Modeling the Energy Efficiency of Heterogeneous Clusters

Lavanya Ramapantulu, Bogdan Marius Tudor, Dumitrel Loghin, Trang Vu, Yong Meng Teo

Department of Computer Science

National University of Singapore

10th September 2014

43rd International Conference on

Parallel Processing, Minneapolis, MN, USA

Outline

- Motivation
- Objective
- Methodology
- Analysis
- Conclusions

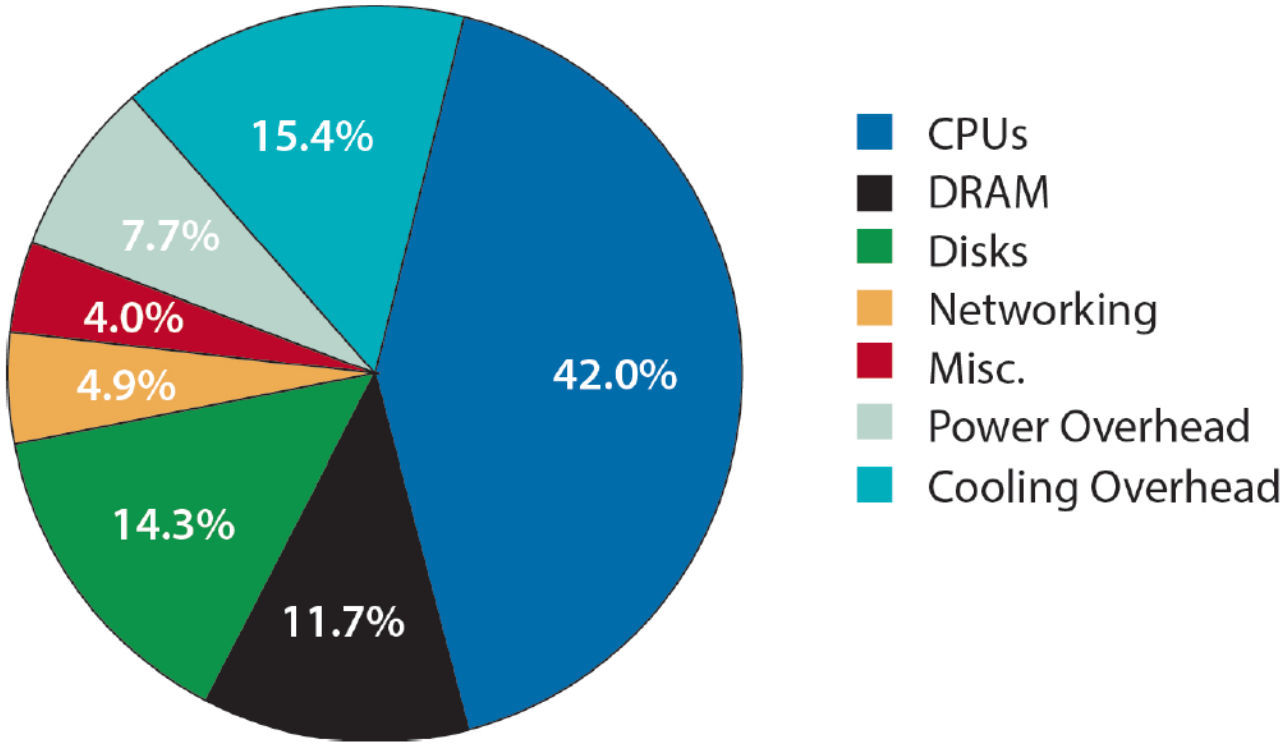
Energy Use of Datacenters

- Energy consumption of large-scale data centers and its costs are significant
  - 2006 - 6,000 data centers in US consumed 61×10^9 KWh of energy, 1.5% of all electricity consumption, at a cost of $4.5 billion
  - 2006-2011 - from 7 GW to 12 GW, 10 new power plants

- 1998-2007: performance of supercomputers (+7,000%) has increased 3.5 times faster than their operating efficiency* (+2,000%)

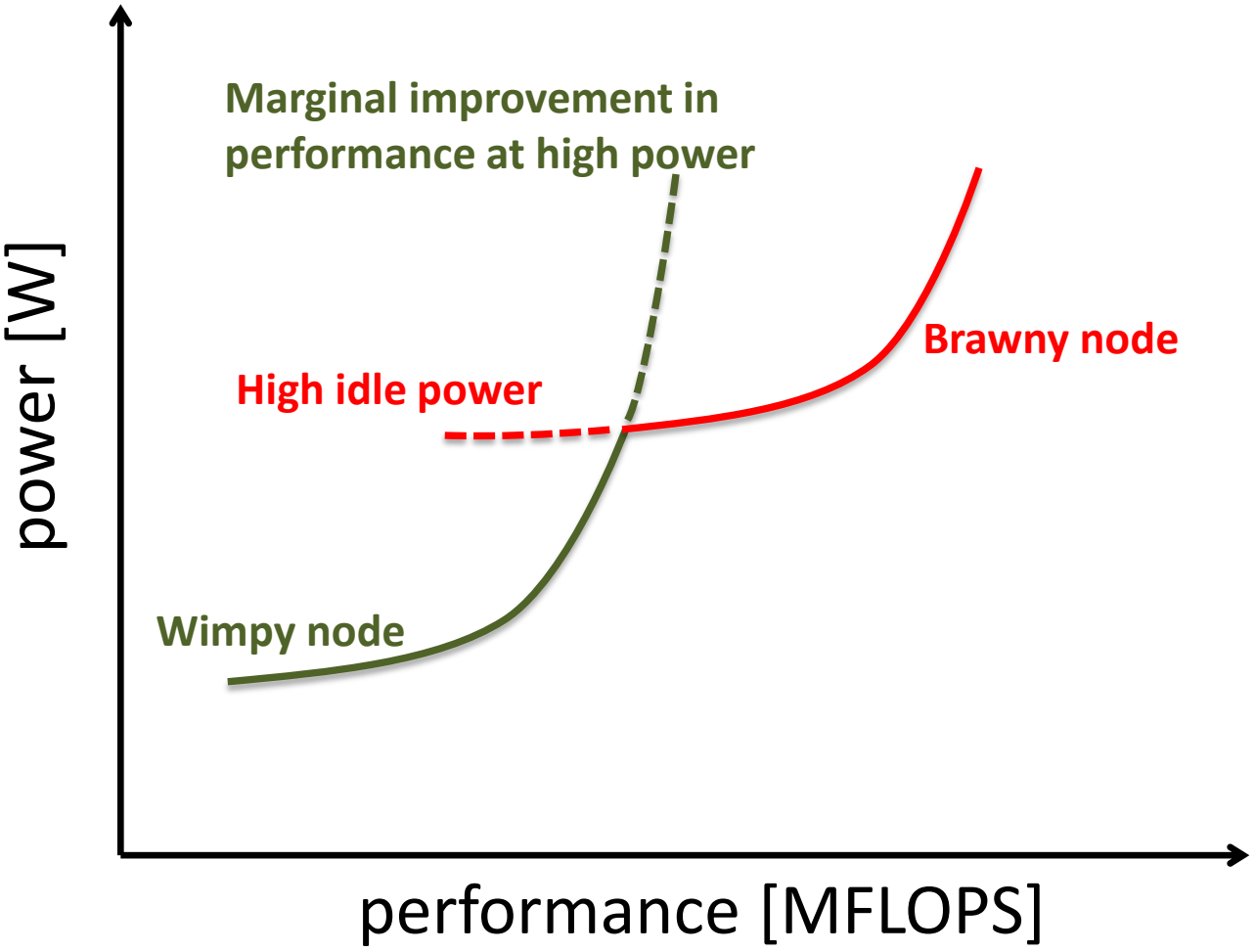
*operating efficiency of a system = *performance per Watt of power*

Datacenter Power Usage



Source: L. Barroso, J Clidaras and U. Hölzle, *The Datacenter as a Computer: An Introduction to Warehouse-Scale Machines*, Morgan Claypool, 2013

Wimpy vs Brawny Servers [Gupta et al. 2013]



Intra-node Heterogeneity

- KnightShift: [Wong et al. 2012]
  - Low utilizations, lower energy proportionality
  - Knight responds to low-utilization requests
  - Enables two energy-efficient operating regions

- Thin servers with smart pipes: [Lim et al. 2013]
  - Accelerator for memcached
  - 6X-16X power-performance improvement

Inter-node Heterogeneity

- Dynamic request allocation on heterogeneous clusters
  - Throughput vs. power [Heath et al. 2005]
  - Pikachu: dynamic load balancing among fast and slow nodes for MapReduce [Gandhi et al. 2013]
- Static analysis of single workload on heterogeneous clusters
  - **Unexplored from energy-time performance perspective**

Objective

- For a given application with a power budget, to determine energy efficient heterogeneous configurations that meet an execution time deadline.
  - Energy efficient configurations meet a given deadline with the minimum energy

Contributions

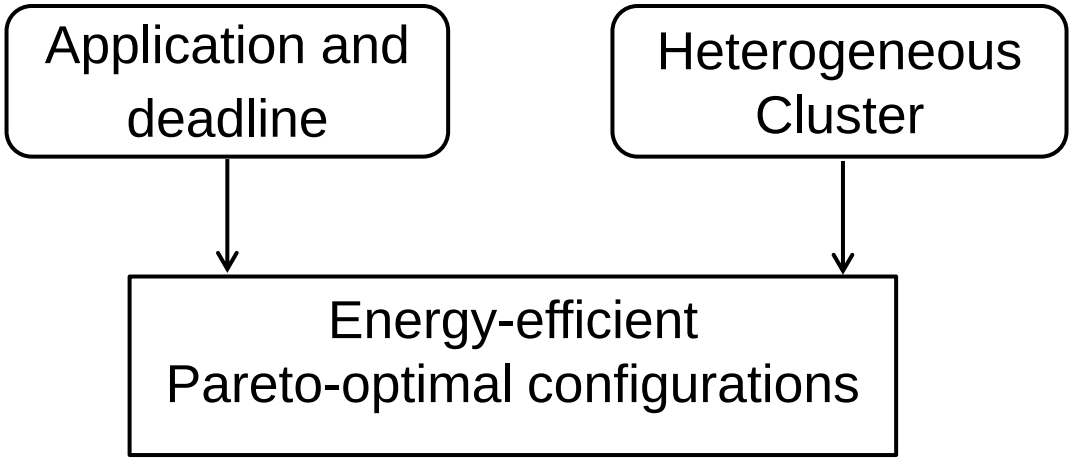
- A measurement-based analytical model to determine energy efficient configurations on a mix of heterogeneous nodes
  - Meets a deadline with minimum energy

- Our analysis shows that energy-deadline Pareto frontier consisting of heterogeneous mixes is almost always more energy-efficient than homogeneous clusters

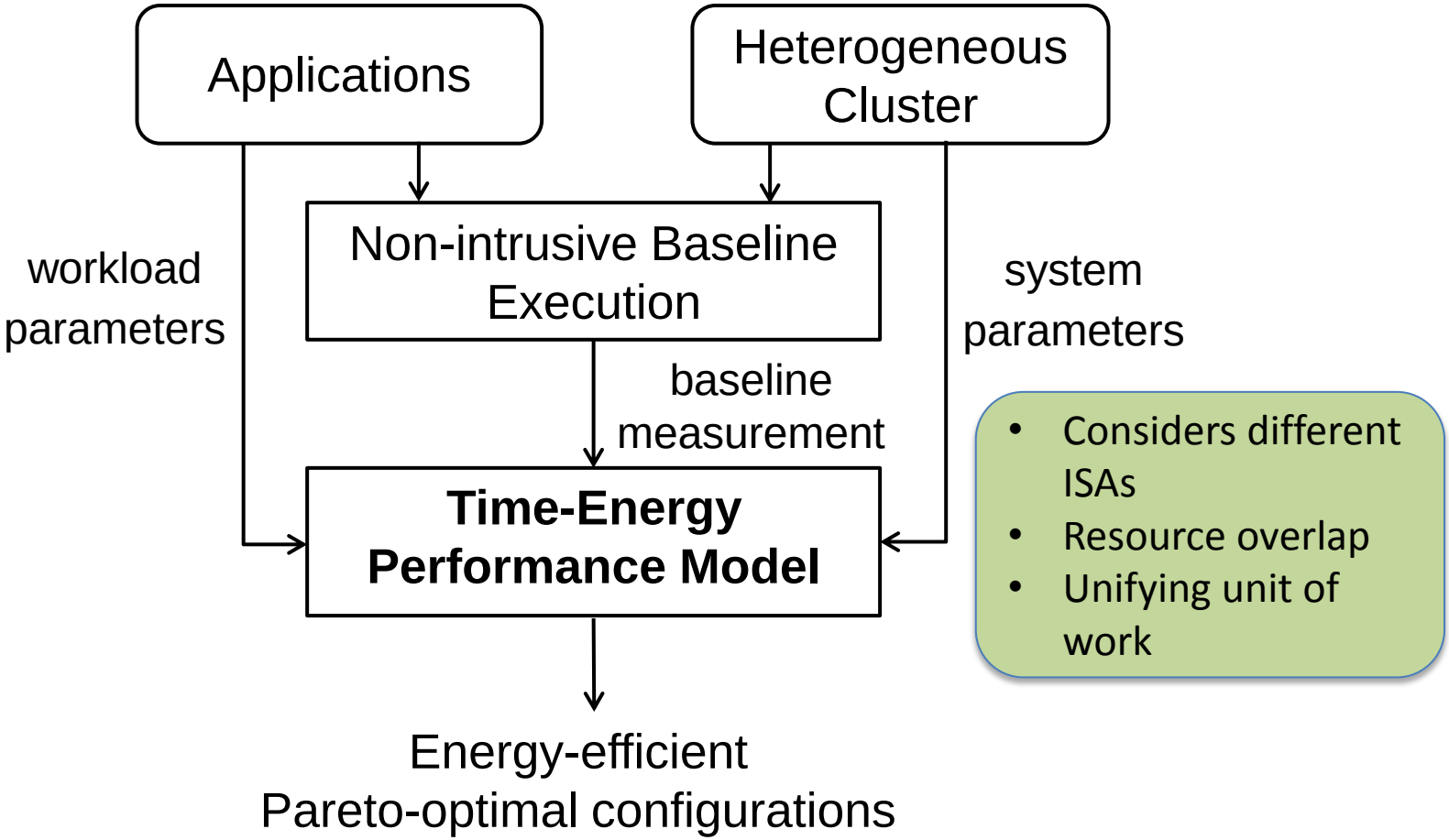
Outline

- Motivation
- Objective
- Methodology
- Analysis
- Conclusions

Approach


Application and deadline
Heterogeneous Cluster
Energy-efficient Pareto-optimal configurations

Approach


Applications
Heterogeneous Cluster
Non-intrusive Baseline Execution
workload parameters
system parameters
baseline measurement
Time-Energy Performance Model
Energy-efficient Pareto-optimal configurations
• Considers different ISAs
• Resource overlap
• Unifying unit of work

Applications

Broad range of datacenter application domains

Domain	Program	Problem Size
HPC	EP	2,147,483,648 random numbers
Web Server	memcached	600,000 GET/SET operations
Streaming video	x264	600 frames 704 × 576
Financial	Black-scholes	500,000 stock options
Speech recognition	Julius	2,310,559 samples
Web security	RSA-2048	5000 keys verifications

Heterogeneous System

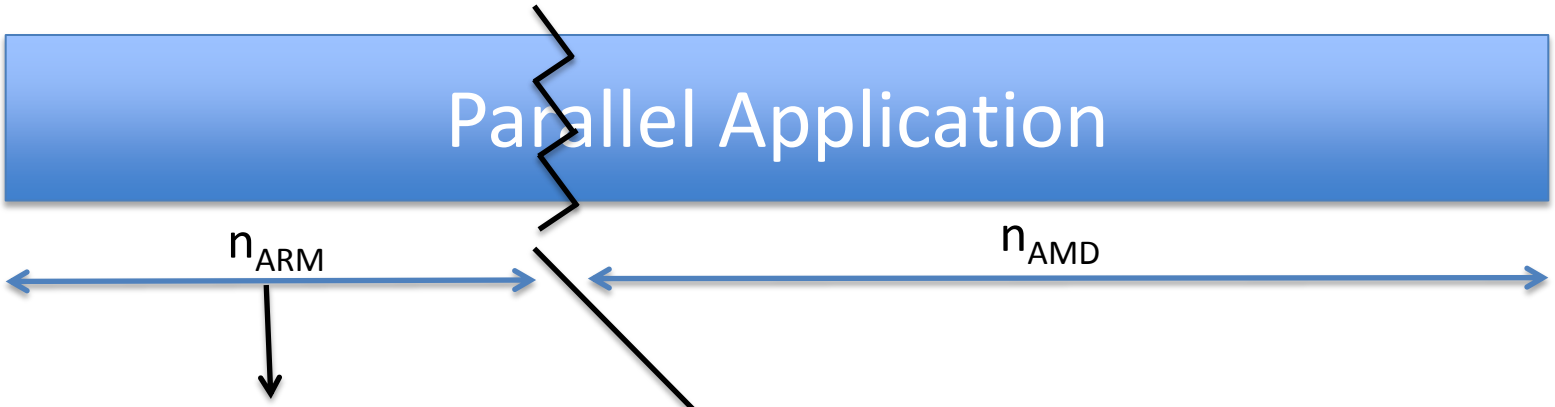
- AMD K10, x86_64
- six-core, 0.8 to 2.1GHz

- ARM v7-A Cortex-A9
- quad-core, 0.2 to 1.4GHz

Baseline Execution

- Measurements needed only for a single node, for each type of node
  - non-intrusive hardware performance counters
- Execute the program for a very small problem size
  - measure instructions, computation cycles and stall cycles
  - Ex: measure instructions per GET operation of memcached
- Execute micro-benchmarks to measure active and stall power of processor cores

Execution Time Model



within a type of node workload is equally divided

$$T(n_{ARM}) \approx \frac{T_{ARM}(1)}{n_{ARM}}$$

match the execution rates between ARM and AMD nodes

$$T(n_{ARM}) \approx T(n_{AMD})$$

$$T(1) \approx \max(T_{CPU}, T_{I/O}) \text{ [CPU and I/O overlap]}^*$$

$$T_{CPU} \approx T_{core,work} + T_{core,stall}$$

* B.M. Tudor and Y.M. Teo, On Understanding the Energy Consumption of ARM-based Multicore Servers, Proceedings of ACM SIGMETRICS, pp 267-278, Carnegie Mellon University, Pittsburgh, USA, June 17 - 21, 2013

Execution Time Model

- $T_{core,work} \approx \frac{\text{computational work cycles}}{\text{clock frequency}}$

- $T_{core,stall} \approx \frac{\text{stall cycles due to memory contention}}{\text{clock frequency}}$

  - Stall cycles increase linearly with
    - increase in core clock frequency
    - increase in the number of cores

Energy Model

- Total Energy = $E_{ARM} \times n_{ARM} + E_{AMD} \times n_{AMD}$
- $E_{node} = E_{core} + E_{mem} + E_{I/O} + E_{idle}$
- $E_{core} = P_{core,act} \times T_{core,work} + P_{core,stall} \times T_{core,stall}$
  - Power × Time
  - uses execution time model
  - measured values for $P_{core,act}$, $P_{core,stall}$

Model Summary

Execution Time Model	
T	$\max(T_{ARM}, T_{AMD})$
T_{ARM}	$\max(T_{CPU,ARM}, T_{I/O,ARM})$
$T_{CPU,ARM}$	$\max(T_{core,ARM}, T_{mem,ARM})$
$T_{core,ARM}$	$\frac{I_{core,ARM} \times (WPI_{ARM} + SPI_{core,ARM})}{f_{ARM}}$
$T_{mem,ARM}$	$\frac{I_{core,ARM} \times (WPI_{ARM} + SPI_{mem,ARM})}{f_{ARM}}$
$T_{i/o,ARM}$	$\max(T_{I/O,ARM}, 1/\lambda_{I/O})$
Energy Model	
E	$E_{ARM} + E_{AMD}$
E_{ARM}	$(E_{core,ARM} + E_{mem,ARM} + E_{I/O,ARM} + E_{idle,ARM}) \times n_{ARM}$
$E_{core,ARM}$	$(P_{core,act,ARM} \times T_{act,ARM} + P_{core,stall,ARM} \times T_{stall,ARM}) \times c_{act,\ ARM}$

Model Validation

Program	Configuration		Execution time	Energy
	ARM nodes	AMD nodes	error[%]	error[%]
EP	8	1	3	10
	8	0	3	2
memcached	8	1	10	8
	8	0	3	1
x264	8	1	11	10
	8	0	13	11
blackscholes	8	1	4	7
	8	0	4	13
Julius	8	1	13	1
	8	0	1	2
RSA-2048	8	1	2	8
	8	0	1	12

Outline

- Motivation
- Objective
- Methodology
- Analysis
- Conclusions

Performance-to-Power Ratio

memory bound on ARM

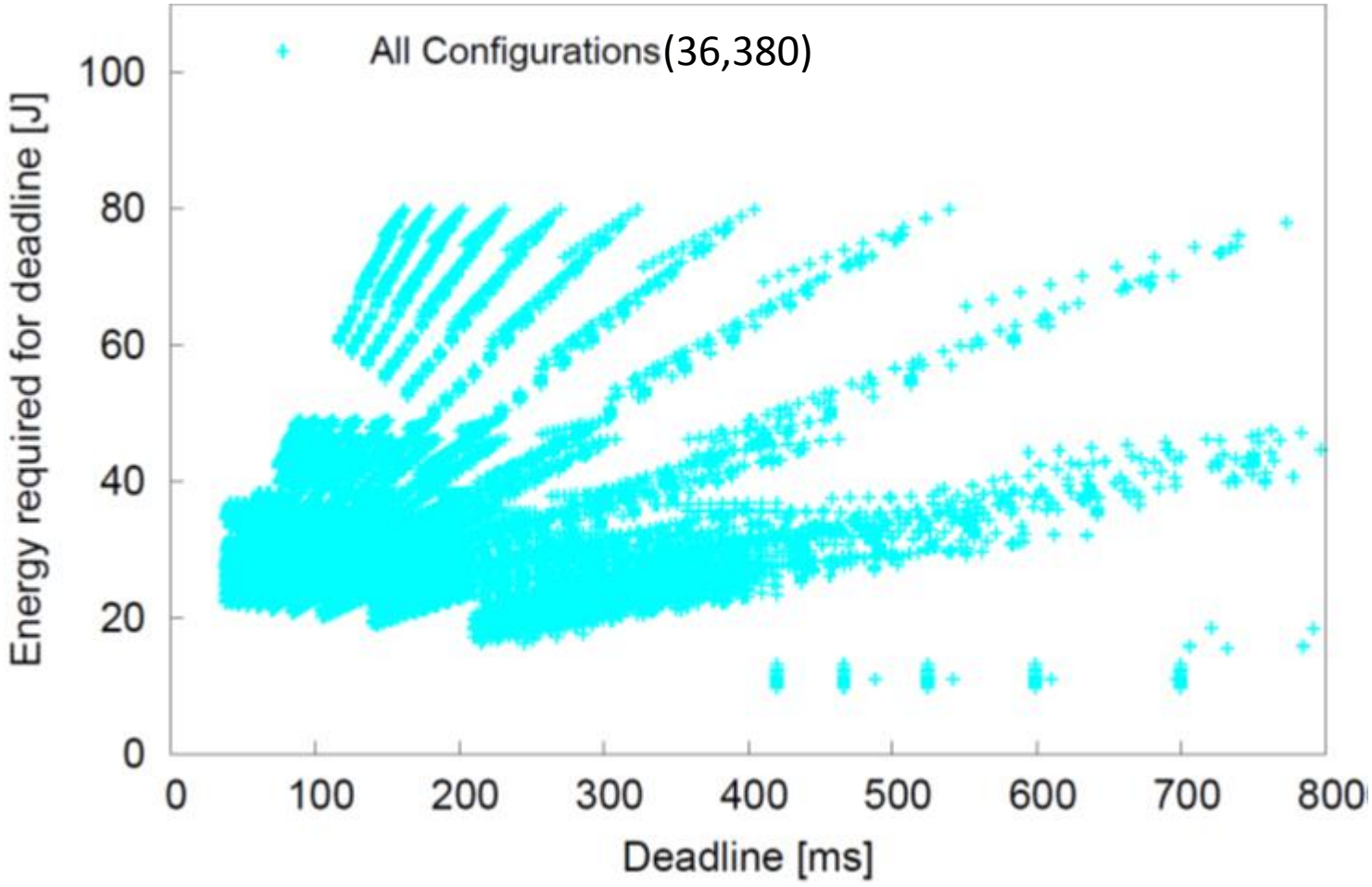
Program	Performance per Watt (PPR)	AMD Node	ARM Node
EP	(random no./s)/W	1,414,922	6,048,057
memcached	(kbytes/s)/W	2,628	5,220
x264	(frames/s)/W	1	0.7
blackscholes	(options/s)/W	2,902	11,413
Julius	(samples/s)/W	21,390	69,654
RSA-2048	(verify/s)/W	9,346	6,877

x86 ISA has special instruction for cryptography

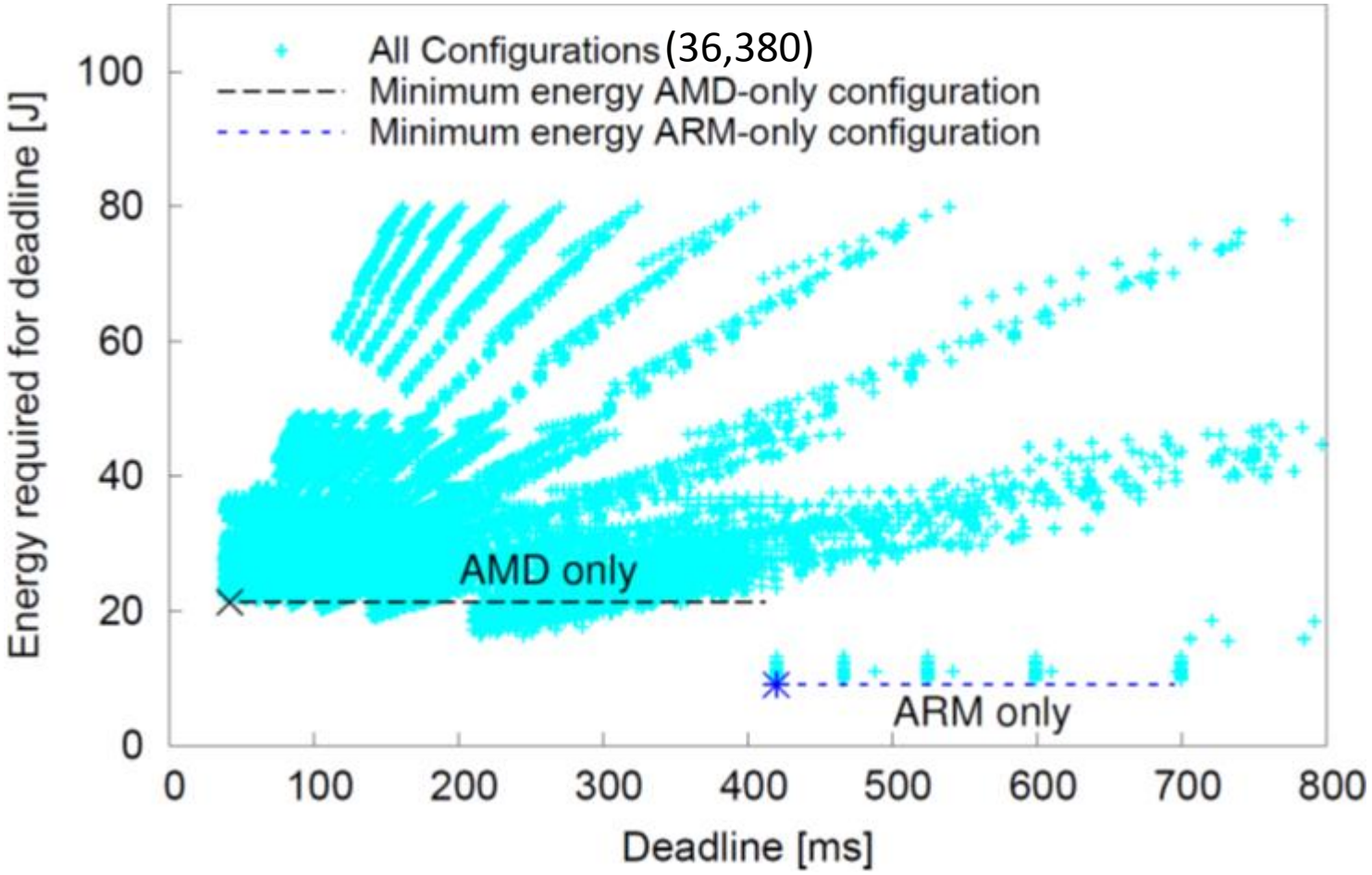
Research Questions

1. Is heterogeneity better than homogeneity ?
2. Are larger mixes of heterogeneous nodes better ?
3. …

Heterogeneity versus Homogeneity



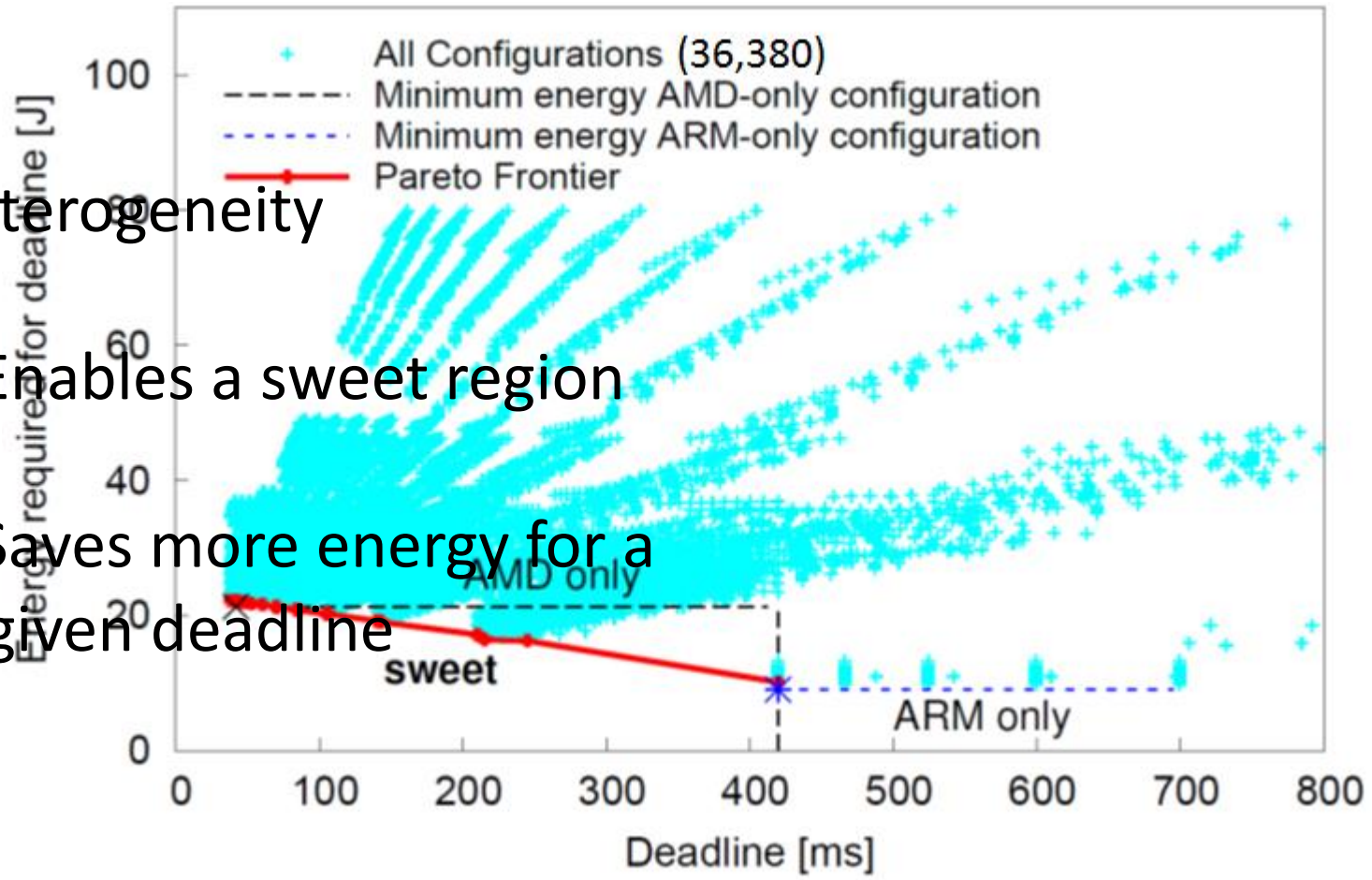
Heterogeneity versus Homogeneity



Heterogeneity versus Homogeneity

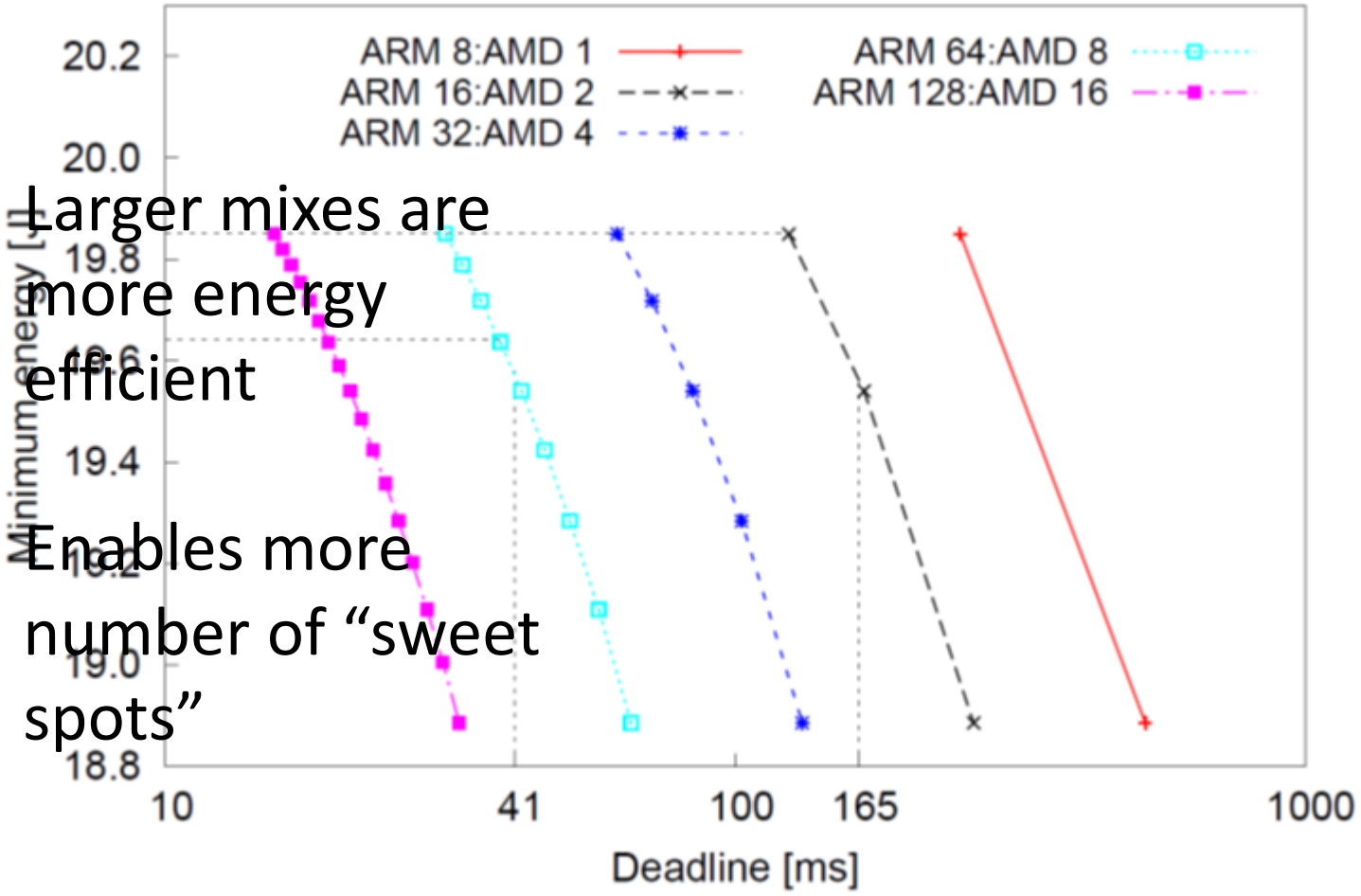
Heterogeneity

- Enables a sweet region
- Saves more energy for a given deadline



Are larger mixes better ?

- Larger mixes are more energy efficient
- Enables more number of “sweet spots”



Observations

1. Heterogeneity allows larger energy savings compared to homogeneous systems.
2. Larger mixes increase the number of configurations in the sweet region.
3. …

Conclusions

- measurement-driven analytical model to determine energy-efficient configurations for a single workload on a heterogeneous mix with different ISA's

- Heterogeneity is almost always more energy-efficient than homogeneity
  - But not for programs with large sequential fraction and high parallel overhead

Questions ?

Thank you

[lavanya,teoym]@comp.nus.edu.sg

L. Ramapantulu, B.M. Tudor, D.Loghin, T. Vu and Y.M. Teo, **Modeling the Energy Efficiency of Heterogeneous Clusters**, Proceedings of 43rd International Conference on Parallel Processing, Minneapolis, USA, Sep 9-12, 2014.

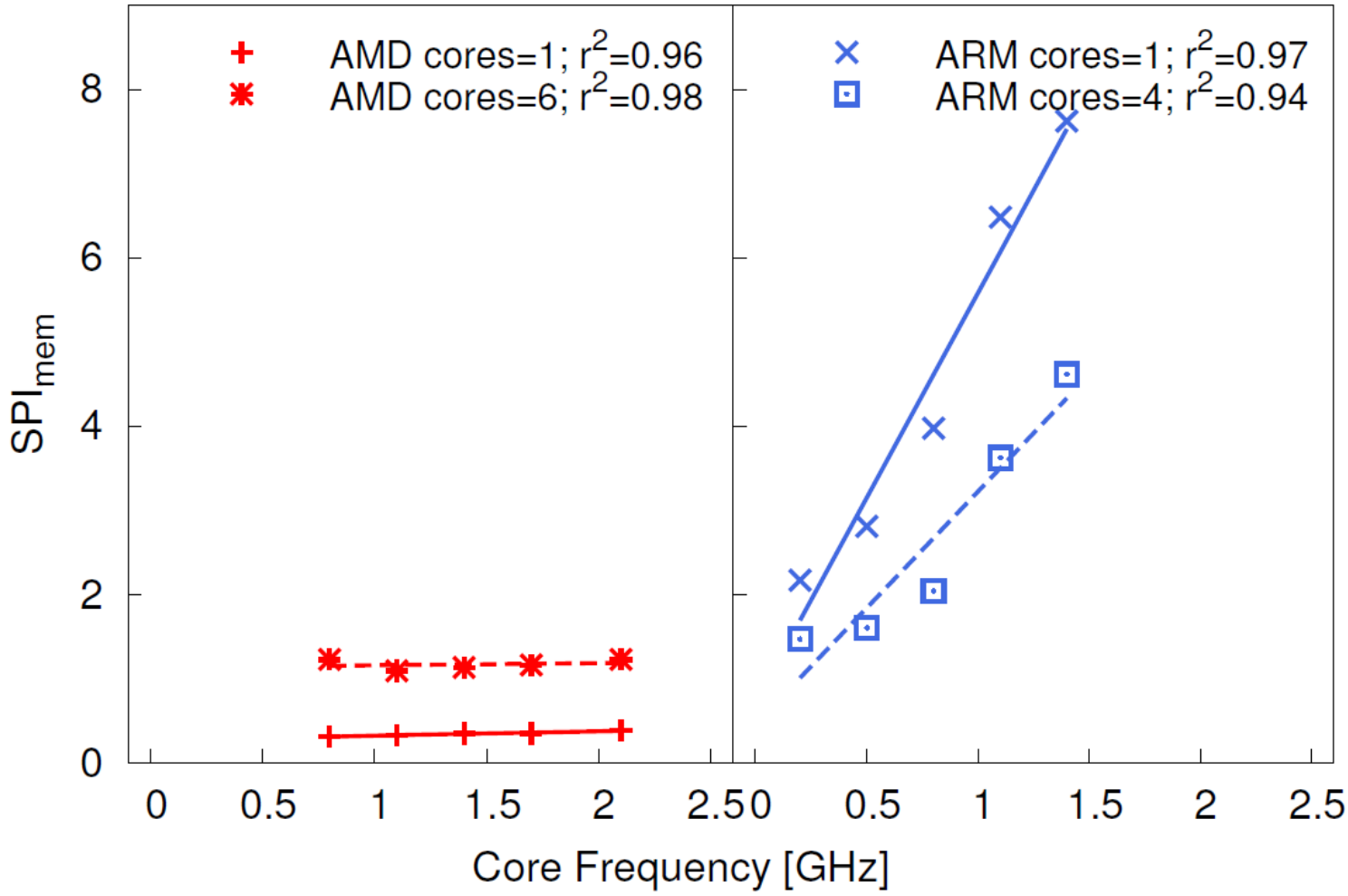
Thank you

BACKUP SLIDES

System Overview

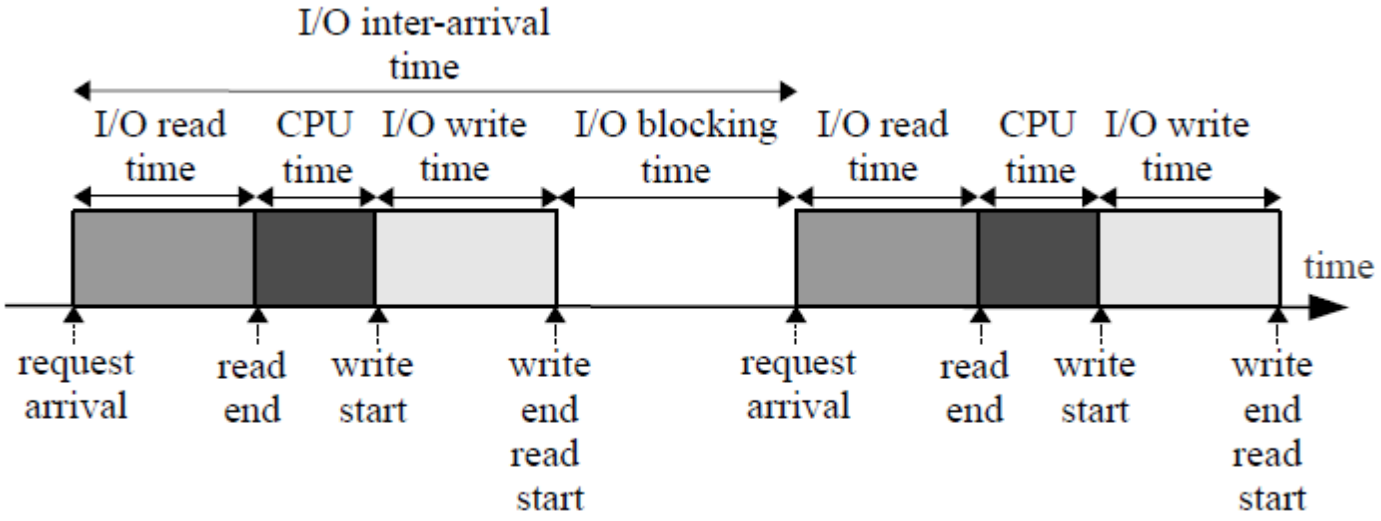
Node	AMD K10	ARM Cortex-A9
ISA	X86_64	ARM v7-A
Cores/node	6	4
Core clock frequency	0.8-2.1 GHz	0.2-1.4 GHz
L1 data cache	64KB/core	32KB/core
L2 cache	512KB/core	1MB/node
L3 cache	6MB /node	NA
Memory	8GB DDR3	1GB LP-DDR2
I/O bandwidth	1Gbps	100MBps

Stalls due to memory contention



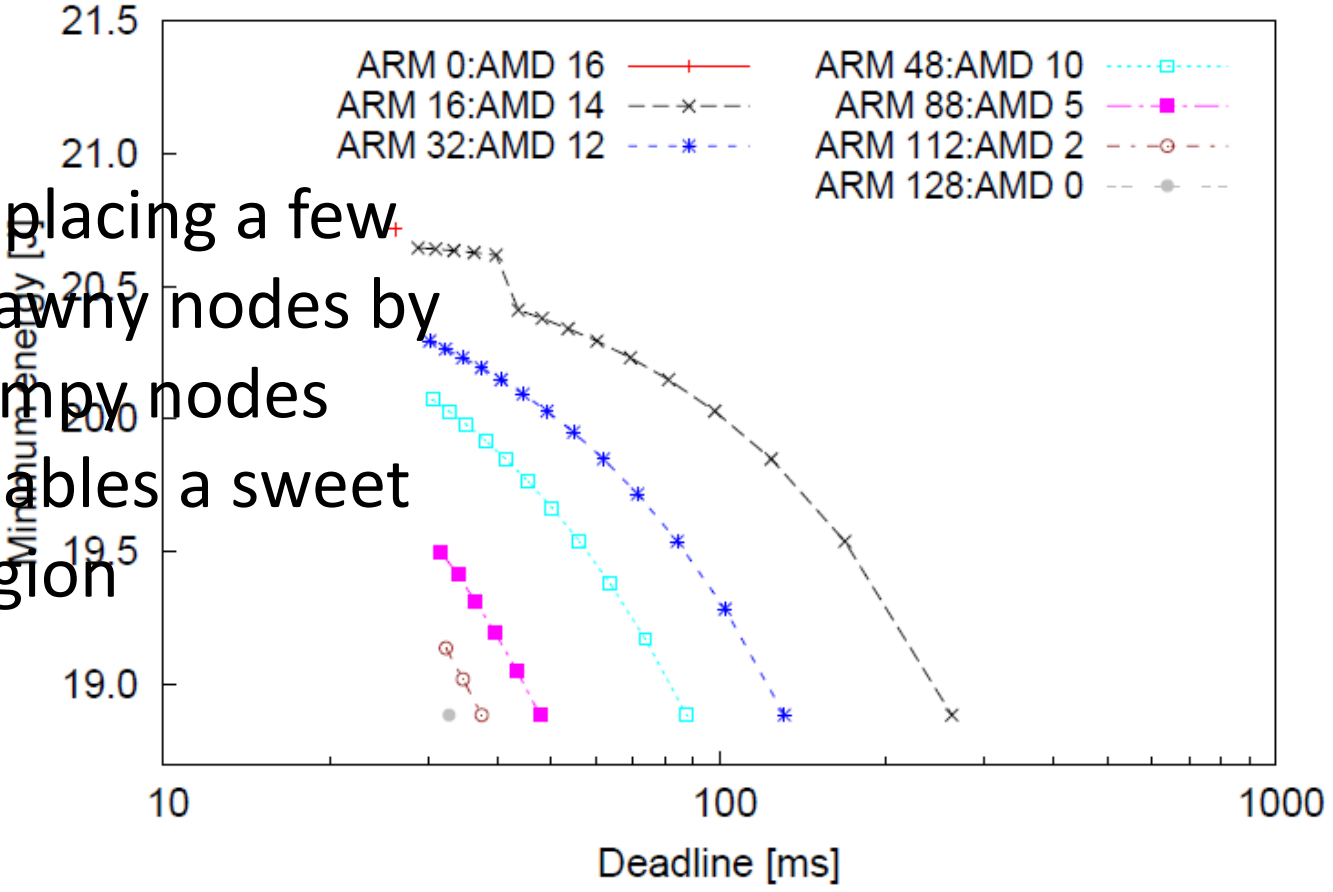
CPU and I/O Overlap

- Tudor et al. [SIGMETRICS'13]
- Server workloads (Ex: memcached)



What is a good mix ?

- Replacing a few brawny nodes by wimpy nodes enables a sweet region



Other Research Questions

- Queuing Delay
  - As cluster utilization increases due to faster job arrivals, the energy savings are further amplified, but the minimal response time achievable is reduced

Queuing Delay