Models and Metrics for Energy-Efficient Computer Systems

Suzanne Rivoire

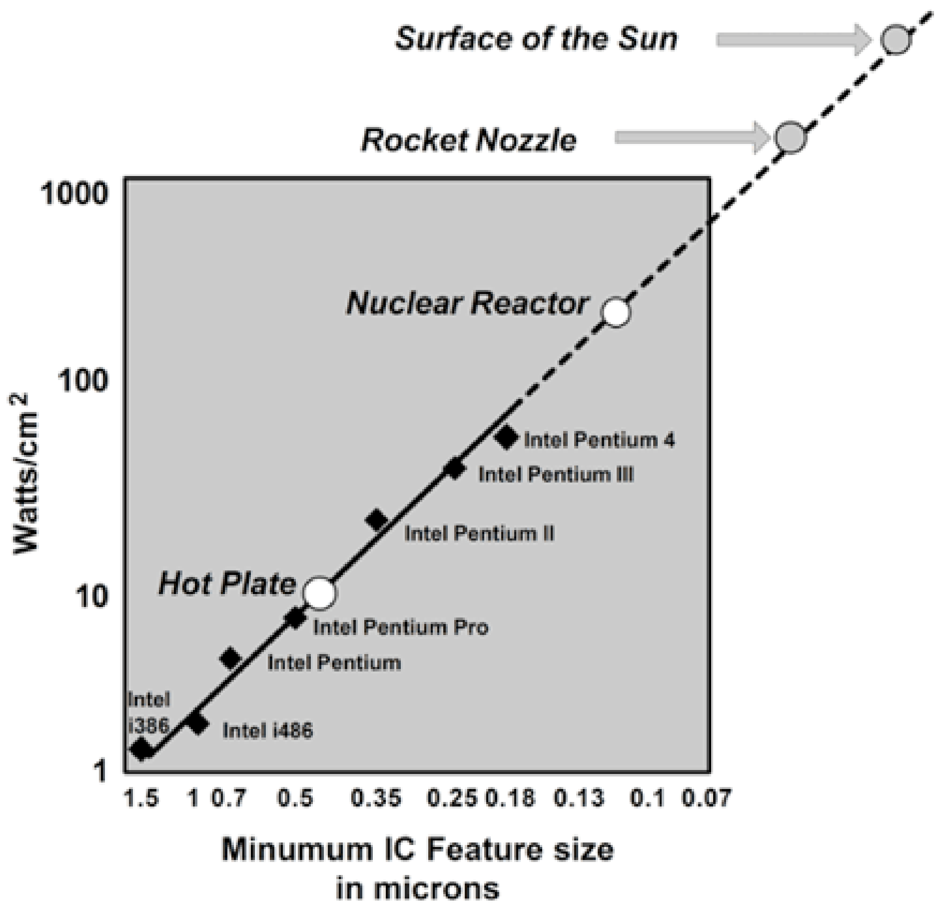
May 22, 2007

Ph.D. Defense

EE Department, Stanford University

Power and Energy Concerns

- Processors: power density



[Borkar, Intel]

Power and Energy Concerns (2)

- Personal computers
  - Mobile devices: battery life/usability
  - Desktops: electricity costs, noise
- Servers and data centers
  - Power and cooling costs
  - Reliability
  - Density/scalability
  - Pollution
  - Load on utilities

Underlying Questions

- Metrics: What are we aiming for?
  - Compare energy efficiency
  - Identify / motivate new designs
- Models: How do we get there?
  - Understand how high-level properties affect power
  - Improve power-aware scheduling policies / usage

Talk Overview

- **Metrics:** JouleSort benchmark
  - First complete, full-system energy-efficiency benchmark
  - Design of winning system
- **Models:** Mantis approach
  - Generates family of high-level full-system models
  - Generic, accurate, portable

JouleSort energy-efficiency benchmark

- JouleSort benchmark specification
  - Workload, metric, guidelines
  - Rationale and pitfalls

- Energy-efficient system design: 2007 “winner”
  - 3.5× better than previous best
  - Insights for future designs

[S. Rivoire, M. A. Shah, P. Ranganathan, C. Kozyrakis, “JouleSort: A Balanced Energy-Efficiency Benchmark,” SIGMOD 2007.]

Why a benchmark?

- Track progress, compare systems, spur innovation
- Current benchmarks/metrics

EMBC the green grid™ ENERGY STAR spec

- Limitations of current metrics:
  - Under-specified or "under construction"
  - Limited to a particular component or domain

Benchmark design goals

- **Holistic** and **balanced**: exercises all core components
- **Inclusive** and **representative**: meaningful and implementable on many different machines
- **History-proof**: meaningful comparisons between scores from different years

Benchmark specification overview

- Workload
- Metric
- Rules

Workload: External sort

- Sort randomly permuted 100-byte records with 10-byte keys
- From file on non-volatile store to file on non-volatile store (“external” storage)

External sort workload

- **Simple** and **balanced**
  - Exercises all core components
  - CPU, memory, disk, I/O, OS, filesystem
  - End-to-end measure of improvement
- **Inclusive** of variety of systems
  - PDAs, laptops, desktops, supercomputers
- **Representative** of sequential I/O tasks
- Technology trend bellwether
  - Supercomputers to clusters, GPU?

Existing sort benchmarks

- Sort benchmarks used since 1985
- Pure performance
  - MinuteSort: How many records sorted in 1 min?
  - Terabyte: How much time to sort 1 TB?
- Price-performance
  - PennySort: How many records sorted for $0.01?
  - Performance-Price: MinuteSort/$$

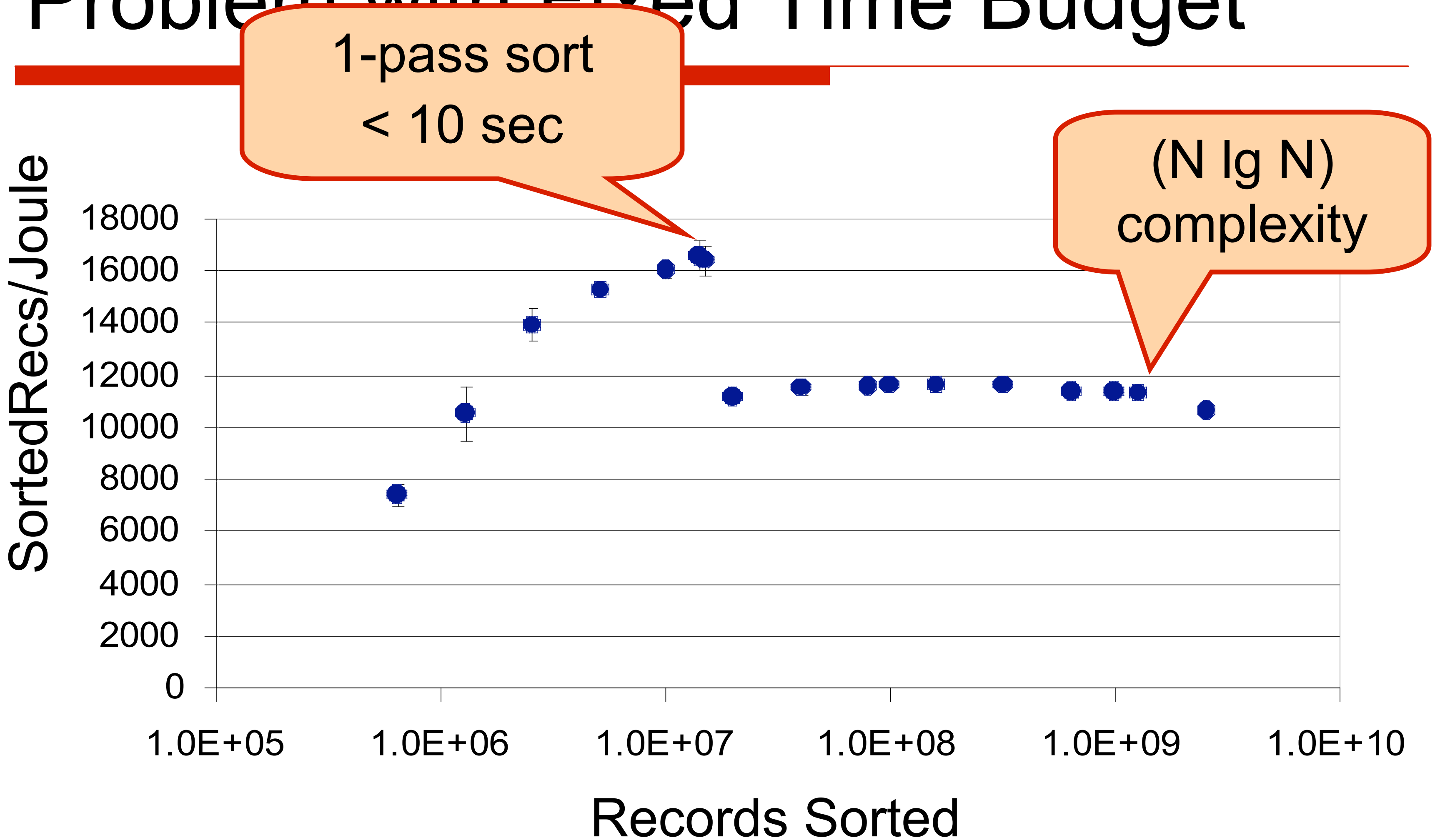
More info at http://research.microsoft.com/barc/SortBenchmark/

JouleSort metric choices

- How to weigh power and performance?
  - **Equally (energy)?**
    - Energy (Joules) = Power (Watts) × Time (sec.)
  - Privilege performance (energy-delay product)?

- What to fix and what to compare?
  - Fix energy budget and compare records sorted?
  - **Fix num. records and compare energy?**
  - Fix time budget and compare records/Joule?

Problem with Fixed Time Budget

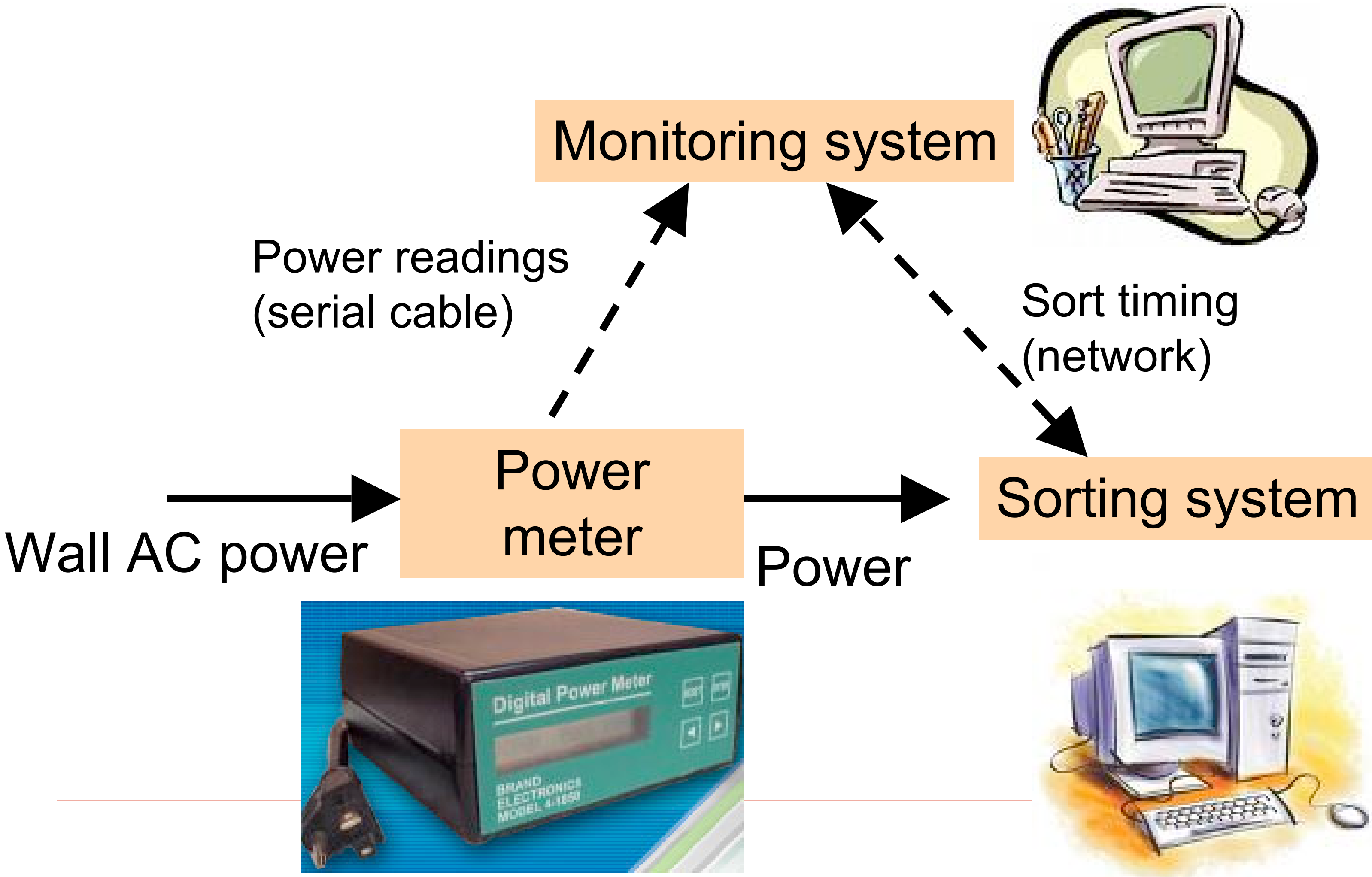


Final metric: Fixed input size

- 3 classes: 10GB, 100GB, 1TB
- Winner: minimum energy
- Report (records sorted / Joule)

- Inter-class comparisons imperfect
- Adjust classes as technology improves

Energy measurement setup



Talk Overview

- **Metrics:** JouleSort benchmark
  - First complete, full-system energy-efficiency benchmark
  - Design of winning system
- **Models:** Mantis approach
  - Generates family of high-level full-system models
  - Generic, accurate, portable

Representative systems

	Disks	CPU %	SRecs	Pwr (W)	SRecs/J
GPUTeraSort (estimated)	9	n/a	59GB	290	~3200
Blade	1	11%	5GB	90	~300
Low-end server	2	26%	10GB	140	~1200
Laptop	1	1%	10GB	22	~3400
Commodity fileserver	12	>90%	10GB	406	~3800

Representative systems

	Disks	CPU %	SRecs	Pwr (W)	SRecs/J
GPUTeraSort (estimated)	9	n/a	59GB	290	~3200
Blade	1	11%	5GB	90	~300
Low-end server	2	26%	10GB	140	~1200
Laptop	1	1%	10GB	22	~3400
Commodity fileserver	12	>90%	10GB	406	~3800

Representative systems

	Disks	CPU %	SRecs	Pwr (W)	SRecs/J
GPUTeraSort (estimated)	9	n/a	59GB	290	~3200
Blade	1	11%	5GB	90	~300
Low-end server	2	26%	10GB	140	~1200
Laptop	1	1%	10GB	22	~3400
Commodity fileserver	12	>90%	10GB	406	~3800

Energy-Efficient Components: Processor

Fileserver

CoolSort

Fileserver		CoolSort
Sort BW: 313 MB/s	75% perf →	Sort BW: 236 MB/s
65W (peak)	52% power	34W (peak)

Energy-Efficient Components: Disks

Fileserver

Our winner

Seagate Barracuda		Hitachi Travelstar
Seq. BW: 80MB/s	50% perf → 15% power	Seq. BW: 40MB/s
13W		2W

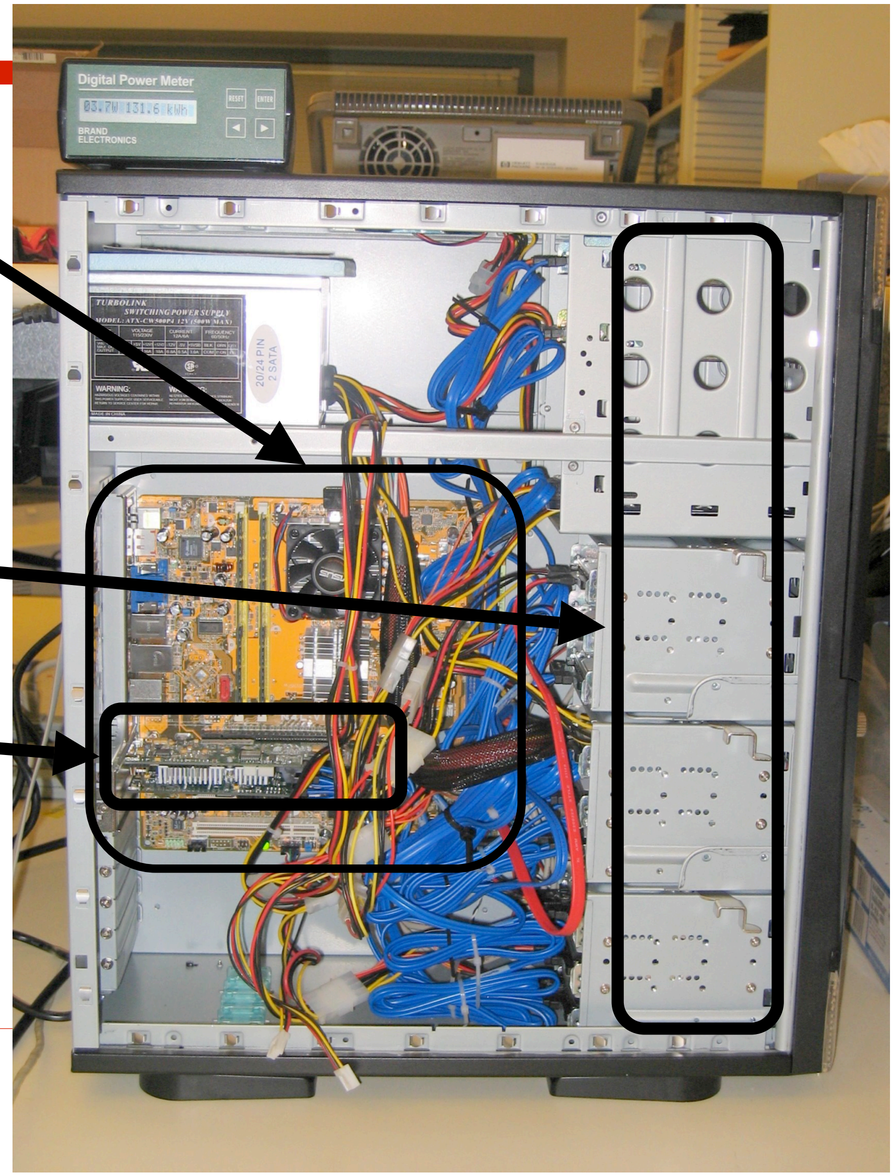
CoolSort Design

Asus motherboard:

Mobile CPU + 2 PCI-e slots

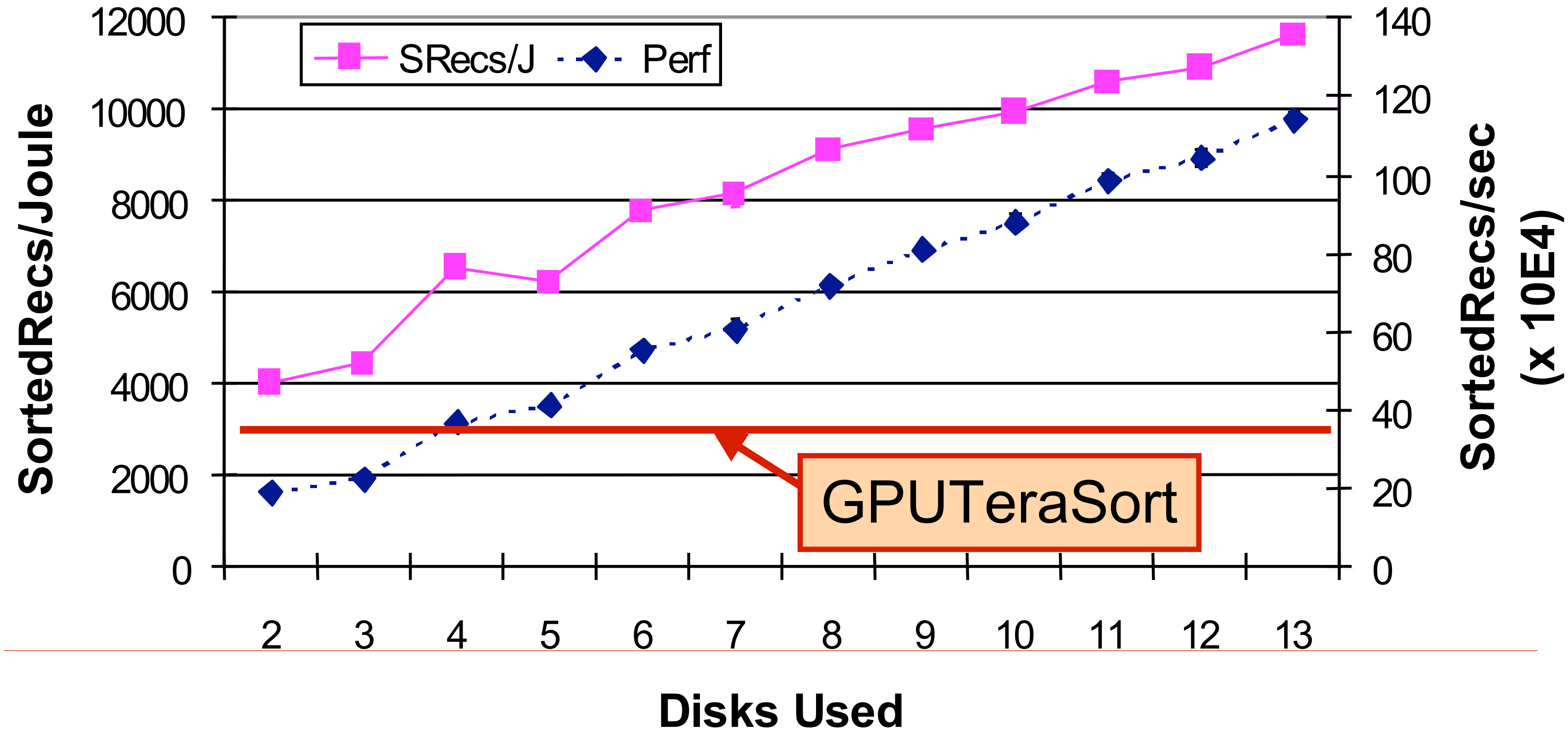
13 Hitachi TravelStar 160GB

RocketRAID Disk Controllers



Maximizing performance

- Balanced sort: enough disks to fully utilize CPU
- Disks running near peak BW



CoolSort: The 100 GB winner

- 11,300 records sorted per Joule
- 3.5× more efficient than GPUTeraSort
- Average sorting power: 100 W

Insights for future designs

- Low-hanging fruit: use low-power HW
  - Best power-performance trade-off
  - Still need to fully utilize resources
  - Challenge: adequate interfaces and “glue” to bring laptop components into servers
- Scaledown efficiency
  - Limited dynamic range
  - For fixed HW: peak efficiency = peak performance
  - How can we design machines that perform equally well in different benchmark classes?

Benchmark limitations

- Tests energy efficiency at high utilization -- but most servers are under-utilized
  - How efficient is system at 50% utilization? 20%?
- Doesn't measure building power/cooling
- Real goal: TCOSort
  - JouleSort and PennySort give pieces of the answer

JouleSort Conclusions

- Need energy-efficiency benchmark
- JouleSort specification
  - Simple, representative, full-system benchmark
  - Workload, metric, measurement rules
- CoolSort system
  - 3.5× better than 2006 estimated winner
  - Mobile components, server-class interfaces
- Part of the sort benchmark suite
  - *joulesort.stanford.edu*

Talk Overview

- **Metrics:** JouleSort benchmark
  - First complete, full-system energy-efficiency benchmark
  - Design of winning system
- **Models:** Mantis approach
  - Generates family of high-level full-system models
  - Generic, accurate, portable

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?

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**

Detailed models: Simulation-based

Input:
- Current state
- Architecture
- Circuit parameters

Simulation →

Output:
Predicted power (component)

- Inexpensive, arbitrarily accurate
- Not full-system
- Slow (not real-time)
- Not portable

Detailed models: Metric-based

Input:
- Design info
- HW counters

Equation $\longrightarrow$

Output:
Predicted power (component)

- Highly accurate
- Not full-system
- Complex, require specialized knowledge
- Not portable

[Contreras and Martonosi, ISLPED 2005]
[Isci and Martonosi, MICRO 2003]

High-level metrics (Mantis)

Input:
Common util. metrics

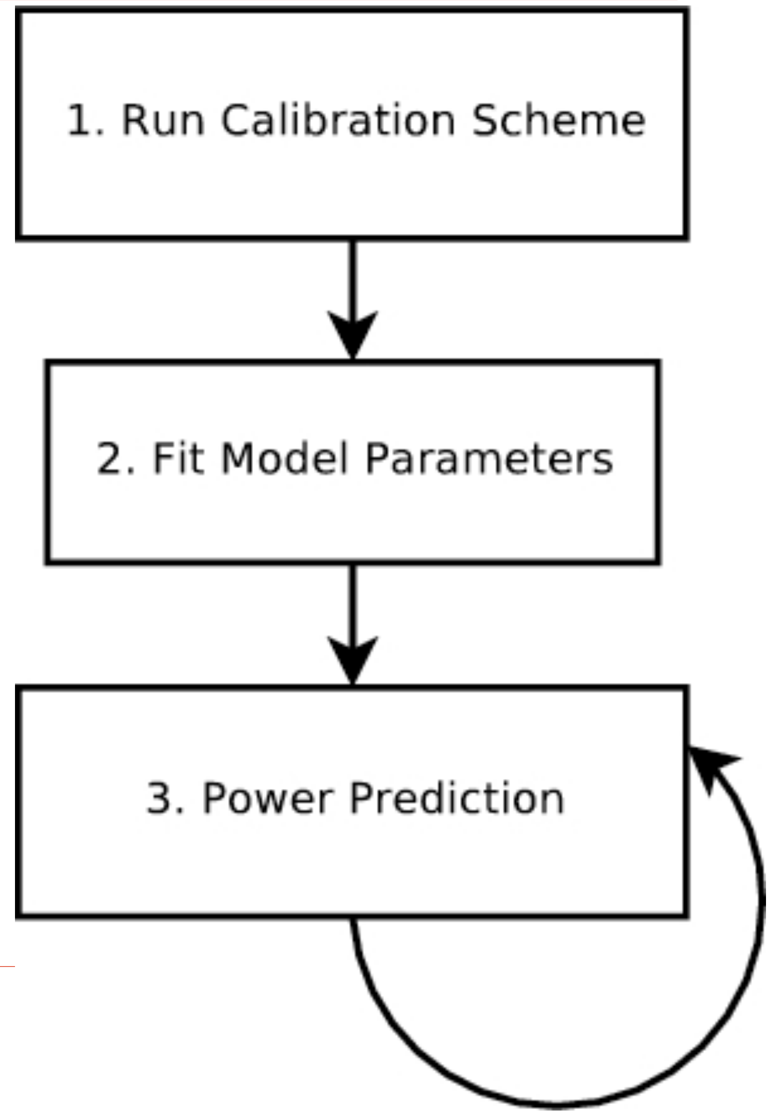
$\xrightarrow{\text{Equation}}$

Output:
Predicted power (system)

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

Power Modeling

- Run **<u>one-time</u>** 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. %

$\xrightarrow{\text{Equation}}$

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

$\xrightarrow{\text{Equation}}$

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
- Requires OS modification
- 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

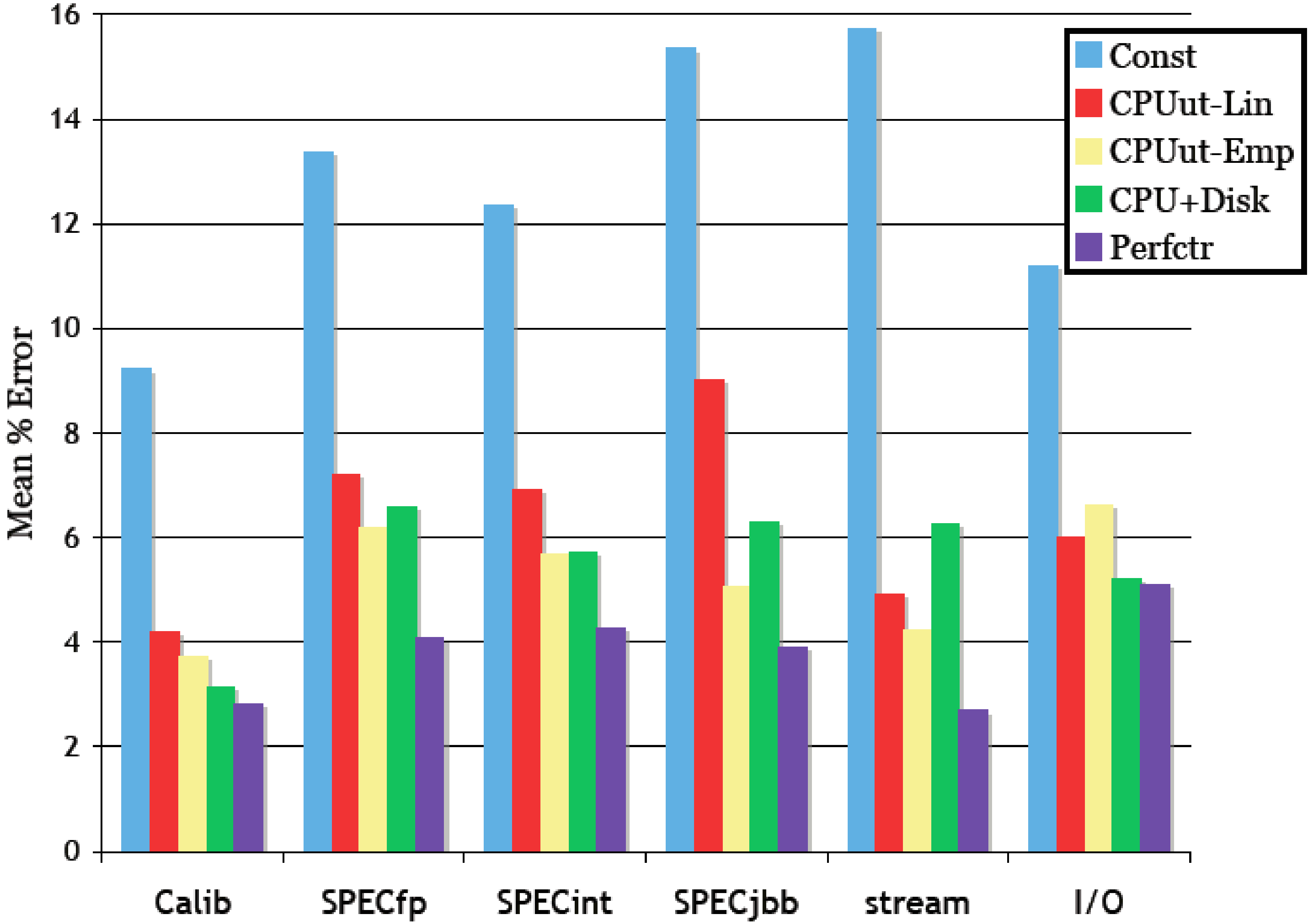
Evaluation machines

- CoolSort 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*

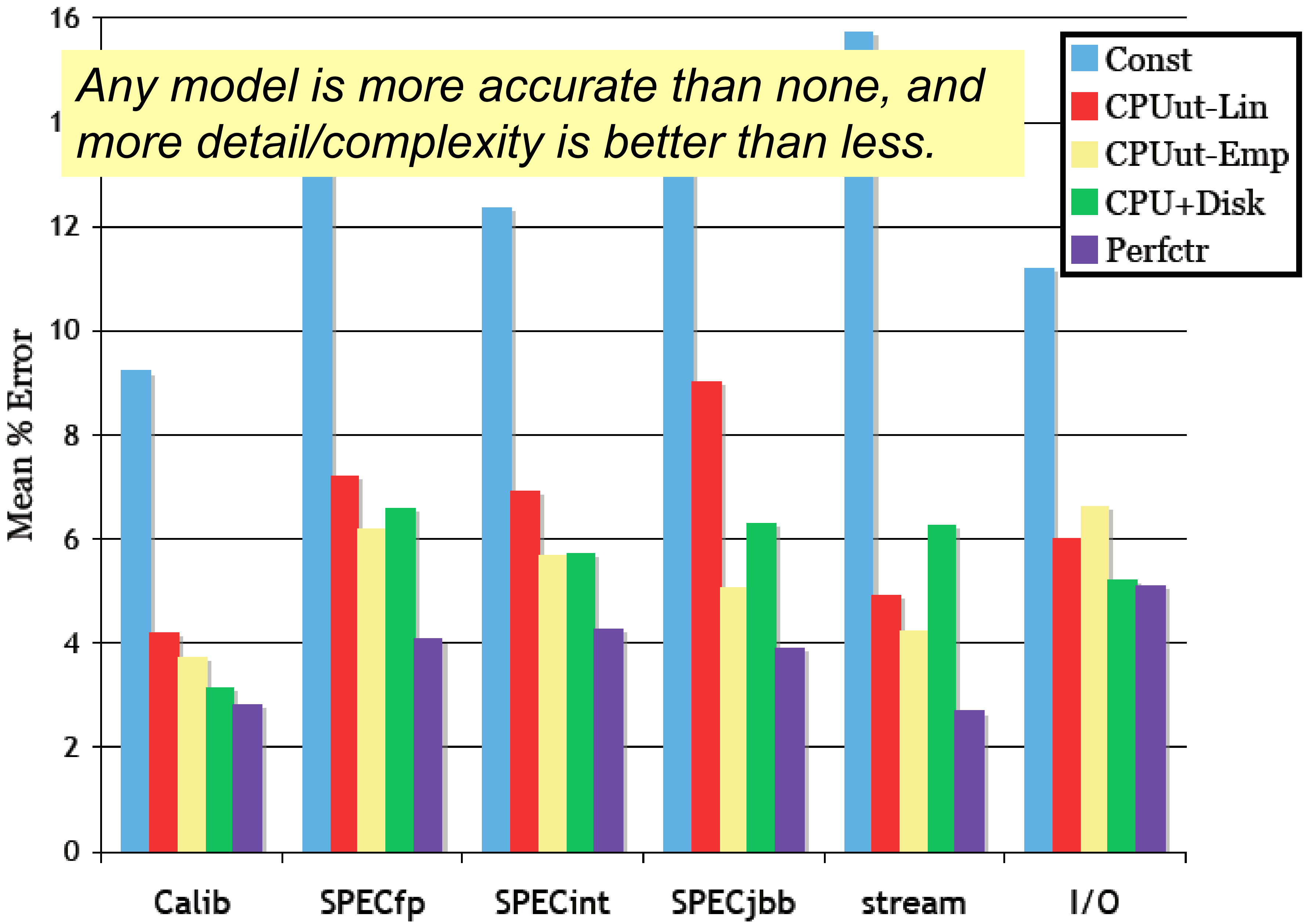
Evaluation benchmarks

- SPECcpu int and fp
  - Laptop: gcc and gromacs only
- SPECjbb
- Stream
- I/O-intensive programs
  - ClamAV
  - Nsort (CoolSort-13 only)
  - SPECweb (Itanium only)

Overall mean % error



Overall mean % error

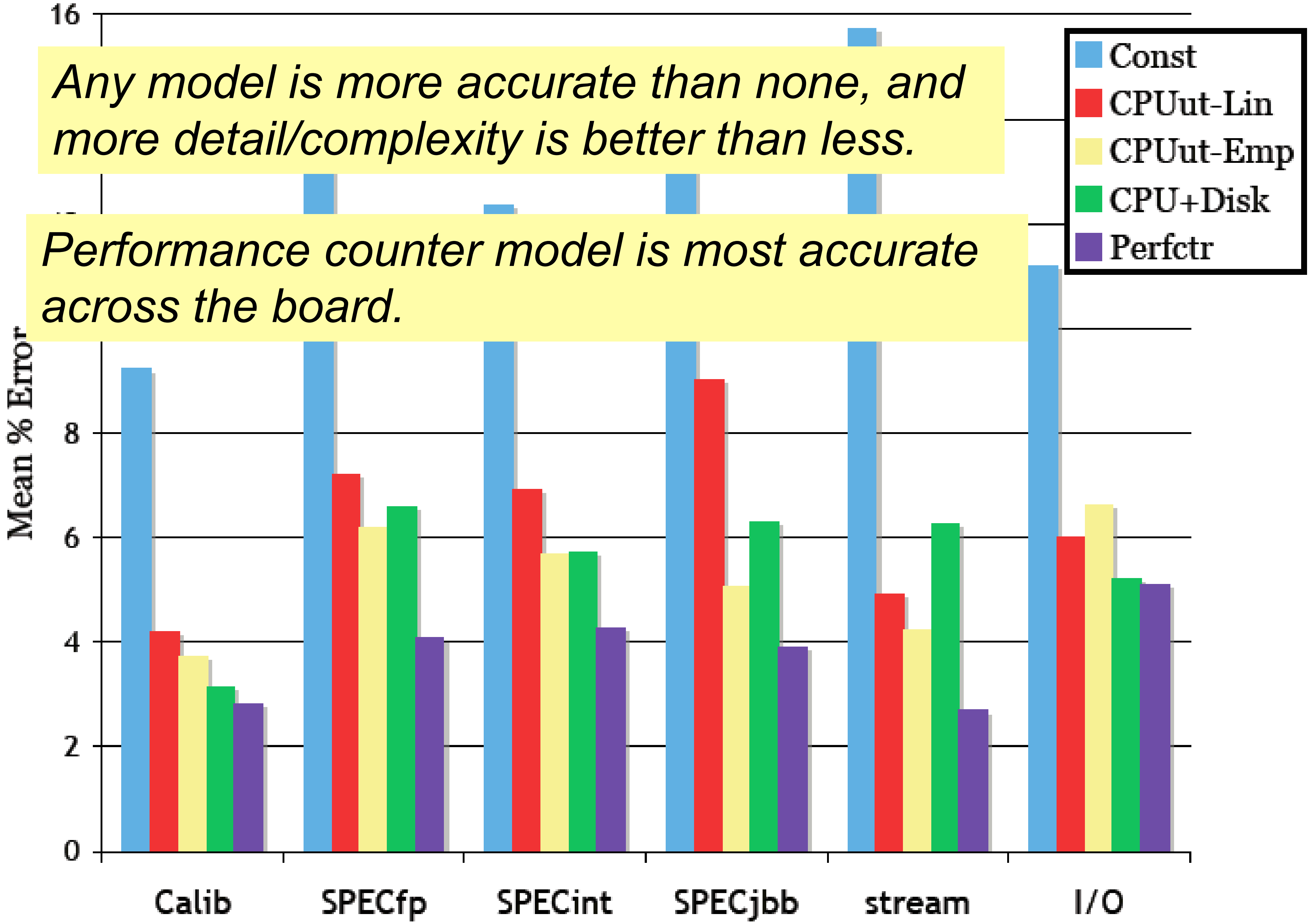


Any model is more accurate than none, and more detail/complexity is better than less.

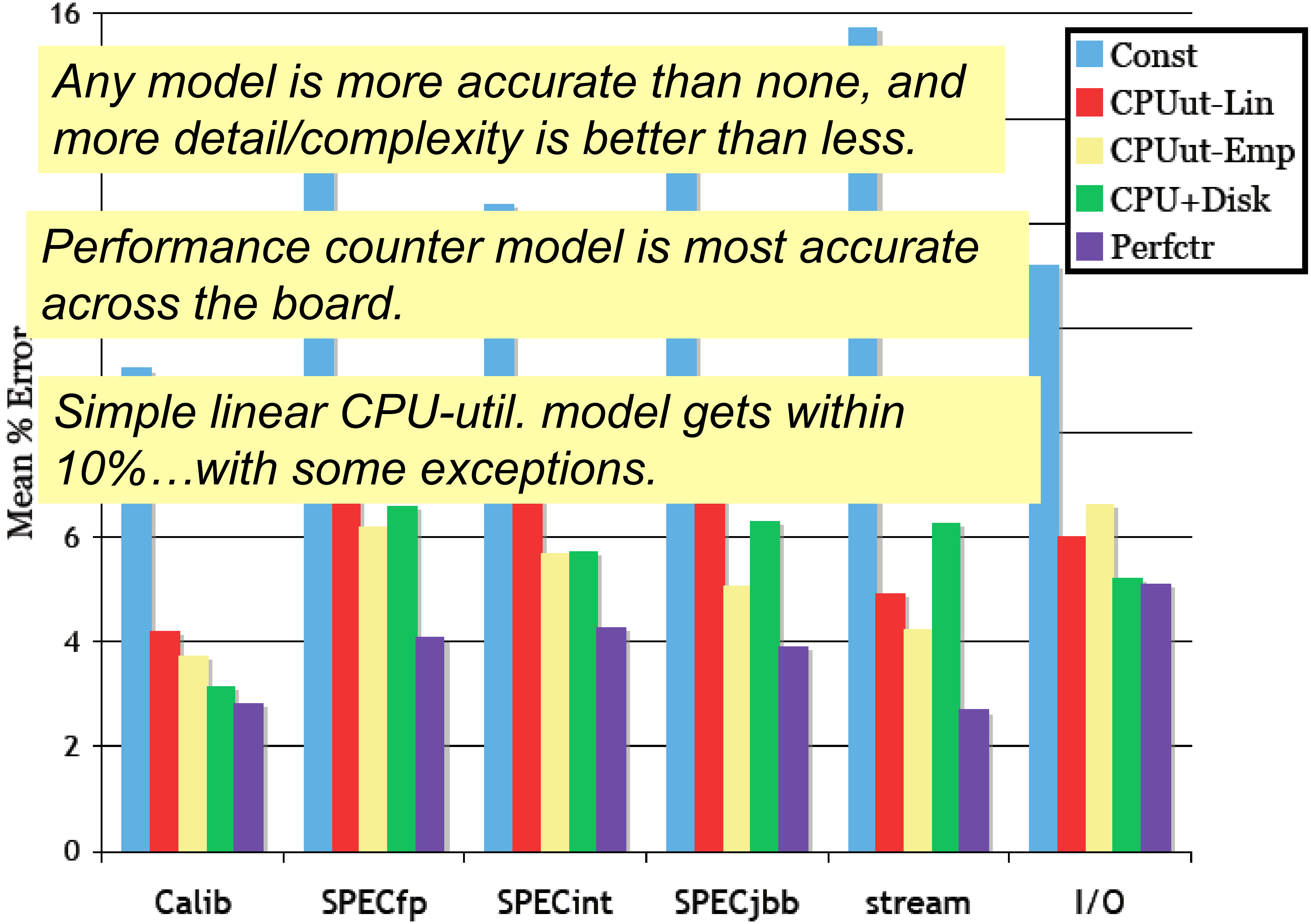
Overall mean % error

Any model is more accurate than none, and more detail/complexity is better than less.

Performance counter model is most accurate across the board.



Overall mean % error

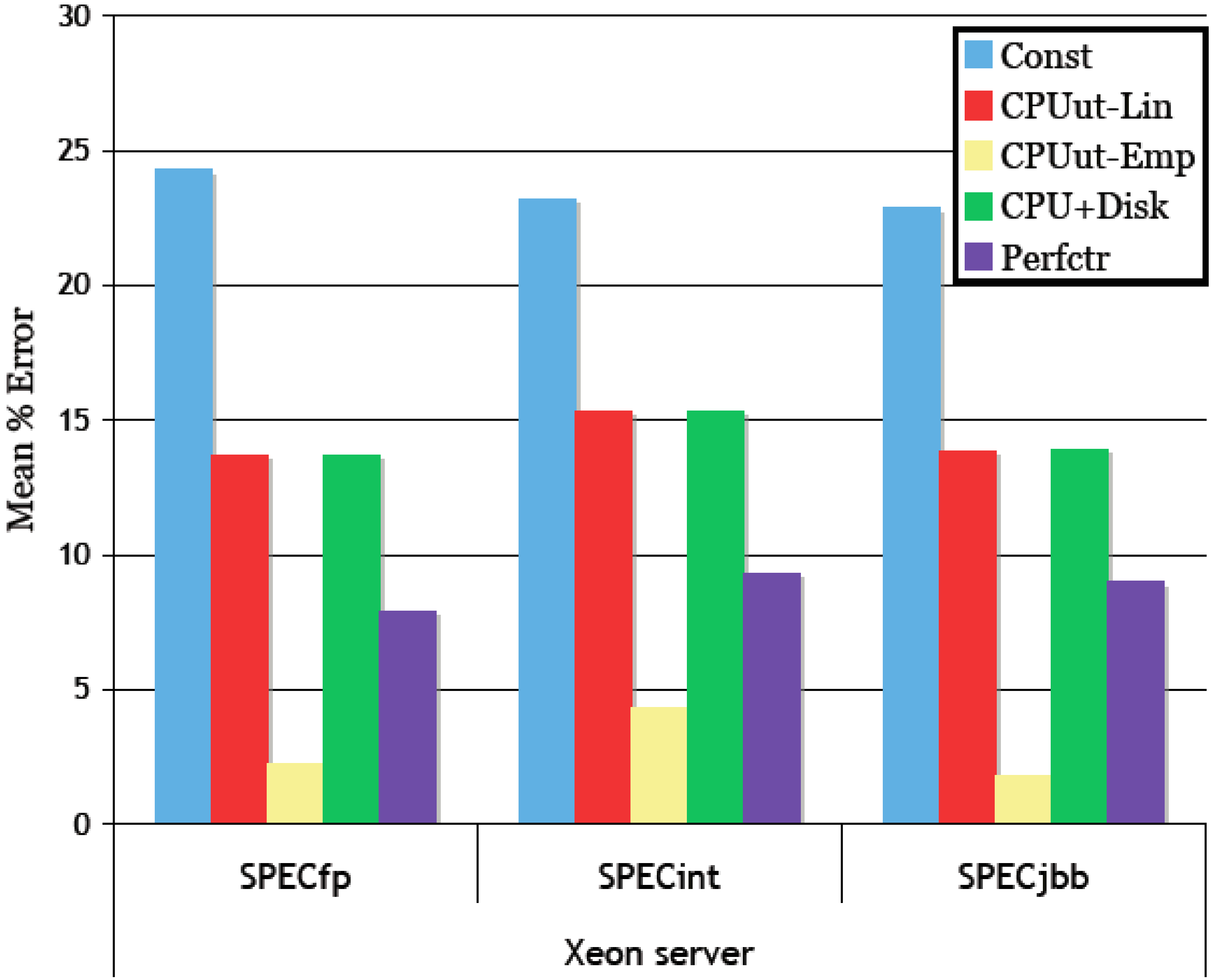


Any model is more accurate than none, and more detail/complexity is better than less.

Performance counter model is most accurate across the board.

Simple linear CPU-util. model gets within 10%…with some exceptions.

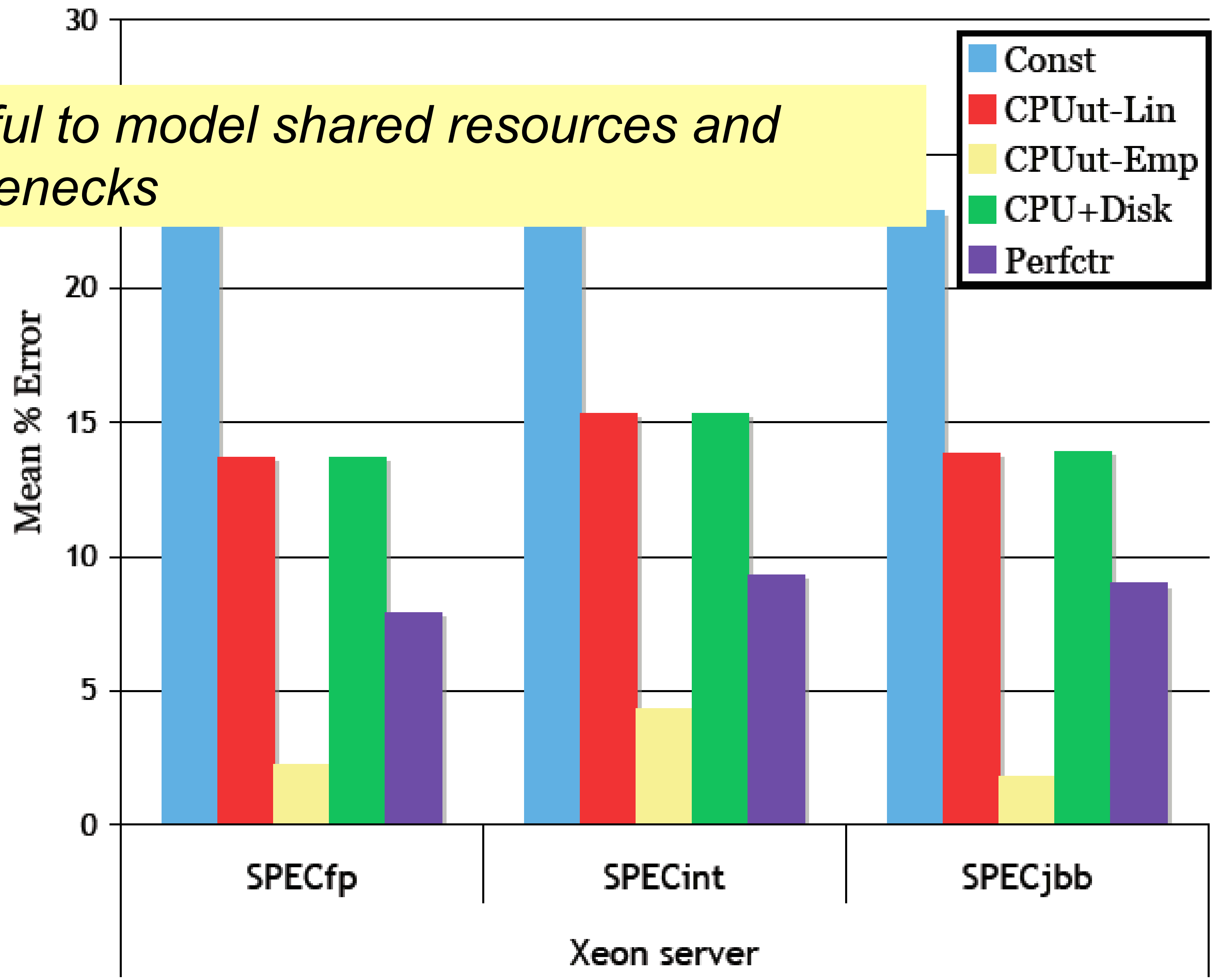
Best case for empirical CPU model
(Xeon server)



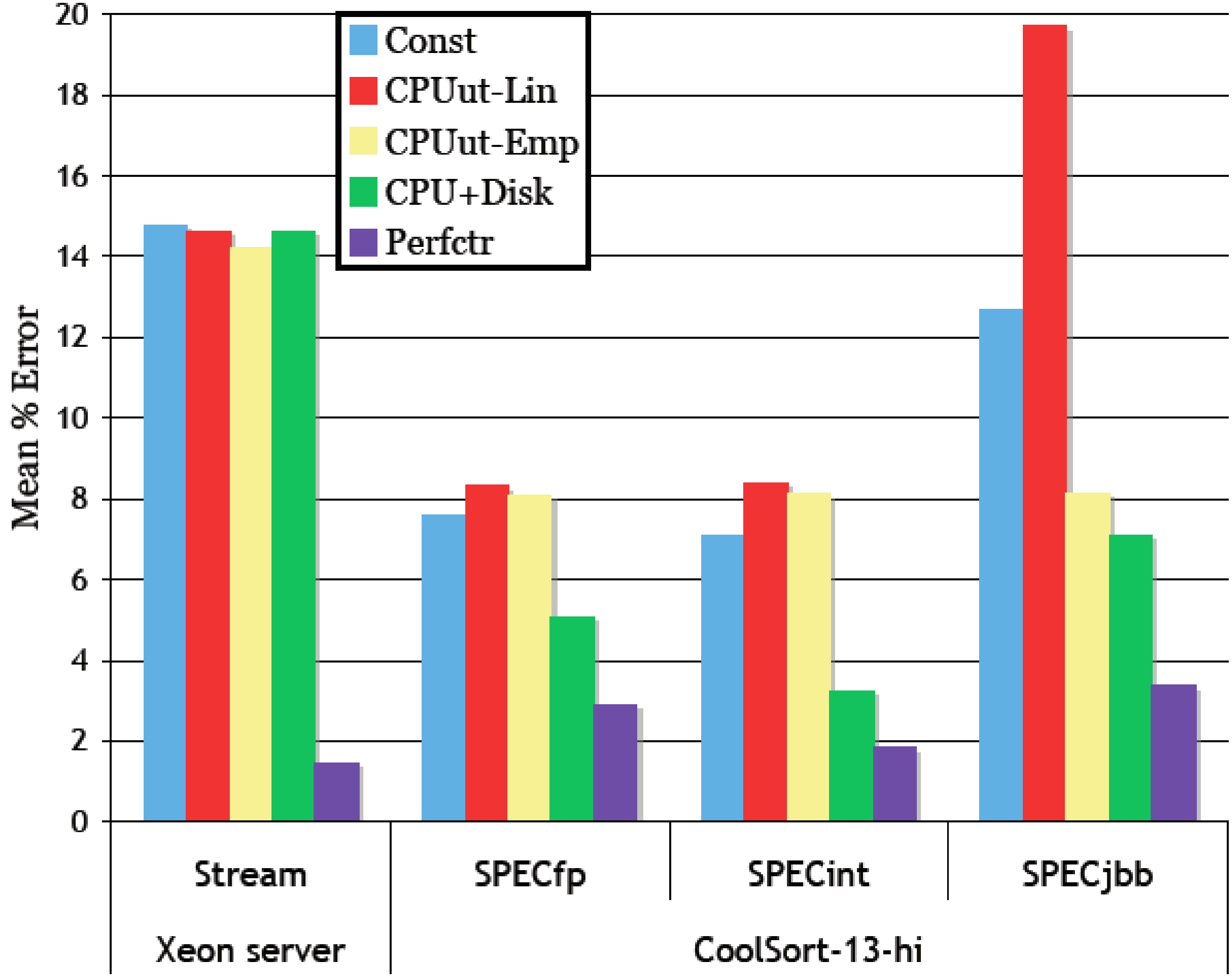
Best case for empirical CPU model

(Xeon server)

Useful to model shared resources and bottlenecks

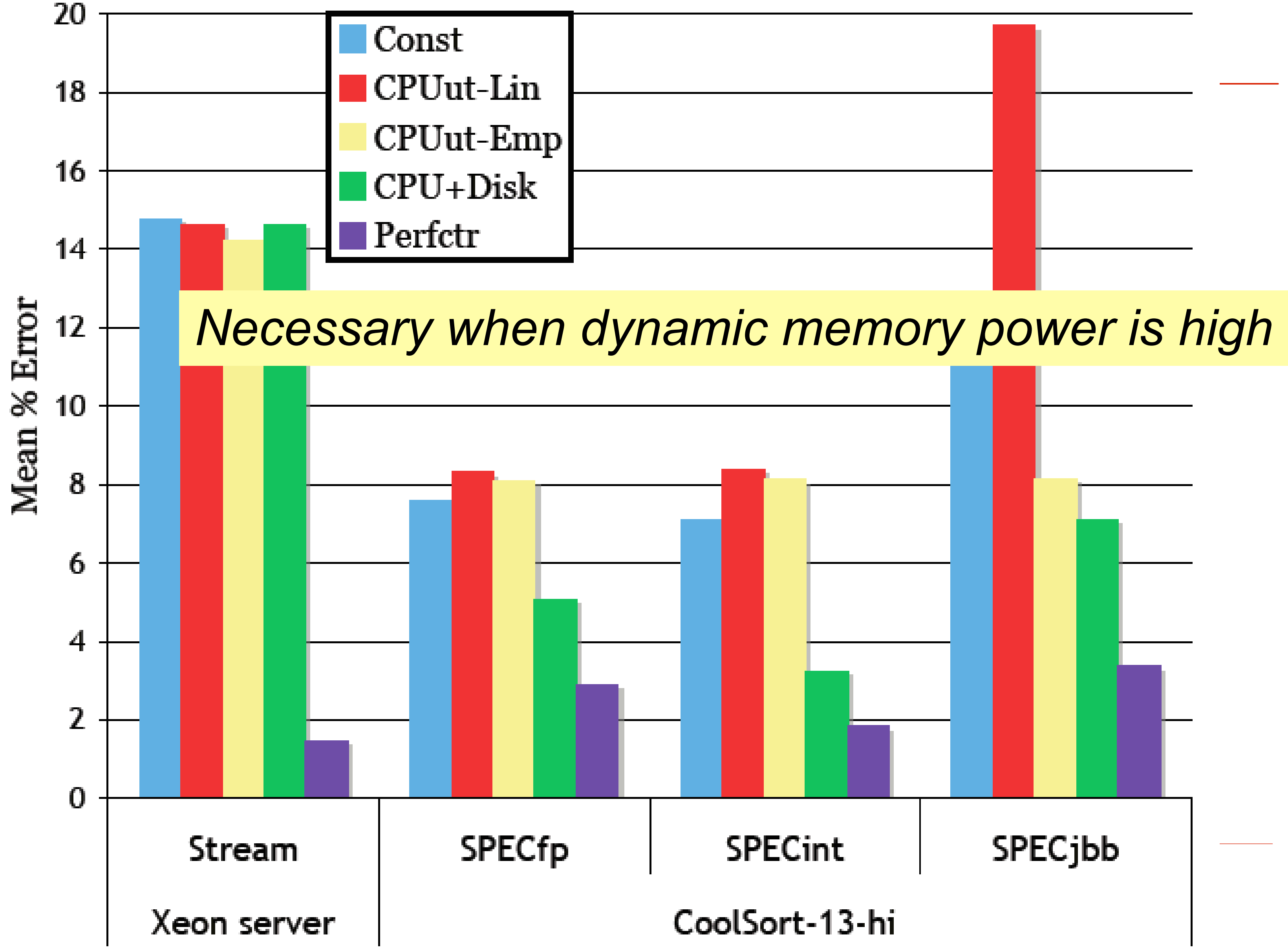


Best case for performance counters
(Xeon server and CoolSort-13)



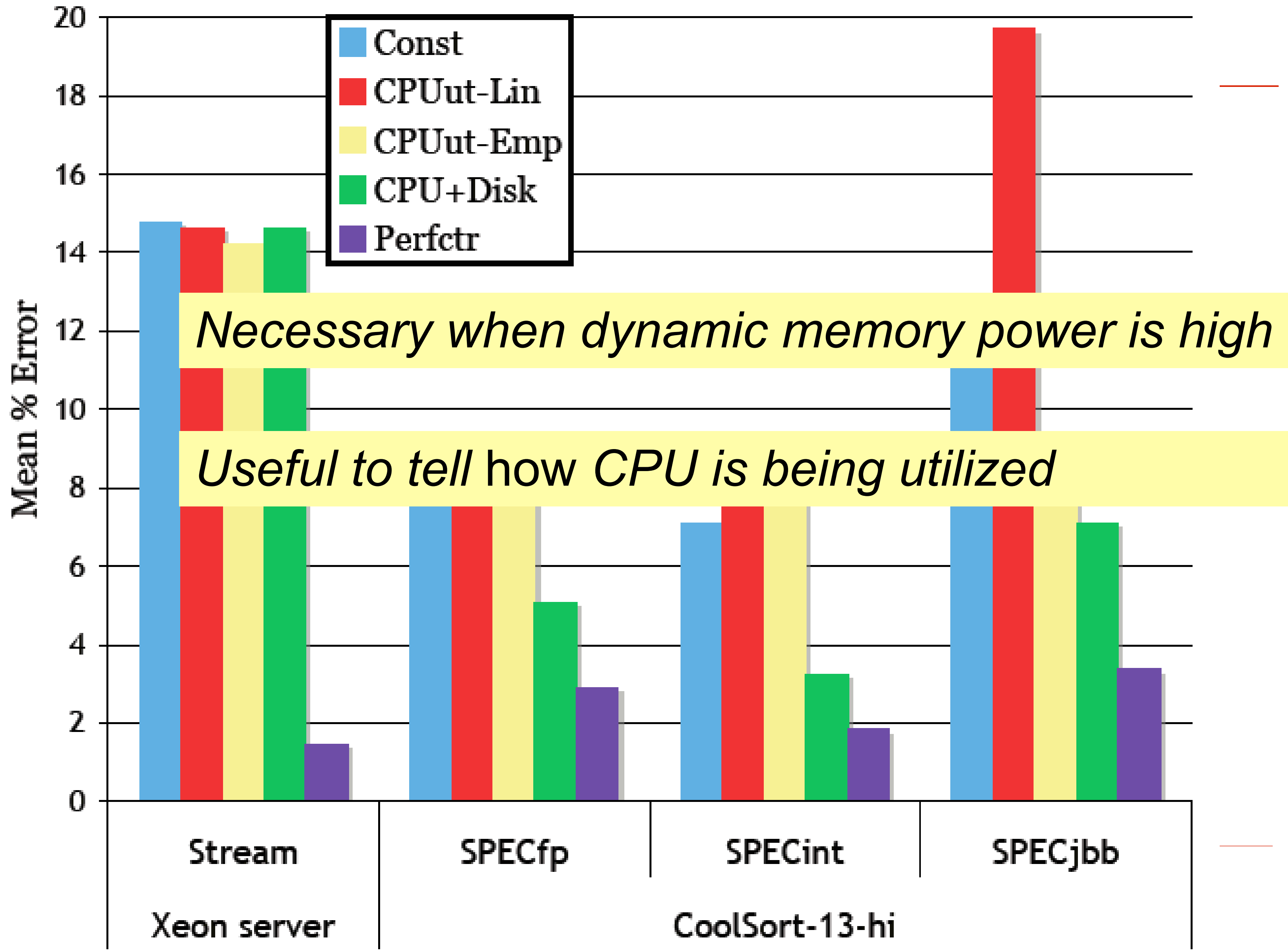
Best case for performance counters

(Xeon server and CoolSort-13)



Best case for performance counters

(Xeon server and CoolSort-13)



Modeling 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.*

Future work

- Beyond CPU, memory, and disk
  - GPUs
  - Network (not a factor today)
- Model complexity
  - Combine exponential CPU model w/ perfctrs?
  - Cooling?

Overall Summary

- Models and metrics needed to improve energy efficiency
- **Metrics:**
  - JouleSort energy-efficiency benchmark specification
  - Winning JouleSort machine
- **Models:**
  - Simple, portable high-level modeling technique
  - Trade-offs between accuracy and simplicity

Acknowledgments

- **Advisor:** Christos Kozyrakis
- **Mentor:** Partha Ranganathan
- **Committee:** Kunle Olukotun & Dwight Nishimura
- **Collaborators:** Mehul Shah, Dimitris Economou, Justin Meza
- **Assistance:** Jacob Leverich, HP Labs, Charlie Orgish, Teresa Lynn
- **Defense food!** Jayanth and Amin
- Architecture grad students
- Grant Gavranovic, Kelley Rivoire, friends & family