NANYANG TECHNOLOGICAL UNIVERSITY

Inaugural Youth Olympic Village

High-Level Event Driven Thermal Estimation for Thermal Aware Task Allocation and Scheduling

presented by

Cui Jin
Center of High Performance Embedded Systems (CHiPES)
School of Computer Engineering
Nanyang Technological University

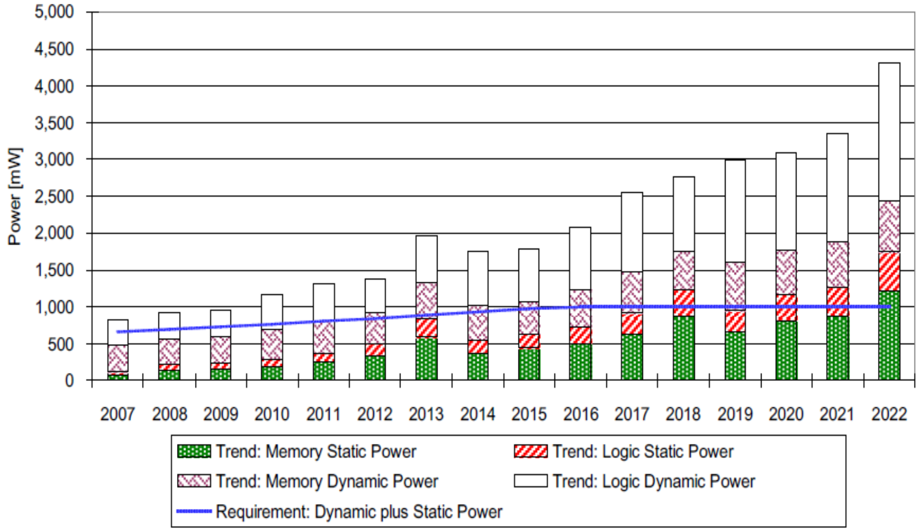
21st Jan, 2010

Outline

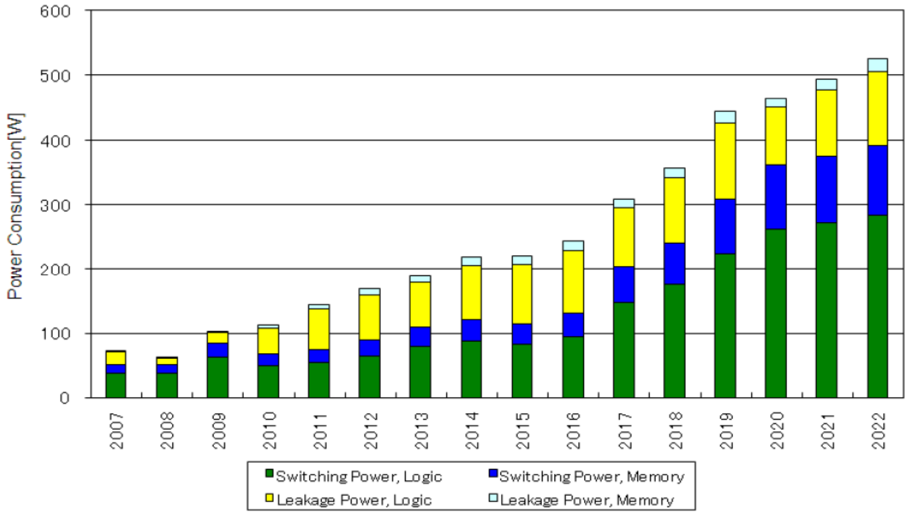
- Introduction & Motivation
- Basic Thermal Model
- Our Event Driven Method
- Heuristic Task Allocation
- Experimental Results
- Conclusion

Introduction & Motivation

- Power/thermal factors are becoming the major design constraints and bottleneck for current and future computing devices.
- ITRS(International Technology Roadmap of Semiconductor) 2008 shows this trend in next two decades.



Power Consumption of Portable Devices



Power Consumption of Stationary Devices

NANYANG TECHNOLOGICAL UNIVERSITY 南洋理工大學

Introduction & Motivation

- Temperature is directly related to power density. The power density in state-of-the-art microprocessors is currently in the range of150--250W/cm^2.
- Thermal related problems:
  - Temperature/Performance Degradation
  - Reliability and Accelerated Aging
  - Cost
  - Temperature/Leakage Power Relationship
- The microprocessor development trends:
  - Uniprocessor ➔ Multi-Core ➔ Many-Core
  - More than 50 cores at 2022 in ITRS prediction
- Power/Thermal Issues + Multiprocessor = ?

Introduction & Motivation

- Power/thermal optimizations at high level:
  - Micro-Architecture/Architecture Level: dynamic voltage and frequency scaling(DVFS), clock gating, pipeline gating, stop & go policy, I-cache toggling, thermal-aware floorplanning.
  - System Level: compiler techniques, power/thermal-aware scheduling, hardware/software partition, application specific optimization(e.g. producer-consumer model, stream media)
- System level power/thermal optimization on multiprocessor is a relatively new research area.
- Current high level power/thermal optimizations are dramatically based on some high level power/thermal models or predictable methodologies.
  - Efficient
  - Accurate

Introduction & Motivation

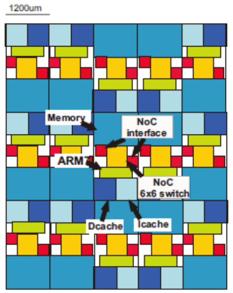
- Some existed thermal models:
  - TEMPEST: average temperature for a single chip; not suitable for multicore
  - TSIC: empirical method; fast, but accuracy depends on the style of functions; other empirical model, like single exponential temperature curve, gives inaccurate estimation for the multicore chip
  - HotSpot: well known; suitable for micro-architecture level thermal simulation; accurate enough for high level estimation, but the efficiency is not designed for embedding into the OS kernel to guide the task scheduling
  - Learning-based models: linear regressive function for 22 system events; dramatically depends on the learning sample set and features of task set; long time to training
- The intention of our paper is to propose a rapid thermal estimation method to assist the OS kernel in applying dynamic TAS at runtime.

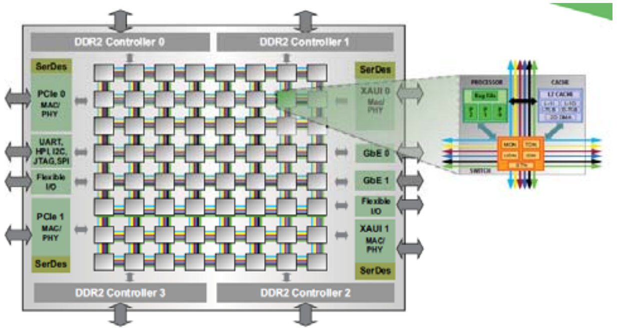
Outline

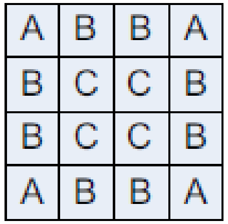
- Introduction & Motivation
- Basic Thermal Model
- Our Event Driven Method
- Heuristic Task Allocation
- Experimental Results
- Conclusion

Basic Thermal Models

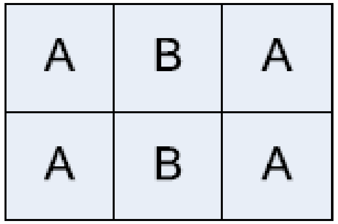
- Objective of Study: CMP and MPSoC
  - Two distinct branches: Chip Multi-Processor(CMP) and Multi-Processor System-on-Chip(MPSoC)
  - Abstracted Layout at Core-Level



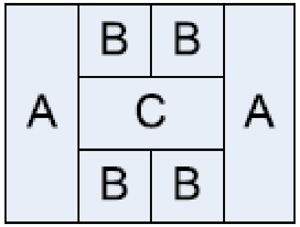




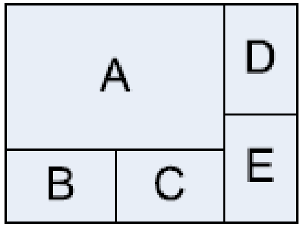
4×4 CMP



2×3 CMP



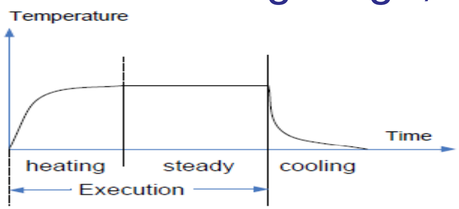
7Cores MPSoC



5Cores MPSoC

Basic Thermal Model

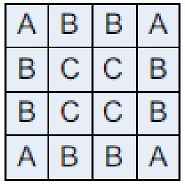
- For thermal-aware scheduling, the task-level power variation and core-level thermal behavior are more concentrated, not like at micro-architecture level, every details over time must be reviewed for better design. From other’s observation, we summarized the following two points:
  - The power consumed by a task varies in the short term, but the variance is small enough that the temperature changes relatively slowly due to the large thermal mass of the CMP. The core level thermal effect of the entire task execution can be treated as a heating stage, a steady temperature stage, and a cooling stage.



  - The power consumption changes dramatically only after task allocation and deallocation, and thus the temperature of each core experiences a rapid change after the occurrence of these system events (i.e. allocation, deallocation, context switch, preemption etc.).

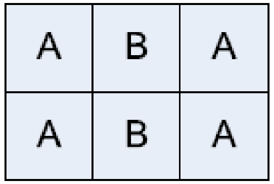
Basic Thermal Model

- Thermally Different Location.
  - *A thermally different location (TDL) is defined for sets of homogeneous cores which are symmetrical in their relative location, and thus have similar thermal characteristics. The important implication of TDL is that if a task is allocated to a core in a TDL, it will produce the same thermal effects and distribution as it would when allocated to another core of the CMP with the same TDL.*



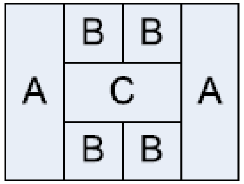
A	B	B	A
B	C	C	B
B	C	C	B
A	B	B	A

4×4 CMP

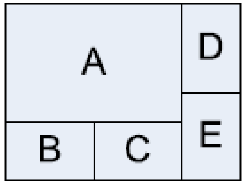


A	B	A
A	B	A

2×3 CMP



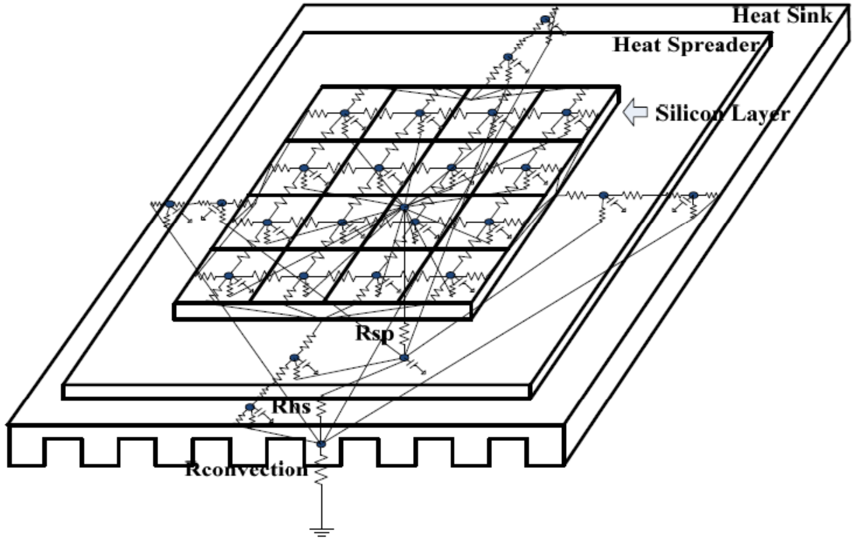
7Cores MPSoC



5Cores MPSoC

Basic Thermal Model

- Thermal RC network is widely used for temperature simulation at micro-architecture level. (HotSpot)



$$C_i \frac{dT_i}{dt} = p_i + \sum_{j \in Path} \frac{T_i - T_j}{R_j}, \;\; node \;\; i \in silicon \; layer$$

$$C_i \frac{dT_i}{dt} = \sum_{j \in Path} \frac{T_i - T_j}{R_j}, \;\; node \;\; i \notin silicon \; layer$$

This is a set of the linear ordinary differential equations.

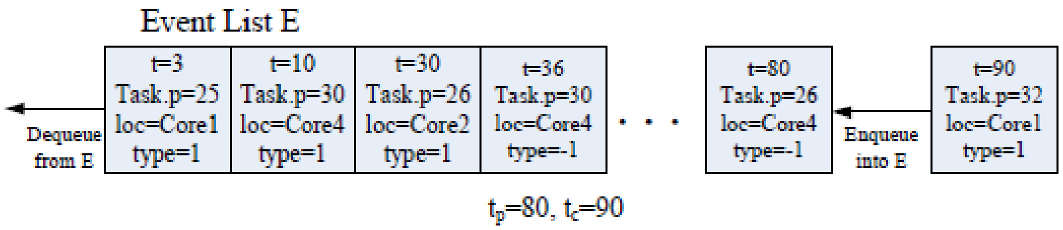
  - Several Layers: Silicon Layer, Heat Spreader, Heat Sink
  - Several Nodes: Power Injection from the Node on Silicon Layer
  - Resistances between the adjacent nodes
  - Capacitances attached to the each node

Outline

- Introduction & Motivation
- Basic Thermal Model
- Our Event Driven Method
- Heuristic Task Allocation
- Experimental Results
- Conclusion

Our Event Driven Method

- Two definitions on ‘Event’
  - High Level Event: A high level event is defined as an event which will induce a change in the power consumption of the system. High level events can be captured and maintained by the OS kernel.
  - Atomic Power Event: An atomic power event is associated with the (relatively instantaneous) increase or decrease in power generated by a core. Any high level event can be decomposed into reasonable combinations of these two atomic events.
- Updating the core temperature only when a high level event occurs. Such events include: task allocation, deallocation, context-switch, migration, preemption, stop-go and DVFS.
- We use a Event List E as a queue to keep the record of the atomic events.



Event List E

Dequeue from E

t=3 Task.p=25 loc=Core1 type=1

t=10 Task.p=30 loc=Core4 type=1

t=30 Task.p=26 loc=Core2 type=1

t=36 Task.p=30 loc=Core4 type=-1

. . .

t=80 Task.p=26 loc=Core4 type=-1

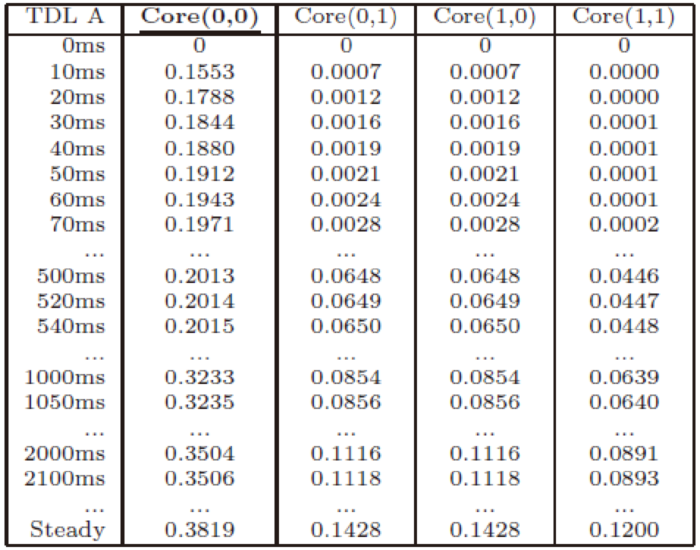
Enqueue into E

t=90 Task.p=32 loc=Core1 type=1

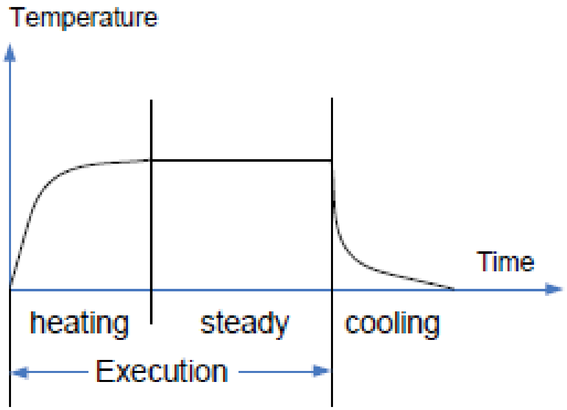
t_p=80, t_c=90

Our Event Driven Method

- Fast enough to calculate the thermal distribution online by using Look-Up Table (LUT)
- One TDL generates one LUT (offline)
- Record the temperature transient in the whole heating stage



TDL A	Core(0,0)	Core(0,1)	Core(1,0)	Core(1,1)
0ms	0	0	0	0
10ms	0.1553	0.0007	0.0007	0.0000
20ms	0.1788	0.0012	0.0012	0.0000
30ms	0.1844	0.0016	0.0016	0.0001
40ms	0.1880	0.0019	0.0019	0.0001
50ms	0.1912	0.0021	0.0021	0.0001
60ms	0.1943	0.0024	0.0024	0.0001
70ms	0.1971	0.0028	0.0028	0.0002
...	...	...	...	...
500ms	0.2013	0.0648	0.0648	0.0446
520ms	0.2014	0.0649	0.0649	0.0447
540ms	0.2015	0.0650	0.0650	0.0448
...	...	...	...	...
1000ms	0.3233	0.0854	0.0854	0.0639
1050ms	0.3235	0.0856	0.0856	0.0640
...	...	...	...	...
2000ms	0.3504	0.1116	0.1116	0.0891
2100ms	0.3506	0.1118	0.1118	0.0893
...	...	...	...	...
Steady	0.3819	0.1428	0.1428	0.1200



Coarser time interval between two adjacent rows over the time

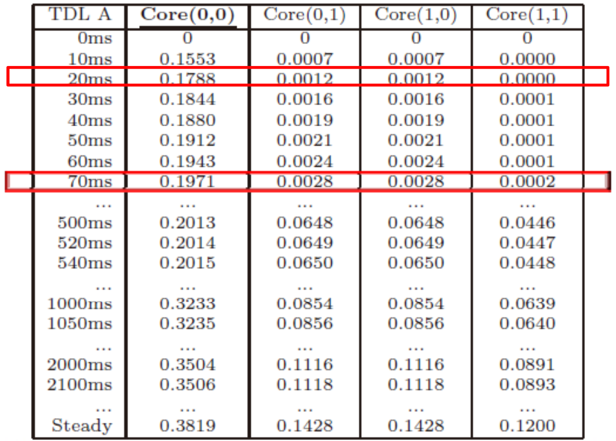
Our Event Driven Method

- Current thermal map is updated from previous thermal map by:

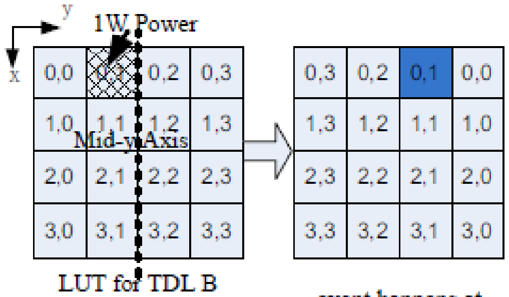
$$T_{t_c} = T_{t_p} + \Delta T_{\Delta t = t_c - t_p}$$

- The temperature increment at one core can be treated as the accumulation of every individual temperature increment at this core induced by each atomic power event (power increasing or decreasing). The thermal distribution induced by atomic power events adheres to the superposition principle.(linear model)

$$\Delta T_{\Delta t = t_c - t_p} = \sum\nolimits_{ae \in E} ae.type \times ae.Task.p \times trans(\Re^{tdl(ae.loc)}_{t_c - ae.t} - \Re^{tdl(ae.loc)}_{t_p - ae.t}, ae.loc)$$



TDL A	**Core(0,0)**	Core(0,1)	Core(1,0)	Core(1,1)
0ms	0	0	0	0
10ms	0.1553	0.0007	0.0007	0.0000
20ms	0.1788	0.0012	0.0012	0.0000
30ms	0.1844	0.0016	0.0016	0.0001
40ms	0.1880	0.0019	0.0019	0.0001
50ms	0.1912	0.0021	0.0021	0.0001
60ms	0.1943	0.0024	0.0024	0.0001
70ms	0.1971	0.0028	0.0028	0.0002
...	...	...	...	...
500ms	0.2013	0.0648	0.0648	0.0446
520ms	0.2014	0.0649	0.0649	0.0447
540ms	0.2015	0.0650	0.0650	0.0448
...	...	...	...	...
1000ms	0.3233	0.0854	0.0854	0.0639
1050ms	0.3235	0.0856	0.0856	0.0640
...	...	...	...	...
2000ms	0.3504	0.1116	0.1116	0.0891
2100ms	0.3506	0.1118	0.1118	0.0893
...	...	...	...	...
Steady	0.3819	0.1428	0.1428	0.1200

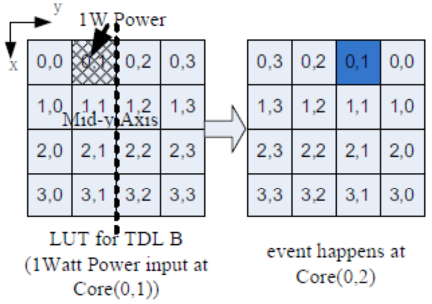


LUT for TDL B
(1Watt Power input at
Core(0,1))

event happens at
Core(0,2)

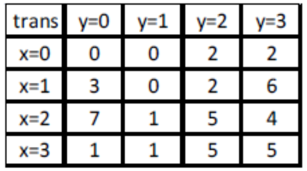
Our Event Driven Method

- There are eight simple transform options: 0: no change; 1: mid-x mirroring; 2: mid-y mirroring; 3: principal diagonal mirroring; 4: secondary diagonal mirroring; 5: center point mirroring; 6: clockwise rotation; and 7: counter-clockwise rotation.



LUT for TDL B
(1Watt Power input at
Core(0,1))

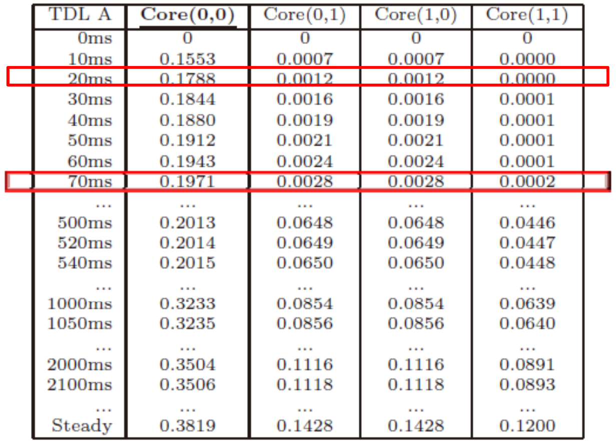
event happens at
Core(0,2)



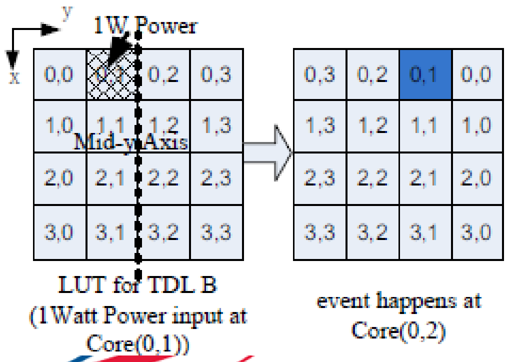
trans	y=0	y=1	y=2	y=3
x=0	0	0	2	2
x=1	3	0	2	6
x=2	7	1	5	4
x=3	1	1	5	5

$$t_c - ae.t > Steady$$

$$\Delta T_{\Delta t=t_c-t_p} = \sum_{ae\in E} ae.type \times ae.Task.p \times trans(\Re^{tdl(ae.loc)}_{t_c-ae.t} - \Re^{tdl(ae.loc)}_{t_p-ae.t}, ae.loc)$$



TDL A	Core(0,0)	Core(0,1)	Core(1,0)	Core(1,1)
0ms	0	0	0	0
10ms	0.1553	0.0007	0.0007	0.0000
20ms	0.1788	0.0012	0.0012	0.0000
30ms	0.1844	0.0016	0.0016	0.0001
40ms	0.1880	0.0019	0.0019	0.0001
50ms	0.1912	0.0021	0.0021	0.0001
60ms	0.1943	0.0024	0.0024	0.0001
70ms	0.1971	0.0028	0.0028	0.0002
...	...	...	...	...
500ms	0.2013	0.0648	0.0648	0.0446
520ms	0.2014	0.0649	0.0649	0.0447
540ms	0.2015	0.0650	0.0650	0.0448
...	...	...	...	...
1000ms	0.3233	0.0854	0.0854	0.0639
1050ms	0.3235	0.0856	0.0856	0.0640
...	...	...	...	...
2000ms	0.3504	0.1116	0.1116	0.0891
2100ms	0.3506	0.1118	0.1118	0.0893
...	...	...	...	...
Steady	0.3819	0.1428	0.1428	0.1200



LUT for TDL B
(1Watt Power input at
Core(0,1))

event happens at
Core(0,2)

Our Event Driven Method

- Our event driven method can not only obtain the current thermal map from the previous thermal map, but also predict the future thermal distribution on chip in next time interval. If we replace the t_p and t_c, using t_c and t_n(time instant we want to estimate):

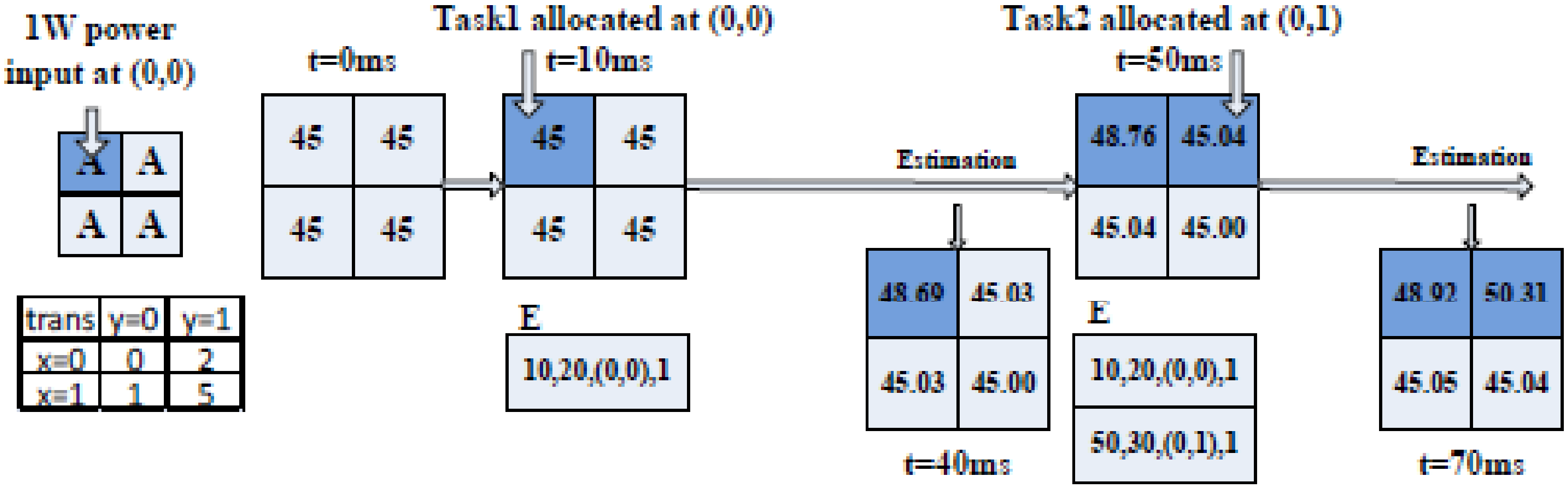
$$T_{t_c} = T_{t_p} + \Delta T_{\Delta t = t_c - t_p}$$

$$\Big\downarrow$$

$$T_{t_n} = T_{t_c} + \Delta T_{\Delta t = t_n - t_c}$$

Our Event Driven Method (Example)

- A Fast Event-Driven Thermal Model for On-line Temperature Prediction



1W power input at (0,0)

A	A
A	A

trans	y=0	y=1
x=0	0	2
x=1	1	5

t=0ms

45	45
45	45

Task1 allocated at (0,0)

t=10ms

45	45
45	45

E

10,20,(0,0),1

Estimation

48.69	45.03
45.03	45.00

t=40ms

Task2 allocated at (0,1)

t=50ms

48.76	45.04
45.04	45.00

E

10,20,(0,0),1

50,30,(0,1),1

Estimation

48.92	50.31
45.05	45.04

t=70ms

Outline

- Introduction & Motivation
- Basic Thermal Model
- Our Event Driven Method
- Heuristic Task Allocation
- Experimental Results
- Conclusion

Heuristic Task Allocation

- Trend of Temperature Transient: One set is the cores with temperature increasing in next interval; Another set is the cores with temperature decreasing in next interval.

$$Weight_{+} = T \times a_{+} \;\; if\ core \in Core_{+}$$
$$Weight_{-} = T \div a_{-} \;\; if\ core \in Core_{-}$$

- The core with lowest weight should be allocated for the task.
- The rationale behind this we use the following criteria:
  - Prefer cooler cores at current point
  - Prefer the cores whose temperature will increase slightly in next time slot
  - Prefer the cores whose temperature will decrease dramatically in next time slot

Outline

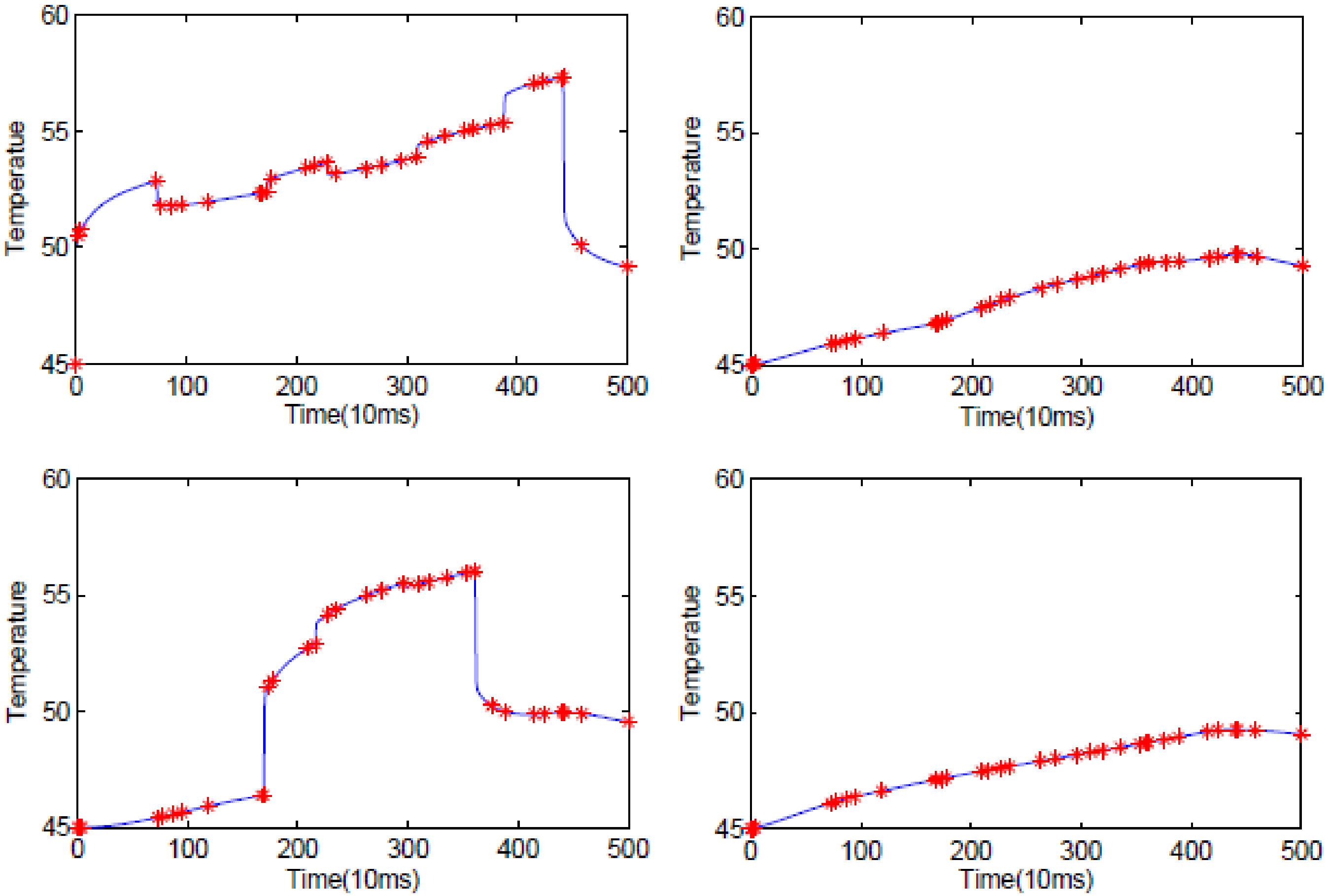
- Introduction & Motivation
- Basic Thermal Model
- Our Event Driven Method
- Heuristic Task Allocation
- Experimental Results
- Conclusion

Experimental Results

- Experimental Setup:
  - 4x4 CMP (Alpha Core)
  - Random Task Arrival in [0s, 60s]
  - Average Power Consumption of Applications in SPEC 2000: [25W, 40W]
  - Task Execution Time: [10ms, 500ms]
  - Task Set contains 200 tasks
  - Initial Chip Temperature: 45
  - Threshold Temperature for DTM: 75
  - If DTM is triggered, task migration will move the task from the hot core to some other cooler idle core. If other cooler cores are not available, the task is blocked and sent back to the waiting list for the next scheduling round. DVFS and stop-go are not used here, however our method can be applied to these DTM policies.

Experimental Results

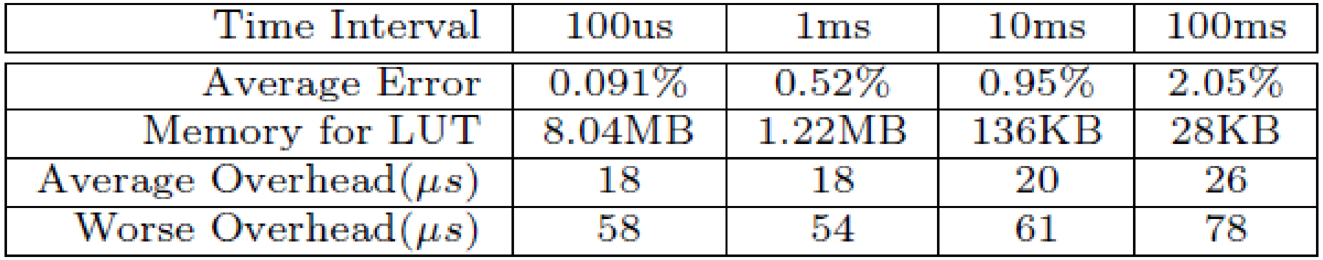
- Validation against HotSpot



NANYANG TECHNOLOGICAL UNIVERSITY 南洋理工大學

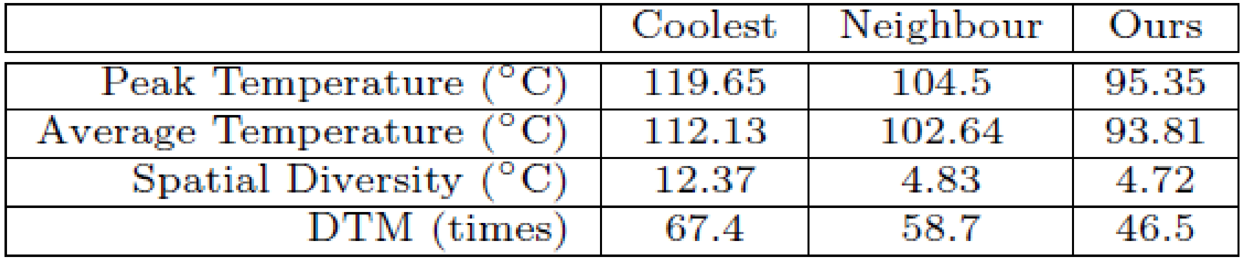
Experimental Results

- The Overhead of Our Method:



Time Interval	100us	1ms	10ms	100ms
Average Error	0.091%	0.52%	0.95%	2.05%
Memory for LUT	8.04MB	1.22MB	136KB	28KB
Average Overhead(μs)	18	18	20	26
Worse Overhead(μs)	58	54	61	78

- Our Algorithms Complexity: O(eN), where e is atomic event number in the event list, and N is the number of core on chip.
- Comparison of Our Method and Others:



	Coolest	Neighbour	Ours
Peak Temperature (°C)	119.65	104.5	95.35
Average Temperature (°C)	112.13	102.64	93.81
Spatial Diversity (°C)	12.37	4.83	4.72
DTM (times)	67.4	58.7	46.5

Outline

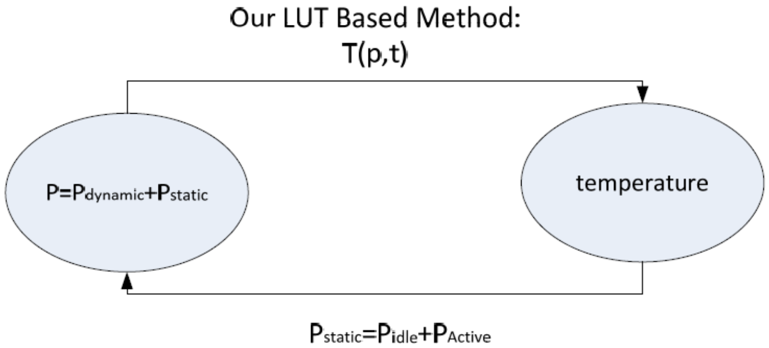
- Introduction & Motivation
- Basic Thermal Model
- Our Event Driven Method
- Heuristic Task Allocation
- Experimental Results
- Conclusion

Conclusion

- Event driven thermal model is much faster and keep the closest results compared with HotSpot.
- Suitable for guiding the dynamic TAS and embedding into OS.
- Heuristic allocation based on predictable method is easier to get better thermal behaviors than other simple heuristic ones that only consider the current thermal distribution on chip (sensor-based method cannot predict future thermal map easily).

Future Work (leakage calibration)

- The leakage power introduces the difficulties and makes the linear problem become the non-linear one due to power-temperature relationship.



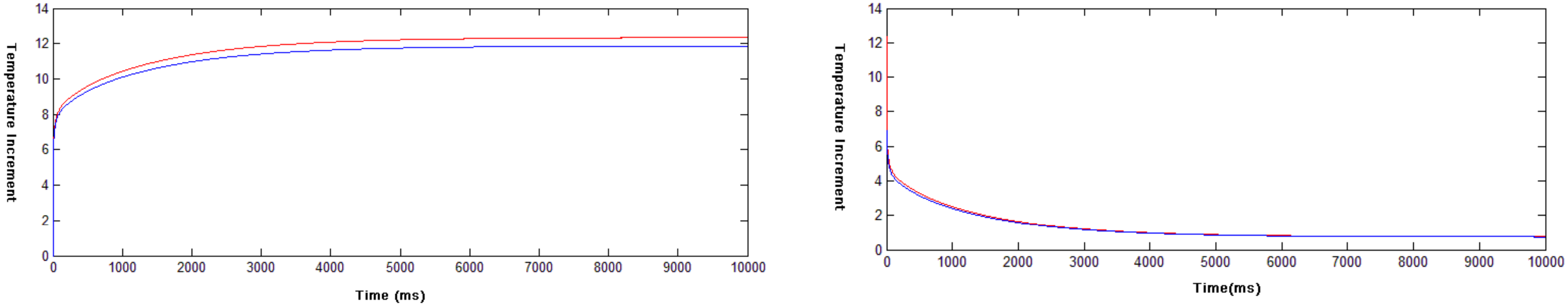
Our LUT Based Method:
T(p,t)

$P=P_{dynamic}+P_{static}$

temperature

$P_{static}=P_{idle}+P_{Active}$

- Heating & Cooling Stage Considering Leakage Power



Thank you!

Q & A

NANYANG
TECHNOLOGICAL
UNIVERSITY

南洋理工大學