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 $61 \times 10^9$ 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

Wimpy vs Brawny Servers
[Gupta et al. 2013]

Marginal improvement in performance at high power
High idle power

power [W]

performance [MFLOPS]
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

Time-Energy Performance Model

Energy-efficient Pareto-optimal configurations

- Considers different ISAs
- Resource overlap
- Unifying unit of work

workload parameters

baseline measurement

system parameters
Applications

Broad range of datacenter application domains

<table>
<thead>
<tr>
<th>Domain</th>
<th>Program</th>
<th>Problem Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>HPC</td>
<td>EP</td>
<td>2,147,483,648 random numbers</td>
</tr>
<tr>
<td>Web Server</td>
<td>memcached</td>
<td>600,000 GET/SET operations</td>
</tr>
<tr>
<td>Streaming video</td>
<td>x264</td>
<td>600 frames 704 x 576</td>
</tr>
<tr>
<td>Financial</td>
<td>Black-scholes</td>
<td>500,000 stock options</td>
</tr>
<tr>
<td>Speech recognition</td>
<td>Julius</td>
<td>2,310,559 samples</td>
</tr>
<tr>
<td>Web security</td>
<td>RSA-2048</td>
<td>5000 keys verifications</td>
</tr>
</tbody>
</table>
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
Parallel Application

Within a type of node workload is equally divided

\[ T(n_{\text{ARM}}) \approx \frac{T_{\text{ARM}}(1)}{n_{\text{ARM}}} \]

Match the execution rates between ARM and AMD nodes

\[ T(n_{\text{ARM}}) \approx T(n_{\text{AMD}}) \]

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

\[ T_{\text{CPU}} \approx T_{\text{core,work}} + T_{\text{core,STALL}} \]

Execution Time Model

\[ T_{\text{core,work}} \approx \frac{\text{computational work cycles}}{\text{clock frequency}} \]

\[ T_{\text{core,STALL}} \approx \frac{\text{stall cycles due to memory contention}}{\text{clock frequency}} \]

– Stall cycles increase linearly with
  
  • increase in core clock frequency
  • increase in the number of cores
Energy Model

• Total Energy = $E_{\text{ARM}} \times n_{\text{ARM}} + E_{\text{AMD}} \times n_{\text{AMD}}$

• $E_{\text{node}} = E_{\text{core}} + E_{\text{mem}} + E_{\text{I/O}} + E_{\text{idle}}$

• $E_{\text{core}} = P_{\text{core,act}} \times T_{\text{core,work}} + P_{\text{core,stall}} \times T_{\text{core,stall}}$
  
  – Power × Time

  – uses execution time model

  – measured values for $P_{\text{core,act}}$, $P_{\text{core,stall}}$
### Execution Time Model

<table>
<thead>
<tr>
<th>T</th>
<th>max($T_{ARM}, T_{AMD}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_{ARM}$</td>
<td>max($T_{CPU,ARM}, T_{I/O,ARM}$)</td>
</tr>
<tr>
<td>$T_{CPU,ARM}$</td>
<td>max($T_{core,ARM}, T_{mem,ARM}$)</td>
</tr>
<tr>
<td>$T_{core,ARM}$</td>
<td>$\frac{l_{core,ARM} \times (WPI_{ARM} + SPI_{core,ARM})}{f_{ARM}}$</td>
</tr>
<tr>
<td>$T_{mem,ARM}$</td>
<td>$\frac{l_{core,ARM} \times (WPI_{ARM} + SPI_{mem,ARM})}{f_{ARM}}$</td>
</tr>
<tr>
<td>$T_{I/O,ARM}$</td>
<td>max($T_{I/O,ARM}, 1/\lambda_{I/O}$)</td>
</tr>
</tbody>
</table>

### Energy Model

<table>
<thead>
<tr>
<th>E</th>
<th>$E_{ARM} + E_{AMD}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_{ARM}$</td>
<td>$(E_{core,ARM} + E_{mem,ARM} + E_{I/O,ARM} + E_{idle,ARM}) \times n_{ARM}$</td>
</tr>
<tr>
<td>$E_{core,ARM}$</td>
<td>$(P_{core,act,ARM} \times T_{act,ARM} + P_{core,stall,ARM} \times T_{stall,ARM}) \times c_{act, ARM}$</td>
</tr>
</tbody>
</table>
## Model Validation

<table>
<thead>
<tr>
<th>Program</th>
<th>Configuration</th>
<th>Execution time</th>
<th>Energy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ARM nodes</td>
<td>AMD nodes</td>
<td>error[%]</td>
</tr>
<tr>
<td>EP</td>
<td>8</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>memcached</td>
<td>8</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>x264</td>
<td>8</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>0</td>
<td>13</td>
</tr>
<tr>
<td>blackscholes</td>
<td>8</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>Julius</td>
<td>8</td>
<td>1</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>RSA-2048</td>
<td>8</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Outline

- Motivation
- Objective
- Methodology
- Analysis
- Conclusions
## Performance-to-Power Ratio

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

*memory bound on ARM*

*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

![Energy vs Deadline Diagram](image)

- All Configurations
- Minimum energy AMD-only configuration
- Minimum energy ARM-only configuration

Energy required for deadline [J]

Deadline [ms]
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

Thank you
BACKUP SLIDES
# System Overview

<table>
<thead>
<tr>
<th>Node</th>
<th>AMD K10</th>
<th>ARM Cortex-A9</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISA</td>
<td>X86_64</td>
<td>ARM v7-A</td>
</tr>
<tr>
<td>Cores/node</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>Core clock frequency</td>
<td>0.8-2.1 GHz</td>
<td>0.2-1.4 GHz</td>
</tr>
<tr>
<td>L1 data cache</td>
<td>64KB/core</td>
<td>32KB/core</td>
</tr>
<tr>
<td>L2 cache</td>
<td>512KB/core</td>
<td>1MB/node</td>
</tr>
<tr>
<td>L3 cache</td>
<td>6MB /node</td>
<td>NA</td>
</tr>
<tr>
<td>Memory</td>
<td>8GB DDR3</td>
<td>1GB LP-DDR2</td>
</tr>
<tr>
<td>I/O bandwidth</td>
<td>1Gbps</td>
<td>100MBps</td>
</tr>
</tbody>
</table>
Stalls due to memory contention

![Graph showing Core Frequency vs. SPI_{mem} for AMD and ARM cores with different configurations and correlation coefficients.

- AMD cores=1; $r^2=0.96$
- AMD cores=6; $r^2=0.98$
- ARM cores=1; $r^2=0.97$
- ARM cores=4; $r^2=0.94$]
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.
• 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

![Graph showing energy for 20s (J) vs. response time per job (ms) at different utilizations (5%, 25%, 50%).]