A Framework for Thermal and Performance Management

Davide B. Bartolini, Filippo Sironi, Martina Maggio, Riccardo Cattaneo, Donatella Sciuto, Marco D. Santambrogio
Politecnico di Milano, Lund University

{**bartolini**, sironi, cattaneo, sciuto, santambrogio}**@elet.polimi.it**,
martina.maggio@control.lth.se

Context

- In server farms, power costs account for up to 80% of TCO [1]
  - 33% of TCO just to operate the cooling infrastructure
- Processors are one of the most power-hungry and hot components for plenty of server workloads

- Chip Multiprocessors (CMPs) are pervasive
- Also CMPs are crashing into the power wall (e.g., dark silicon [2]), power density is increasing and we need to exhaust the heat

- Keeping processors cool is crucial [3] (high working temperatures lead to reduced MTTF and higher leakage power)

- Traditional Dynamic Thermal Management (DTM) techniques used for emergency situations, not for normal runtime

---

[1] U. Hoelzle et al. *The Datacenter as a Computer […]*.Morgan and Claypool Publishers, 2009.

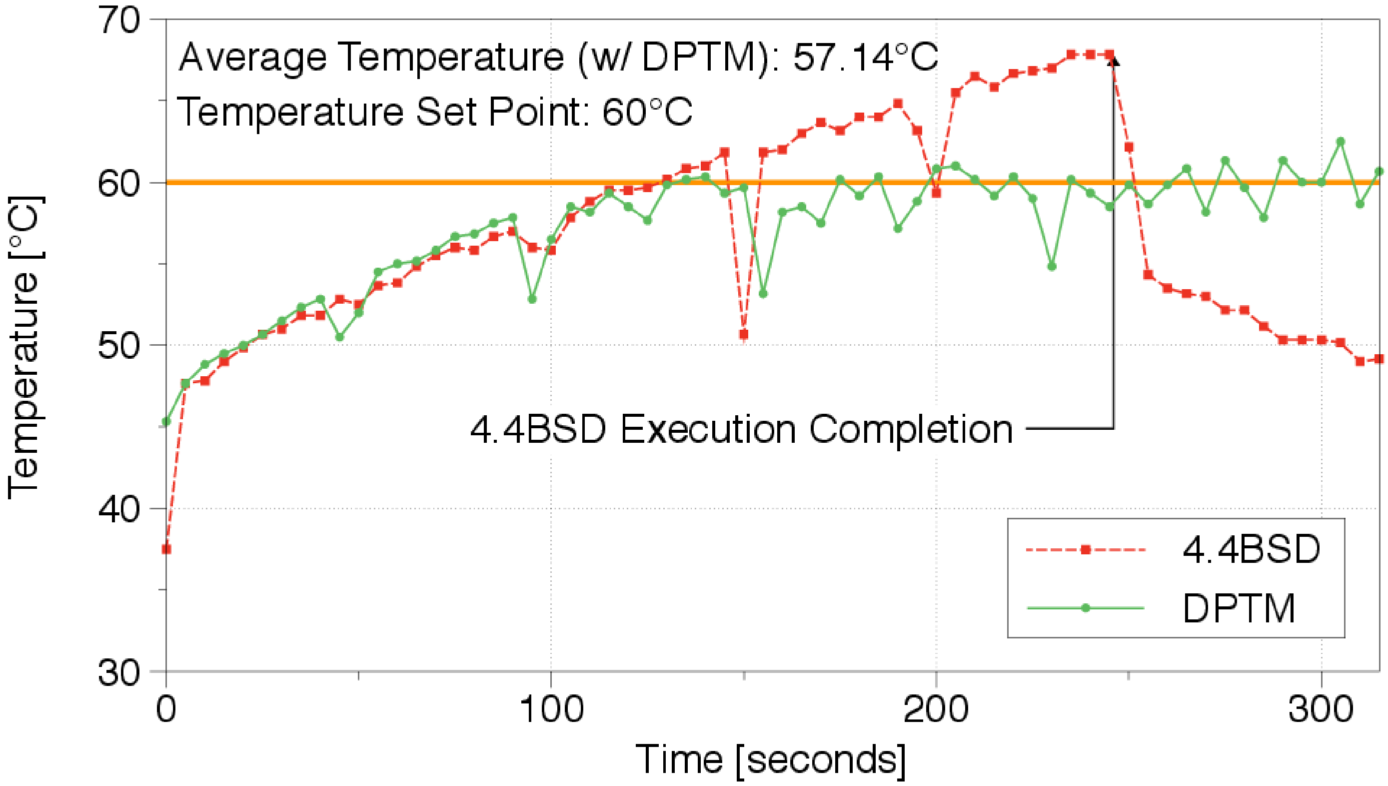
[2] H. Esmaeilzadeh, et al. *Dark Silicon and the End of Multicore Scaling*. In Proc. ISCA 2011.

[3] J. Srinivasan, et al. *The Case for Lifetime Reliability-Aware Microprocessors*. In Proc. ISCA 2004.

Rationale

Common approach in commodity processor scheduling: run to idle

- energy efficient, but leads to peaks in power draw and temperature [4]



Reducing performance can keep temperature under control

---

[4] M. Garrett. *Powering Down*. ACM Queue, 5(7), 2007.

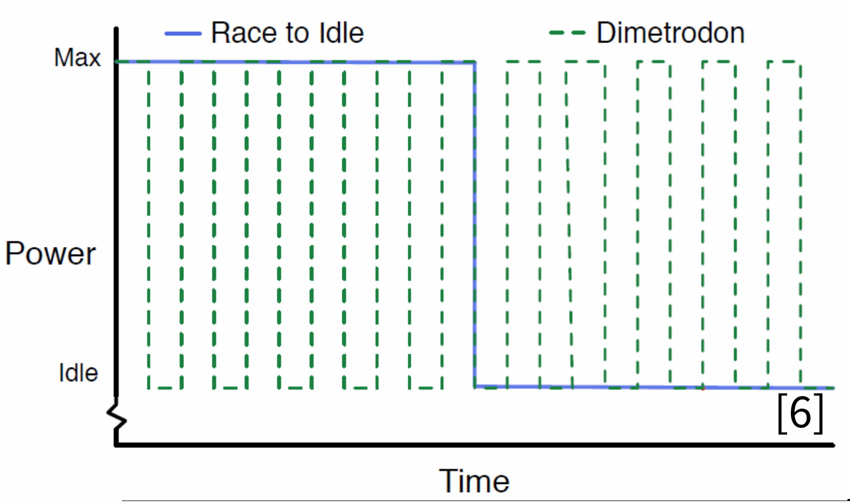
State of the Art

Commonly used techniques for DTM:

- Dynamic Frequency and Voltage Scaling (DVFS) [5]
- Idle-cycle injection [6]

The *Dimetrodon* framework [6] exploited the idea of Preventive Thermal Management (PTM) for long-term thermal management through idle-cycle injection

- Reduce average power draw by injecting idle-cycles with a certain probability, resulting in cooler (but longer) execution



- Open-loop control (no temperature set point)
- Performance traded for cooler execution (but can we afford it?)

---

[5] N. Gupta and R. Mahapatra. *Temperature Aware Energy Management for Real-Time Scheduling*. In 12th ISQED, 2011.
[6] P. Bailis et al. Dimetrodon: *Processor-level Preventive Thermal Management via Idle Cycle Injection*. In Proc. DAC 2011.

Methodology – Key Ideas

Use closed-loop control to drive idle-cycle injection, triggering low power mode (C-states) and reduce temperature

- Users specify a temperature set-point
- A controller decides how much idle time is needed

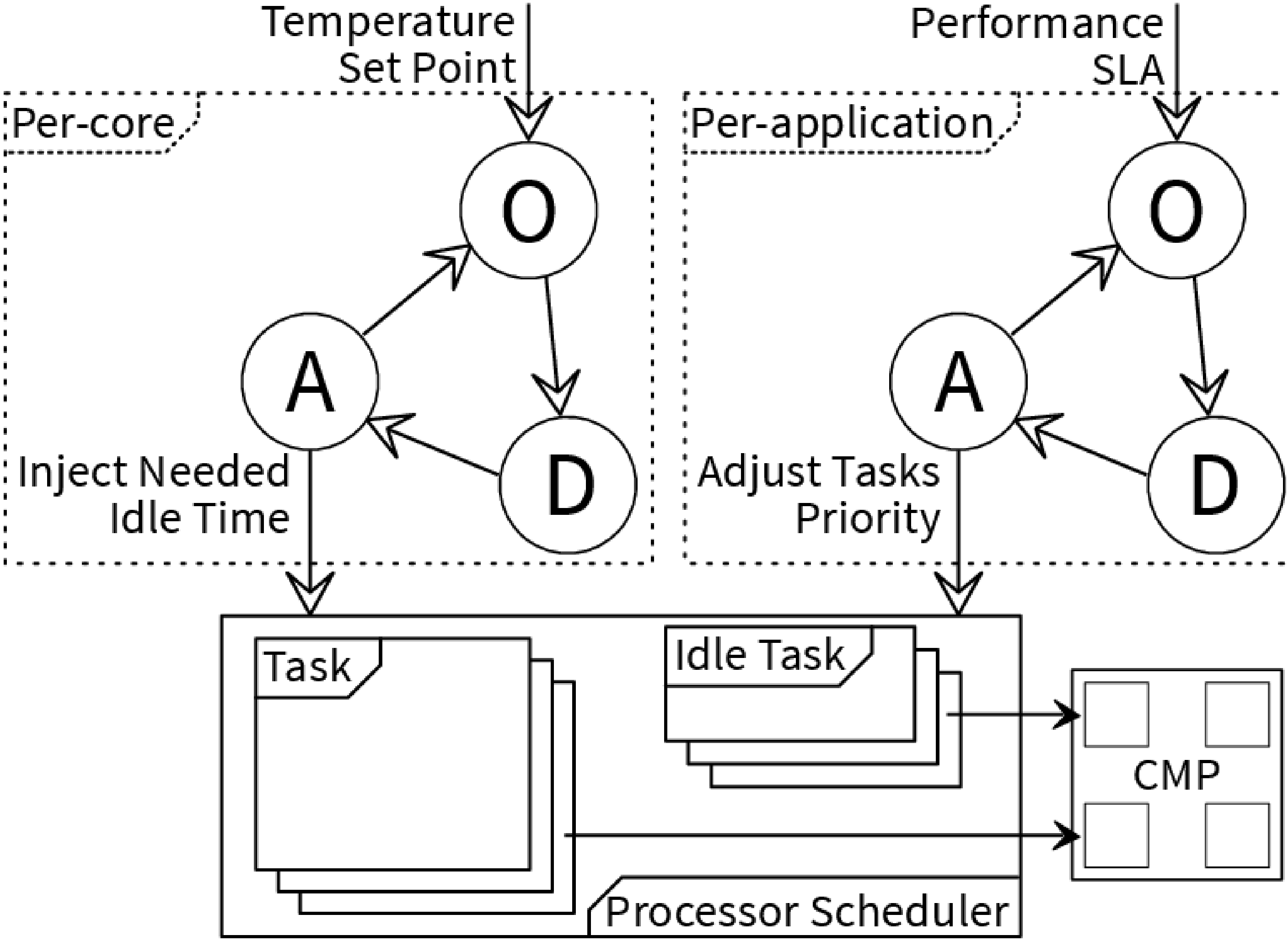
Also account for performance and Service-Level Agreements (SLAs)

- Selectively charge SLA-bound tasks for the idle time, so as to avoid breaking contracts
- Drive tasks' priorities to meet QoS requirements

Coordinate thermal and performance control

Methodology – Overview

Observe-Decide-Act (ODA) control loops [7] for closed-loop control



---

[7] F. Sironi et al. *Metronome: operating system level performance management via self-adaptive computing.* In Proc. DAC 2012.

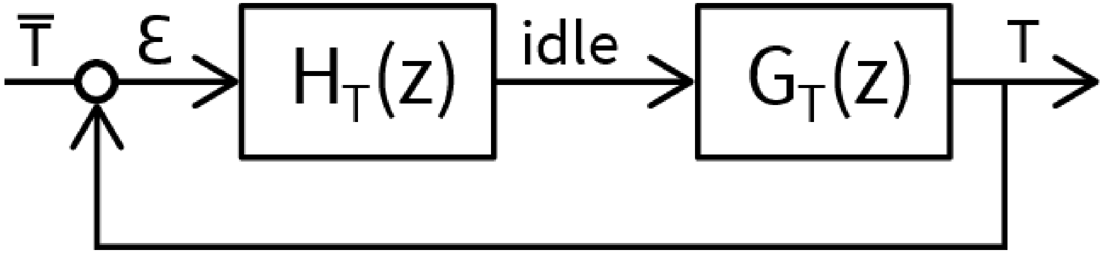
Thermal Model and Controller

We assume the following thermal model, per core i

$$T_i(k+1) = T_i(k) + \mu_i(k) \cdot \mathit{idle}_i(k)$$

μ_i is an unknown parameter; we estimate it with an Exponential Weighted Average (EWA) adaptive filter:

$$\mu_i(k) = \sum_{j=0}^{n} \hat{\mu}_i(k) = \sum_{j=0}^{n} \frac{(\bar{T} - T(k-j))}{\mathit{idle}_i(k-j)} \cdot e^{\lambda j}$$



We use a standard control-theoretical deadbeat controller:

$$\mathit{idle}_i(k) = (1/\mu_i(k)) \cdot (\bar{T} - T(k - j))$$

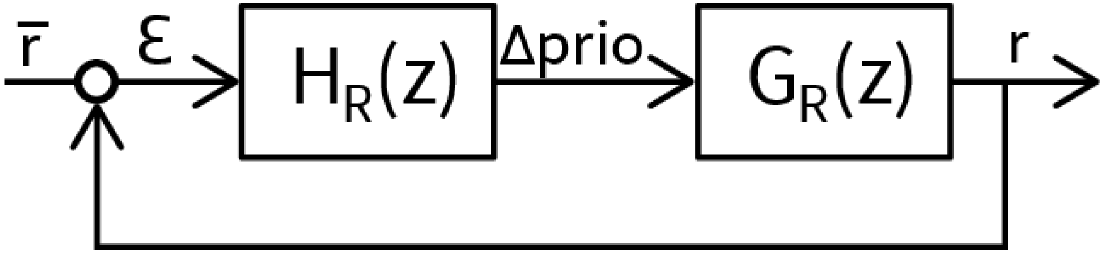
Performance Model and Controller

We assume the following performance model, per application i

$$r_i(k+1) = r_i(k) + \eta_i(k) \cdot \Delta \mathit{prio}_i(k)$$

η_i is an unknown parameter; we estimate it with an Exponential Weighted Average (EWA) adaptive filter:

$$\eta_i(k) = \sum_{j=0}^{n} \hat{\eta}_i(k) = \sum_{j=0}^{n} \frac{(\bar{r} - r(k-j))}{\Delta \mathit{prio}_i(k-j)} \cdot e^{\lambda j}$$



We use a standard control-theoretical deadbeat controller:

$$\Delta \mathit{prio}_i(k) = (1/\eta_i(k)) \cdot (\bar{r} - r(k - j))$$

Performance – Temperature Trade Off

We devised a simple heuristics to couple thermal and performance control

- Respecting SLAs has the priority: tasks of applications not meeting their QoS always have precedence over the idle task

- Idle time is charged to tasks of non SLA-bound applications or to those currently meeting their QoS

Implementation Details

We realized a port of the Heart Rate Monitor (HRM) [7] to FreeBSD 7.2 to get throughput measurements

- Throughput is measured on 1 second Moving Averages (MAs)

Processor temperature is measured on a per-core base by reading the appropriate Model Specific Register (MSR) with a high-priority kernel thread

The thermal and performance controllers run with a period of 100ms, and the 4.4BSD scheduler was modified to set priorities and schedule the idle task as computed by the controllers

---

[7] F. Sironi et al. *Metronome: operating system level performance management via self-adaptive computing.* In Proc. DAC 2012.

Thermal-Aware Policy Evaluation

Intel Core i7-990X six-core processor, FreeBSD 7.2, applications from the PARSEC 2.1 benchmark suite [8]

We evaluated the thermal-aware policy alone and compared it with *Dimetrodon* (no performance control in this experiment)

- *Dimetrodon* was run and resulting temperature was recorded
- DPTM temperature goal set to *Dimetrodon* outcome

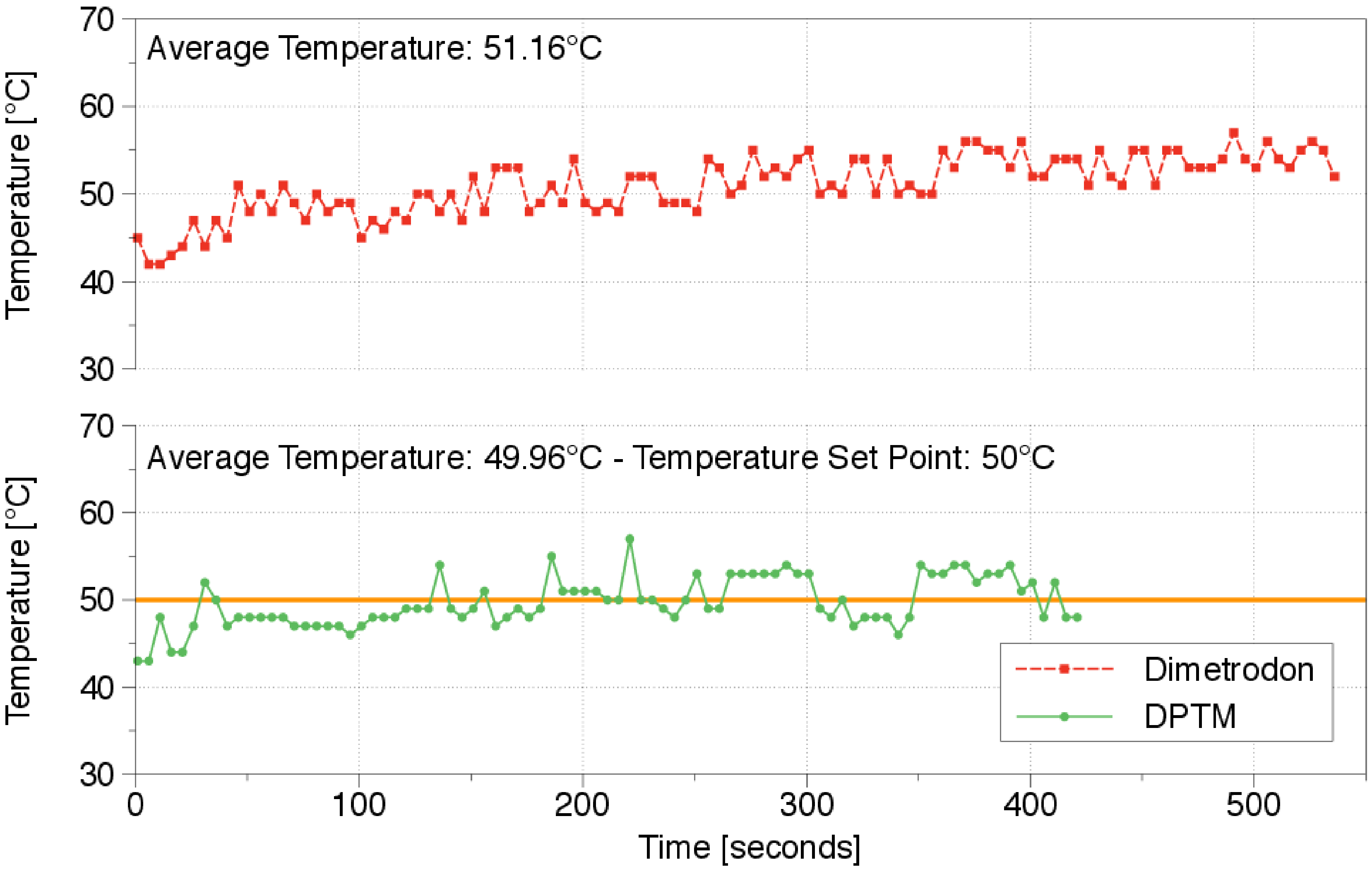
Application		Dimetrodon		DPTM framework			$\frac{\text{Dimetrodon}}{\text{DPTM}}$
	Idle Inj. Prob.	Avg [° C]	Std. Dev. [° C]	Set Point [° C]	Avg [° C]	Std. Dev. [° C]	Perf. Speedup
blackscholes	50%	51.16	3.45	50	49.96	2.98	1.27×
ferret	20%	58.02	5.01	55	56.36	2.97	1.12×
fluidanimate	10%	60.48	2.36	60	58.86	3.20	1.32×
swaptions	50%	59.00	4.60	55	54.03	3.33	1.57×

---

[8] C. Bienia. Benchmarking Modern Multiprocessors. PhD thesis, Princeton University, 2011..

Thermal-Aware Policy Sample Run

Blackscholes benchmark application, six-threaded



DPTM Framework Evaluation - Setup

Intel Core i7-870 quad-core processor @2.93 GHz, FreeBSD 7.2, applications from the PARSEC 2.1 benchmark suite [7]

This time, both the thermal and performance control are active, coordinated by the chosen heuristics

Four multithreaded applications in execution at the same time

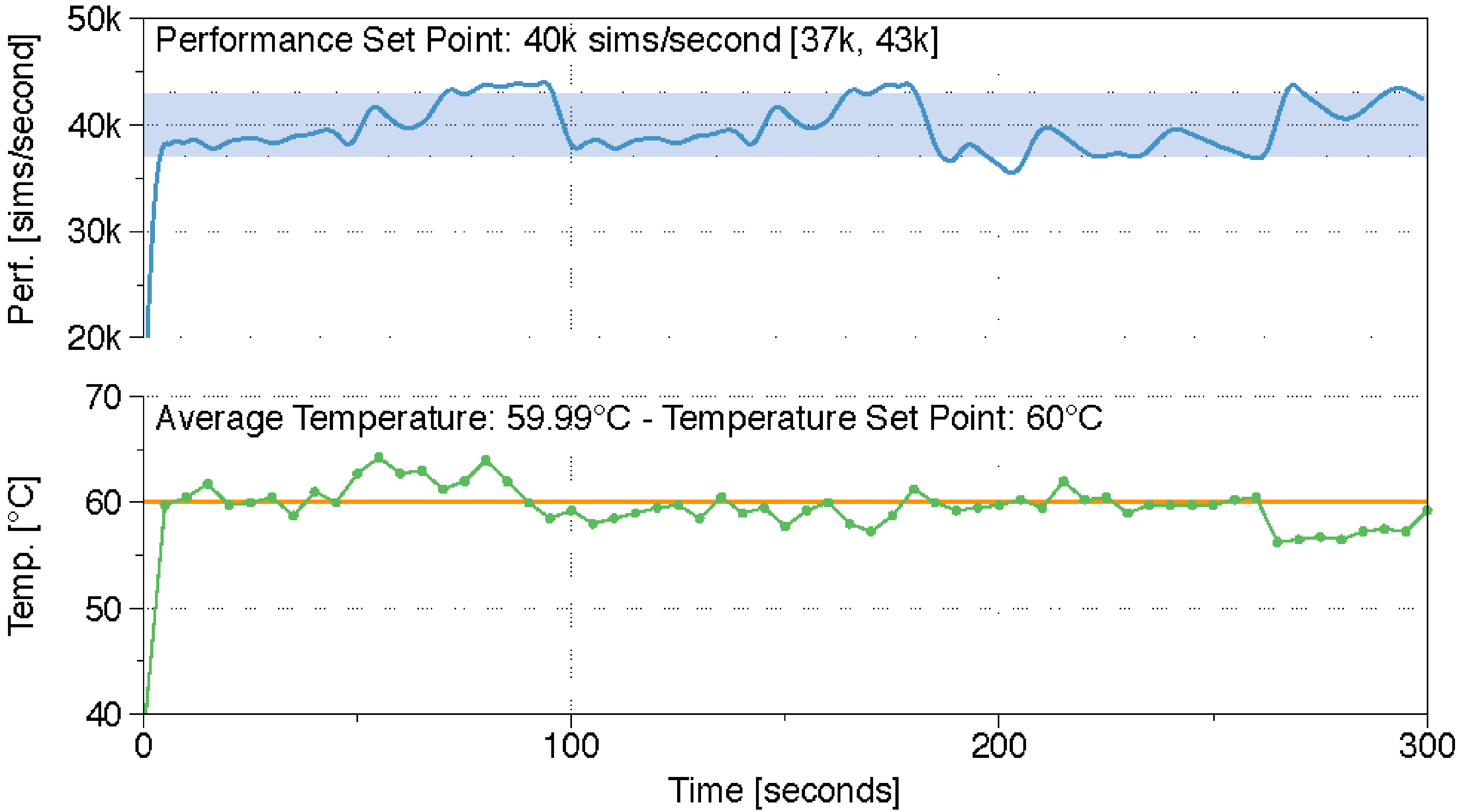
- One application is bound to an SLA on performance
- The thermal-aware policy is active towards a temperature set point

---

[7] C. Bienia. Benchmarking Modern Multiprocessors. PhD thesis, Princeton University, 2011..

Results – DPTM Framework

Four instances of the *Swaptions* benchmark, each four-threaded



Discussion and Future Work

The DPTM framework couples thermal and performance management, allowing to reduce temperature while respecting SLAs

The closed-loop thermal control policy overcomes limitations of *Dimetrodon*, allowing a goal-oriented approach

We show the soundness of the methodology; refinements are possible:

- Improve the thermal model to account for thermal interactions among cores
- Improve the idle-cycle injection mechanism to act evenly on multithreaded applications
- Improve the performance model
- Try different coupling strategies (e.g., for managing situations of resources scarceness)

That's All, Folks

A Framework for Thermal and Performance Management

Davide B. Bartolini, Filippo Sironi, Martina Maggio, Riccardo Cattaneo, Donatella Sciuto, Marco D. Santambrogio

Politecnico di Milano, Lund University

`{`**`bartolini`**`, sironi, cattaneo, sciuto, santambrogio}`**`@elet.polimi.it`**`,`
`martina.maggio@control.lth.se`