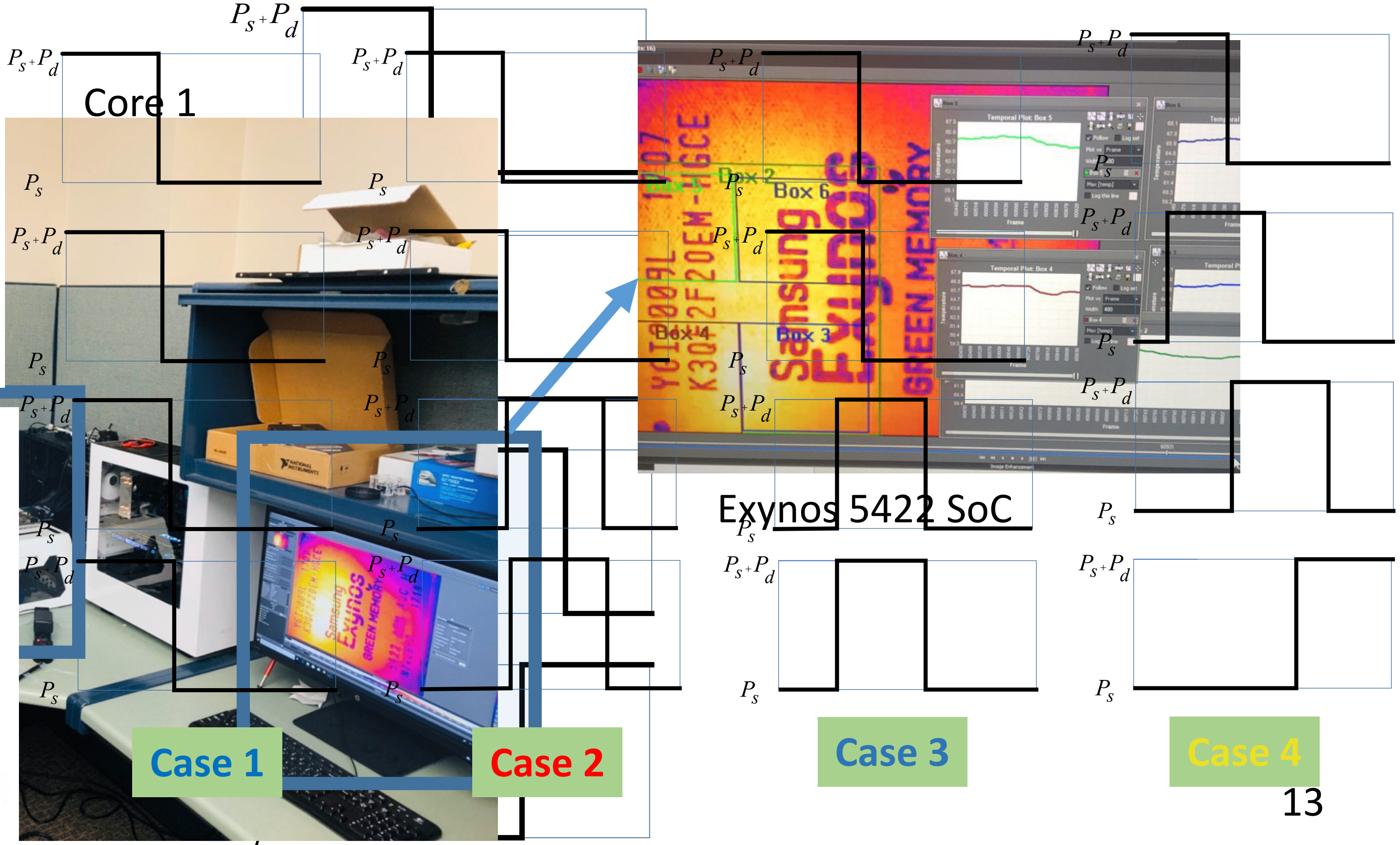
Worst Task Execution Pattern (Multi-Core)

- **Theorem 2:** The minimum value of waking time for a maximum temperature constraint is when the server on each core exhausts all its budget at once, simultaneously

Table I: Descriptions of workload executions on big cores.

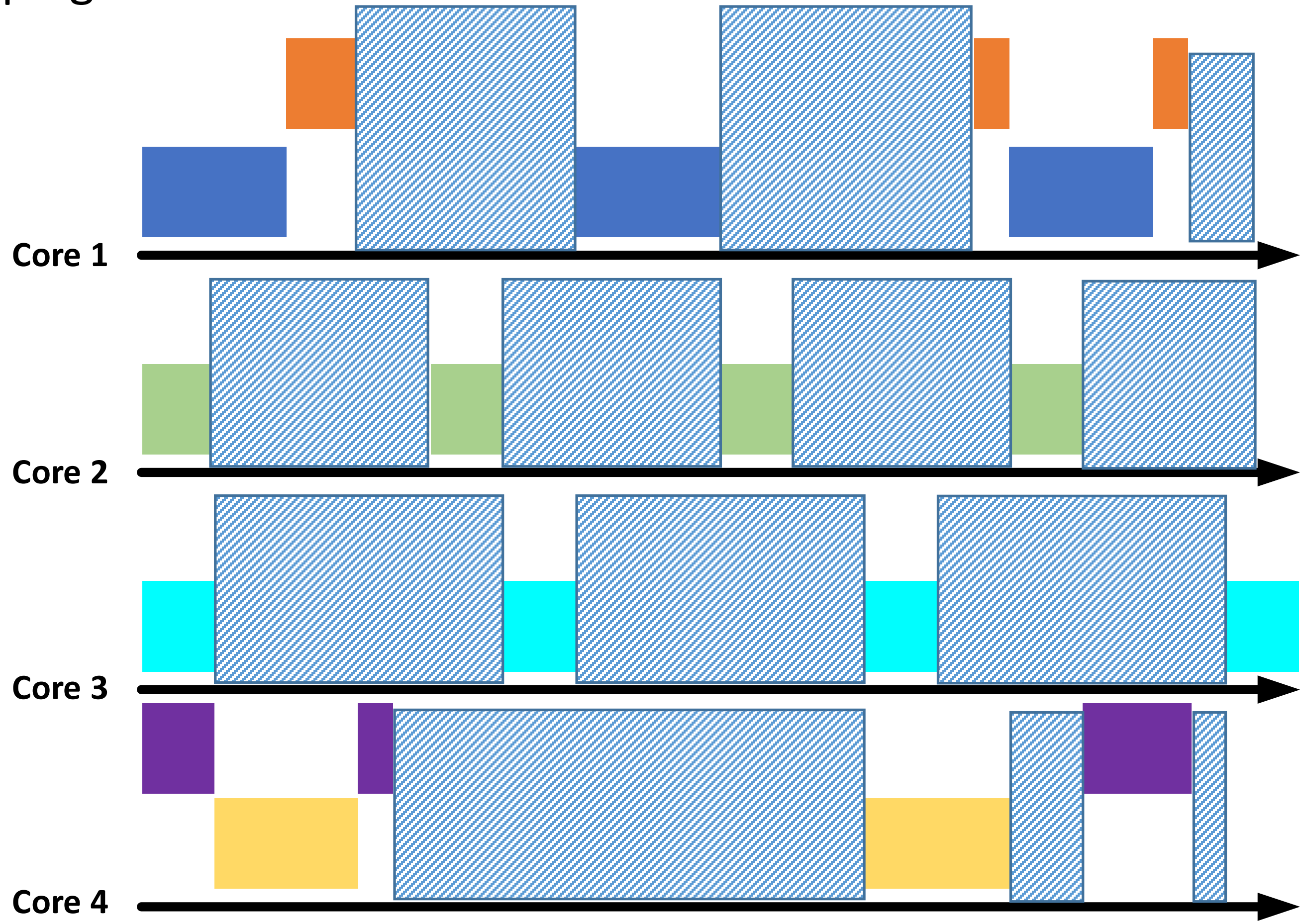
Name	description	workloads delay settings (seconds)
case 1	all workloads execute at the same time	$\phi(1) = \phi(2) = 0$ $\phi(3) = \phi(4) = 0$
case 2	workloads on two cores begin after those of other cores	$\phi(1) = \phi(2) = 0$ $\phi(3) = \phi(4) = 4$
case 3	workloads on two cores begin 1 seconds before finishing those of on other cores	$\phi(1) = \phi(2) = 0$ $\phi(3) = \phi(4) = 3$
case 4	workloads on each core execute with a 2-second overlap	$\phi(1) = 0$ $\phi(2) = 2$ $\phi(3) = 4$ $\phi(4) = 6$

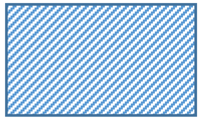




Multiple Server Analysis (I)

Assumption : servers are given
Problem: schedulable without thermal violation?
Constraint: no periodic enforced sleeping time



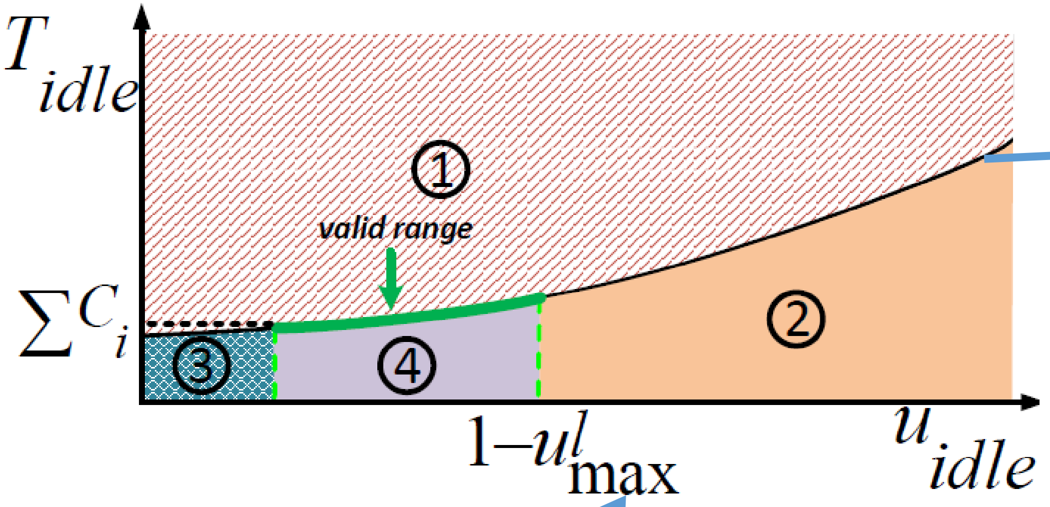


Sleeping time slot of the core

Multiple Server Analysis (II)

- **Intuition:** Modeling multiple servers using the notion of **idle server**:
  - Does not actually exist on the CPU cores
  - Its budget represents the amount of time that the CPU core needs to be idle in the cooling phase.
  - Lowest-priority server (no enforced sleeping; no delay to regular servers)

- **Optimal server setting range**



$$T_{idle} = \frac{\ln\left(1+\alpha \frac{e^{1-u_{idle}+\frac{1}{2\beta}}-1}{\theta_M}\right)}{2\beta}$$

Maximum CPU utilization of idle server at the critical level l

Evaluation

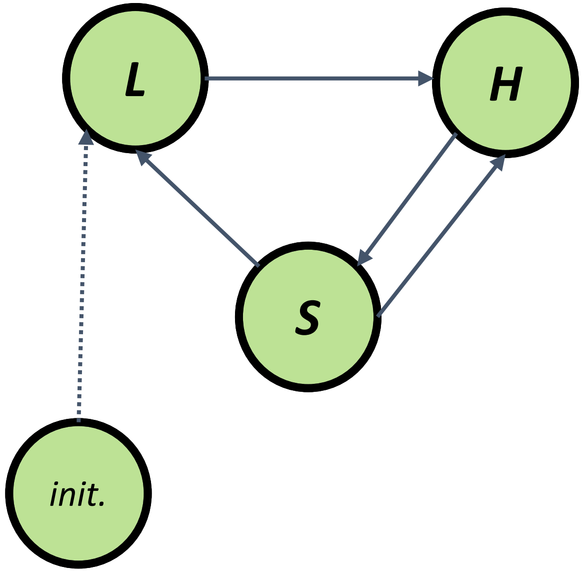
- Platform ODROID-XU4
  - Samsung Exynos5422 SoC
  - 4 Cortex-A15 cores (big cluster)
  - 4 Cortex-A7 cores (little cluster)
    - Used for only system maintenance & monitoring processes, etc.
  - Nordic Semiconductor Thingy:52™IoT sensor developme kit to capture the ambient temperature

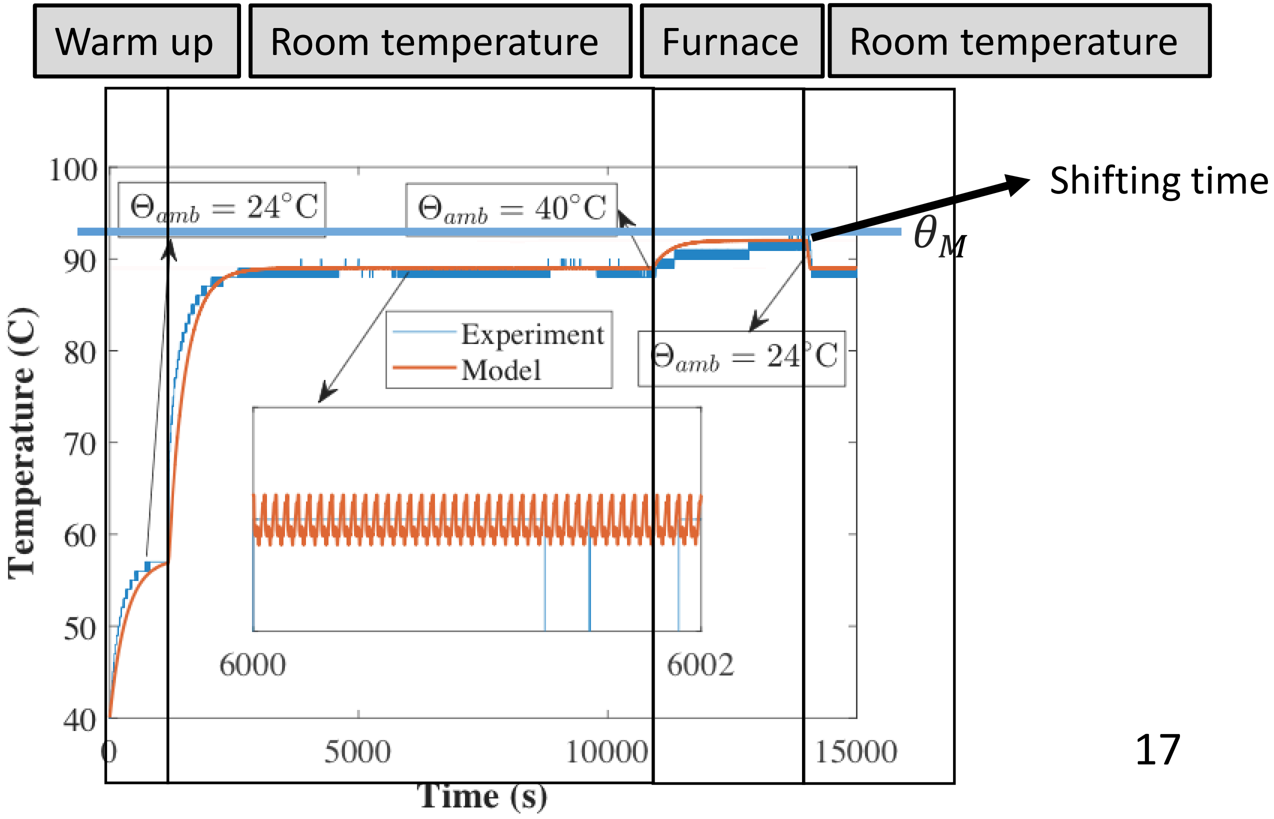
- Benchmark: Flight Management System (FMS) application
- Server’s budget replenishment policy: polling
- Scheduling: Rate Monotonic (RM)
- Tasks assignment: worst-fit decreasing (WFD)

Purpose	CL	Count	Period (ms)	Exec.Time (ms)
Sensor data acquisition	HI	5	200	10
Localization	HI	3	200	10
	HI	3	1000	50
	HI	1	5000	50
Flight-plan management	HI	4	1000	50
	LO	4	1000	50
Flight-plan computation	HI	2	1000	50
	HI	1	5000	750
	HI	1	5000	180
	HI	1	5000	150
	HI	1	5000	90
	HI	1	5000	75
Guidance	HI	1	200	10
Nearest Airport	LO	1	1000	50

Case study

- Server setting (budget, period):
  - Low-criticality server: (27, 50) *ms*
  - High-criticality server: (15, 50) *ms*
- Critical ambient temperatures:
  - Low-criticality ambient temperature: 24°C
  - High-criticality ambient temperature: 40°C





Conclusions

- Mixed-criticality thermal-aware server framework
  - Effective in bounding the maximum temperature at every criticality level
  - Fully preemptive and priority-based server/task scheduling
  - Analyzing the amount of slack between execution with the notion of idle servers
  - Directly applicable to any task models with critical sections under hierarchical scheduling

- Future directions
  - Analysis to allow different idle server settings per CPU core
  - Study on the thermal behavior of tasks to alleviate the pessimism of thermal schedulability
    - Tasks have different thermal footprints (memory-intensive *vs.* computationally-intensive)

Thank you!

On Dynamic Thermal Conditions in Mixed-Criticality Systems

Seyedmehdi Hosseinimotlagh, Ali Ghahremannezhad and Hyoseung Kim

University of California, Riverside

{shoss007, aghah001, hyoseung}@ucr.edu

UCR
UNIVERSITY OF CALIFORNIA, RIVERSIDE