# Tradeoffs between Power Management and Tail Latency in Warehouse-Scale Applications

Svilen Kanev, Gu-Yeon Wei, David Brooks
Harvard University

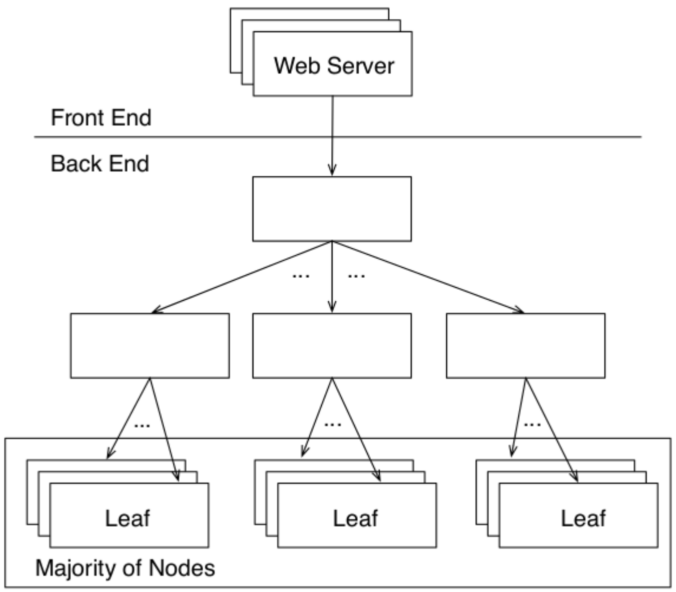
Kim Hazelwood
Google Inc.

# Warehouse-scale computers (WSC)

Datacenters built for a specific class of workloads

Heterogeneous, multi-tiered distributed services, tightly coupled

Overall service must provide latency guarantees
often in the ~100ms




[Meisner et al. 2011]

# WSC performance metric is most often tail latency

| | 50%ile latency | 95%ile latency | 99%ile latency |
|---|---|---|---|
| One random leaf finishes | 1ms | 5ms | 10ms |
| 95% of all leaf requests finish | 12ms | 32ms | 70ms |
| 100% of all leaf requests finish | 40ms | 87ms | 140ms |

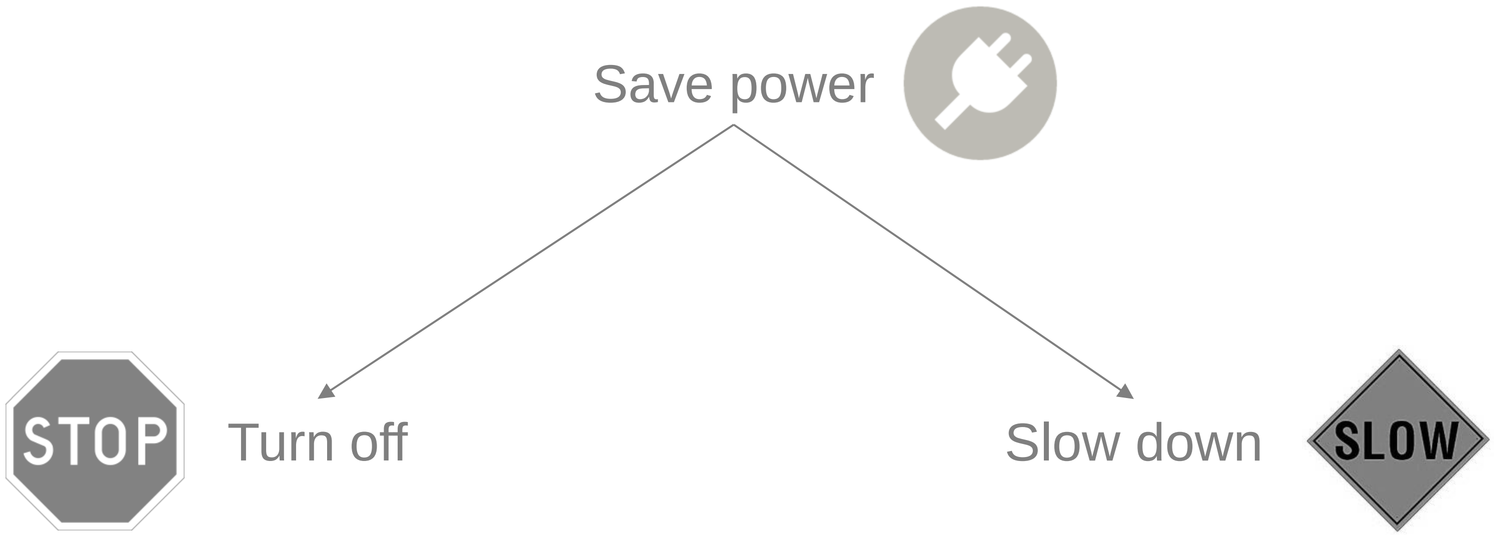
[Dean et al. 2012]

Many services require a response from all leaves

Which could be orders of magnitude slower than average responses

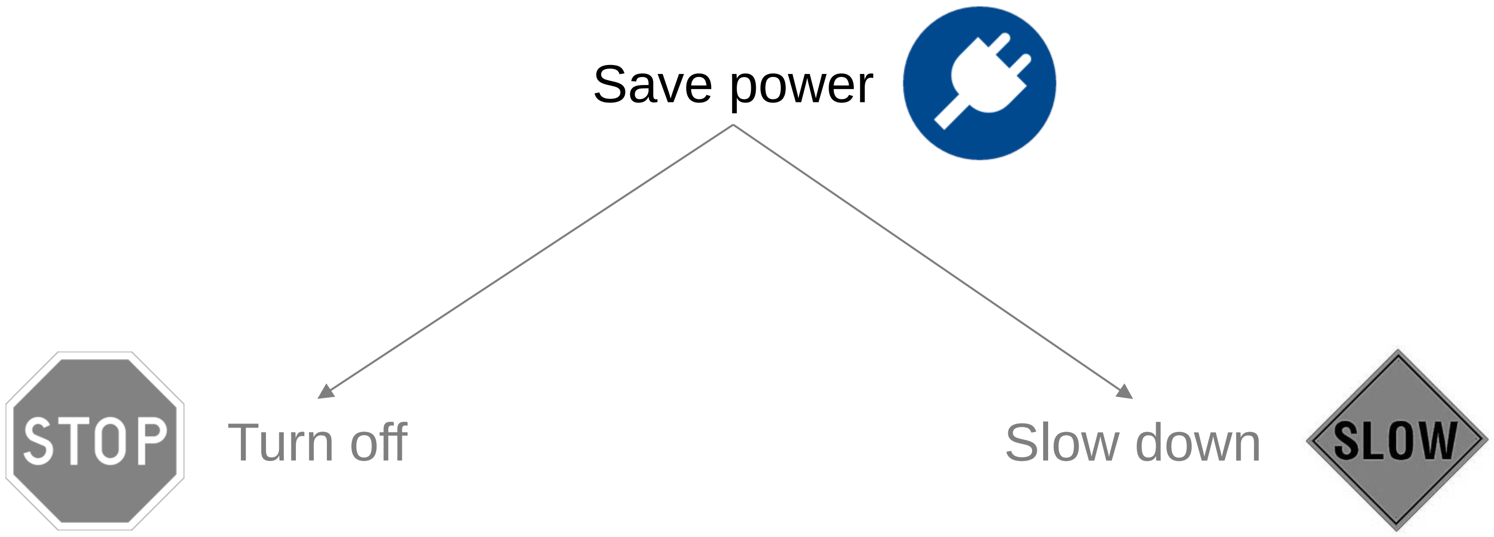
And very sensitive to variability

# Power management leads to performance variability

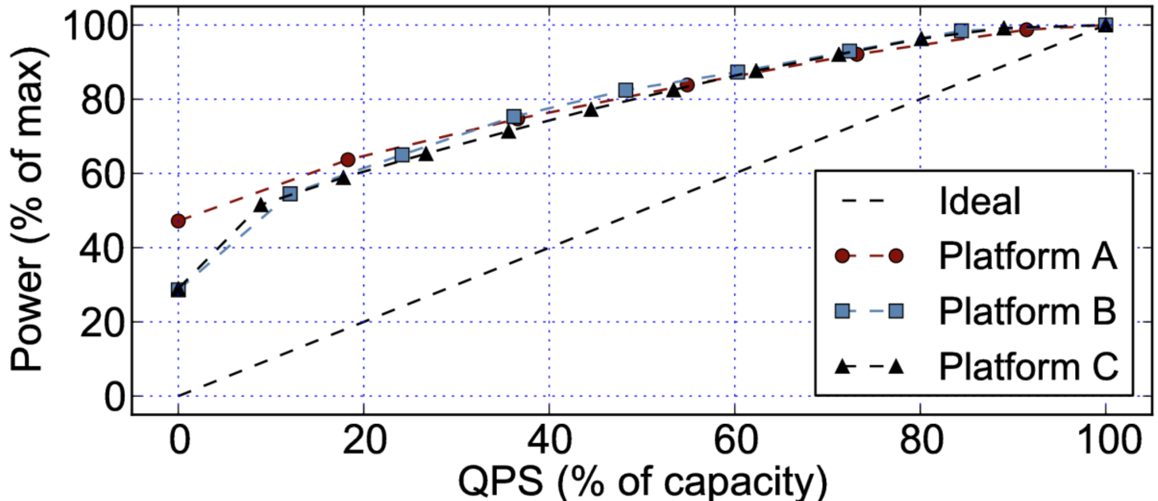



A space between power savings and worse tail latency

# Opportunities for power management



# Energy proportionality

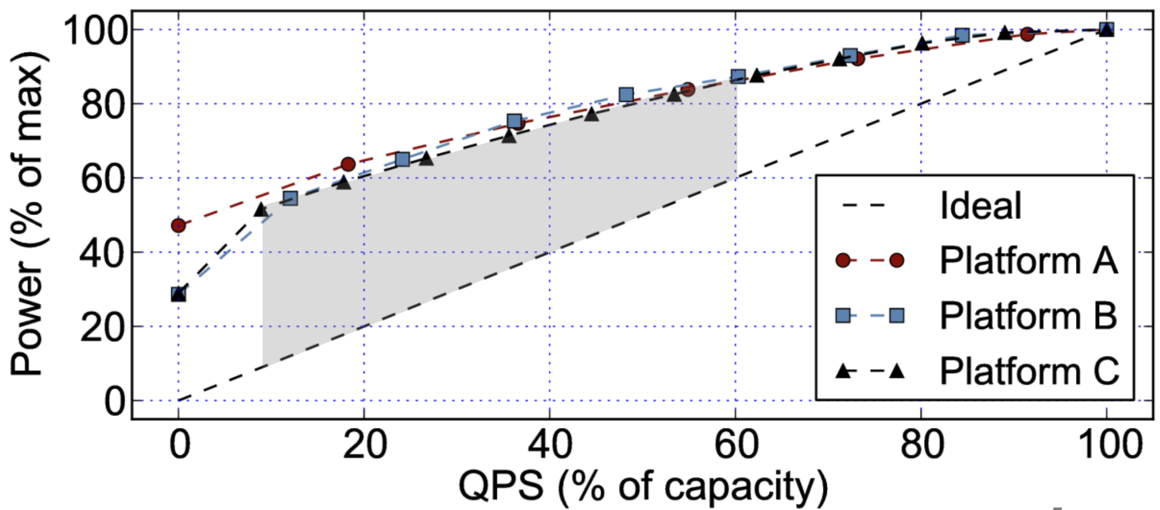



[search leaf node]

Energy proportionality: scale server power with load

Stable across platform generations

# Energy proportionality

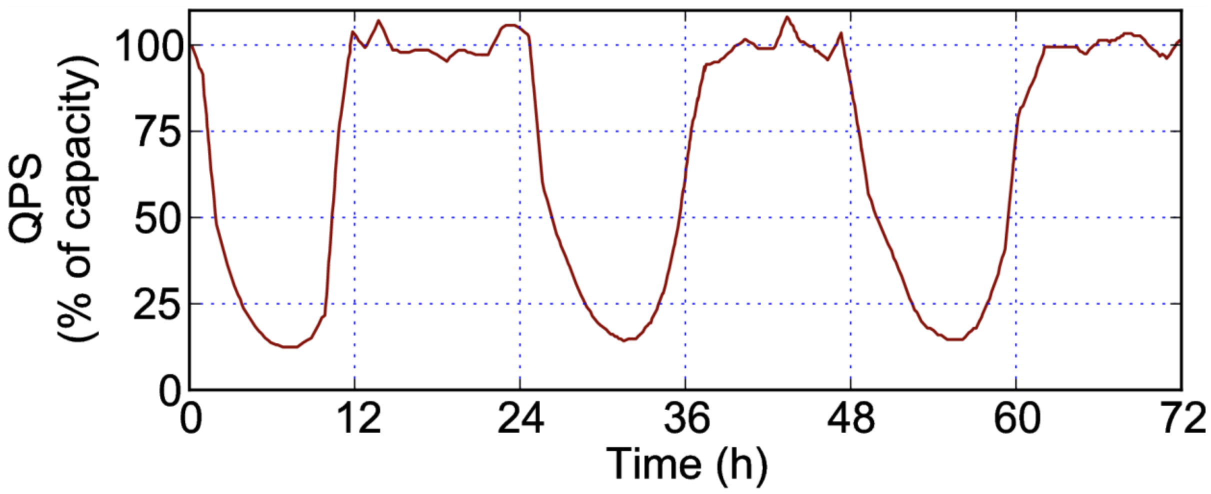



[search leaf node]

Energy proportionality: scale server power with load

Worst in mid-range utilization

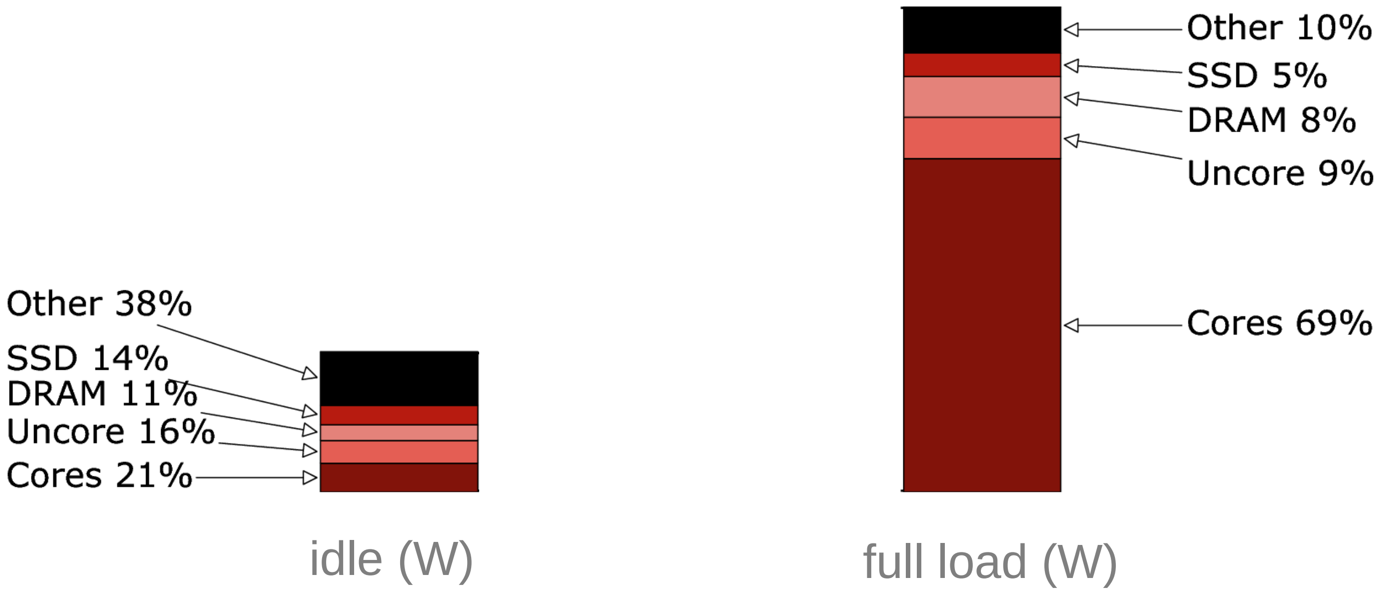
# Servers see the full range of utilization




[Content ads cluster in North America]

15-100% swings during a single day

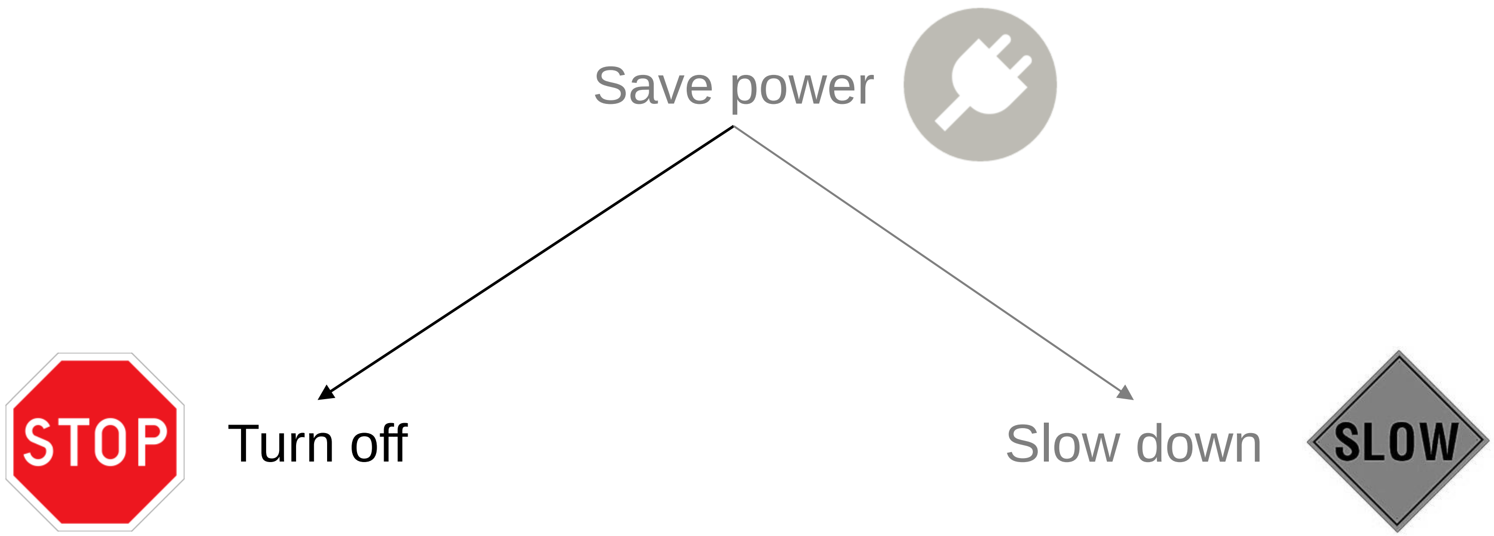
# Server components are differently energy-proportional




Processors are still the major power consumers
but cores also scale best with load

At low-utilization, non-proportional components (disks, flash, DRAM) matter more

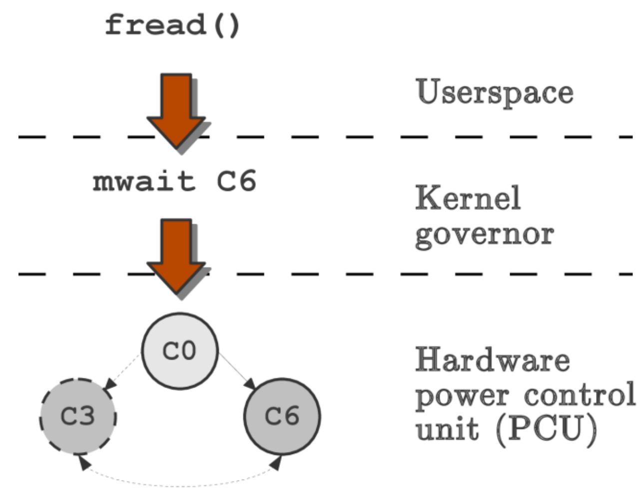
# Idle power management



# Processor idle power management (C-states)

OS-exposed mechanism to exploit idle periods

but still HW-controlled



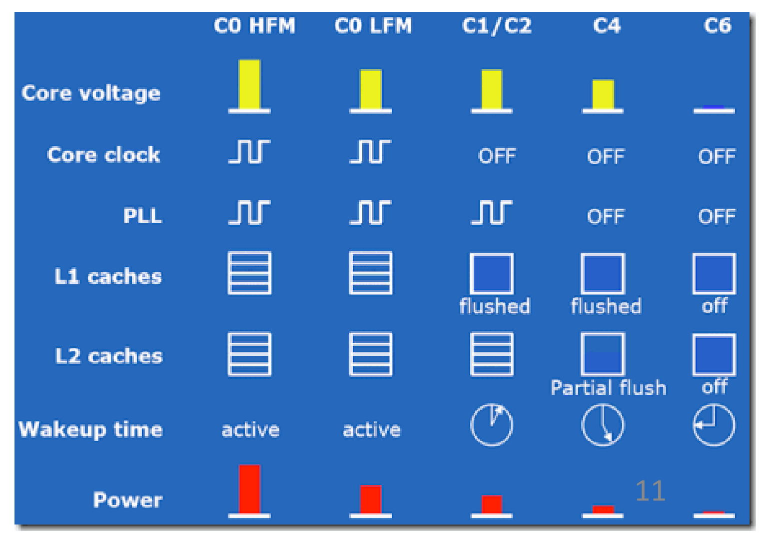

Mostly responsible for current processors' proportionality

Various degrees of power gating

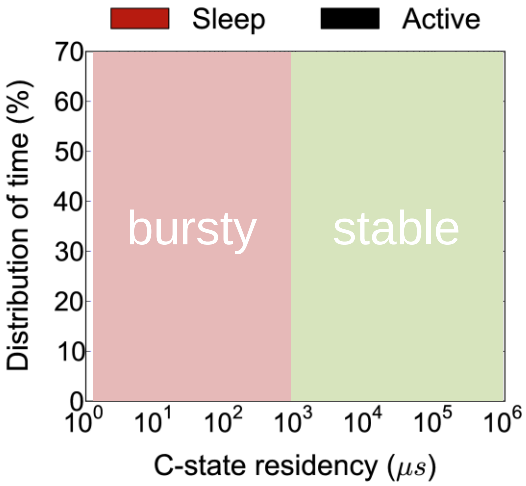
increasing power savings

increasing wakeup latency

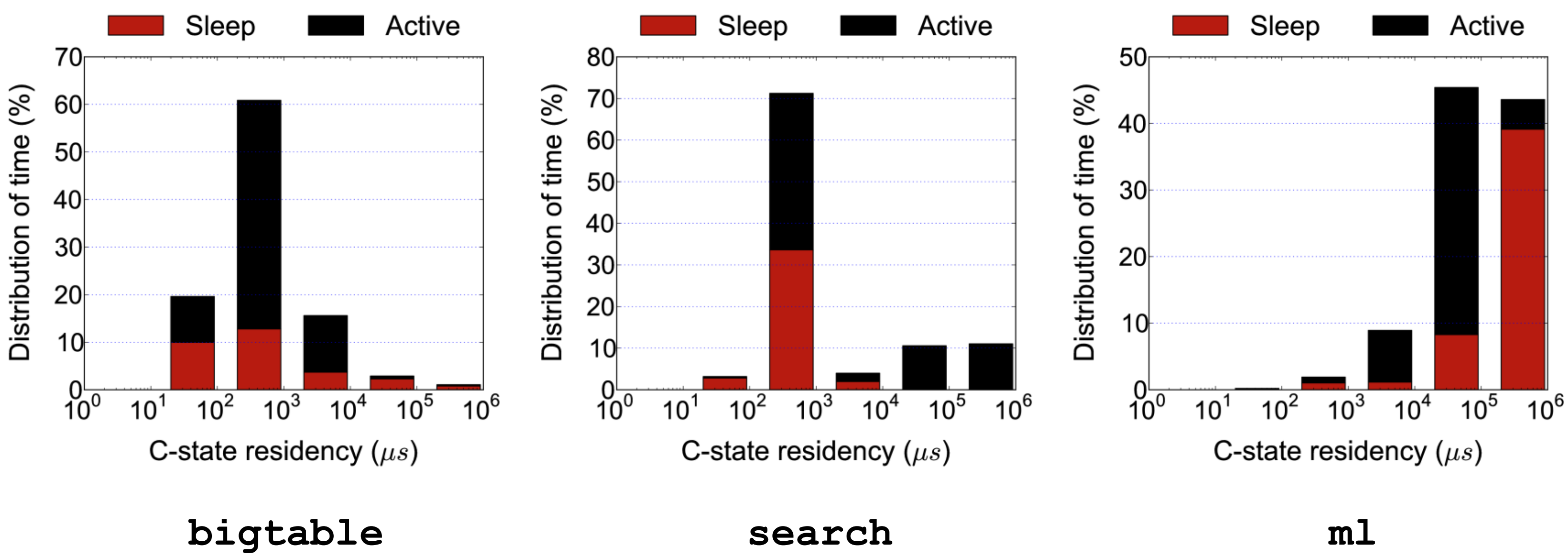
[1-200 μs]



# Some WSC workloads sleep in short bursts

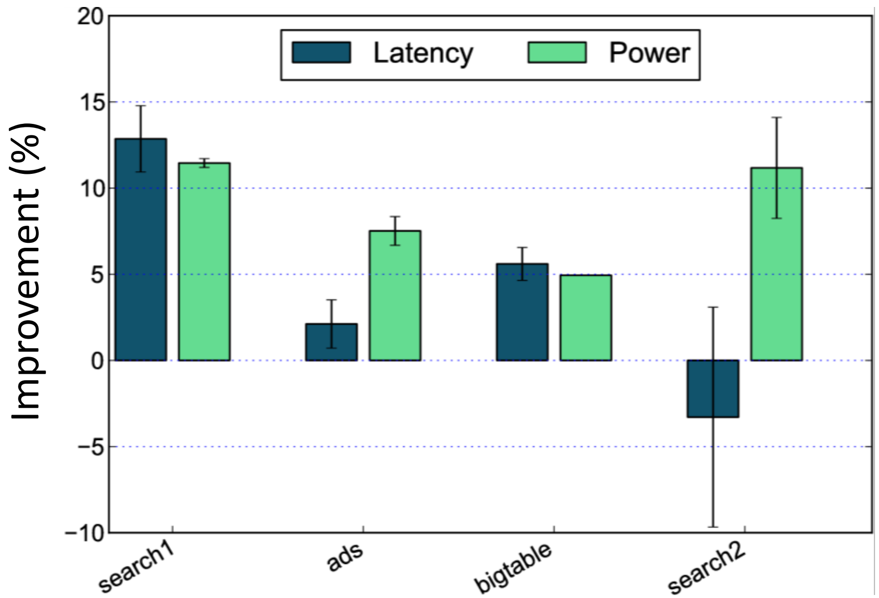


# Some WSC workloads sleep in short bursts




Application sleep activity length can be comparable to wakeup latencies → deep sleep can hurt service latency

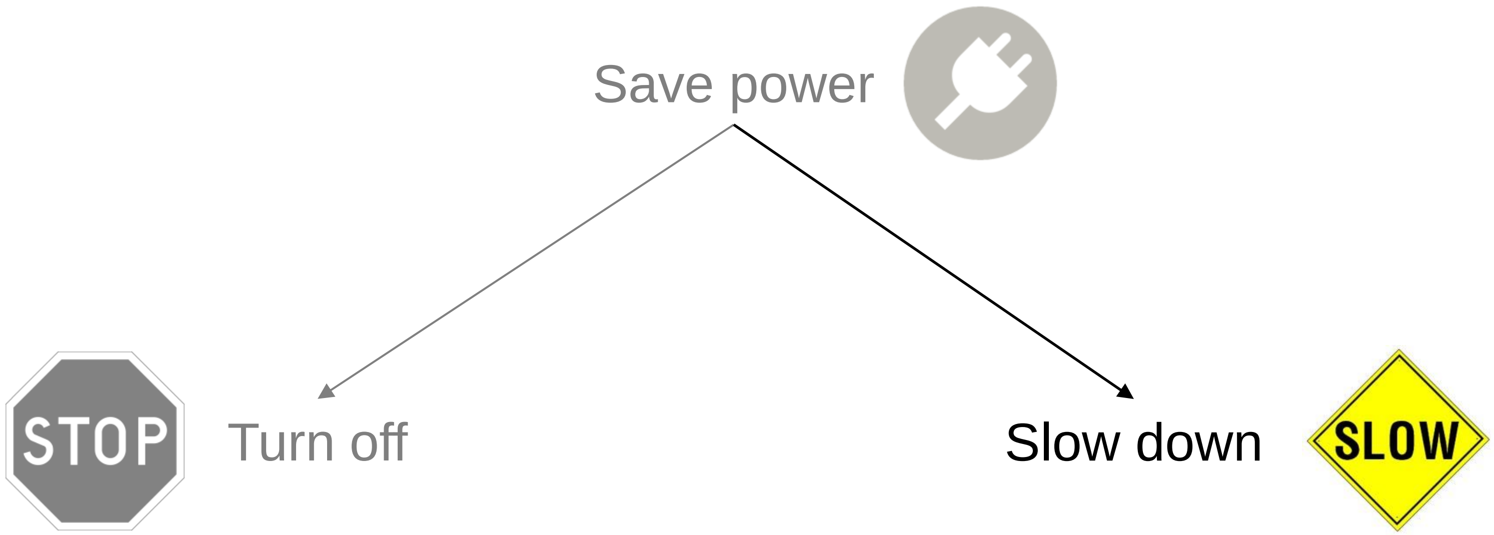
# Effects of sleep state selection



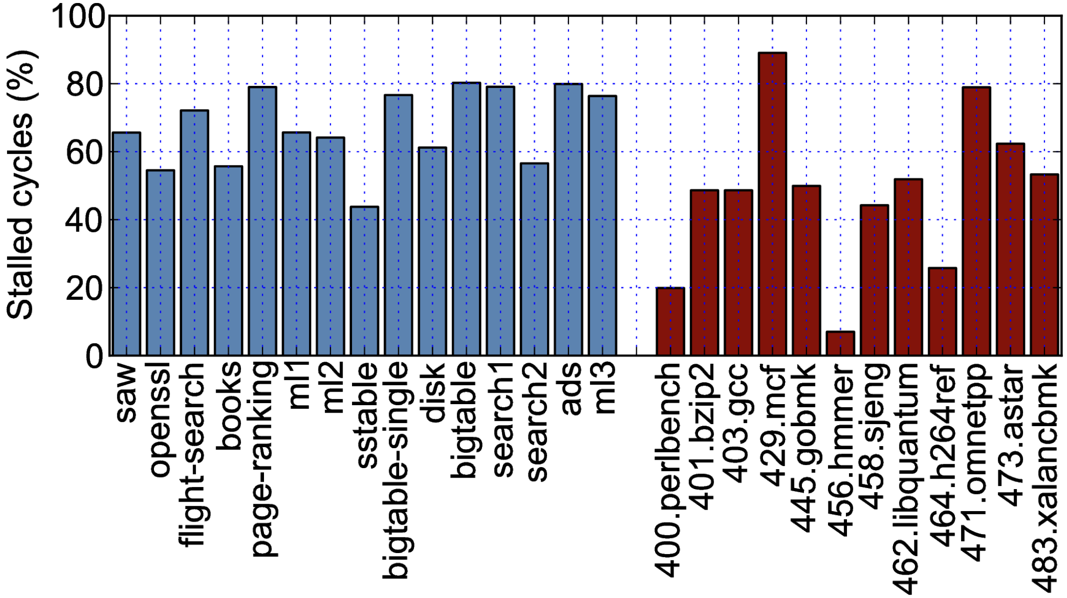

Deep sleep does save significant power (up to 15%)

But also hurts tail latency (up to 15%)

# Active power management



# WSC services are often stalled on memory




A good candidate for voltage and frequency scaling (DVFS)

# Wishlist for server DVFS

Zero tolerance

latency degradation is evil

Workload independence

thousands of relevant workloads

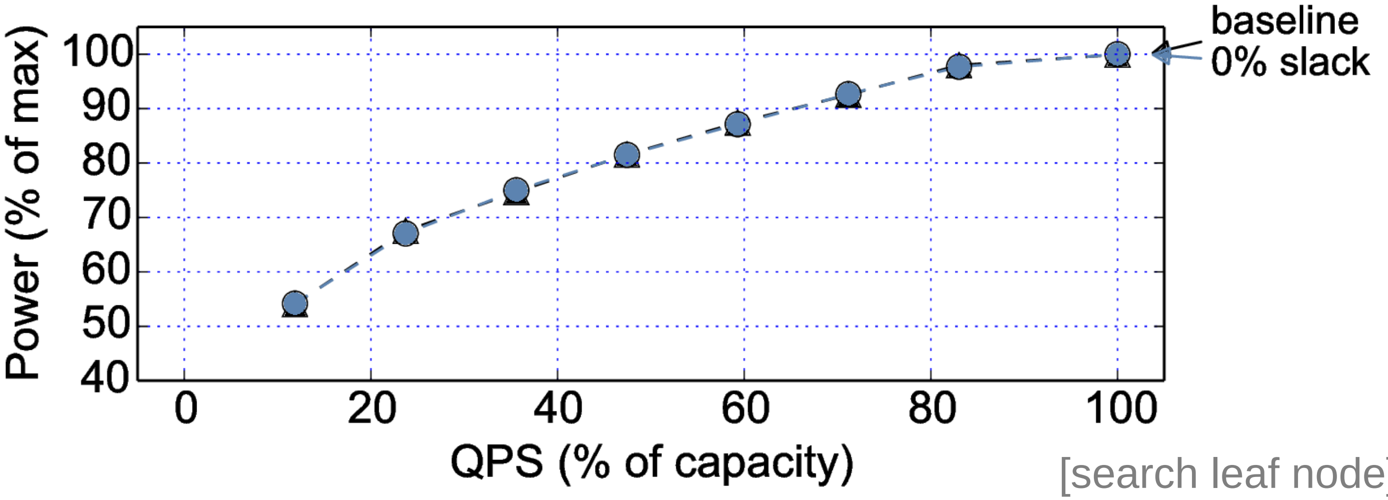
Fine-grained

requests handled in O(1ms)

Per-core

scalable services have independent threads handling independent requests

# Importance of fine granularity

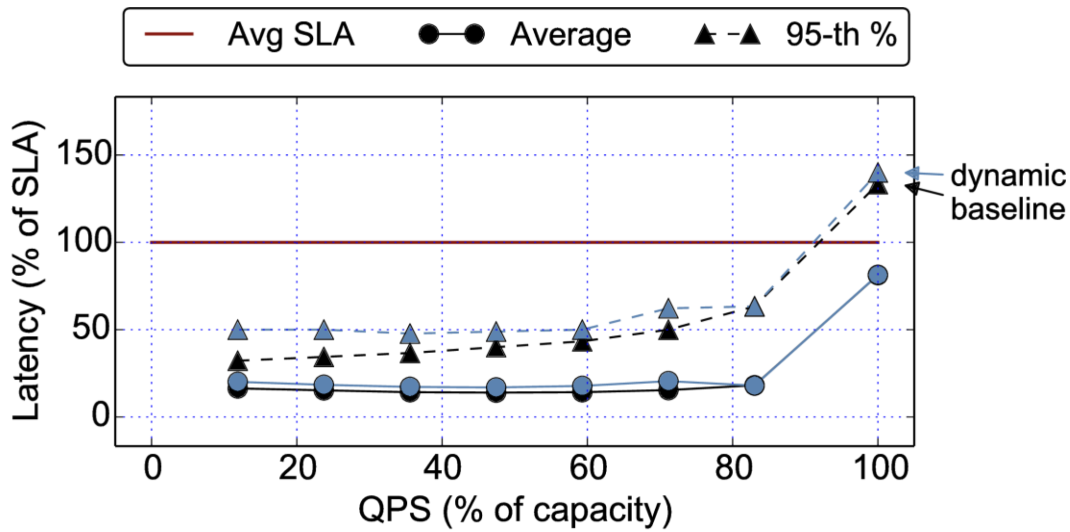


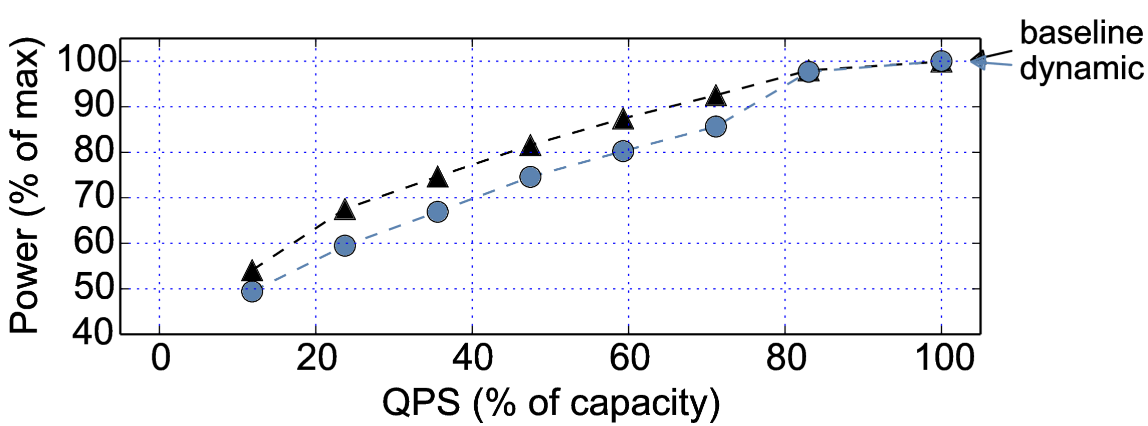

[search leaf node]

No power savings for control as fast as 100 μs

Execution phases are likely more fine-grained
and would be best exploited in hardware

# Importance of workload (in)dependence



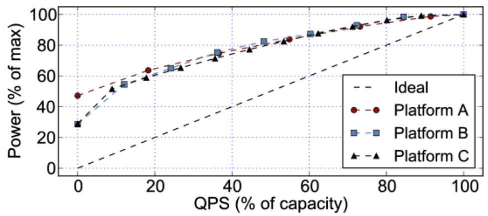




Monitor per-workload latencies, compared to SLA

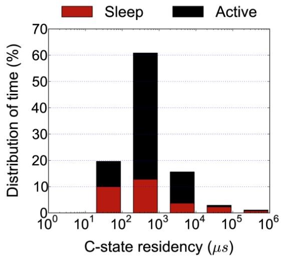
Easily save >10% power without performance implications

Service-specific

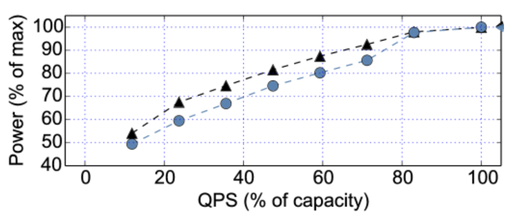
# Takeaways




Current server hardware is not universally energy proportional. Especially related to components like flash, DRAM, or disks.

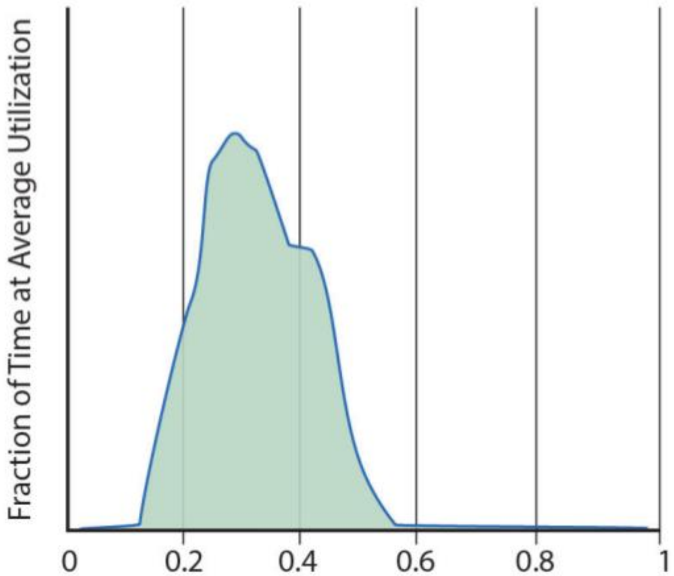



Core sleep states (clock & power gating) are mostly responsible for power savings. But their effects on latency should be treated with care.

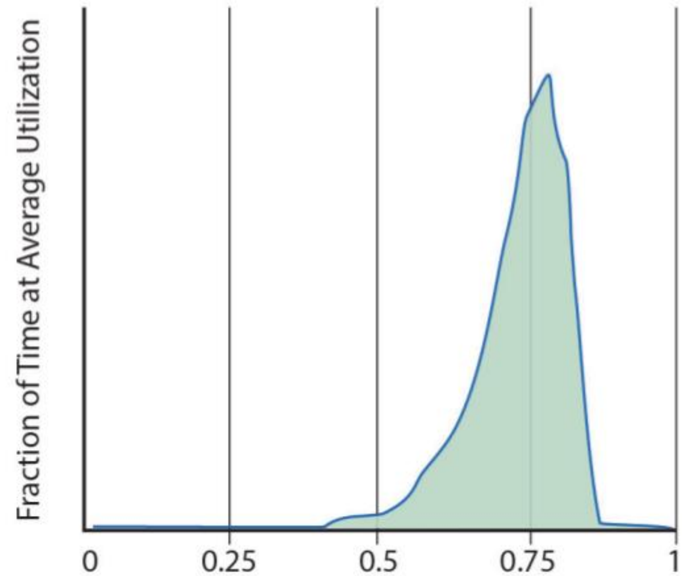



Active power savings are possible either on a very fine granularity, with additional hardware, or on ubiquitous individual workloads, exploiting latency slack.

# Servers are often underutilized




shared cluster

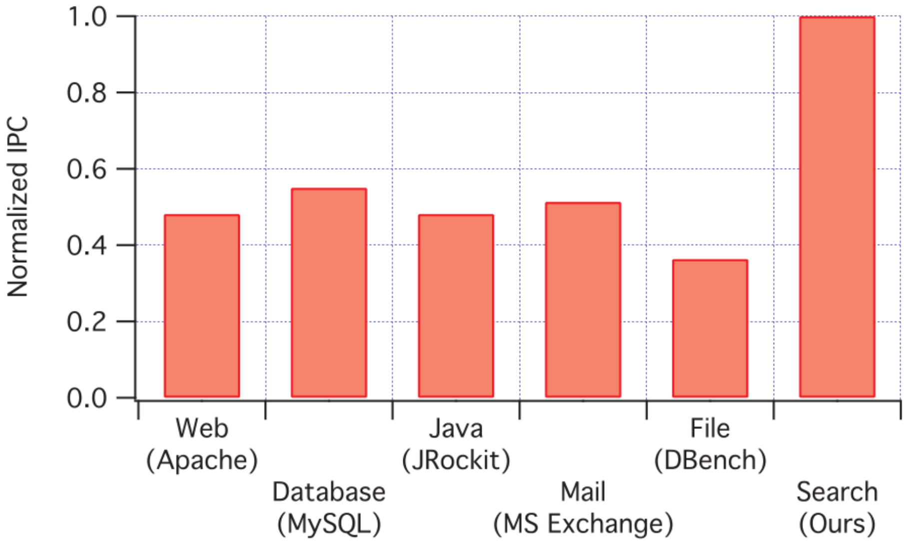



dedicated cluster

Operating in power inefficient regions

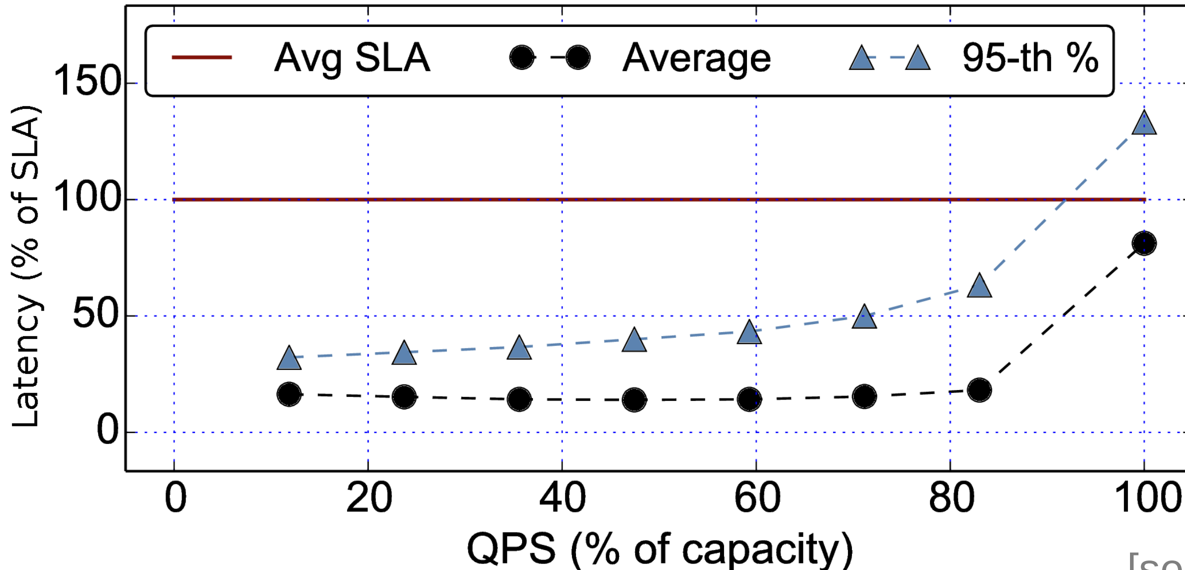
[Barroso et al., 2013, several thousand machines over 3 months]

# … but also require a lot of computation




[Reddi et al. 2010, Bing websearch]

# Some services can be overdesigned



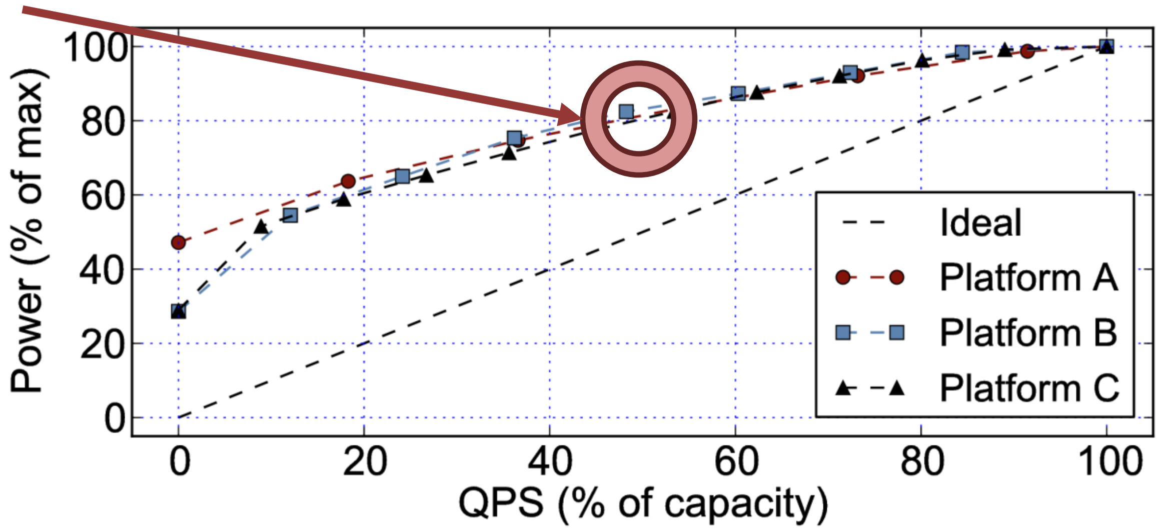

[search leaf node]

Specifically, to handle the peak utilization case

There is no benefit in beating service agreements (SLAs) at low utilization

# Energy proportionality

~80% power
at ~50% load




Energy proportionality: scale server power with load

Relatively stable across platform generations