Datacenter Simulation Methodologies
Case Studies

Tamara Silbergleit Lehman, Qiuyun Wang, Seyed Majid Zahedi
and Benjamin C. Lee

This work is supported by NSF grants CCF-1149252, CCF-1337215, and STARnet, a Semiconductor
Research Corporation Program, sponsored by MARCO and DARPA.
<table>
<thead>
<tr>
<th>Time</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>09:00 - 09:30</td>
<td>Introduction</td>
</tr>
<tr>
<td>09:30 - 10:30</td>
<td>Setting up MARSSx86 and DRAMSim2</td>
</tr>
<tr>
<td>10:30 - 11:00</td>
<td>Break</td>
</tr>
<tr>
<td>11:00 - 12:00</td>
<td>Spark simulation</td>
</tr>
<tr>
<td>12:00 - 13:00</td>
<td>Lunch</td>
</tr>
<tr>
<td>13:00 - 13:30</td>
<td>Spark continued</td>
</tr>
<tr>
<td>13:30 - 14:30</td>
<td>GraphLab simulation</td>
</tr>
<tr>
<td>14:30 - 15:00</td>
<td>Break</td>
</tr>
<tr>
<td>15:00 - 16:15</td>
<td>Web search simulation</td>
</tr>
<tr>
<td><strong>16:15 - 17:00</strong></td>
<td><strong>Case studies</strong></td>
</tr>
</tbody>
</table>
Big data demands big computing, yet we face challenges...

- Architecture Design
- Systems Management
- Research Coordination
Toward Energy-Efficient Datacenters

Heterogeneity
- Tailors hardware to software, reducing energy
- Complicates resource allocation and scheduling
- Introduces risk

Sharing
- Divides hardware over software, amortizing energy
- Complicates task placement and co-location
- Introduces risk
Heterogeneity for Efficiency
Heterogeneous datacenters deploy mix of server- and mobile-class hardware
Heterogeneity for Efficiency
Heterogeneous datacenters deploy mix of server- and mobile-class hardware

Heterogeneity and Markets
Agents bid for heterogeneous hardware in a market that maximizes welfare
Heterogeneity for Efficiency
Heterogeneous datacenters deploy mix of server- and mobile-class hardware

Heterogeneity and Markets
Agents bid for heterogeneous hardware in a market that maximizes welfare

Sharing and Game Theory
Agents share multiprocessors with game-theoretic fairness guarantees
Heterogeneity for Efficiency

Heterogeneous datacenters deploy mix of server- and mobile-class hardware

- “Web search using mobile cores” [ISCA'10]
- “Towards energy-proportional datacenter memory with mobile DRAM” [ISCA'12]
Mobile versus Server Processors

- **Simpler Datapath**
  - Issue fewer instructions per cycle
  - Speculate less often

- **Smaller Caches**
  - Provide less capacity
  - Provide lower associativity

- **Slower Clock**
  - Reduce processor frequency
Applications in Transition

Conventional Enterprise
- Independent requests
- Memory-, I/O-intensive
- Ex: web or file server

Emerging Datacenter
- Inference, analytics
- Compute-intensive
- Ex: neural network

Reddi et al., “Web search using mobile cores” [ISCA’10]
Web Search

- Distribute web pages among index servers
- Distribute queries among index servers
- Rank indexed pages with neural network
Query Efficiency

- Joules per second :: ↓ 10× on Atom versus Xeon
- Queries per second :: ↓ 2×
- Queries per Joule :: ↑ 5×

Reddi et al., “Web search using mobile cores” [ISCA’10]
Case for Processor Heterogeneity

Mobile Core Efficiency
- Queries per Joule $\uparrow \times 5$

Mobile Core Latency
- 10% queries exceed cut-off
- Complex queries suffer

Heterogeneity
- Small cores for simple queries
- Big cores for complex queries

Reddi et al., “Web search using mobile cores” [ISCA’10]
Memory Architecture and Applications

Conventional Enterprise
- High bandwidth
- Ex: transaction processing

Emerging Datacenter
- Low bandwidth (< 6% DDR3 peak)
- Ex: search [Microsoft], memcached [Facebook]
Memory Capacity vs Bandwidth

- **Online Services**
  - Use < 6% bandwidth, 65-97% capacity
  - Ex: Microsoft mail, map-reduce, search [Kansal+]

- **Memory Caching**
  - 75% of Facebook data in memory
  - Ex: memcached, RAMCloud [Ousterhout+]

- **Capacity-Bandwidth Bundles**
  - Server with 4 sockets, 8 channels
  - Ex: 32GB capacity, >100GB/s bandwidth
Mobile-class Memory

- **Operating Parameters**
  - Lower active current (130mA vs 250mA)
  - Lower standby current (20mA vs 70mA)

- **Low-power Interfaces**
  - No delay-locked loops, on-die termination
  - Lower bus frequency (400 vs 800MHz)
  - Lower peak bandwidth (6.4 vs 12.8GBps)
Activity Example

- 16% DDR3 peak

Energy per Bit

- Large power overheads
- High cost per bit

“Calculating memory system power for DDR3” [Micron]
Case for Memory Heterogeneity

Mobile Memory Efficiency
- Bits / Joule $\uparrow \times 5$

Mobile Memory Bandwidth
- Peak B/W $\downarrow 0.5\times$

Heterogeneity
- LPDDR for search, memcached
- DDR for databases, HPC

Malladi et al., “Towards energy-proportional datacenter memory with mobile DRAM” [ISCA’12]
Heterogeneity and Markets
Agents bid for heterogeneous hardware in a market that maximizes welfare

- “Navigating heterogeneous processors with market mechanisms” [HPCA’13]
- “Strategies for anticipating risk in heterogeneous system design” [HPCA’14]
Datacenter Heterogeneity

Systems are heterogeneous
  • Virtual machines are sized
  • Physical machines are diverse

Heterogeneity is exposed
  • Users assess machine price
  • Users select machine type

Burden is prohibitive
  • Users must understand hardware-software interactions

Elastic Compute Cloud (EC2) [Amazon]
Managing Performance Risk

Risk: the possibility that something bad will happen

Understand Heterogeneity and Risk
- What types of hardware?
- How many of each type?
- What allocation to users?

Mitigate Risk with Market Allocation
- Ensure service quality
- Hide hardware complexity
- Trade-off performance and power
Market Mechanism

- User specifies value for performance
- Market shields user from heterogeneity
Proxy Bidding

**User Provides...**
- Task stream
- Service-level agreement

**Proxy Provides...**
- $\mu$architectural insight
- Performance profiles
- Bids for hardware

---

Guevara et al. “Navigating heterogeneous processors with market mechanisms” [HPCA’13]

Wu and Lee, “Inferred models for dynamic and sparse hardware-software spaces” [MICRO’12]
Visualizing Heterogeneity (2 Processor Types)

- Ellipses represent hardware types
- Points are combinations of processor types
- Colors show QoS violations

Guevara et al. “Navigating heterogeneous processors with market mechanisms” [HPCA’13]
Further Heterogeneity (4 Processor Types)

- Best configuration is heterogeneous
- QoS violations fall 16% → 2%
- Trade-offs motivate design for manageability

Guevara et al. “Navigating heterogeneous processors with market mechanisms” [HPCA’13]
Guevara et al. “Strategies for anticipating risk in heterogeneous datacenter design” [HPCA’14]
Sharing and Game Theory
Agents share multiprocessors with game-theoretic fairness guarantees

- “REF: Resource elasticity fairness with sharing incentives for multiprocessors” [ASPLOS’14]
Case for Sharing

Big Servers
- Hardware is under-utilized
- Sharing amortized power

Heterogeneous Users
- Tasks are diverse
- Users are complementary
- Users prefer flexibility

Sharing Challenges
- Allocate multiple resources
- Ensure fairness

Image: Intel Sandy Bridge E die [www.anandtech.com]
Motivation

• Alice and Bob are working on research papers
• Each has $10K to buy computers
• Alice and Bob have different types of tasks
• Alice and Bob have different paper deadlines
Strategic Behavior

- Alice and Bob are strategic

- Which is better?
  - Small, separate clusters
  - Large, shared cluster

- Suppose Alice and Bob share
  - Is allocation fair?
  - Is lying beneficial?

Image: [www.websavers.org]
Users must share
  • Overlooks strategic behavior

Fairness policy is equal slowdown
  • Fails to encourage envious users to share

Heuristic mechanisms enforce equal slowdown
  • Fail to give provable guarantees
"If an allocation is both equitable and Pareto efficient, ... it is fair." [Varian, Journal of Economic Theory (1974)]
Resource Elasticity Fairness (REF)

REF is an allocation mechanism that guarantees game-theoretic desiderata for shared chip multiprocessors

- **Sharing Incentives**
  Users perform no worse than under equal division

Zahedi et al. “REF: Resource elasticity fairness with sharing incentives for multiprocessors” [ASPLOS’14]
Resource Elasticity Fairness (REF)

REF is an allocation mechanism that guarantees game-theoretic desiderata for shared chip multiprocessors

- **Sharing Incentives**
  Users perform no worse than under equal division

- **Envy-Free**
  No user envies another’s allocation

Zahedi et al. “REF: Resource elasticity fairness with sharing incentives for multiprocessors” [ASPLOS’14]
Resource Elasticity Fairness (REF)

REF is an allocation mechanism that guarantees game-theoretic desiderata for shared chip multiprocessors

- **Sharing Incentives**
  Users perform no worse than under equal division

- **Envy-Free**
  No user envies another’s allocation

- **Pareto-Efficient**
  No other allocation improves utility without harming others

---

Zahedi et al. “REF: Resource elasticity fairness with sharing incentives for multiprocessors” [ASPLOS’14]
REF is an allocation mechanism that guarantees game-theoretic desiderata for shared chip multiprocessors

- **Sharing Incentives**
  Users perform no worse than under equal division

- **Envy-Free**
  No user envies another’s allocation

- **Pareto-Efficient**
  No other allocation improves utility without harming others

- **Strategy-Proof**
  No user benefits from lying

Zahedi et al. “REF: Resource elasticity fairness with sharing incentives for multiprocessors” [ASPLOS’14]
Cobb-Douglas Utility

\[ u(x) = \prod_{r=1}^{R} x_r^{\alpha_r} \]

- \( u \) utility (e.g., performance)
- \( x_r \) allocation for resource \( r \) (e.g., cache size)
- \( \alpha_r \) elasticity for resource \( r \)

- Cobb-Douglas fits preferences in computer architecture
- Exponents model diminishing marginal returns
- Products model substitution effects

Zahedi et al. “REF: Resource elasticity fairness with sharing incentives for multiprocessors” [ASPLOS’14]
Example Utilities

\[ u_1 = x_1^{0.6} y_1^{0.4} \quad u_2 = x_2^{0.2} y_2^{0.8} \]

- \( u_1, u_2 \) performance
- \( x_1, x_2 \) allocated memory bandwidth
- \( y_1, y_2 \) allocated cache size

Zahedi et al. “REF: Resource elasticity fairness with sharing incentives for multiprocessors” [ASPLOS’14]
Possible Allocations

- 2 users
- 12MB cache
- 24GB/s bandwidth
Envy-Free (EF) Allocations

- Identify EF allocations for each user
  - $u_1(A_1) \geq u_1(A_2)$
  - $u_2(A_2) \geq u_2(A_1)$

---

Zahedi et al. “REF: Resource elasticity fairness with sharing incentives for multiprocessors” [ASPLOS’14]
Pareto-Efficient (PE) Allocations

• No other allocation improves utility without harming others

Zahedi et al. “REF: Resource elasticity fairness with sharing incentives for multiprocessors” [ASPLOS’14]
Fair Allocations

Fairness = Envy-freeness + Pareto efficiency

Many possible fair allocations!

Zahedi et al. “REF: Resource elasticity fairness with sharing incentives for multiprocessors” [ASPLOS’14]
Mechanism for Resource Elasticity Fairness

Profile preferences

Fit utility function

Normalize elasticities

Allocate proportionally

Guarantees desiderata

- Sharing incentives
- Envy-freeness
- Pareto efficiency
- Strategy-proofness

Zahedi et al. “REF: Resource elasticity fairness with sharing incentives for multiprocessors” [ASPLOS’14]
Profiling for REF

Profile preferences

Fit utility function

Normalize elasticities

Allocate proportionally

Off-line profiling
- Synthetic benchmarks

Off-line simulations
- Various hardware

Machine learning
- $\alpha = 0.5$, then update

Zahedi et al. “REF: Resource elasticity fairness with sharing incentives for multiprocessors” [ASPLOS'14]
Fitting Utilities

Profile preferences

Fit utility function

Normalize elasticities

Allocate proportionally

\[ u = \prod_{r=1}^{R} x_r^{\alpha_r} \]

\[ \log(u) = \sum_{r=1}^{R} \alpha_r \log(x_r) \]

Use linear regression to find \( \alpha_r \)

Zahedi et al. “REF: Resource elasticity fairness with sharing incentives for multiprocessors” [ASPLOS’14]
- Utility is instructions per cycle
- Resources are cache size, memory bandwidth

Zahedi et al. “REF: Resource elasticity fairness with sharing incentives for multiprocessors” [ASPLOS’14]
Normalizing Utilities

- Profile preferences
- Fit utility function
- Normalize elasticities
- Allocate proportionally

- Compare users’ elasticities on same scale
- \( u = x^{0.2}y^{0.3} \rightarrow u = x^{0.4}y^{0.6} \)

Zahedi et al. “REF: Resource elasticity fairness with sharing incentives for multiprocessors” [ASPLOS’14]
Allocating Proportional Shares

Profile preferences

Fit utility function

Normalize elasticities

Allocate proportionally

\[ u_1 = x_1^{0.6} y_1^{0.4} \quad u_2 = x_2^{0.2} y_2^{0.8} \]

\[ x_1 = \left( \frac{0.6}{0.6+0.2} \right) \times 24 = 18 \text{GB/s} \]

\[ x_2 = \left( \frac{0.2}{0.6+0.2} \right) \times 24 = 6 \text{GB/s} \]

Zahedi et al. “REF: Resource elasticity fairness with sharing incentives for multiprocessors” [ASPLOS’14]
Equal Slowdown versus REF

- Equal slow-down provides neither SI nor EF
- Canneal receives < half of cache, memory

Zahedi et al. “REF: Resource elasticity fairness with sharing incentives for multiprocessors” [ASPLOS’14]
Equal Slowdown versus REF

- Equal slow-down provides neither SI nor EF
- Canneal receives < half of cache, memory

- Resource elasticity fairness provides both SI and EF
- Canneal receives more cache, less memory

Zahedi et al. “REF: Resource elasticity fairness with sharing incentives for multiprocessors” [ASPLOS’14]
Performance versus Fairness

- Measure weighted instruction throughput
- REF incurs < 10% penalty

Zahedi et al. “REF: Resource elasticity fairness with sharing incentives for multiprocessors” [ASPLOS’14]
Heterogeneity for Efficiency

Heterogeneous datacenters deploy mix of server- and mobile-class hardware

- “Web search using mobile cores” [ISCA’10]
- “Towards energy-proportional datacenter memory with mobile DRAM” [ISCA’12]

Heterogeneity and Markets

Agents bid for heterogeneous hardware in a market that maximizes welfare

- “Navigating heterogeneous processors with market mechanisms” [HPCA’13]
- “Strategies for anticipating risk in heterogeneous system design” [HPCA’14]

Sharing and Game Theory

Agents share multiprocessors with game-theoretic fairness guarantees

- “REF: Resource elasticity fairness with sharing incentives for multiprocessors” [ASPLOS’14]
Heterogeneity for Efficiency
Heterogeneous datacenters deploy mix of server- and mobile-class hardware
- Processors – hardware counters for CPI stack
- Memories – simulator for cache, bandwidth activity

Heterogeneity and Markets
Agents bid for heterogeneous hardware in a market that maximizes welfare
- Processors – simulator for core performance
- Server Racks – queueing models (e.g., M/M/1)

Sharing and Game Theory
Agents share multiprocessors with game-theoretic fairness guarantees
- Memories – simulator for cache, bandwidth utility