In collaboration with Massachusetts Institute of Technology

MWSCAS Abu Dhabi 2016

16-19 October

iMicro مركز أبحاث النظم الدقيقة

Masdar INSTITUTE

Autoscaling of Cores in Multicore Processors Using Power and Thermal Workload Signatures

Rupesh Karn and Ibrahim (Abe) M. Elfadel

Institute Center of Microsystems(iMicro)

Masdar Institute of Science and Technology

Abu Dhabi, UAE

Date: 18th Oct 2016

OUTLINE

- Introduction
- Motivation
- Experimental Setup
- Results
- Conclusion
- Acknowledgements

INTRODUCTION

- Multicore processors have been commonly used for over a decade now.
- In multicore, execution speed depends on shared resources like high-level cache memory, data and address buses and communication between cores.
- Switching cores off and on between sleep and active states increases the dynamic power consumption and temperature of the whole processor.
- To tightly manage the trade-off between power/temperature and performance, advanced scaling algorithms are required.

INTRODUCTION (contd.)

What is scaling ?

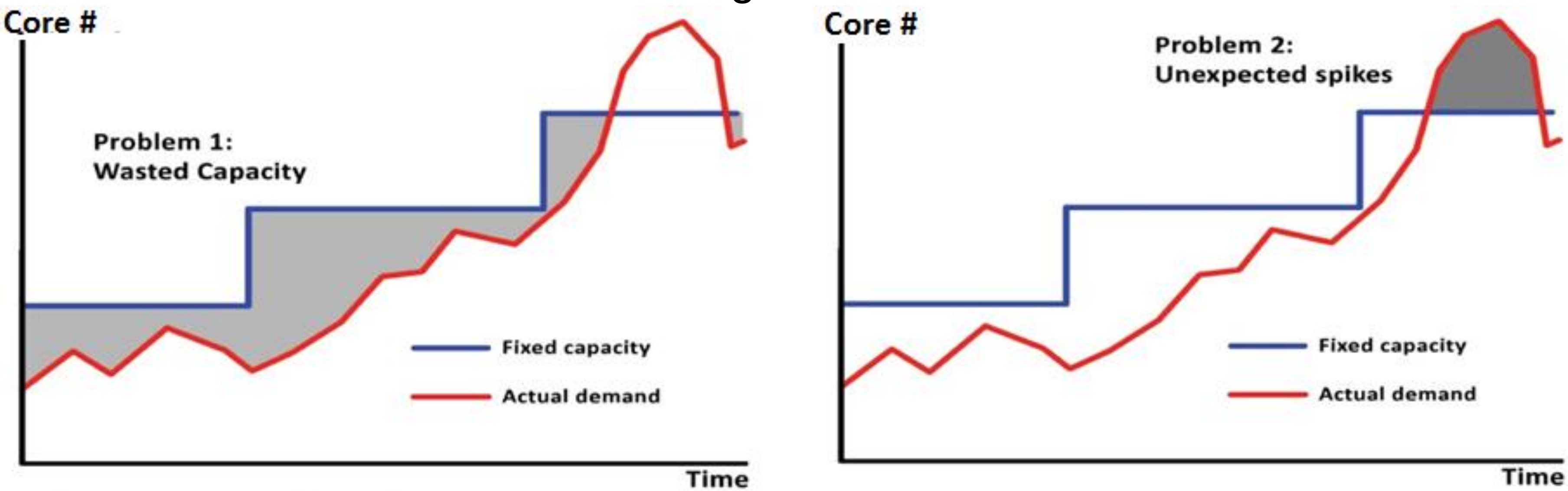
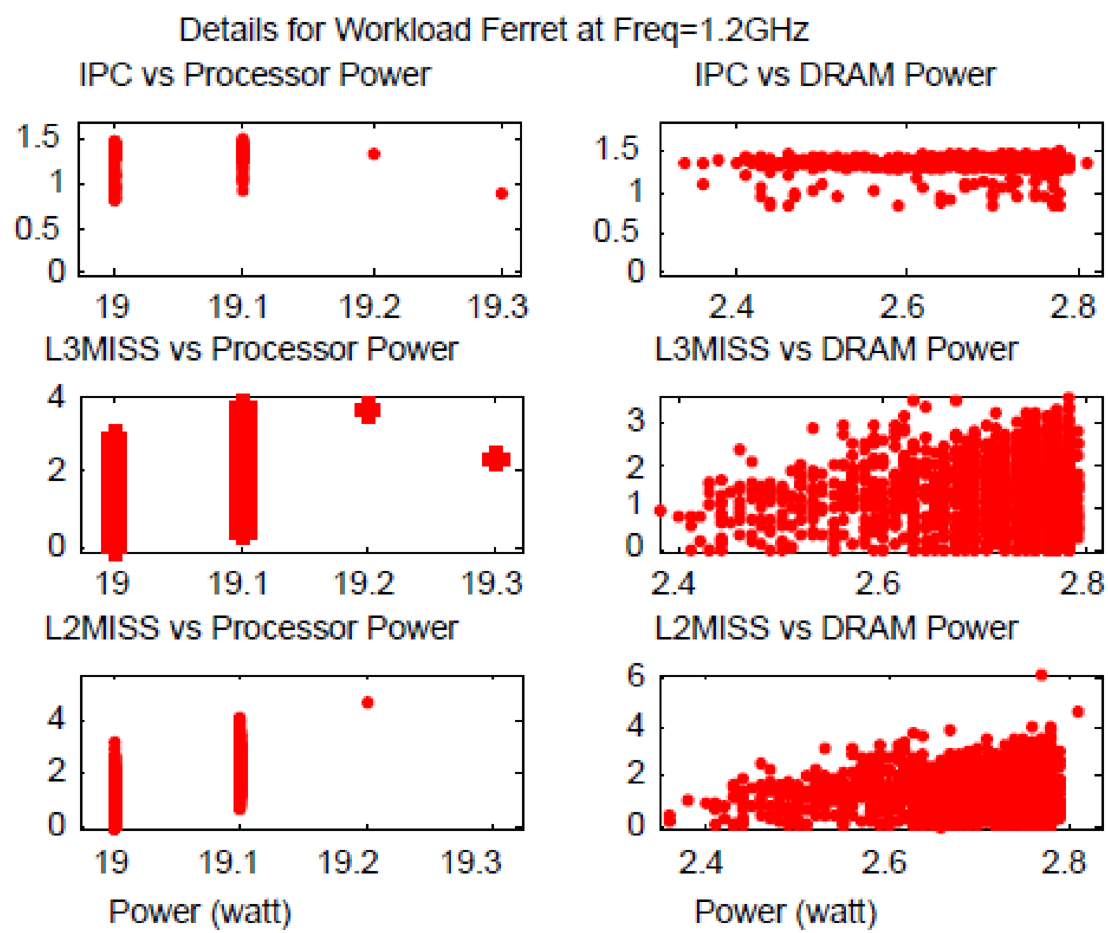


Image Source: http://harish11g.blogspot.com/

- ‘Autoscaling’ refers to a scaling decision that is made autonomously based on:
  - the type of load
  - user specified performance
  - energy/power or temperature setpoint

MOTIVATION

- Traditional scaling has several disadvantages:
  - It doesn't take memory throughput and power into consideration.
  - It only considers core's private resources but not shared ones such as high-level caches used by complex workloads.
  - It uses generic performance counters(e.g., IPC, Clock freq.) irrespective of workload characteristics.



Performance Counter vs Power

RESEARCH CONTRIBUTION

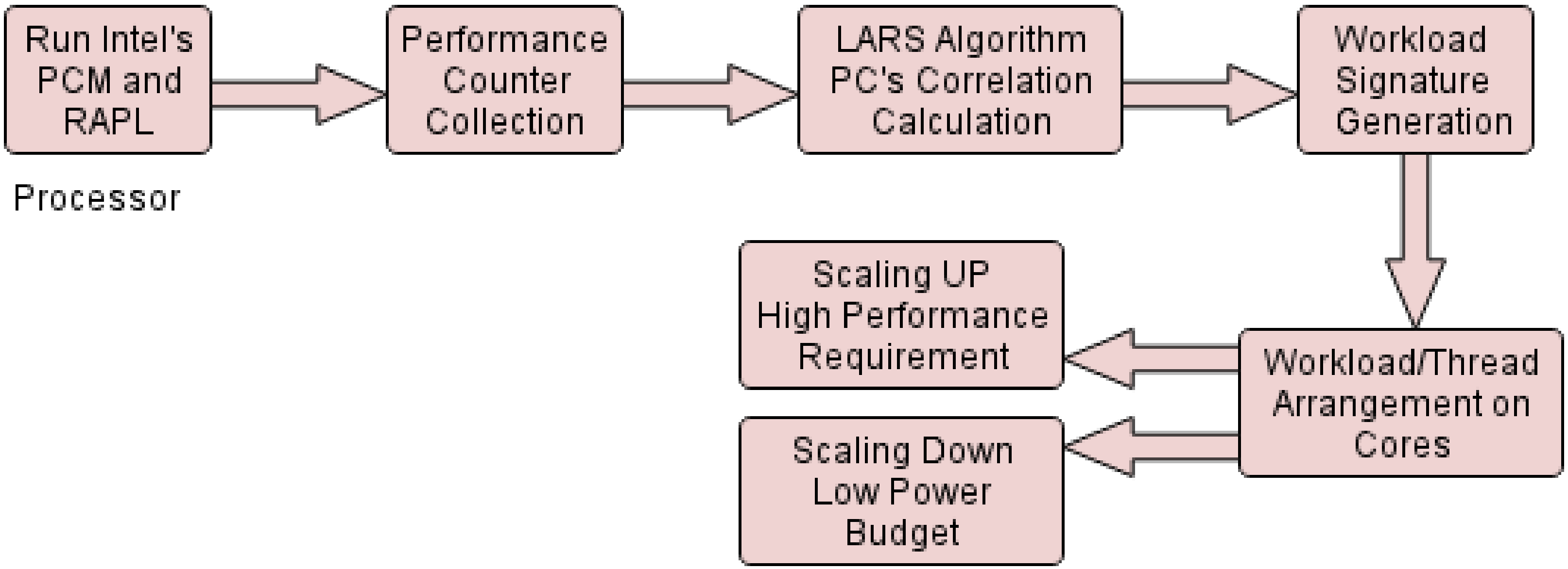
- A real-time power-aware and thermal-aware autoscaling algorithm.
- Taking various performance counters (PCs) into account, e.g., the cache access rate, sleep states, IPC, frequency.
- Power-aware workload signature generation using those PCs.
- Best arrangement of workloads and threads among the cores for high performance under a given power budget.

EXPERIMENTAL SETUP

- Operating System: RED HAT Linux.
- Performance Counters and Temperature Measurement : Intel's PCM.
- Power and Energy Measurement : RAPL Interface.
- Hardware Specification :

Feature	Value
Processor(s)	2x Intel Xeon E5-2630
Number of Cores	6 per processor (12 total)
Frequency	2.3 GHz
L1 I-cache	32 KB/core
L1 D-cache	32 KB/core
L2 Cache	256 KB/core
L3 Cache	15 MB
Main Memory	16 GB DDR3-1333
Bus Speed	1333 MHz

EXPERIMENTAL SETUP (contd.)



FFREQ	CPU Frequency
IPC	Instructions per CPU cycle
L3MISS	L3 cache misses
L2MISS	L2 cache misses
L3HIT	L3 cache hit ratio
L2HIT	L2 cache hit ratio
L3CLK	Ratio of CPU cycles lost due to L3 cache misses
L2CLK	Ratio of CPU cycles lost due to missing L2 cache but still hitting L3 cache
C0RES	C0 Residency State
C6RES	C6 Residency State

LARS Ref: Rupesh R. Karn and Ibrahim (Abe) M. Elfadel, "Multicore power proxies using least-angle regression," In the 2015 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 2872-2875, Lisbon, Portugal, May 2015.

Workloads of PARSEC & SPEC CPU 2006 benchmarks are used.

WORKLOAD SIGNATURE

- Any workload can be characterized as memory intensive, CPU intensive or I/O-intensive.
- The correlation coefficients between power/temperature of the cores and PCs triggered for the workloads have been calculated.
- Least Angle Regression is used for correlation calculation.

$$corr(X,Y) = X'Y$$

X is a matrix containing all PCs' occurrences and Y is the vector containing either power or temperature values.

Workload Signatures for Power

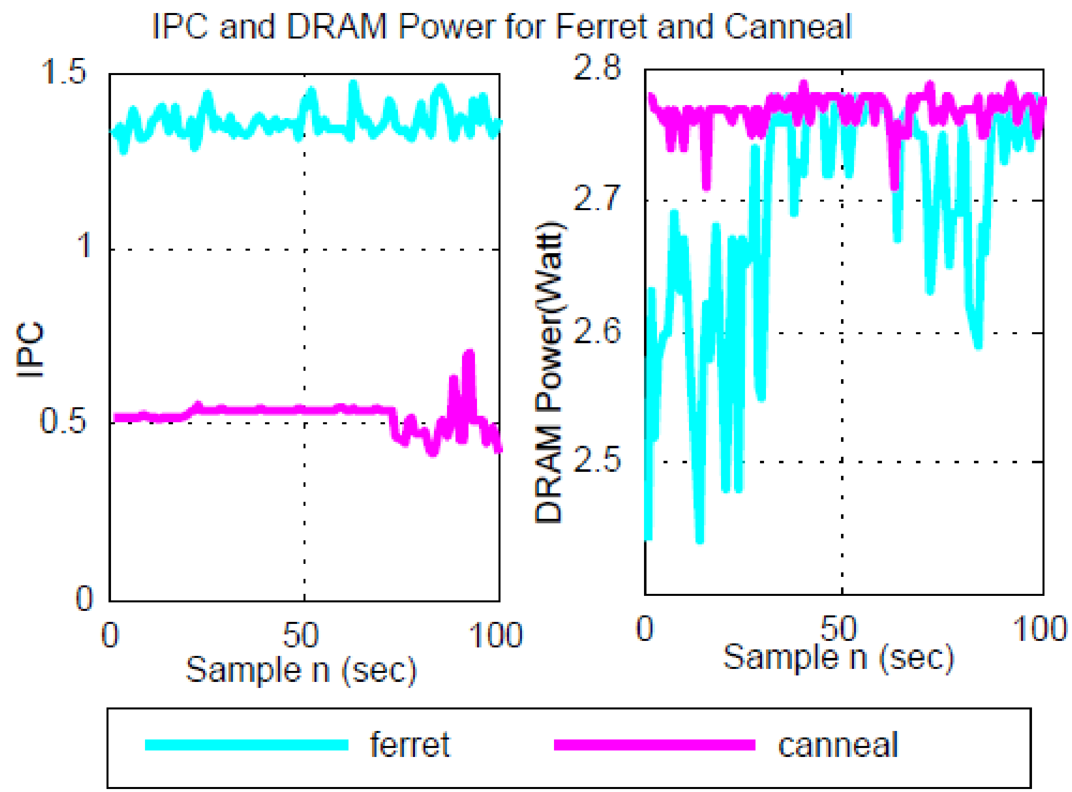
Workload	PCs' Correlation Ranking				
	1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}
Bodytrack	CORES	FFREQ	L2MISS	IPC	L3MISS
Canneal	CORES	L2MISS	FFREQ	L3CLK	L2HIT
Ferret	CORES	IPC	L2MISS	FFREQ	L2HIT
X264	CORES	L2MISS	FFREQ	IPC	L3MISS
Gamess	CORES	FFREQ	IPC	L3HIT	L2MISS
Libquantum	CORES	L2MISS	IPC	FFREQ	L3CLK
Gobmk	CORES	FFREQ	IPC	L3HIT	L2MISS

Workload Signatures for Temperature

Workload	PCs' Correlation Ranking				
	1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}
Bodytrack	CORES	T(n-1)	FFREQ	L2MISS	IPC
Canneal	CORES	T(n-1)	L2MISS	FFREQ	IPC
Ferret	CORES	IPC	T(n-1)	FFREQ	L2HIT
X264	CORES	T(n-1)	L2MISS	FFREQ	IPC
Gamess	CORES	T(n-1)	FFREQ	IPC	L3HIT
Libquantum	CORES	T(n-1)	L2MISS	L3MISS	IPC
Gobmk	CORES	T(n-1)	FFREQ	IPC	L2MISS

WORKLOAD SIGNATURE (contd.)

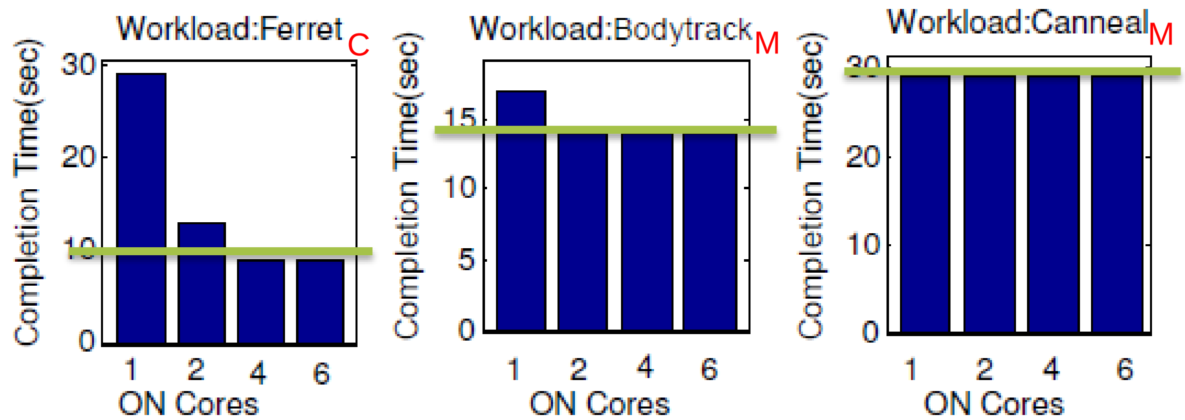
- The rankings/signatures vary notably from one workload to another.
- For memory-intensive workloads (‘canneal’ and ‘x264’), the cache miss rate (L2MISS) is the one most correlated with power and temperature.
- For CPU-intensive workloads (e.g., ‘ferret’), it is the IPC that is most correlated.



Workload Signatures of ‘ferret’C and ‘canneal’M

SCALING UP FOR SEPARATE WORKLOAD

- Separate workloads that run in multiple threads may fully utilize a multicore processor.



Runtime Comparison of Workloads

- For memory intensive workloads (e.g 'bodytrack' and 'canneal'), the completion time varies only slightly.

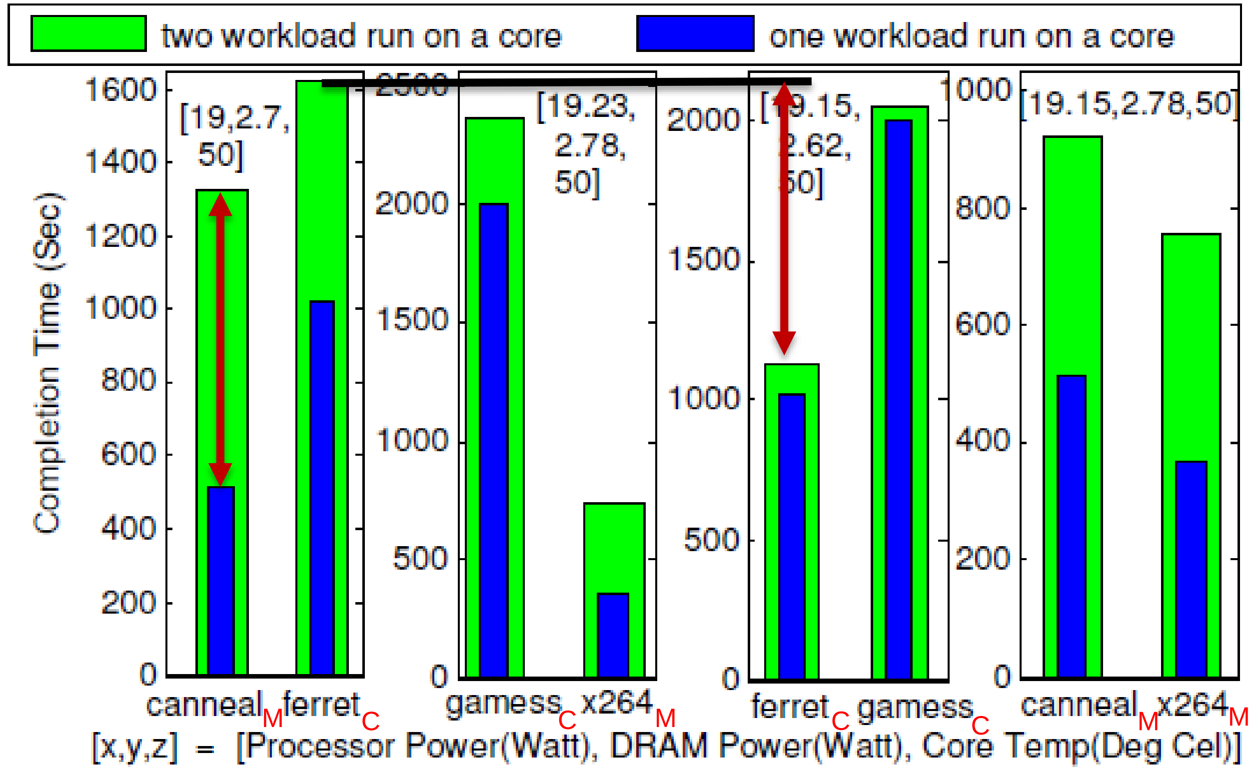
- CPU-intensive workload: Performance depends mainly on instruction throughput.

- Since the instruction pipeline is private to each core, 'ferret' can fully utilize the core's resources and run its threads in parallel with high performance.

- Core scaling increases performance significantly for CPU-intensive.

SCALING DOWN FOR MULTIPLE WORKLOADS

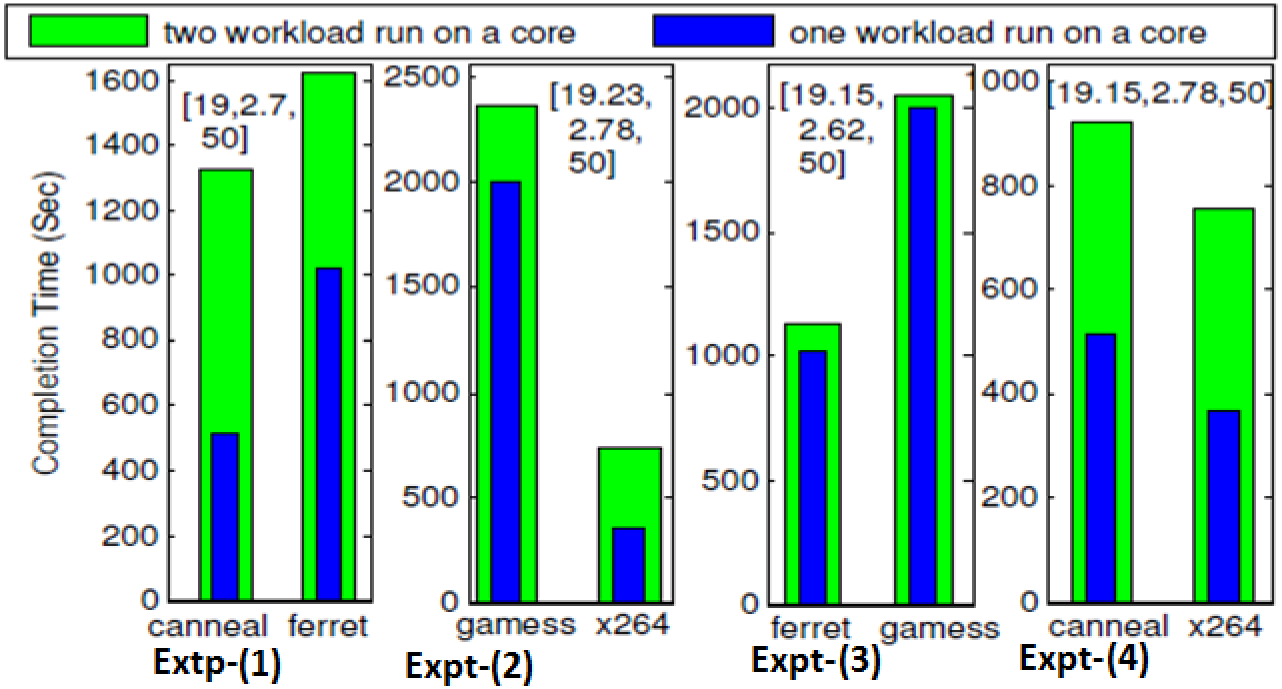
- The overall performance can be improved by merging the workloads onto particular core based on its signatures.
- Two memory intensive (canneal & X264) and two CPU intensive (ferret & games) workloads are used for experiment.
- The shortest completion time can be observed in the third and fourth chart/set.

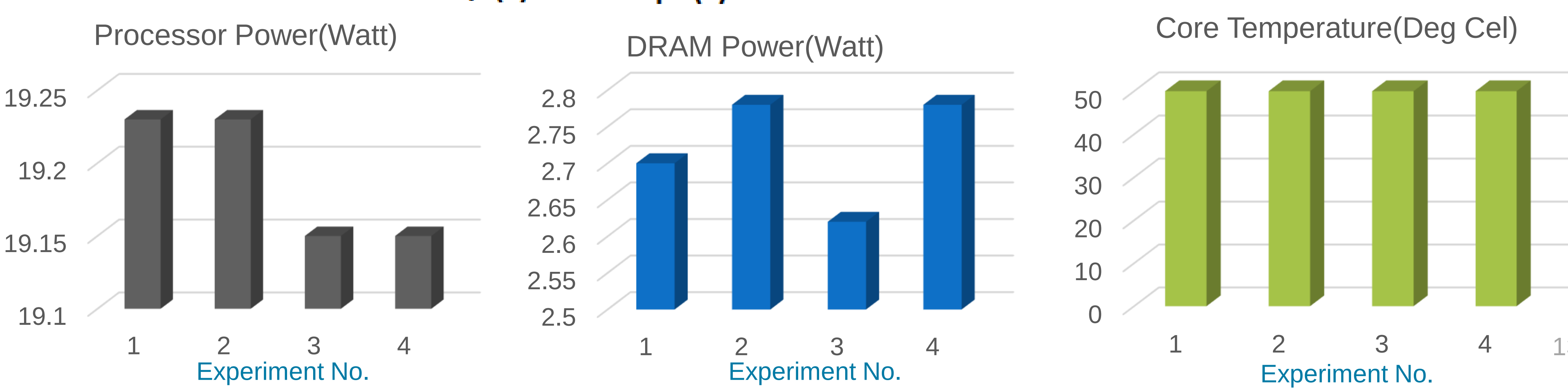


Runtime Comparison for Multi-Workload Execution on Single Cores

- Two workloads running on the same core can provide relatively high performance assuming that they share the same signature.

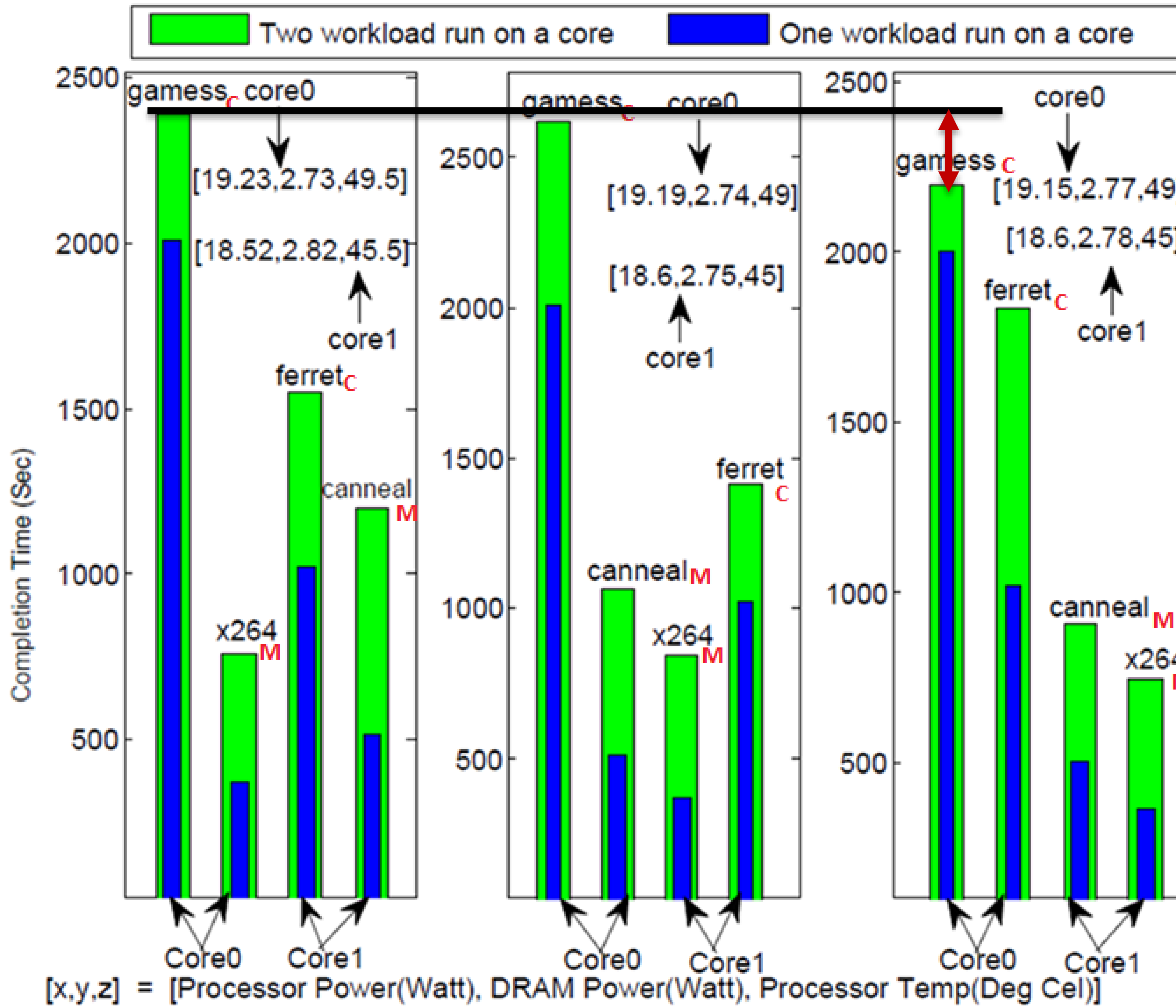
SCALING DOWN FOR MULTIPLE WORKLOADS



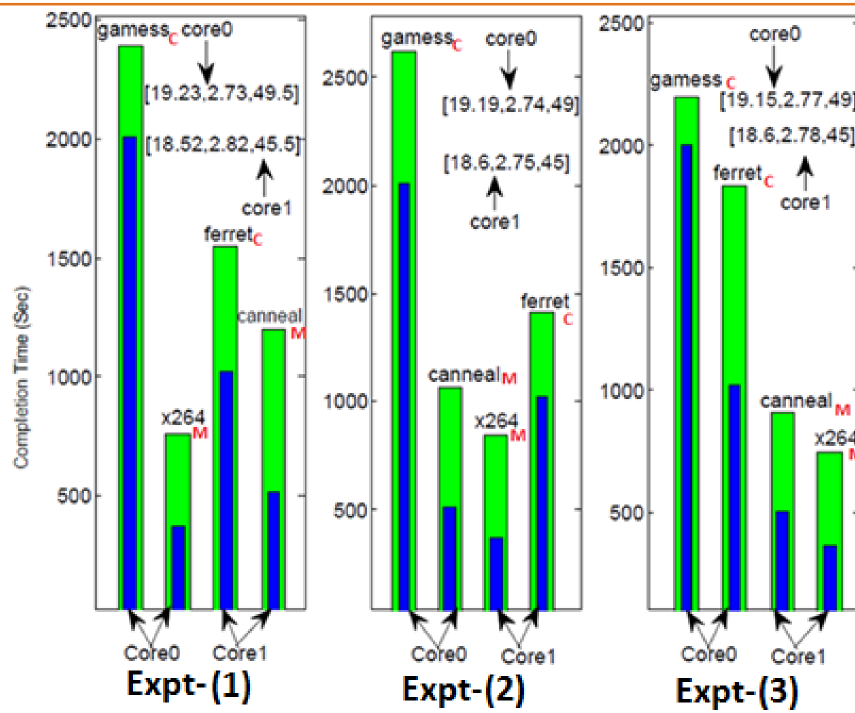


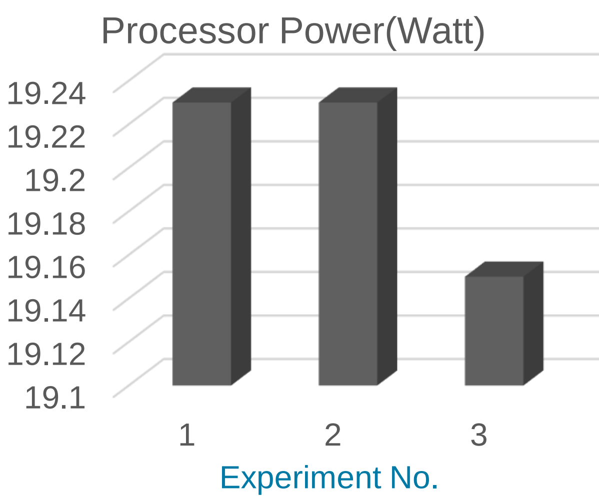
SCALING DOWN FOR MULTIPLE WORKLOADS

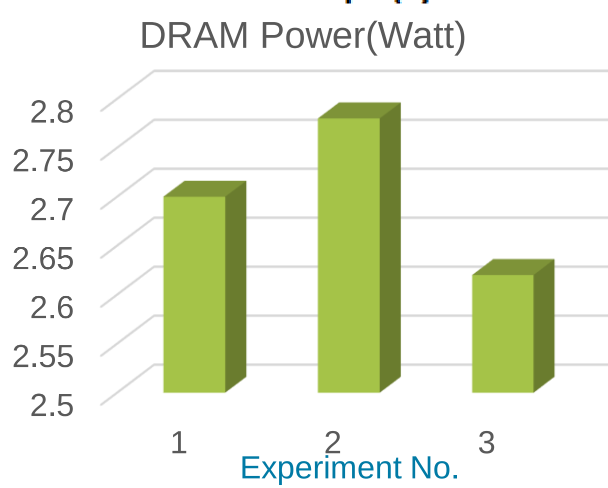
- Similar experiments have been conducted for multicore setups.
- The shortest completion time is observed for the third set/chart.
- The CPU-intensive workloads (‘ferret’ and ‘gamess’) run on one core $k = 0$ and the memory-intensive workloads (‘canneal’ and ‘x264’) run on another core $k = 1$.

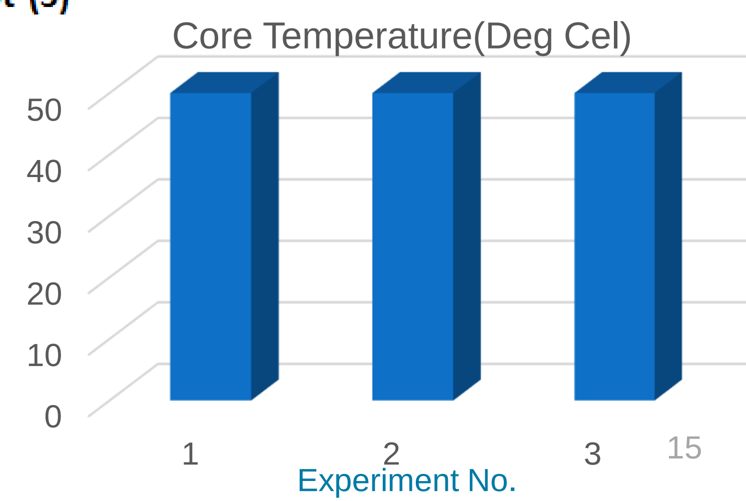


SCALING DOWN FOR MULTIPLE WORKLOADS









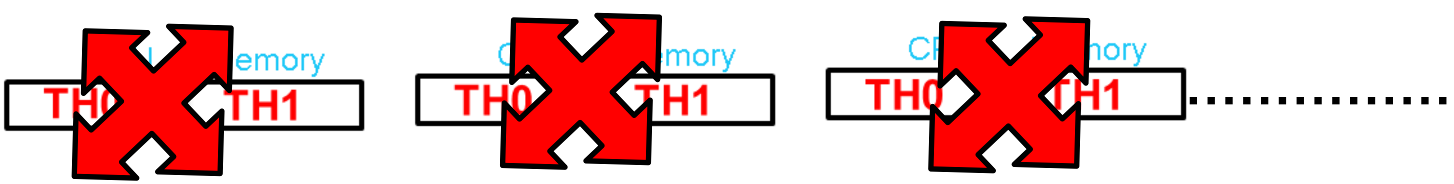
SCALING ACROSS THREADS

- Performance can be even enhanced by arranging the threads of workloads based on its signatures.

- Note that complex, large-scale workloads comprise many different computational phases.

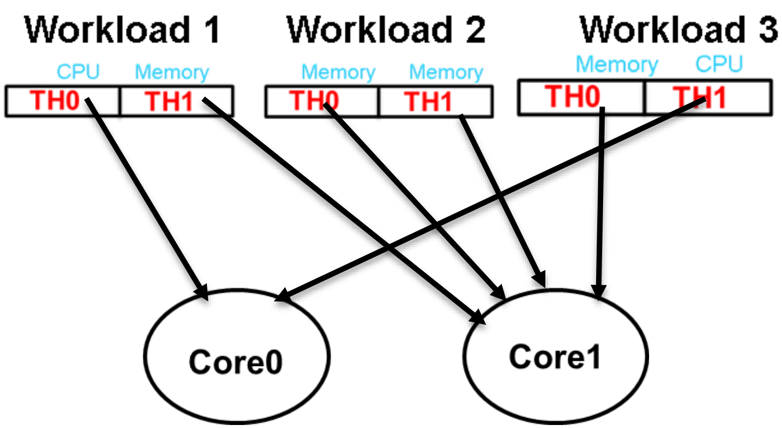
- E.g. the workload ‘ferret’ mimics applications tailored for content-based similarity search of feature-rich data such as audio, images, video and 3D shapes.

- For ‘ferret’, in the early phase the threads are CPU intensive while at later stages the threads become memory intensive.

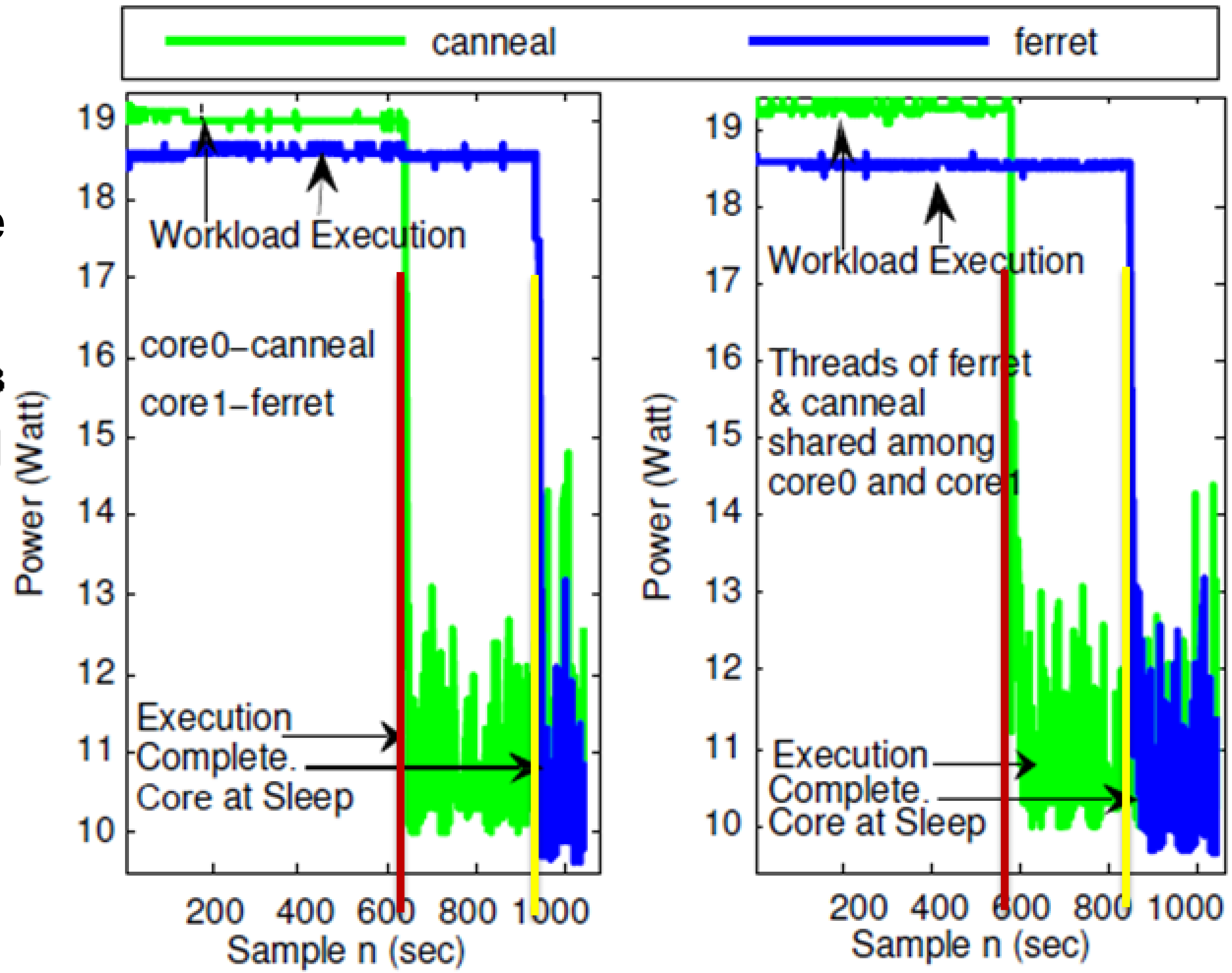


SCALING ACROSS THREADS

- For improved performance, run threads of different workloads but with the same (thread's) signature on a same core.



- This simple but effective technique enables shorter completion times of workloads, along with similar power consumption.



Power and Runtime Comparison for Thread Swapping Across Workloads

CONCLUSION

- Power-aware workload signatures are defined using least angle regression on hardware-based performance counters.

- Signatures enable the accurate prediction of workload characteristics.

- They are used to assign the workloads and their threads to particular cores to improve the overall performance and/or to lower the power consumption and temperature.

- The proposed autoscaling technique can be applied to “workload driven autoscaling of virtual machines” on cloud servers.

- To the best of our knowledge, this is the first work that determines and applies power-aware workload signatures in real-time to the autoscaling of cores in multicore processors.

ACKNOWLEDGEMENTS

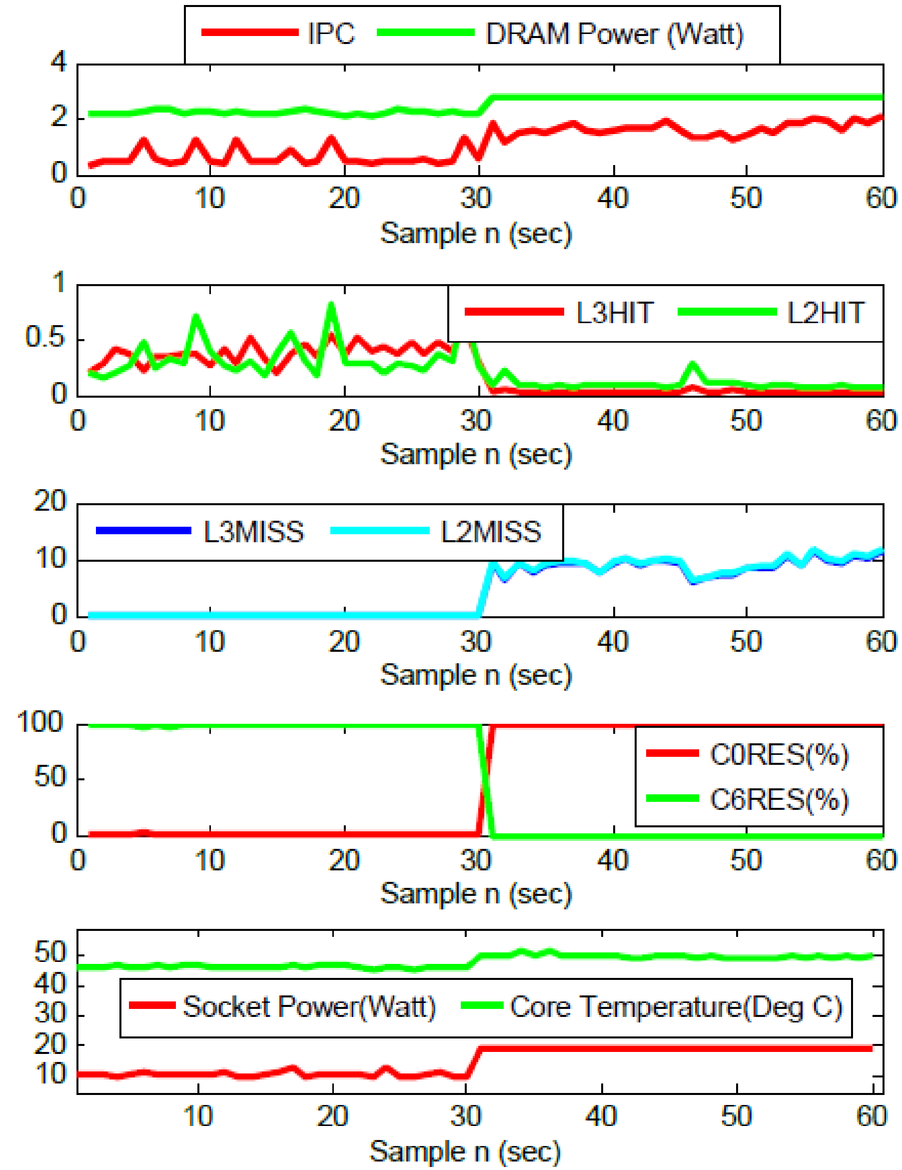
- Industrial Liaisons:
  - Intel:
    - Andrew D. Henroid
  - IBM:
    - Architecture:
      - P. Bose
      - A. Buyuktosunoglu
      - C. Isci
    - Software:
      - M. A. Silva
      - H. Franke
      - P. Kudva
      - G. Kandiraju

- Mubadala (ATIC)

THANK YOU

PCs PROFILE Vs WORKLOADS

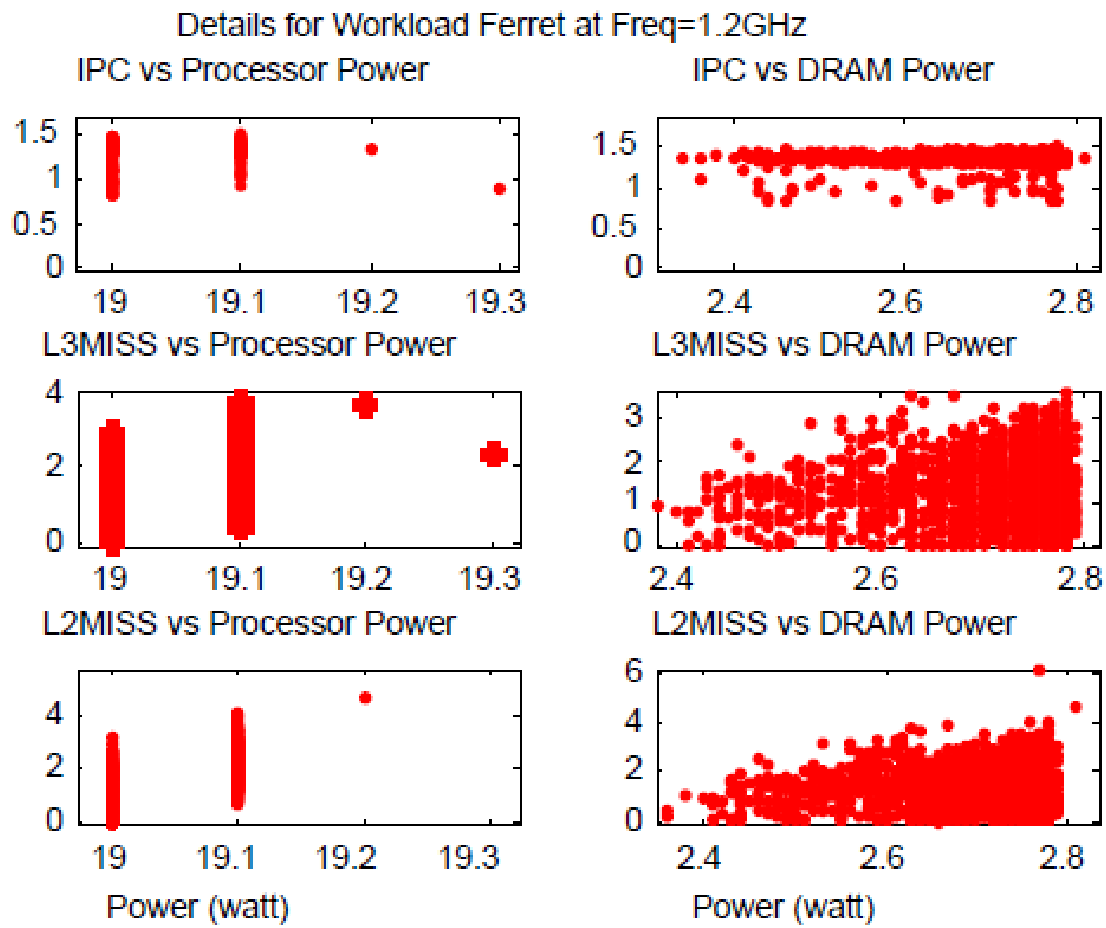
- Workload: Ferret.
- PCs samples collected per second.
- Increment in DRAM & processor power along with higher IPC and temperature is observed after the load acts on the core.
- The processor also switches from C6 (sleeping state) to C0 (active state).



PCs PROFILE Vs POWER

- Workload: Ferret.
- PCs samples collected per second.
- The plots reveal typically low P-states for the processor but high P-states for the DRAM.
- The values of PCs remain for longer times at low P-state for the processor but at high P-state for the DRAM.

Memory throughput along with memory power consumption should be considered when distrusting workloads via autoscaling.



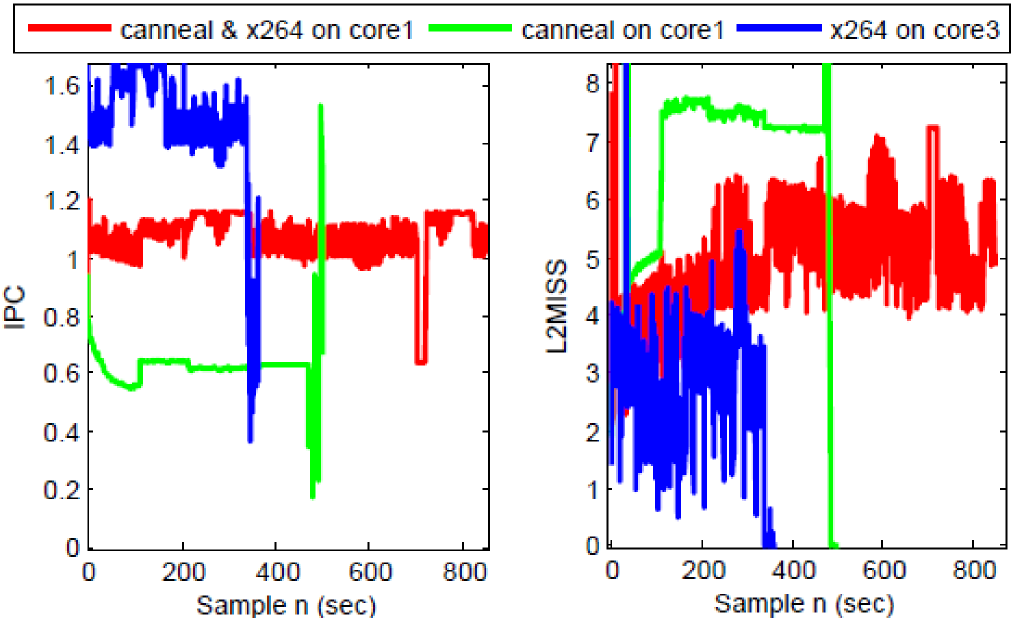
Performance Counter vs Power

POWER & ENERGY COMPARISION

Workloads Completion Time

Workload	Completion Time ($\triangle t$) (sec)	Power (Watt)
'canneal', 'x264' on $core_1$; other cores idle	'canneal': 843; 'x264': 708	36
'canneal' on $core_1$; 'x264' on $core_3$; other cores idle	'canneal': 497; 'x264': 362	36.8
'bodytrack', 'ferret' on $core_1$; other cores idle	'bodytrack': 896; 'ferret': 1419	36.4
'bodytrack' on $core_1$; 'ferret' on $core_3$; other cores idle	'bodytrack': 407; 'ferret': 1015	37.2

Completion Time $(\triangle t) = t_{\text{Final Output from Workload}} - t_{\text{Workload Run Request}}$



IPC and L2MISS for 'ferret' and 'canneal'

- In row 1, $Energy = 36W \ x \ 843s = 30.348KJ$.
- In row 2, $Energy = 36.8W \ x \ 497s = 18.289KJ$.

- It is economical to run different workloads on separate cores, which avoids excessive use of the core's shared resources.