Navigating Heterogeneous Processors with Market Mechanisms

Marisabel Guevara
Duke University

Benjamin Lubin
Boston University

Benjamin C. Lee
Duke University
Growth of Cloud Applications

- Data volumes are growing geometrically
- Cloud applications are diversifying rapidly
- Computing capability must grow
- Datacenters consume tens of megawatts of power
Datacenter Scaling Limitations

- Dennard scaling is over
  - On-chip power density is a constraint
- Adding servers is expensive
  - Power determines operating costs

- Heterogeneity improves energy efficiency
  - Small cores consume a fraction of big core power
  - Big cores ensure service quality
Executive Summary

Goal:
• Improve service quality in heterogeneous datacenters

Methods:
• Leverage processor heterogeneity
• Mitigate performance risk using market

Evaluation:
• Big/Small core heterogeneity improves service quality
• Three core types reduce service violations by 12x
Risks of Heterogeneity

- Heterogeneity introduces performance risk
- Yet it can improve service quality
Managing Risk

1. Types of processors?
2. Number of each?
3. How to allocate?

- Resource allocation that mitigates performance risk
  - Hide hardware complexity
  - Trade-off performance and power
  - Allocate small cores when possible
- Coordinate design and management
Allocating Time

Market periodically allocates time on hardware resource

\[
\frac{\text{Tasks}}{\text{Sec}} = \frac{\text{Tasks}}{\text{Cycles}} \times \frac{\text{Cycles}}{\text{Sec}}
\]

Right-size datacenter via server activation and DVFS
Accommodating Heterogeneity

Profile task-specific performance on each processor:

\[
\frac{\text{Tasks}}{\text{Sec}} = \frac{\text{Tasks}}{\text{Inst}} \times \frac{\text{Insts}}{\text{Cycle}} \times \frac{\text{Cycles}}{\text{Sec}}
\]

Store \( \text{IPC}_{a,m} \) as scaling factor relative to baseline \( \text{IPC}_0 \)
Market Resource Allocation

\[ \text{maximize} \sum_{a \in \text{App}} (\text{Value}_a - \text{Cost}) \]

**Proxy**
- \( \lambda \leftrightarrow \text{predict demand} \)
- \( T \leftrightarrow \text{predict wait time} \)
- \( \lambda \rightarrow \mu \)
- \( V \leftrightarrow \text{predict value} \)

**Cost Model**
- \( \text{Power} \)
- \( \text{Utilization} \)
- \( \text{energy} \times \text{price} \)
- \( \text{peak} \)
- \( \text{idle} \)

**Big/Core**
- Increase Heterogeneity

**Market Mechanism**

**Cases for Heterogeneity**
- \( \text{App A} \)
- \( \text{req/t} \)
- \( t \)
- \( \text{App B} \)
- \( \text{req/t} \)
- \( t \)
- \( \text{App Z} \)
- \( \text{req/t} \)
- \( t \)
Experimental Methodology

Real tasks

Java implementation, CPLEX solver for optimization

Diurnal arrivals

System profiles

Simulation
Defining Big and Small Cores

- Within fixed power budget, vary number of:
  - 4-core Xeon servers
  - 16-core Atom servers

- Core measurements [ISCA’10]
  - 0.3 – 1.0 relative IPC
  - 1.5 W Atom vs. 15 W Xeon

- System model
  - Equal die area
  - Fixed system power overhead (65 W)
Modeling Application Behavior

- Processor Sensitive (PS) – Atom throughput $\frac{1}{3}$ that of Xeon
- Processor Insensitive ($\neg$PS) – same throughput

- One week of requests
- Diurnal pattern
Understanding Datacenter Dynamics

- Vary Atom to Xeon ratio
- Examine allocations to each task
- Identify a balanced mix (e.g. 147:55)
Improving Service Quality

- Xeon-only has insufficient resources
- Atom-only incurs violations due to cost
Improving Service Quality

- Xeon/Atom mix reduces waiting time
  - Atoms mostly allocated to ¬PS
  - Xeons freed to service PS peaks
Defining Greater Heterogeneity

- Within fixed power budget, vary core designs
  - Dynamic scheduling (IO vs OO)
  - Issue Width (1, 2, 4, 6, 8)
  - Frequency (1.0, 2.4 GHz)

- Processor simulation
  - 0.4 – 1.5 relative IPC (gem5)
  - 1.1 W – 28 W (McPAT)

- System model
  - Equal die area
  - Fixed system power overhead (65 W)
Clustering Heterogeneous Processors

- Cluster cores with similar SPEC performance
- Select core with lowest performance variation from each cluster
- Evaluate with diverse SPEC task streams
Visualizing the Design Space

- Ellipses represent core types
- Points are combinations of cores
- Colors represent service violations
Design Space with Four Processors

- Identify right core types
- Prune design space
- Best configuration is heterogeneous
  - RT violations reduced from 15.5% to 1.6%
Conclusions and Future Directions

- Leverage market to mitigate heterogeneity’s risk
  - Embed microarchitectural insight into the market
  - Allocate multiple resources

- Deploy heterogeneous hardware in a datacenter
  - Optimal balance improves service
  - Sophisticated trade-offs require further study

- Propose a datacenter research methodology
  - Simulating detailed server architecture
  - Modeling user and datacenter dynamics
  - Tractability for web search, memcached, map/reduce
Navigating Heterogeneous Processors with Market Mechanisms

Marisabel Guevara
Duke University

Benjamin Lubin
Boston University

Benjamin C. Lee
Duke University