A Comparison of High-Level Full-System Power Models

Suzanne Rivoire, *Sonoma State University*
Partha Ranganathan, *HP Labs*
Christos Kozyrakis, *Stanford University*

HotPower 2008

Who needs power models?

- Component and system designers
  - How do design decisions affect power?
- Users
  - How do my usage patterns affect power?
- Data center schedulers
  - How will workload distribution decisions affect power?

Talk Overview

- Power modeling goals and approaches
- Models compared
- Model generation and evaluation methodology
- Evaluation results

Power modeling goals

- Goal: Online, full-system power models
- Model requirements
  - Non-intrusive and low-overhead
  - Easy to develop and use
  - Fast enough for online use
  - Reasonably accurate (within 10%)
  - Inexpensive
  - Generic and portable

Power modeling approaches

- Detailed component models
  - Simulation-based
  - Hardware metric-based

- **High-level full-system models**

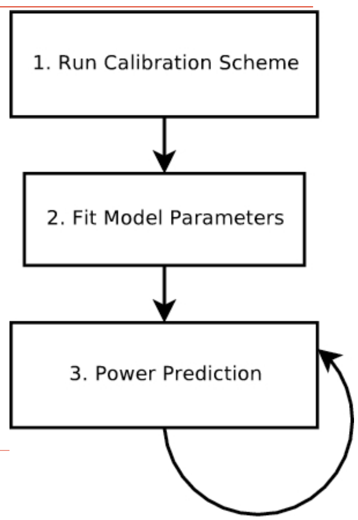
High-level models (Mantis)

Input: Common util. metrics → Equation → **Output:** Predicted power (system)

- How accurate?
- How portable?
- Tradeoff between model parameters/complexity and accuracy?

Power Modeling

- Run **one-time** calibration scheme (possibly at vendor)
  - Stress individual components: CPU, memory, disk
  - *Outputs:* time-stamped performance metrics & AC power measurements
- Fit model parameters to calibration data
- Use model to predict power
  - Inputs: performance metrics at each time *t*
  - Output: estimation of AC power at each time *t*



Models studied

- Constant power (the null model): $P = C_0$

- CPU utilization-based models

Input: CPU util. % → Equation → **Output:** Predicted power (system)

CPU utilization-based models

- Linear in CPU utilization

$$P = C_0 + C_1 u$$

- Empirical power model

$$P = C_0 + C_1 u + C_2 u^r$$

[Fan et al, ISCA 2007]

CPU + disk utilization

Input:
- CPU util. %
- Disk util. %

Equation →

Output:
Predicted power (system)

$$P = C_0 + C_1 u_{CPU} + C_2 u_{disk}$$

[Heath et al, PPoPP 2005]

CPU + disk util. + performance ctrs

Input:
- CPU util. %
- Disk util. %
- CPU perfctrs

Equation →

Output:
Predicted power (system)

$$P = C_0 + C_1 u_{CPU} + C_2 u_{disk} + \sum C_i P_i$$

[D. Economou, S. Rivoire, C. Kozyrakis, P. Ranganathan, MoBS 2006]

CPU performance counters

- Configurable processor registers to count microarchitectural events
- In this study:
  - Memory bus transactions
  - Unhalted CPU clock cycles
  - Instructions retired/ILP
  - Last-level cache references
  - Floating-point instructions

Evaluation methodology

- Run calibration suite and develop models on a variety of machines
- Run benchmarks, collecting metrics and AC power
- Compare predicted power from metrics with measured AC power

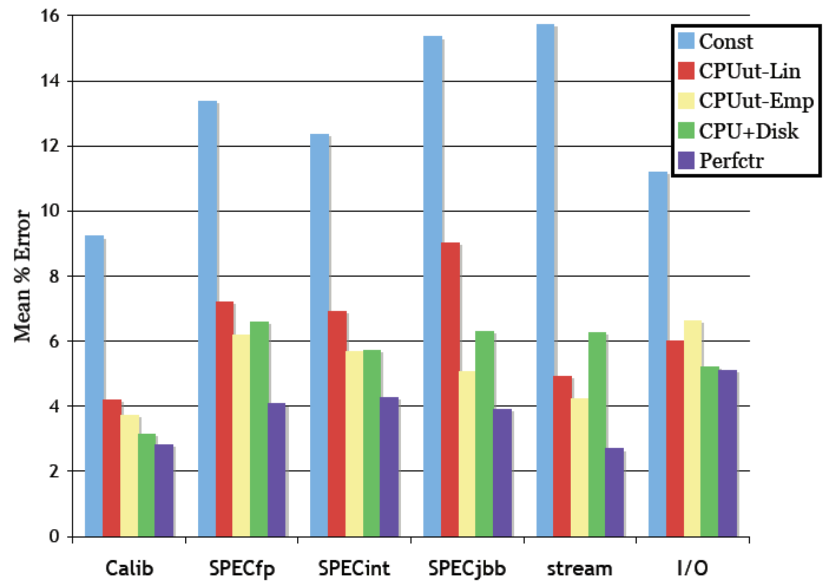
Evaluation machines

- Mobile fileserver with 1 and 13 disks
  - Highest and lowest frequencies
- 2005-era AMD laptop
  - Highest and lowest frequencies
- 2005-era Itanium server
- 2008-era Xeon server with 32 GB FBDIMM
- *Variety in component balance, processor, domain, dynamic range*

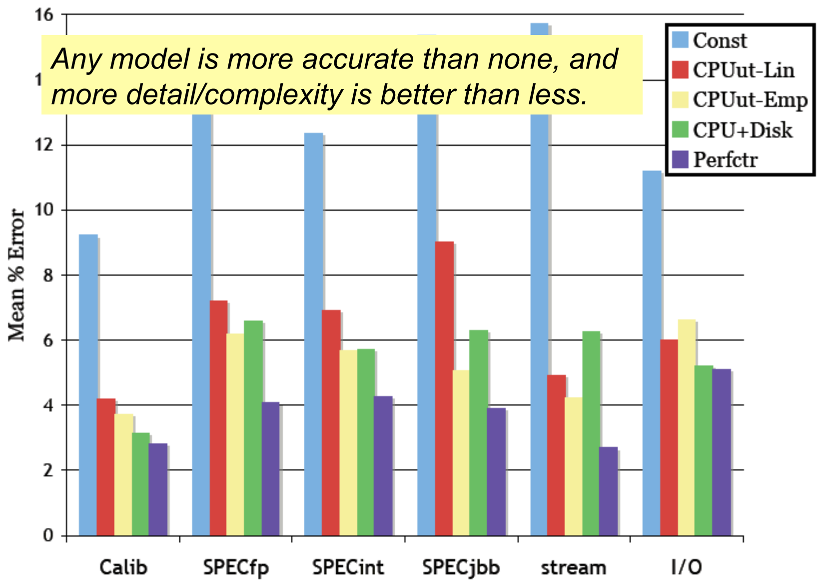
Evaluation benchmarks

- SPECcpu int and fp
  - Laptop: gcc and gromacs only
- SPECjbb
- Stream
- I/O-intensive programs
  - ClamAV
  - Nsort (mobile fileserver only)
  - SPECweb (Itanium only)

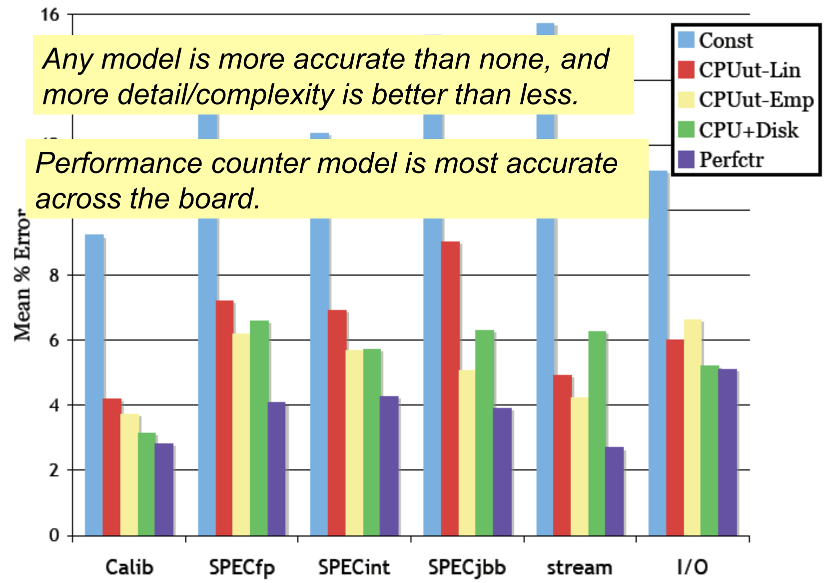
Overall mean % error



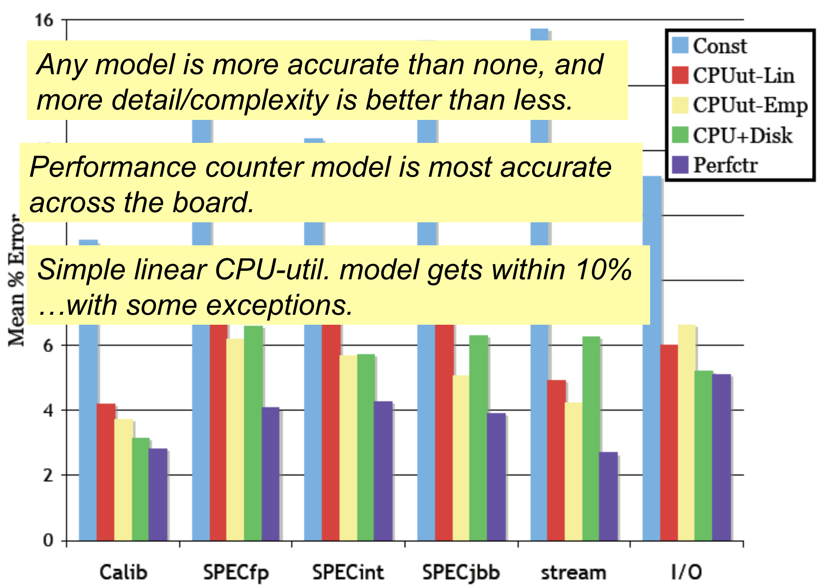
Overall mean % error



Overall mean % error

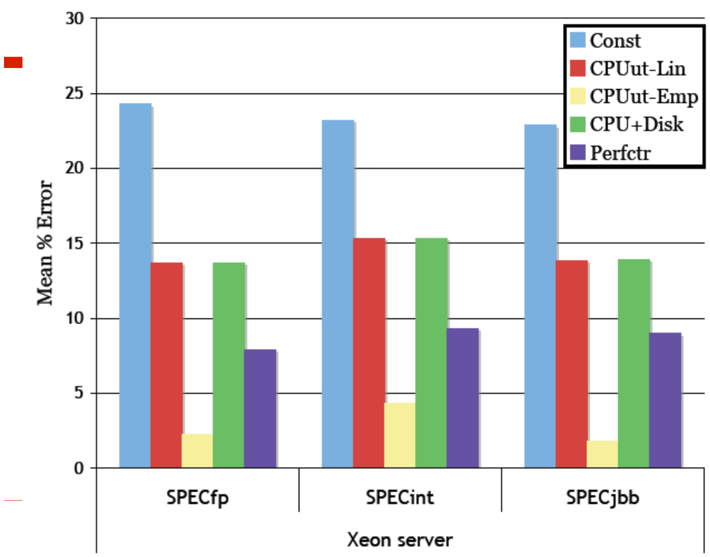


Overall mean % error

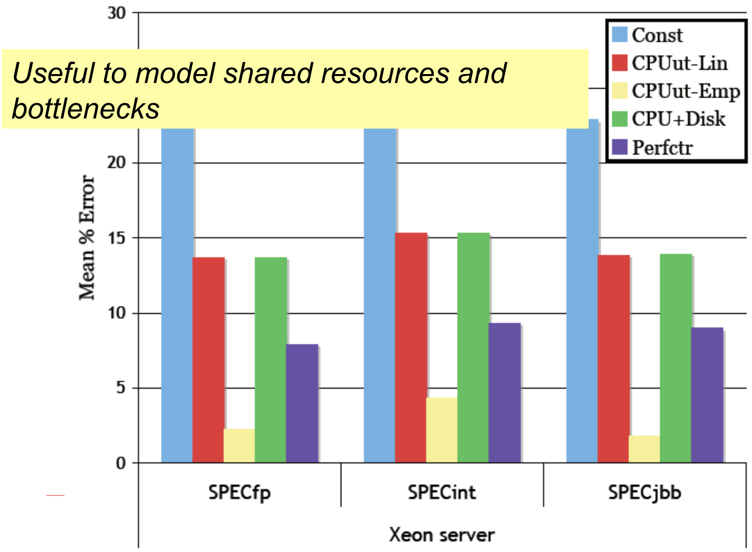


Best case for empirical CPU model

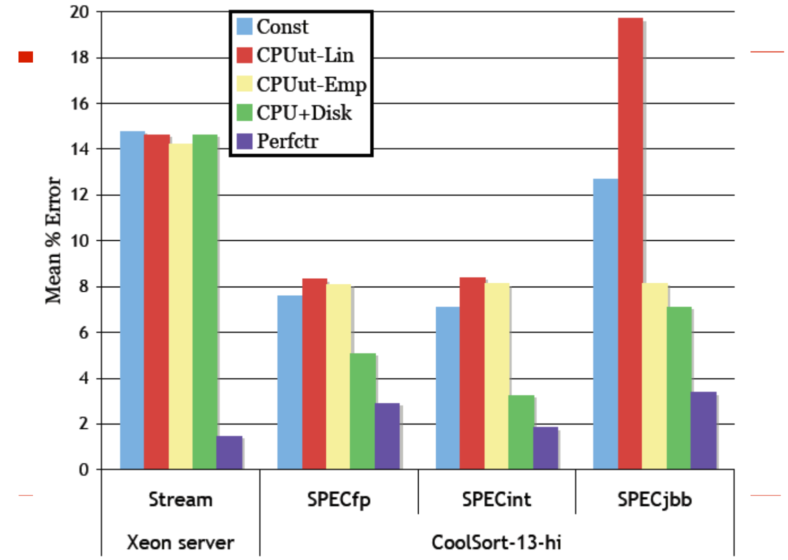
(Xeon server)



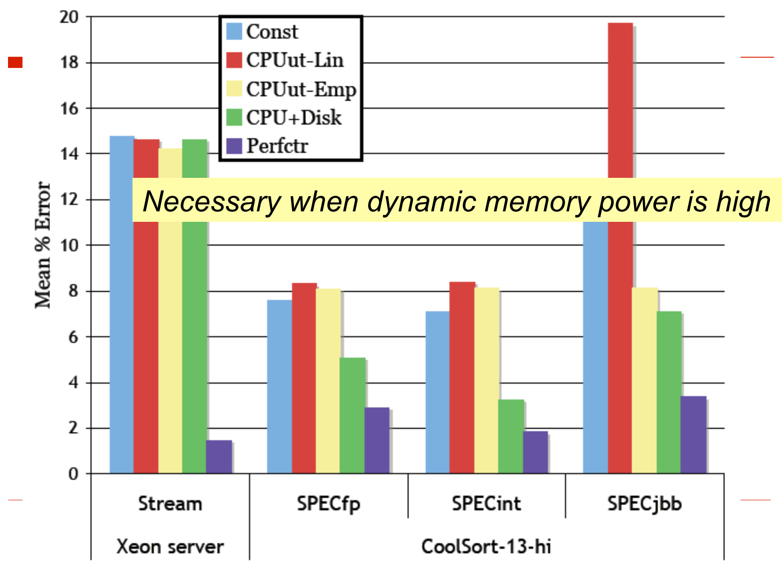
Best case for empirical CPU model
(Xeon server)



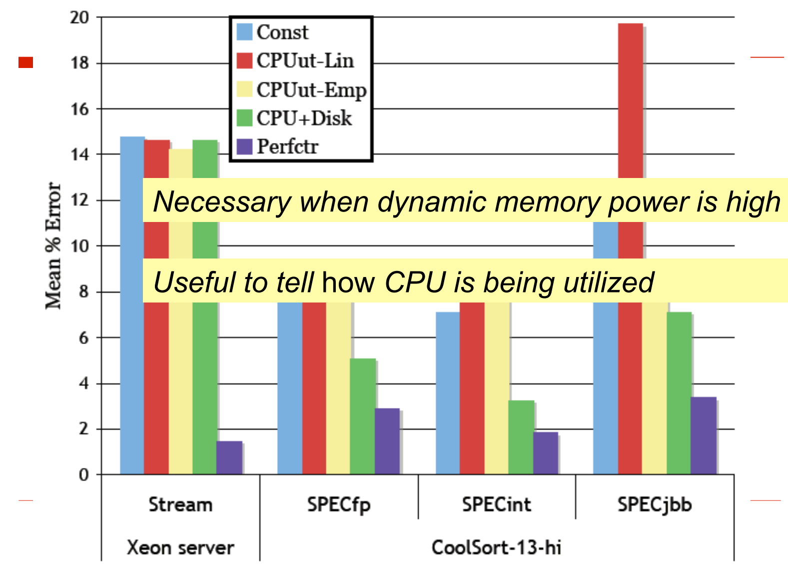
Best case for performance counters
(Xeon server and mobile fileserver-13)



Best case for performance counters
(Xeon server and mobile fileserver-13)



Best case for performance counters
(Xeon server and mobile fileserver-13)



Future work

- Beyond CPU, memory, and disk
  - GPUs
  - Network (not a factor today)
- Model complexity
  - Combine exponential CPU model w/ perfctrs?
  - Cooling – fan power is cubic function of speed

Conclusions

- Generic approach to power modeling yields accurate results
  - Simple models overall have < 10% error
  - Same parameters across very different machines
  - More information ➔ better models
- Linear CPU util. model not enough for…
  - Machines and workloads that are not CPU-dominated
  - CPUs with shared resource bottlenecks
  - Aggressively power-optimized CPUs
  - *…all of which reflect hardware trends.*