Web Search Using Mobile Cores

Quantifying and Mitigating the Price of Efficiency

Vijay Janapa Reddi
Engineering & Applied Science
Harvard University

Benjamin Lee
Electrical Engineering
Stanford University

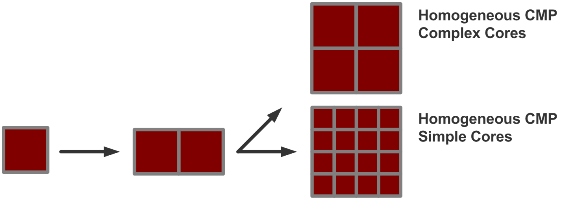
Trishul Chilimbi
Runtime Analysis & Design
Microsoft Research

Kushagra Vaid
Global Foundation Services
Microsoft Corporation

International Symposium on Computer Architecture
22 June 2010

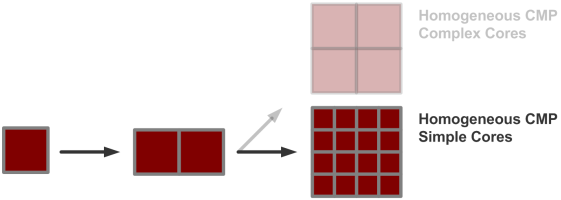
Conventional Wisdom

- Moore's Law provides transistors
- Simple cores improve energy efficiency
- Parallelism recovers lost performance



Simple Cores

- Pursue aggregate throughput, energy efficiency
- Assume task parallelism
- Assume latency tolerance

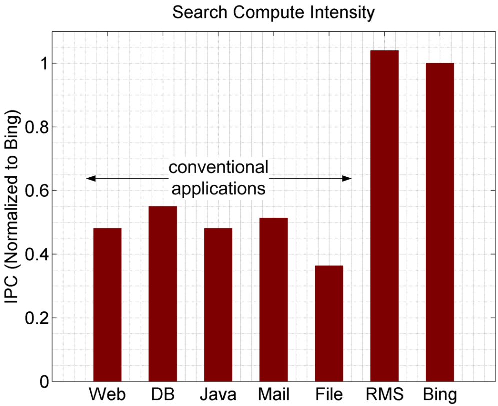


Applications in Transition

- **Conventional Enterprise**
  - Process independent requests
  - Exhibit high memory, I/O intensity
  - Ex: web, database, Java, mail, file servers
- **Emerging Cloud**
  - Extract information, value from data
  - Exhibit high compute intensity
  - Ex: analytics, machine learning

Computational Intensity

- Microsoft Bing ranks pages with neural network
- RMS foreshadows future analytic workloads



Cloud Efficiency

- **Challenges**
  - Migrate computation, data to cloud
  - Choose efficient components
  - Understand application, component interaction
- **Case Study**
  - Mobile cores for efficiency, parallelism for performance?
  - Achieve efficiency with mobile cores (Intel Atom)
  - Quantify price of efficiency (Microsoft Bing)

Efficiency

Atom is more energy, cost efficient than Xeon

Price of Efficiency

Atom limitations impact latency, relevance, flexibility

Mitigating Price of Efficiency

Atom over-provisioning should consider platform overheads

Efficiency

Atom is more energy, cost efficient than Xeon

Price of Efficiency

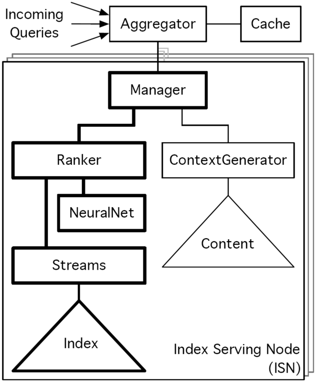
Atom limitations impact latency, relevance, flexibility

Mitigating Price of Efficiency

Atom over-provisioning should consider platform overheads

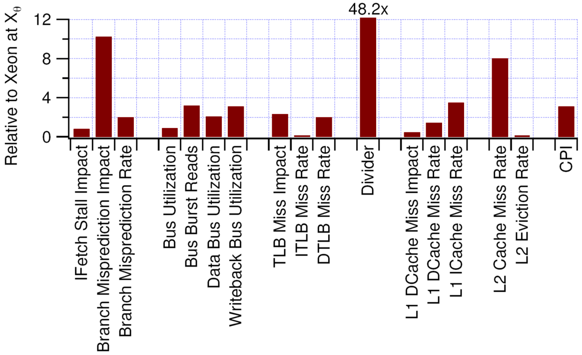
Search Architecture

- Rank pages using neural network
- Deploy on server (Xeon), mobile (Atom) processors



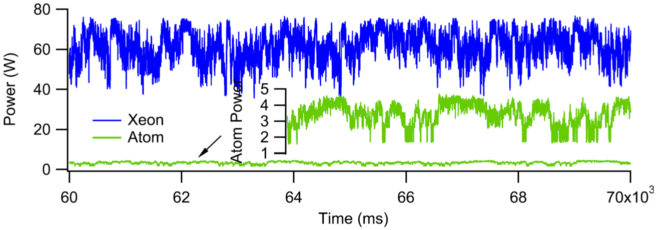
Processor Activity

- Compare Xeon (4-issue, OOO) and Atom (2-issue, IO)
- Measure μarch activity with hardware counters



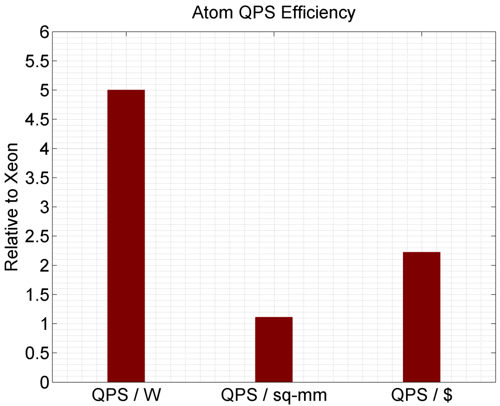
Processor Power

- Compare Xeon (15W per core) and Atom (1.5W per core)
- Measure processor power at voltage regulator



Processor Efficiency

- Demonstrate energy, cost efficiency with Atom
- Measure max QPS within QoS target



Efficiency

Atom is more energy, cost efficient than Xeon

Price of Efficiency

Atom limitations impact latency, relevance, flexibility

Mitigating Price of Efficiency

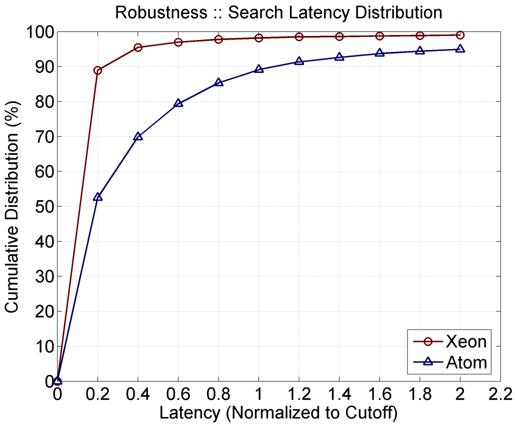
Atom over-provisioning should consider platform overheads

Price of Efficiency

- **Latency**
  - Cut-off latency limits refinement opportunities
  - Per query latency impacts quality-of-service
- **Relevance**
  - Search rank orders documents
  - Choice, ordering of results impact relevance
- **Flexibility**
  - Query activity, complexity increase load
  - Processor resources impact flexibility

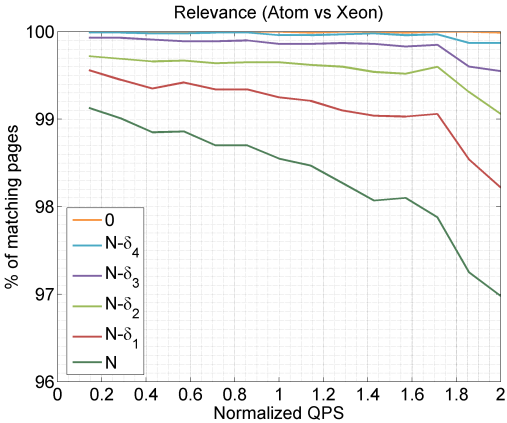
Latency

- Atom increases latency average (μ) by 3$\times$
- Atom increases latency variance (σ^2)



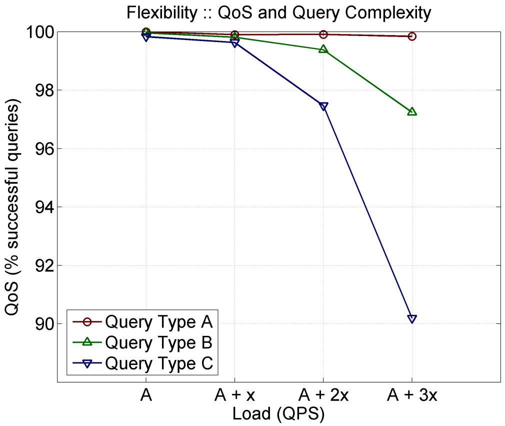
Relevance

- Consider choice, ordering of top N documents
- Atom impacts relevance under all query loads



Flexibility

- Consider activity, complexity of queries
- Atom harms QoS for more complex queries



Efficiency

Atom is more energy, cost efficient than Xeon

Price of Efficiency

Atom limitations impact latency, relevance, flexibility

Mitigating Price of Efficiency

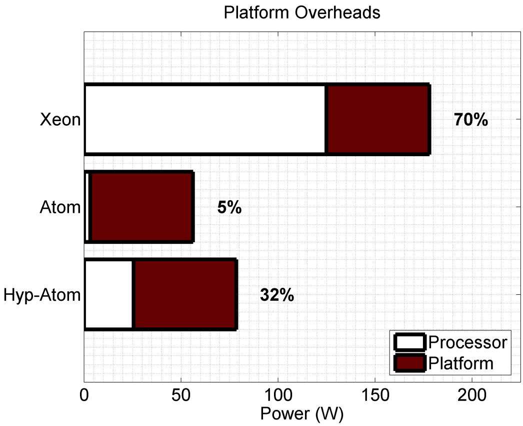
Atom over-provisioning should consider platform overheads

Mitigating Price of Efficiency

- **Addressing Latency & Relevance**
  - Address μarchitectural limitations
  - Integrate application-specific accelerators
  - Manage heterogeneous servers
- **Addressing Flexibility**
  - Over-provision Atoms
  - Mitigate platform overheads
  - Integrate more cores per chip

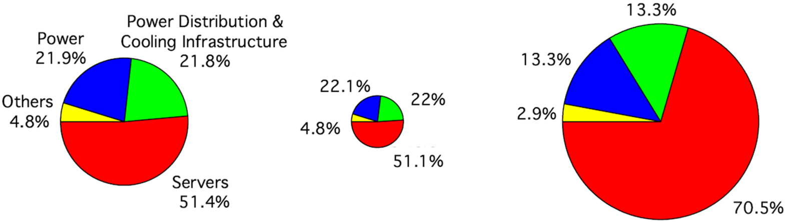
Platform Overheads

- Xeon: 4-core, 2-socket
- Atom: 2-core, 1-socket ⇒ Hyp-Atom: 8-core, 2-socket



Total Cost of Ownership (TCO)

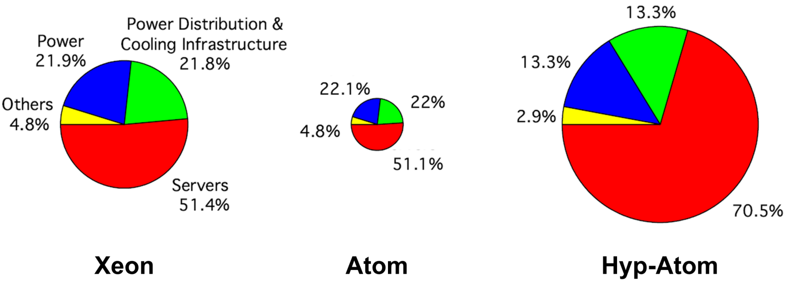
- Pie slice shows breakdown of TCO $
- Pie size shows throughput per TCO $



Xeon **Atom** **Hyp-Atom**

Case for Integration

- Hyp-Atom attributes more per TCO $ to servers
- Hyp-Atom achieves greater throughput per TCO $



Efficiency

Atom is more energy, cost efficient than Xeon

Price of Efficiency

Atom limitations impact latency, relevance, flexibility

Mitigating Price of Efficiency

Atom over-provisioning should consider platform overheads

Also in the paper ...

- **μarchitecture**
  - Processor activity from hardware counters
  - μarchitectural bottlenecks
- **Search**
  - Application phases in computation
  - Execution time breakdown
- **Mitigating Price of Efficiency**
  - μarchitectural enhancements
  - Heterogeneous, accelerated processors

Conclusion

- **Emerging Cloud Applications**
  - Extract value from data
  - Increase compute intensity
- **Energy Efficiency**
  - Improve efficiency by $5\times$ with mobile processors
  - Exact price in latency, relevance, flexiblity
- **Future Challenges**
  - Pursue efficiency given compute intensity
  - Consider heterogeneous, accelerated processors

Web Search Using Mobile Cores

Quantifying and Mitigating the Price of Efficiency

Vijay Janapa Reddi
Engineering & Applied Science
Harvard University

Benjamin Lee
Electrical Engineering
Stanford University

Trishul Chilimbi
Runtime Analysis & Design
Microsoft Research

Kushagra Vaid
Global Foundation Services
Microsoft Corporation

International Symposium on Computer Architecture
22 June 2010