Star-Cap: Cluster Power Management Using Software-Only Models

John D. Davis

Suzanne Rivoire (rivoire@sonoma.edu)

Moisés Goldszmidt (Microsoft Research)

ICPP Workshop on Power-aware Algorithms, Systems, and Architectures (PASA)

Sept. 10, 2014

Power capping motivation

- Reduce waste from overprovisioning
- Provision for actual maximum power instead of sum of nameplate power
- Have a mechanism to throttle power consumption
- Major server manufacturers offer this feature; Intel offers at chip level (RAPL)

[Femal *ICAC* '05, Ranganathan *ISCA* '06, Lefurgy *ICAC* '07…]

The problem with vendor solutions

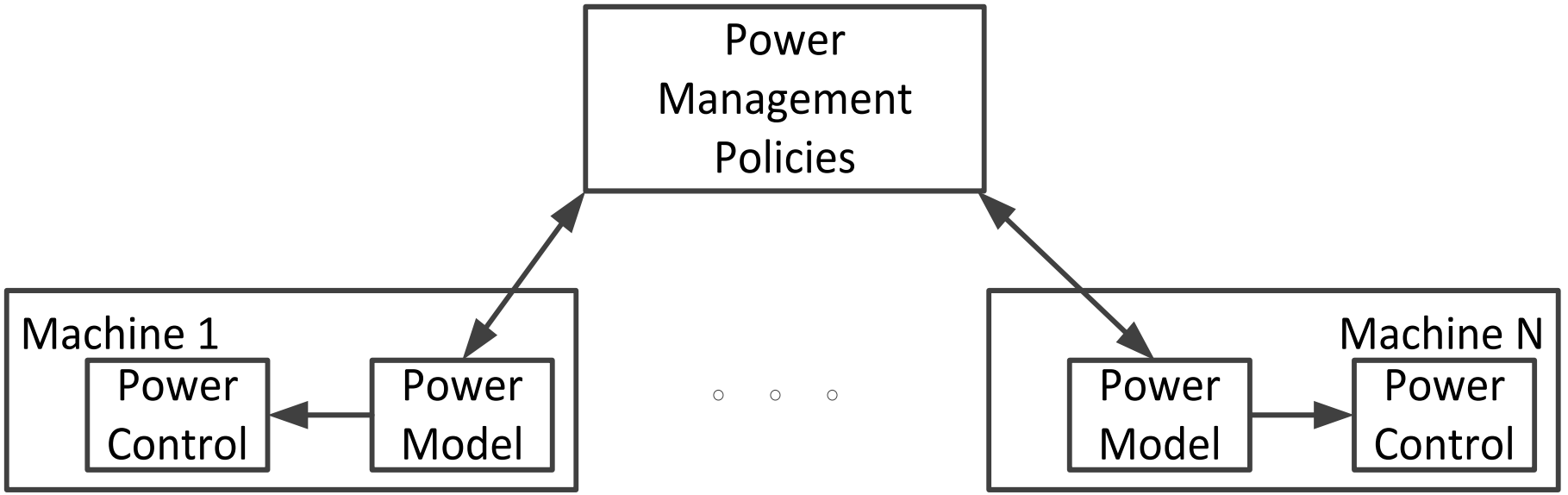
- Additional management hardware, additional cost *or* limited to chip
- Compare to trend of customized bare-bones servers…
- …and "wimpy nodes" for data-intensive workloads

Goal: eliminate cost of hardware instrumentation

Outline

- *Star-Cap overview*
- Software-only power models
- Power capping schemes
- Evaluation

Two-level scheme



- Top level: determine node power budgets
- Node level: enforce and report

Sensors and Actuators

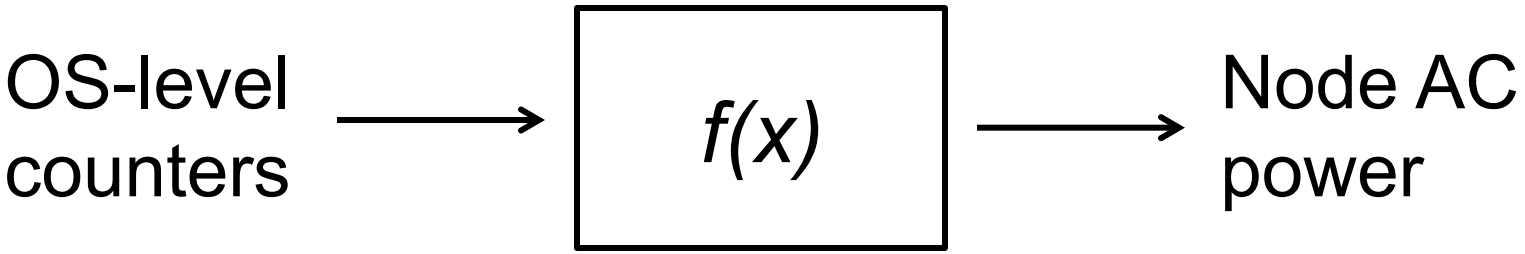
- Sensors: OS-level, architecture-independent performance counters

- Actuators:
  - For this work, DVFS states
  - Nothing prevents other mechanisms from being used

Outline

- Star-Cap overview
- *Software-only power models*
- Power capping schemes
- Evaluation

OS-level counters



- Full-system, not a specific component
- OS-level, architecture-independent counters
- Piecewise quadratic model, fit with MARS

[Davis et al., *IISWC* '12]

Model training process

1. ETW (Event Tracing for Windows)
   - Architecture counters: **~250**
   - Processor, physical and logical disk, network, memory, filesystem
2. Remove redundant counters: **~45**
   - Correlation Matrix (> |0.95|)
   - Performance counter definitions
3. Select features: **~10**
   - R glmpath with L1 regularization
   - Stepwise refinement

Outline

- Star-Cap overview
- Software-only power models
- *Power capping schemes*
- Evaluation

Star-Cap Overview

- Inputs to all schemes
  - Target node-level power consumption (set at top level)
  - Current power (modeled or measured)
  - List of available frequency states
- Outputs
  - List of frequency states available to OS
  - Let current OS policy select from available states

Threshold-based

- If $P_{current} < P_{lo}$
  - Make the next highest frequency state available
- If $P_{current} > P_{hi}$
  - Remove highest frequency state from available list
- Our thresholds:
  - P_{hi} = 95% of cap
  - P_{lo} = 90% of cap

Reactive Capping (ReCap)

- Adjust frequency state based on $P_{current}$
- After making a change, wait for it to settle before making another (reduce oscillations)
- Three versions:
  - M-ReCap: $P_{current}$ is measured power
  - L-ReCap: $P_{current}$ is predicted by a CPU-utilization-based linear model
  - C-ReCap: $P_{current}$ is predicted by quadratic power model in previous section

Proactive Capping (ProCap)

- Use quadratic power model to predict $P_{current}$
- Before changing available frequencies, predict P_{next}
  - Using next allowable frequency state
  - Keeping all other counters constant (oversimplification!)
- If P_{next} would violate threshold, don't bother adjusting available frequencies

Outline

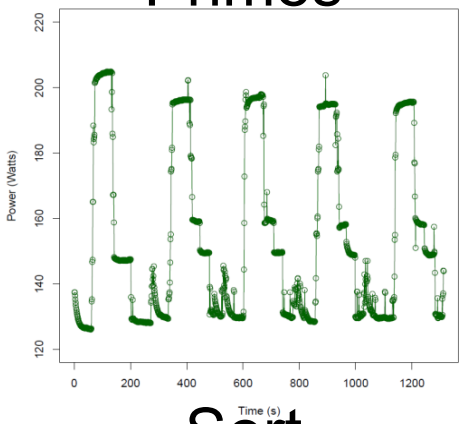
- Star-Cap overview
- Software-only power models
- Power capping schemes
- *Evaluation*

Workloads

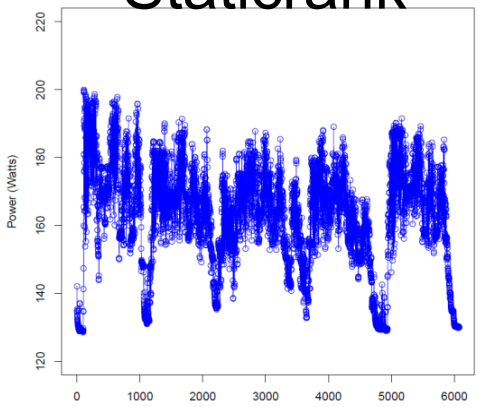
- Primes (CPU)
- Staticrank (Net)
- Sort (Disk, Net)
- Wordcount (Disk)

- All run across 5 homogeneous nodes

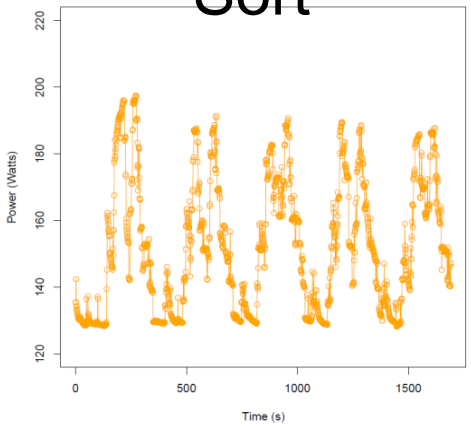
Primes



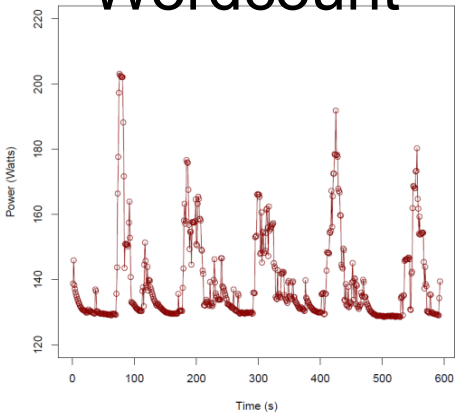
Staticrank



Sort



Wordcount



Hardware Systems

Cluster	Intel Core 2 Duo (laptop)	AMD Opteron (server)
CPU	Intel Core 2 Duo X2 2.26 GHz	AMD Opteron 2X4 2.0 GHz
Storage	SSD	HDD
Idle Power (W)	25	135
Dyn Power range (W)	20	55
OS	Windows Server 2008 R2	

Hardware Systems

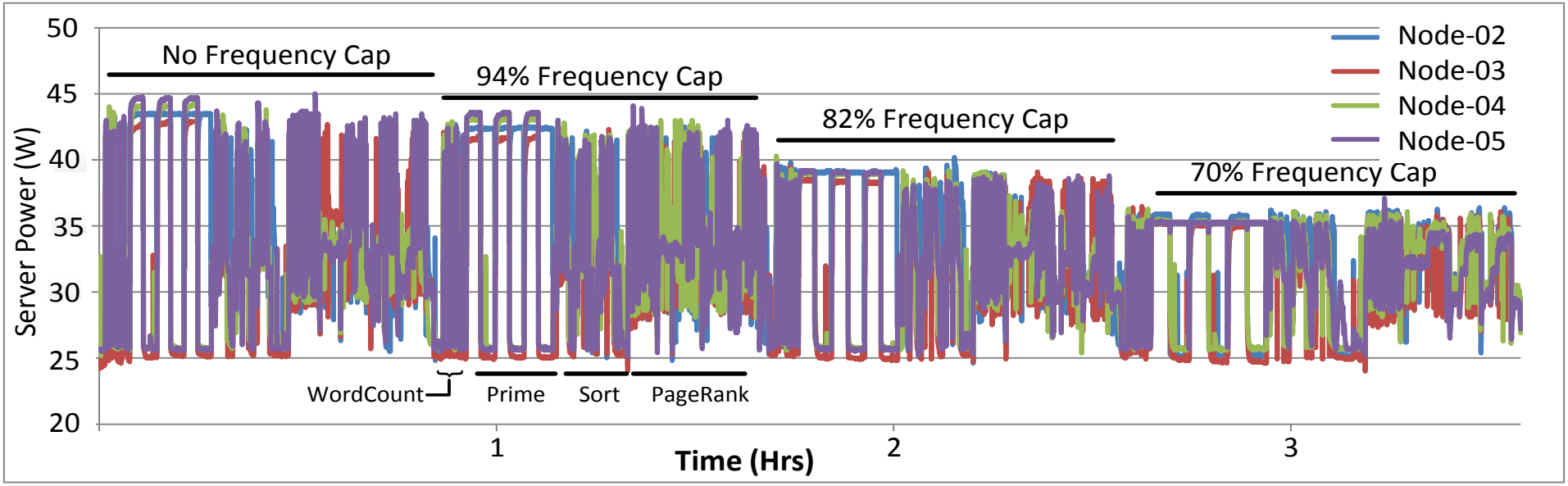
Cluster	Intel Core 2 Duo (laptop)	AMD Opteron (server)
CPU	Intel Core 2 Duo X2 2.26 GHz	AMD Opteron 2X4 2.0 GHz
Storage	SSD	HDD
Idle Power (W)	25	135
Dyn Power range (W)	20	55
OS	Windows Server 2008 R2	

Hardware Systems

Cluster	Intel Core 2 Duo (laptop)	AMD Opteron (server)
CPU	Intel Core 2 Duo X2 2.26 GHz	AMD Opteron 2X4 2.0 GHz
Storage	SSD	HDD
Idle Power (W)	25	135
Dyn Power range (W)	20	55
OS	Windows Server 2008 R2	

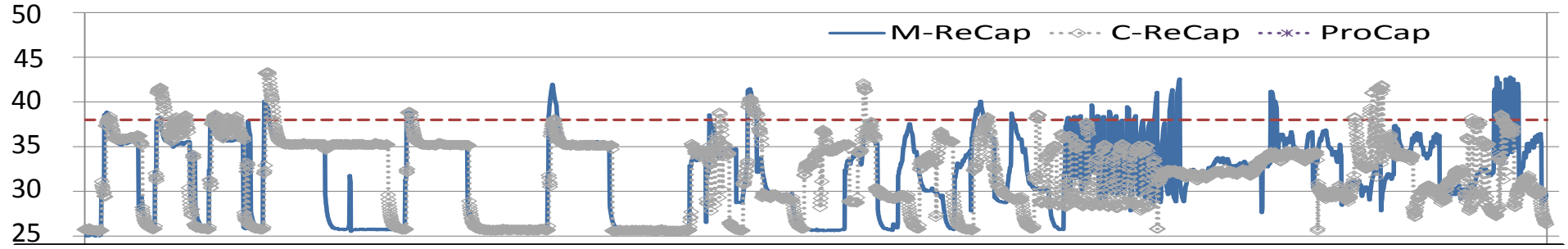
4 frequency states: 100%, 94%, 82%, 70%

Power profiles



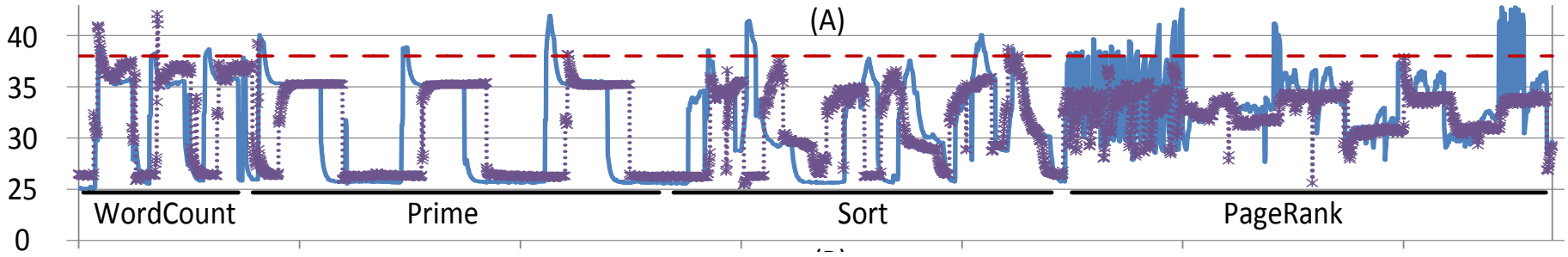
If DVFS is the only actuator, some power budgets will be much easier to deal with than others.

Reactive capping: modeled vs. measured power



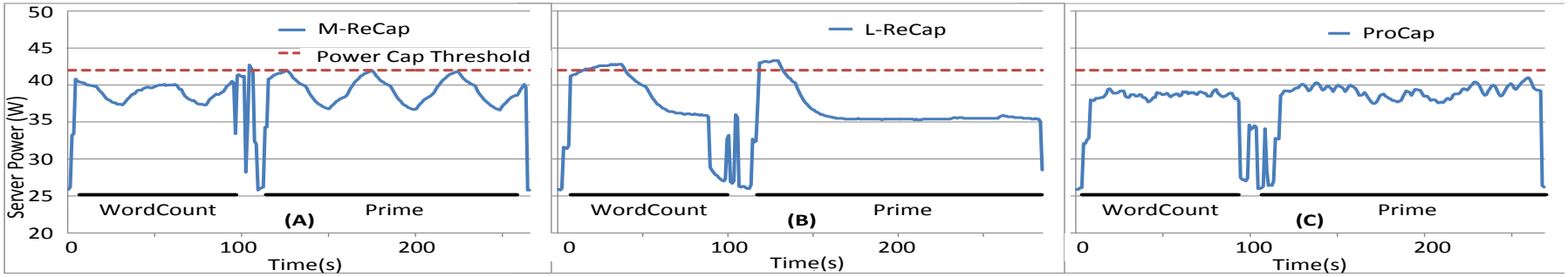
- Low power cap (38 W)
- Graph shows 1 node
- Blue: ReCap based on measured power
- Gray: ReCap based on model power

Reactive vs. proactive capping



- Same power cap
- Blue: ReCap based on measured power
- Purple: ProCap

Higher power cap



- 42W cap
- Left: M-Recap
  Center: L-Recap
  Right: ProCap
- Model accuracy matters!

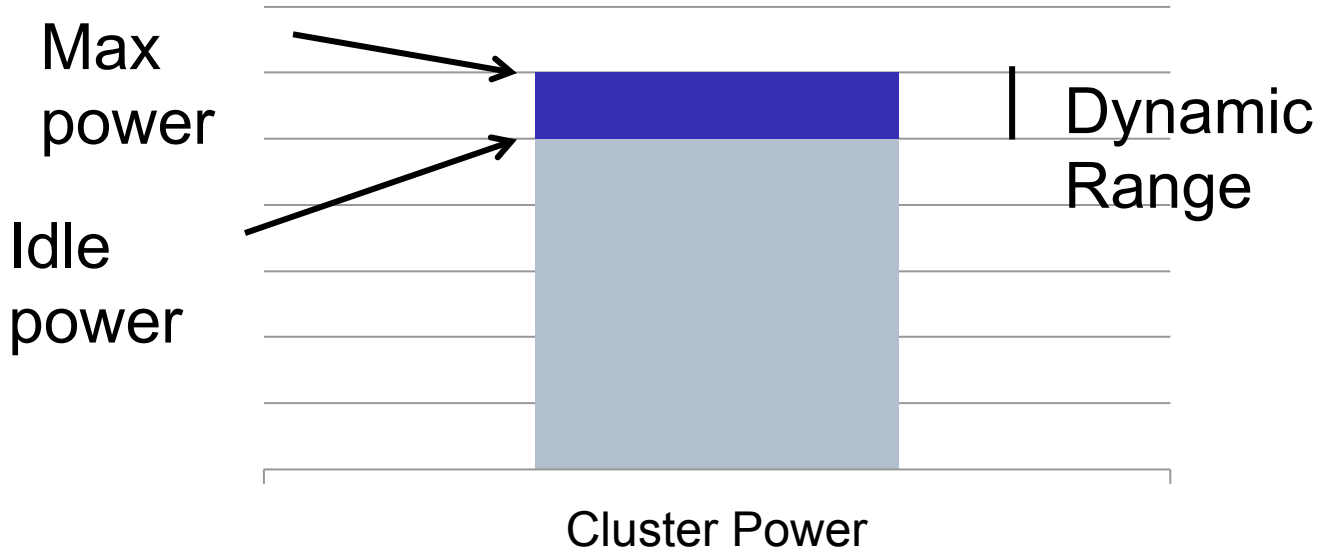
Conclusion

- Demonstrated the potential of high-accuracy, software-only models for server-level power capping
- Suitable for low-power, low-cost “wimpy nodes”
- Extensible to other power management hooks and policies

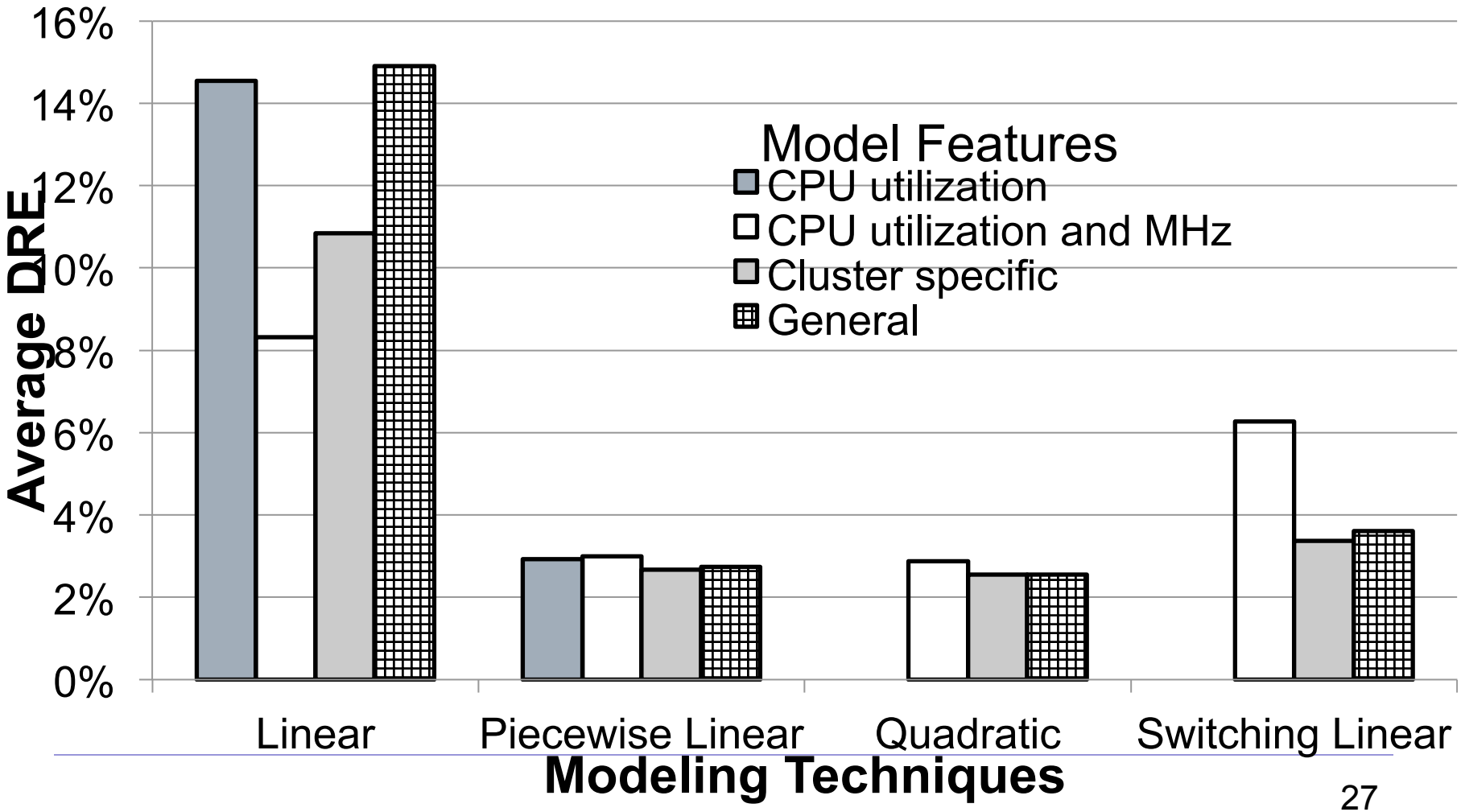
Backup slides

Dynamic Range Error

Report error as a percent of the dynamic range – idle power shouldn't count.



Model Accuracy



Model Features

- Automatically selected from over 200 OS counters
- **Processor:** utilization, frequency
- **Memory:** cache faults/sec; pool nonpaged allocations
- **Disk:** total disk time %
- **Filesystem and virtual memory:** file system pin read/sec, peak page file bytes