BigHouse: A Simulation Infrastructure for Data Center Systems

David Meisner, Junjie Wu, Thomas F. Wenisch
The University of Michigan

ISPASS 2012
Data center research is challenging

- Tools only evaluate a single component
  - Processor, caches, DRAM, disks, etc...
- Evaluating an entire server is difficult
  - Requires coordination of many slow tools
  - Models far too much detail

Simulations finish just after PhD awarded...
Requirements for data center simulation

• Handle scope of data center problems
  – E.g., Performance, power, thermal, reliability
  – **Goal**: General modeling infrastructure

• Publicly distributable
  – **Goal**: Does not rely on proprietary binaries

• Needs to simulate 1 to 10,000 servers
  – **Goal**: Scalable to clusters of machines

_Solution greatly differs from traditional systems tools_
The BigHouse Simulator

• Framework for simulating data center systems
  – Discrete-event simulation w/ statistical rigor
  – Easily extensible to new domains

• Claim: Queuing theory is correct abstraction
  – Simulate at the granularity of requests/tasks
    (i.e., request enters/exits a server)
  – Represent workloads as statistical distributions
  – Stochastic approach allows parallelization

Can simulate systems in minutes instead of hours or days
Outline

• Introduction
• BigHouse
• Case Studies
• Evaluation
• Conclusion
BigHouse Overview

Stochastic queuing simulation methodology:
1. Workload characterization
2. System modeling
3. Statistically-rigorous discrete-event simulation

BigHouse implements Step 3
BigHouse Features

BigHouse provides:

- Base objects for modeling (extend with object-oriented programming)
- *Statistical probes* to instrument object metrics (per-task)
- Statistically-rigorous sampling of outputs w/ parallel histogram approx.
- Automatic parallelization across cores/machines

User provides:

- Model to represent new behavior (e.g., Markov chain to modulate queue)
- New workload distributions (or reuse existing workload suite)

*User burden is minimized to modeling new behaviors*

*Statistical tricks are encapsulated into simulator*
Workload Characterization

Interarrival Times (Live Service)

Service Times (Offline)

- May need to capture more complicated effect (e.g., correlation)
- Similar to statistical simulation [Oskin 2000, Eeckhout 2004]

*Used to create a stochastically generated trace file*
System Modeling

Observations of system model fed to statistical probes
Statistics Sub-system

Power Model

150W

Statistics Manager

Probes

Sojourn Time

Server Power

10ms

Automatically determines simulation convergence
Outline

• Introduction
• BigHouse
• Case Studies
• Evaluation
• Conclusion
Case Study 1: Google Web search [ISCA2011]

What happens if we change power-perf. of leaf nodes?
Web Search Model

CPU and Memory performance modulate service rate

\[
\text{Power} = F(S_{\text{Mem}}, S_{\text{CPU}})
\]

\[
\text{Slowdown} = F(S_{\text{Mem}}, S_{\text{CPU}})
\]
Web Search Model Validation

Simulation accurately predicts saturation points
Average error of 9.2%
Case Study 2: Power capping

- Better use of power infrastructure = large gains
  - Throttle power w/ DVFS during infrequent spikes
  - Power Capping - ~40% more servers [Fan '07]

How to model power capping system w/ Bighouse?
Power Capping Model

Power Models
\[ P_{\text{Server}} = f(U_{\text{server}}, f_{\text{cpu}}) \]
\[ P_{\text{Rack}} = \sum P_{\text{server}}[i] \]

Performance Models
\[ \text{Service\_rate} = f(f_{\text{cpu}}) \]
\[ \text{Latency} = f(f_{\text{cpu}}) + \text{Queuing} \]

Power Management Policy
Set \( f_{\text{cpu}} \) s.t. \( P_{\text{server}} \propto U_{\text{server}} \)

**CPU and Memory performance modulate service rate**
Outline

- Introduction
- BigHouse
- Case Studies
- Evaluation
- Conclusion
Scalability

10,000 server systems can be simulated in hours
Parallel execution

• Sims can be distributed across cores/servers

Can achieve ~7x speedup on 16 slave for a single sim.
Conclusion

• BigHouse: Simulation tool for data center systems
  – Turnaround of hours rather than days
  – Statistically rigorous
  – Parallelizable

• Community engagement
  – Workload repository
  – New system models
  – Validation at scale
How to get BigHouse

- Download at: www.eecs.umich.edu/bighouse
- Will include:
  - Base server models
  - Workload distributions (inc. Google Web search)
- Will be available in ~1 week
Thank you!