Physical-Aware Task Migration Algorithm for Dynamic Thermal Management of SMT Multi-Core Processors

Bagher Salami †, Mohammadreza Baharani §, Hamid Noori †, Farhad Mehdipour ‡

† School of Engineering, Ferdowsi University of Mashhad, Mashhad, Iran

§ School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran

‡ E-JUST Center, Graduate School of Information Science and Electrical Eng., Kyushu University, Fukuoka, Japan

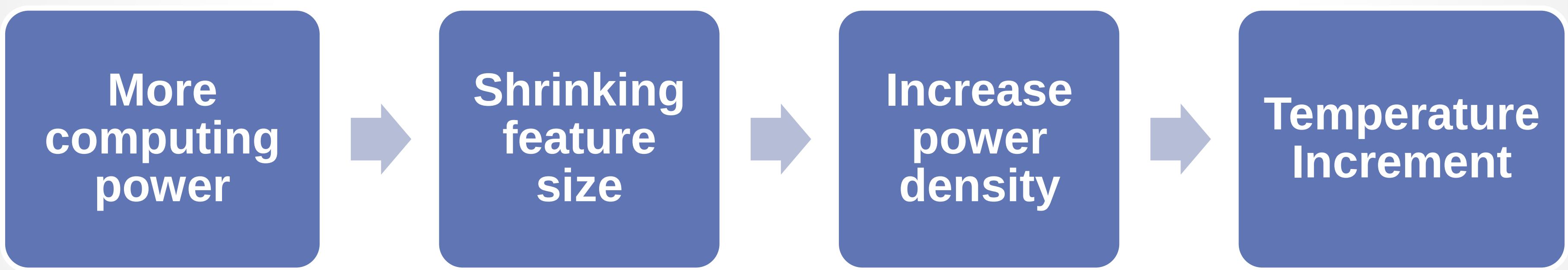
ASIA SOUTH PACIFIC DAC DESIGN AUTOMATION CONFERENCE

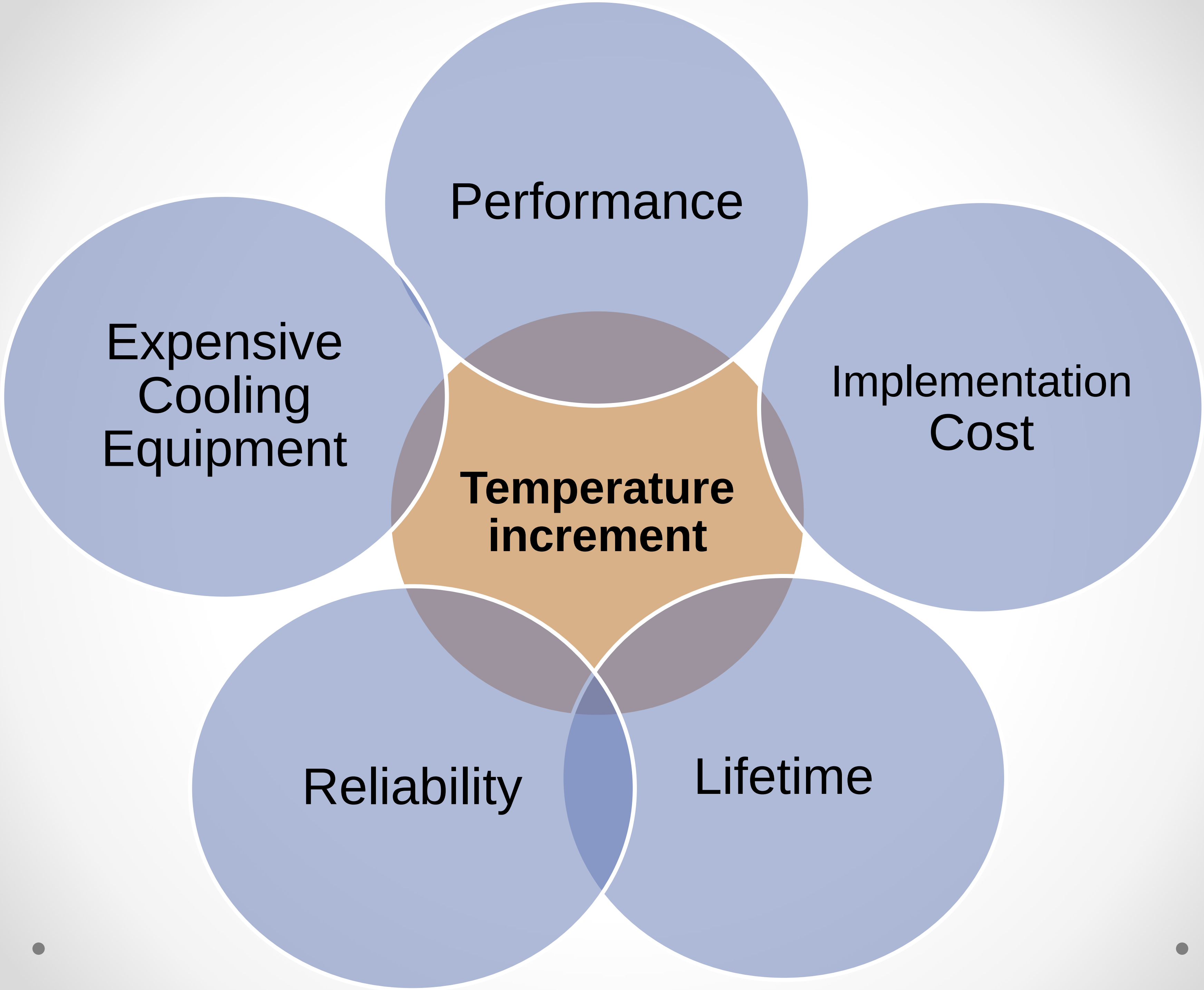
Outline

- Introduction
- SMT-multi core processors
- Problem description
- Contributions
- Proposed algorithm
- Experimental results
- Summary

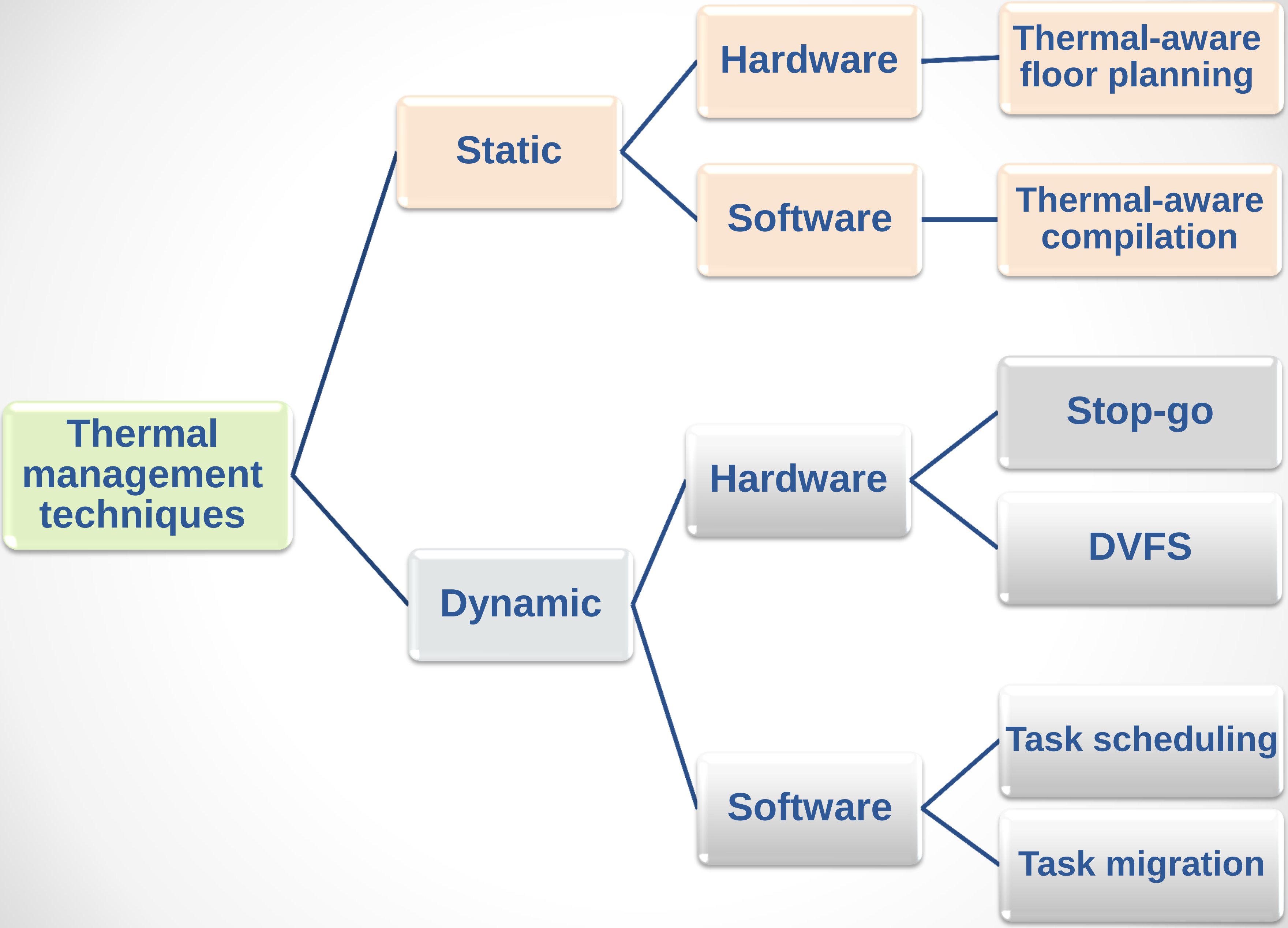
Outline

- Introduction
- SMT-multi core processors
- Problem description
- Contributions
- Proposed algorithm
- Experimental results
- Summary


More computing power
Shrinking feature size
Increase power density
Temperature Increment



Performance
Expensive Cooling Equipment
Implementation Cost
Temperature increment
Reliability
Lifetime


Thermal management techniques
Static
Dynamic
Hardware
Software
Thermal-aware floor planning
Thermal-aware compilation
Hardware
Software
Stop-go
DVFS
Task scheduling
Task migration

Outline

- Introduction
- SMT-multi core processors
- Problem description
- Contributions
- Proposed algorithm
- Experimental results
- Summary

Simultaneous Multi Threading (SMT) (1/2)

- **A single physical processor appears as two logical processors**
- **The physical execution resources are shared and the architecture state is duplicated**

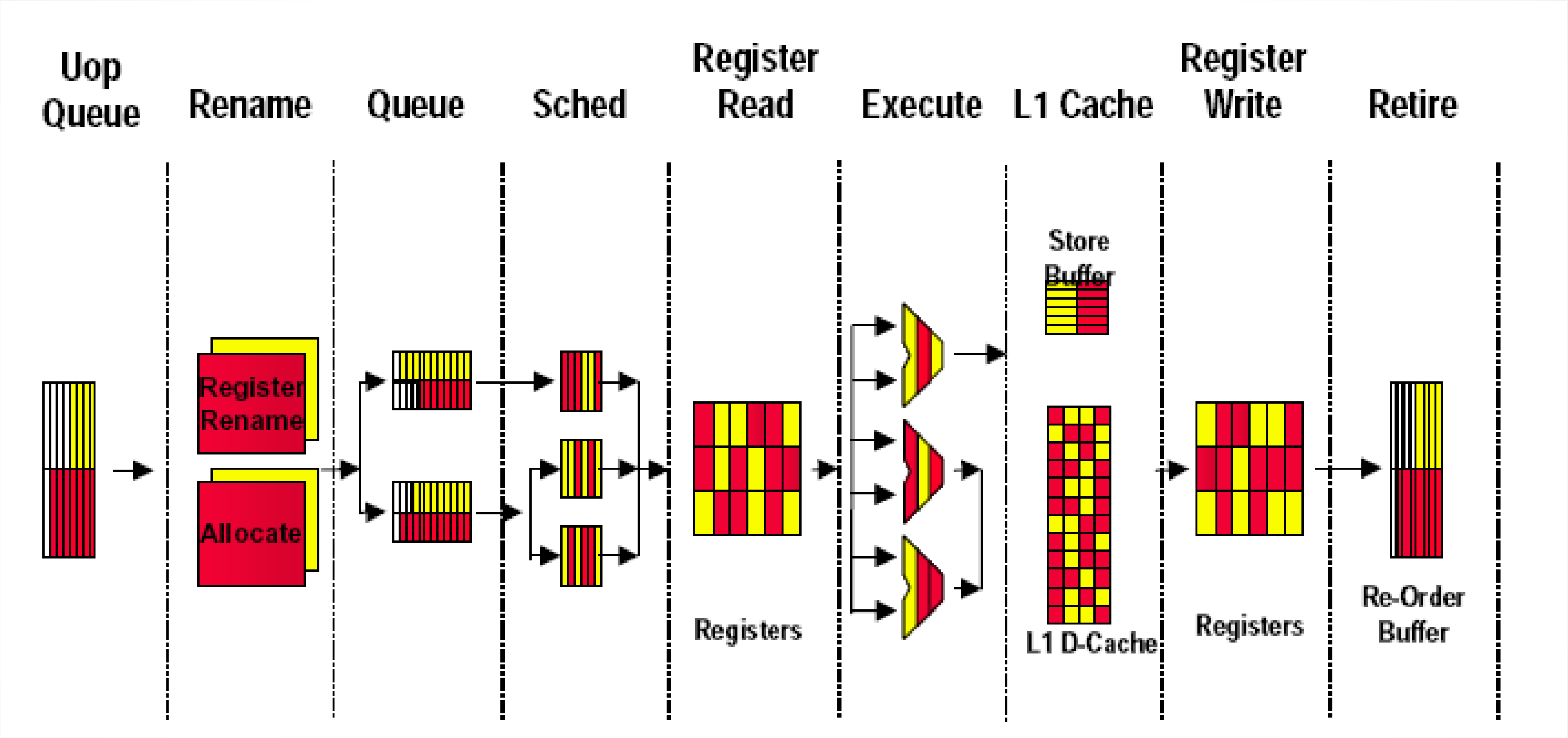
Dedicated resources

- **General purpose registers**
- **Control registers**
- **Machine state registers**

Shared resources

- **Caches**
- **Execution units**
- **Buses**

Simultaneous Multi Threading (SMT) (2/2)



A two-context SMT pipeline*

* Marr, Deborah T, “Hyper-threading technology architecture and microarchitecture: A hypertext history,“ Intel Technology Journal, 2002

Outline

- Introduction
- SMT-multi core processors
- Problem description
- Contributions
- Proposed algorithm
- Experimental results
- Summary

Problem Description

Target System Specification

- An SMT multi-core processor with N cores
- There are $N+1$ to $2 \times N$ tasks for execution
- Each core can execute up to two threads simultaneously (two-context SMT cores)
- Each of N cores has a physical thermal sensor

The problem:

How to schedule tasks among cores dynamically so that:

- The average and peak temperature under minimum performance loss is **minimized**
- Temperature does not violate T_{max}.

Outline

- Introduction
- SMT-multi core processors
- Problem description
- Contributions
- Proposed algorithm
- Experimental results
- Summary

Contributions

- Considering features of SMT-multicores in task scheduling, that leads to optimal thermal management
- Task migration based on **physical features of the cores**
- Thermal management of off-the-shelf SMT multi-core products

Physical Features of Cores (1/3)

Different cores of a processor do not have similar thermal behavior due to:

- Process variation,
- Packaging, and
- The temperature effect of neighbor components,

→ The temperature difference between cores of a processor running the same application can be as much as 10~15°C*

* Inchoon Yeo, Chih Chun Liu, and Eun Jung Kim, "Predictive dynamic thermal management for multicore systems," DATE, 2008.

Physical Features of Cores (2/2)

Benchmark	Executed on core 0	Executed on core 1	Executed on core 2	Executed on core 3
gcc	59°C	58°C	61°C	64°C
hmmer	69°C	62°C	63°C	66°C
bzip2	69°C	67°C	69°C	71°C

Peak temperature differential between cores on Intel Core i7-2600

Benchmark	Executed on core 0	Executed on core 1	Executed on core 2	Executed on core 3
gcc	56◦C	56◦C	57◦C	55◦C
hmmer	60◦C	60◦C	62◦C	58◦C
bzip2	59◦C	59◦C	60◦C	58◦C

Peak temperature differential between cores on Intel Core i7-3770

Physical Features of Cores (3/3)

Benchmark	Executed on core 0	Executed on core 1	Executed on core 2	Executed on core 3	Executed on core 4	Executed on core 5	Executed on core 6	Executed on core 7
gcc	41.5	42	45	44.5	42	42	43	43.5
hmmer	42	43.5	44.5	44.5	42	41.5	43.5	42.5
bzip2	41	41.5	43.5	43.5	41	41	41.5	42.5

Peak temperature differential between cores on AMD's 8-core Bulldozer

Task Scheduling of SMT Multi-Core Processors

Main challenge in task scheduling of SMT multi-core processors:

- Co-scheduling of complementary threads on individual SMT cores*
  - Better use of shared pipeline resources
  - Improving performance
  - Higher heat generation due to higher utilization of pipeline resources

In this work:

- Five different strategies are studied,
- Purpose: finding the most suitable pairs of tasks that should be co-scheduled

* Donald, James, and Margaret Martonosi. "Techniques for multicore thermal manage-ment: Classification and new exploration," ACM SIGARCH Computer Architecture News 34, pp. 78-88, 2006.

Different Strategies of Task Scheduling of SMT Multi-Core Processors (1/4)

First Strategy: Sorting cores based on the **temperature of physical sensors**

- Sort tasks based on **performance counter values,**
- Hottest and coolest tasks are co-scheduled to the coldest core,
- The second hottest and coolest tasks are co-scheduled to the second coolest core,
- The above process continues.

Complimentary task scheduling → **Higher performance**

Execution on cold cores → **Lower temperature**

Different Strategies of Task Scheduling of SMT Multi-Core Processors (2/4)

Second Strategy: Sorting cores based on the **physical features of cores**

- Sort tasks based on **performance counter values,**
- Hottest and coolest tasks are co-scheduled to the coldest core,
- The second hottest and coolest tasks are co-scheduled to the second coolest core,
- The above process continues.

Complimentary task scheduling → **Higher performance**

Execution on cold cores → **Lower temperature**

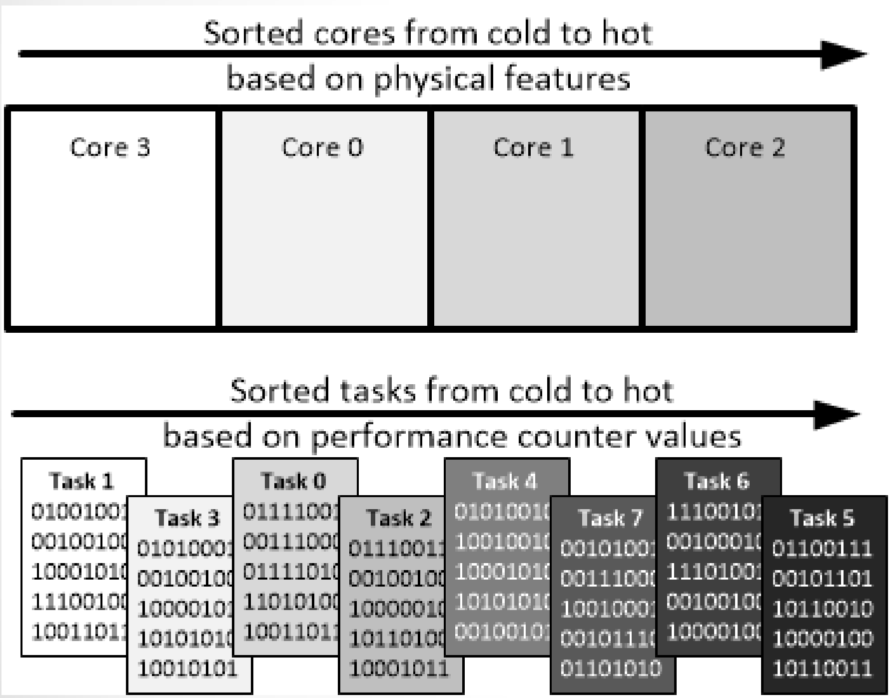
Different Strategies of Task Scheduling of SMT Multi-Core Processors (3/4)

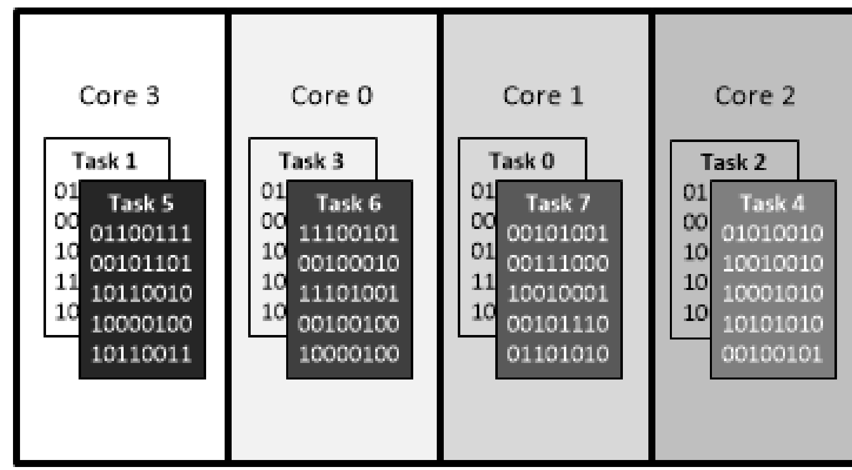
- **Third Strategy:**
  - After sorting cores according to their physical features, the first two hottest tasks are co-scheduled to the coldest core.
- **Fourth Strategy:**
  - Similar to the third strategy, except
  - the first two coldest tasks are assigned to the coldest core.
- **Fifth Strategy:**
  - Reschedules tasks between cores only in ***critical situation*** (the core that has $t_r < t_{res}$)

the response time for the algorithm to decrease the core temperature

Acc… **leads to the best**

Different Strategies of Task Scheduling of SMT Multi-Core Processors (4/4)



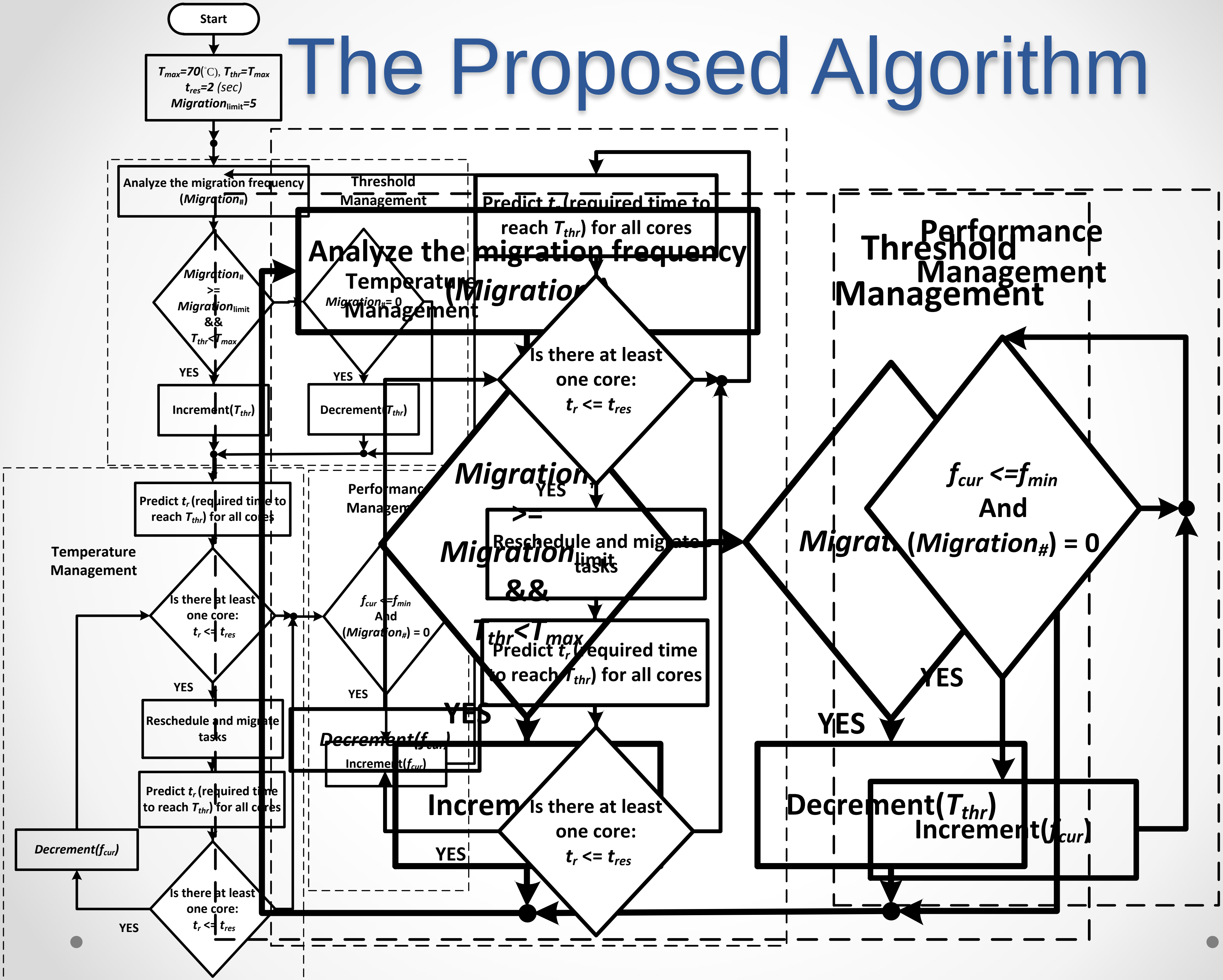


More details on the second strategy

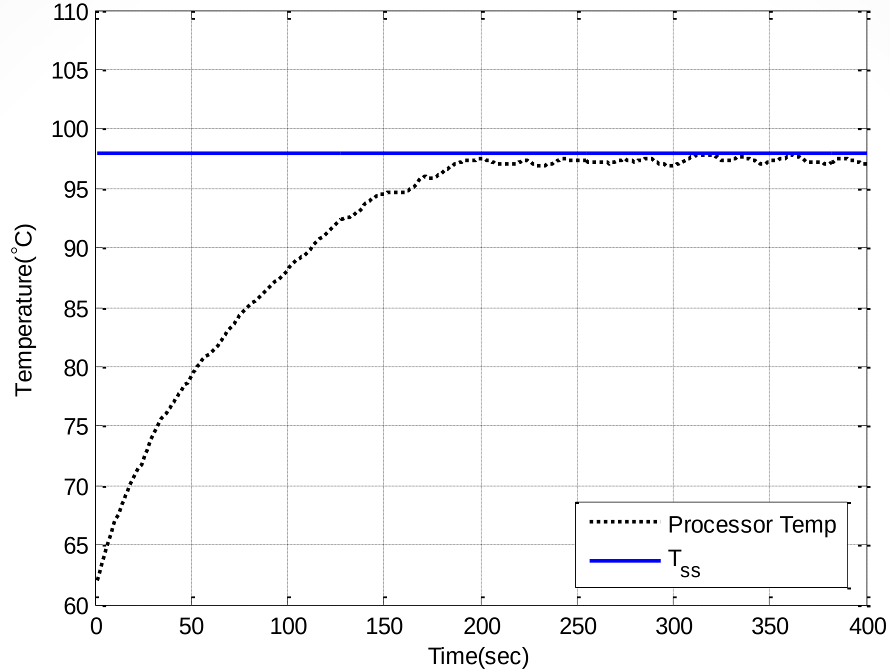
Outline

- Introduction
- SMT-multi core processors
- Problem description
- Contributions
- Proposed algorithm
- Experimental results
- Summary

The Proposed Algorithm



Temperature Prediction*



Steady state temperature of *hmmer* on intel Core i7-2600

a s t

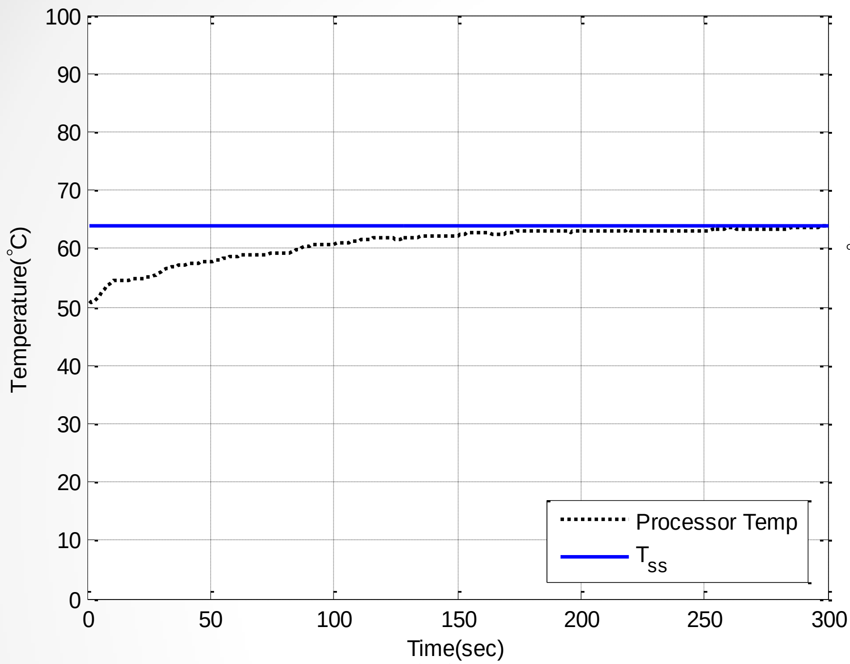
current temperature

initial processor temperature

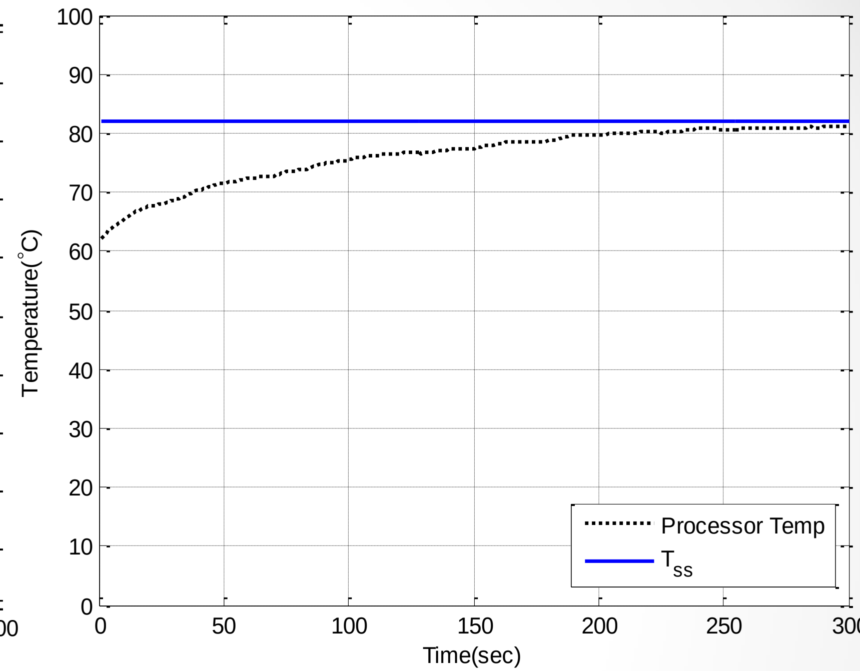
$$\frac{dT}{dt} = c \times (T_{ss} - T), \qquad T(t) = T_{SS} - (T_{SS} - T_{init}) \times e^{-c \times t}.$$

* Inchoon Yeo and Eun Jung Kim, “Temperature-aware scheduler based on thermal behavior grouping in multicore systems,” DATE, 2009.

Improved Temperature Prediction



(a)



(b)

(a) T_{ss} of hmmer on core 3 and other cores are idle.

(b) T_{ss} of hmmer on core 3 and core 2 and core 1 executing gcc.

core activity

$$\frac{dT}{dt} = c \times w \times (T_{ss} - T\ ),\ T(t) = T_{SS} - (\ T_{SS} - T_{init}) \times\ e^{-c \times w \times t}.$$

Outline

- Introduction
- SMT-multi core processors
- Problem description
- Contributions
- Proposed algorithm
- Experimental results
- Summary

Experimental Setup

Benchmarks	hmmer	libquantum	sjeng	perlbench	gobmk	gcc	mcf	bzip2
Avg. Temp (°C)	68.2	67	65.7	65	63.9	63.9	63	62.9

System Specifications	Value
Number of cores	4
Processor	Intel Core i7-3770
Main memory size	GB 8
OS	Linux 3.2.0

Performance Counter Analysis (1/3)

We use performance counters to:

1) Distinguish the cold thread from the hot thread on a two-context SMT core, and
2) To sort applications according to their temperature.

In order to decrease performance overhead using performance counter:

Seeking performance counter with the greatest correlation with temperature of programs

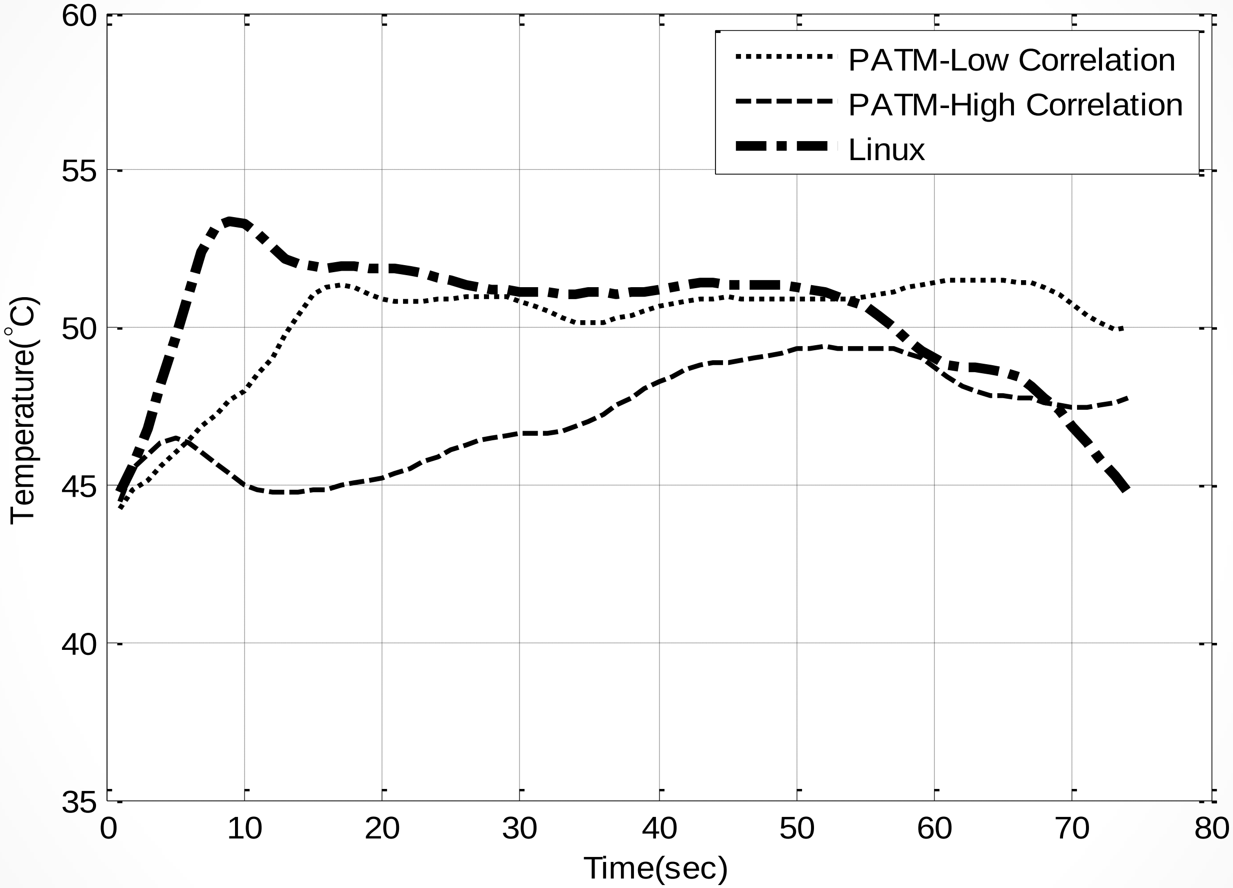
Performance Counter Analysis (2/3)

greatest correlation with temperature

Events	Correlation
stalled-cycles-backend	-0.37
cache-references	-0.35
stalled-cycles-frontend	-0.35
cache-misses	-0.33
Cycles	-0.29
task-clock	-0.24
context-switches	-0.03
branches	-0.03
page-faults	-0.01
branch-misses	0.02
CPU-migrations	0.04
Instructions	0.29
IPC	0.30

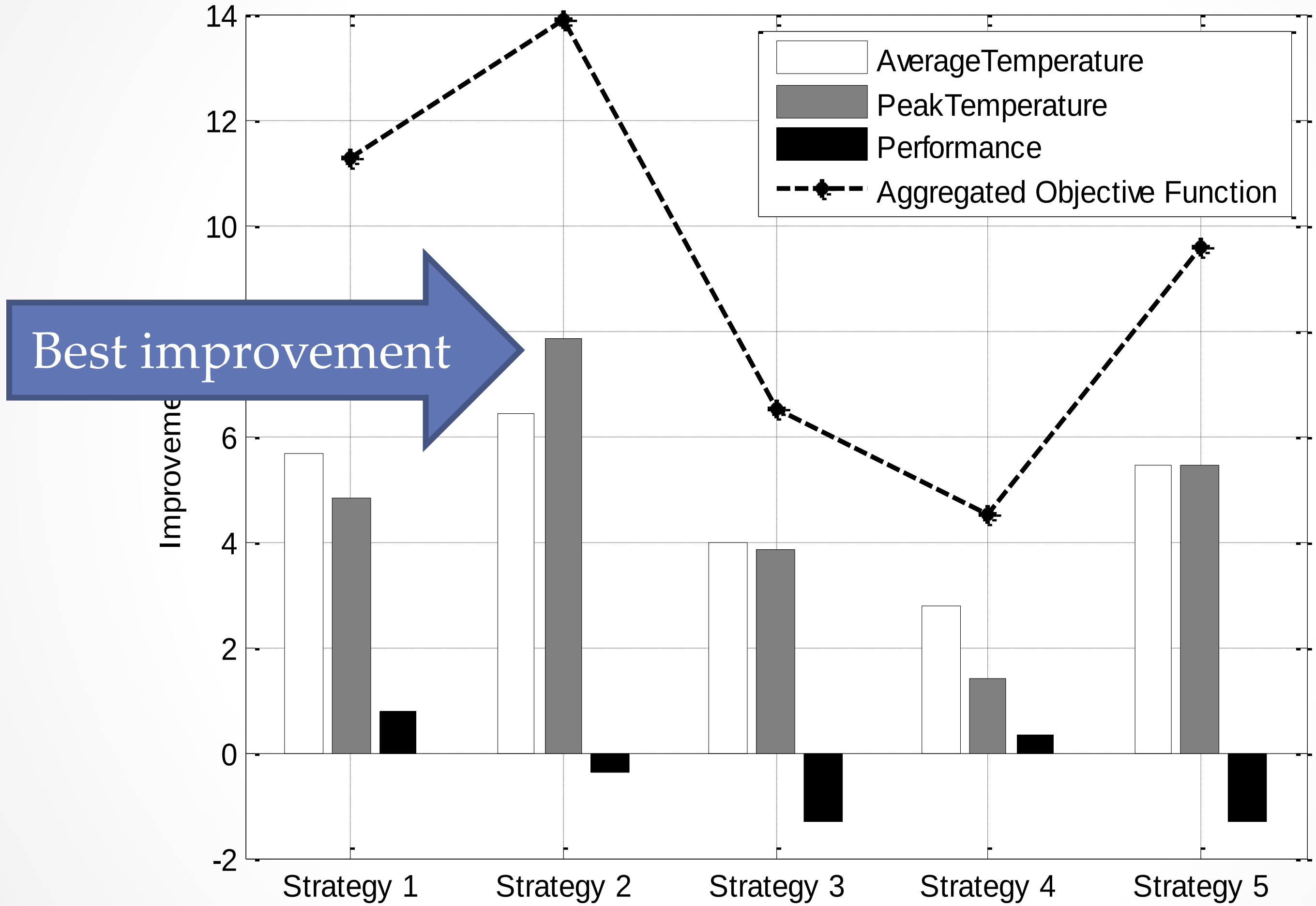
Correlation between events and applications temperature

Performance Counter Analysis (3/3)



PATM (Physical-Aware Task Migration - our proposed method)'s average temperature using high and low correlation counter for application ordering vs. Linux standard scheduler.

Comparison of Task Scheduling Strategies



Performance, and average and peak temperature improvement of different strategies compared to Linux standard scheduler

Experimental Results

Compared methods:

(1) PDTM (Predictive Dynamic Thermal Management):

One of the first attempts that predicts core temperature and migrates applications from the possible overheated core to the future coolest core in order to maintain the processor temperature below a threshold temperature.

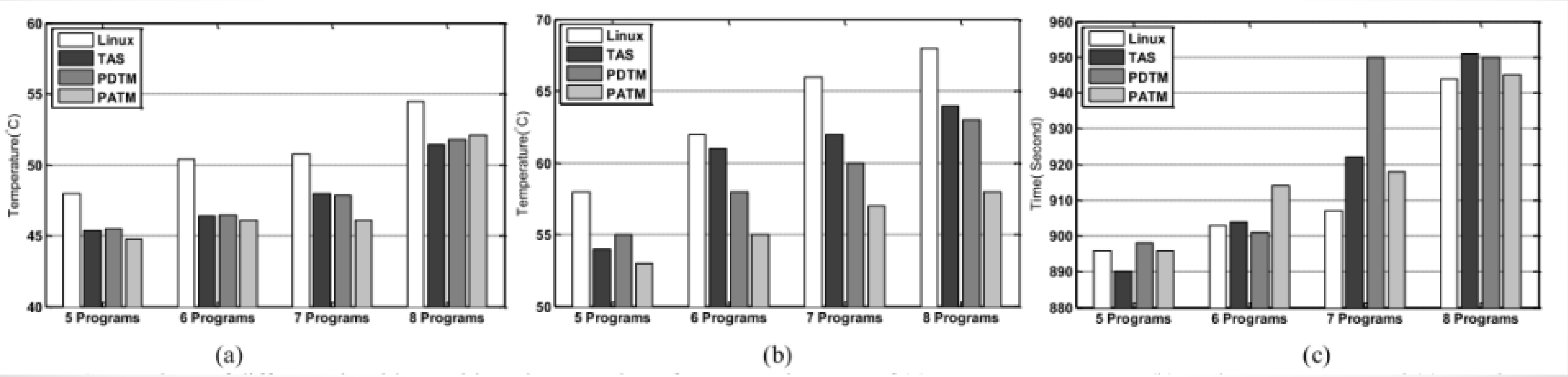
(2) TAS (Thermal Aware Scheduler):

Similar PDTM but categorizes applications according to their thermal behavior.

Experimental Results

Temperature management

- Running 5 to 8 program simultaneously with four different algorithms: Linux, TAS, PDTM, and our proposed method PATM.
- Comparing average and peak temperature and run-time.



Comparison of different algorithms with various number of programs in terms of (a) average temperature (b) peak temperature, and (c) run-time.

Experimental Results

Temperature management

DTM Algorithm	Average Temp.	Peak Temp.	Run Time(Second)
Linux	50.9(°C)	64(°C)	912.5(Sec)
PDTM	47.9(°C)	59(°C)	924.8(Sec)
TAS	47.8(°C)	60(°C)	916.8(Sec)
PATM	47.3(°C)	56(°C)	928.3(Sec)
Improvement PATM vs. PDTM	1.3%	5.8%	-0.4%
Improvement PATM vs. TAS	1.1%	8.1%	-1.3%
Improvement PATM vs. Linux	7.7%	13.9%	-1.7%

Total comparison results of PATM against Linux, TAS, and PDTM.

Outline

- Introduction
- SMT-multi core processors
- Problem description
- Contributions
- Proposed algorithm
- Experimental results
- Summary

Summary

- A dynamic thermal management algorithm for SMT multicore processors.
- The algorithm manages processor temperature taking into account **the workload phase (being hot or cold)** and **physical features of cores**.
- Experimental results indicate that:
  - Our algorithm outperforms Linux standard scheduler, TAS, and PDTM in terms of thermal management,
  - Performance overhead is negligible.

Thank You!