A Scalable Priority-Aware Approach to Managing Data Center Server Power

Yang Li, Charles R. Lefurgy, Karthick Rajamani, Malcolm S. Allen-Ware, Guillermo J. Silva, Daniel D. Heimsoth, Saugata Ghose, Onur Mutlu

Importance of Data Center Power Infrastructure

- Critical impact on availability: **Numerous** service down due to power outage

Data center power efficiency increases, but so do power outages

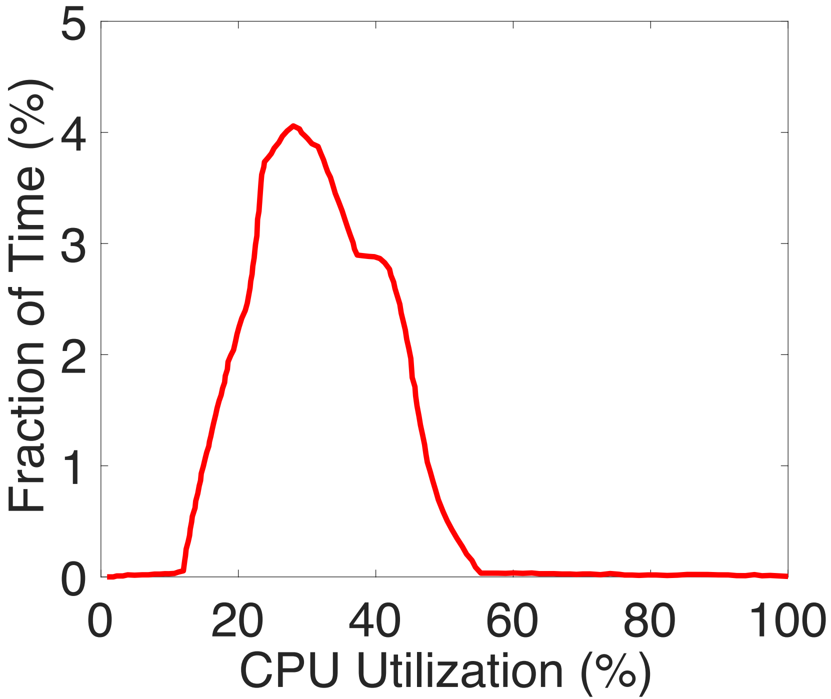
An Uptime Institute survey finds the power usage effectiveness of data centers is better than ever. However, power outages have increased significantly.

By Andy Patrizio, Network World, 2018

- Significant capital cost: **Tens of millions of US dollars**

Underutilized Data Center Power Infrastructure

- Considering ***peak*** demand, not ***typical*** demand for capacity sizing
- Employing redundant power infrastructures
- **Power capacity is sized for 2X peak demand!**



Reproduced from Barroso et al., The Data Center as a Computer, 2013.

Boosting Data Center Performance

- Why not add more servers to boost data center performance?

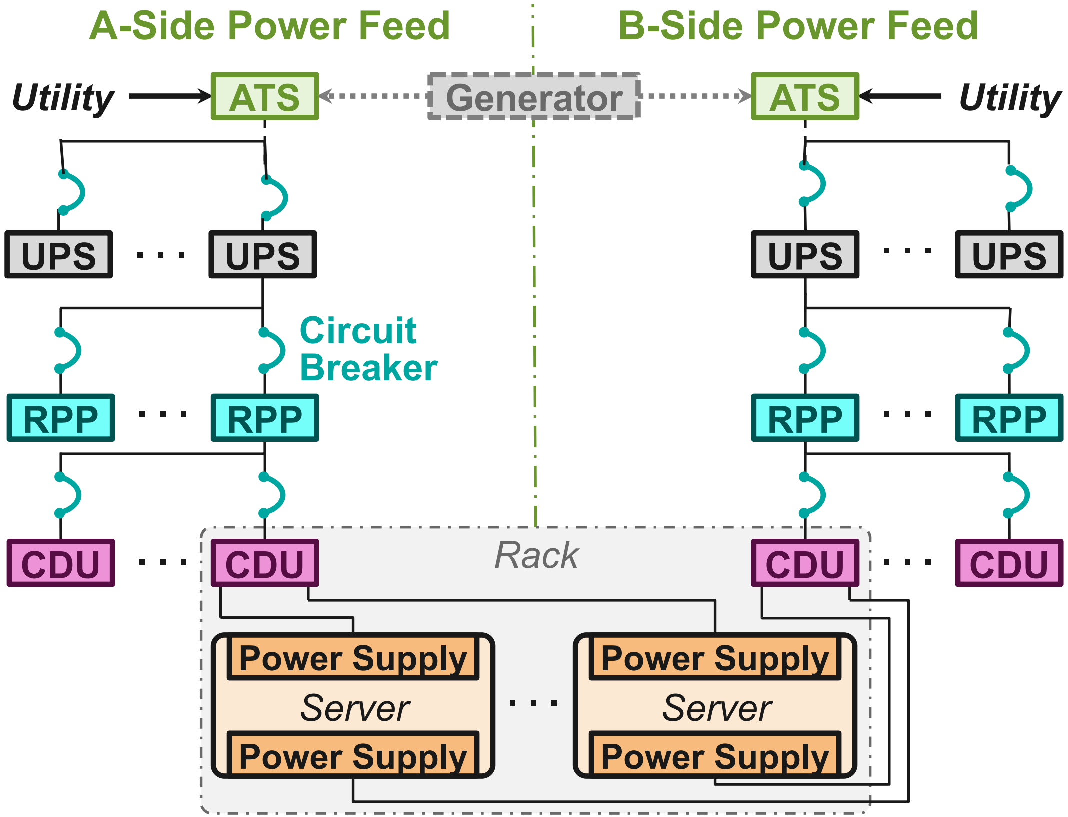
- During normal operation, let these servers utilize the spare capacity
- During the ***rare*** worst cases, let a power management system kick in
  - Throttle the servers with less important workloads (within seconds)
  - Protect the circuit breakers from tripping
- **Problem:**

  Build a data center power management system using real-time control

Outline

- Problem
- **Background**
- Current Practice & Design Challenges
- Key Solutions
- Implementations
- Evaluations
- Open Challenges
- Conclusions

Layout for Data Center Power Infrastructure



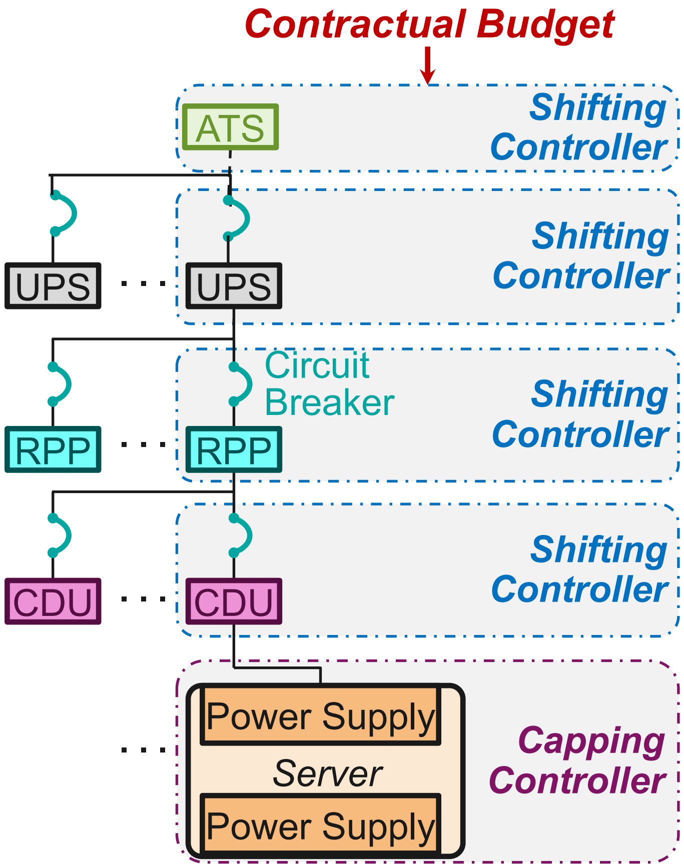
We need to protect:

- DC Contractual budget
- Circuit breakers of UPS, RPP, CDU

Outline

- Problem
- Background
- **Current Practice & Design Challenges**
- Key Solutions
- Implementations
- Evaluations
- Open Challenges
- Conclusions

Hierarchical Control Framework



- Design Challenge #1:

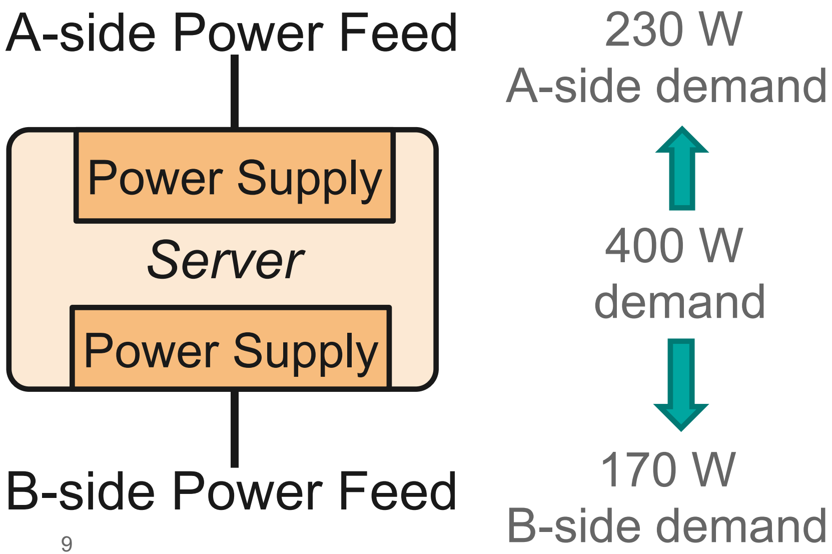
  Consider the redundant connections

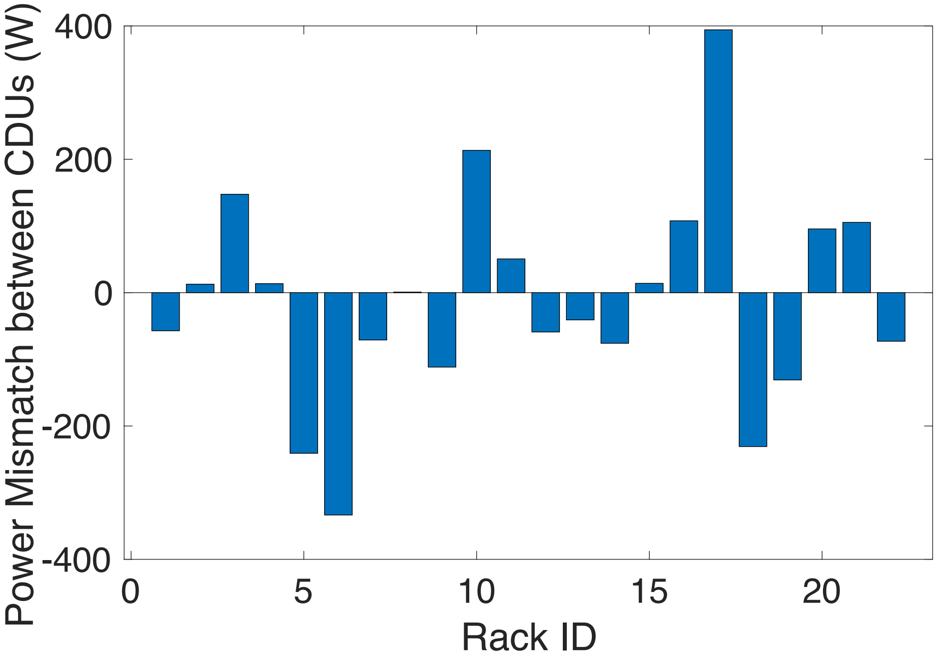
- Design Challenge #2:

  Enforce priority-aware power capping ***globally***

Design Challenges #1: Consider Redundant Connections

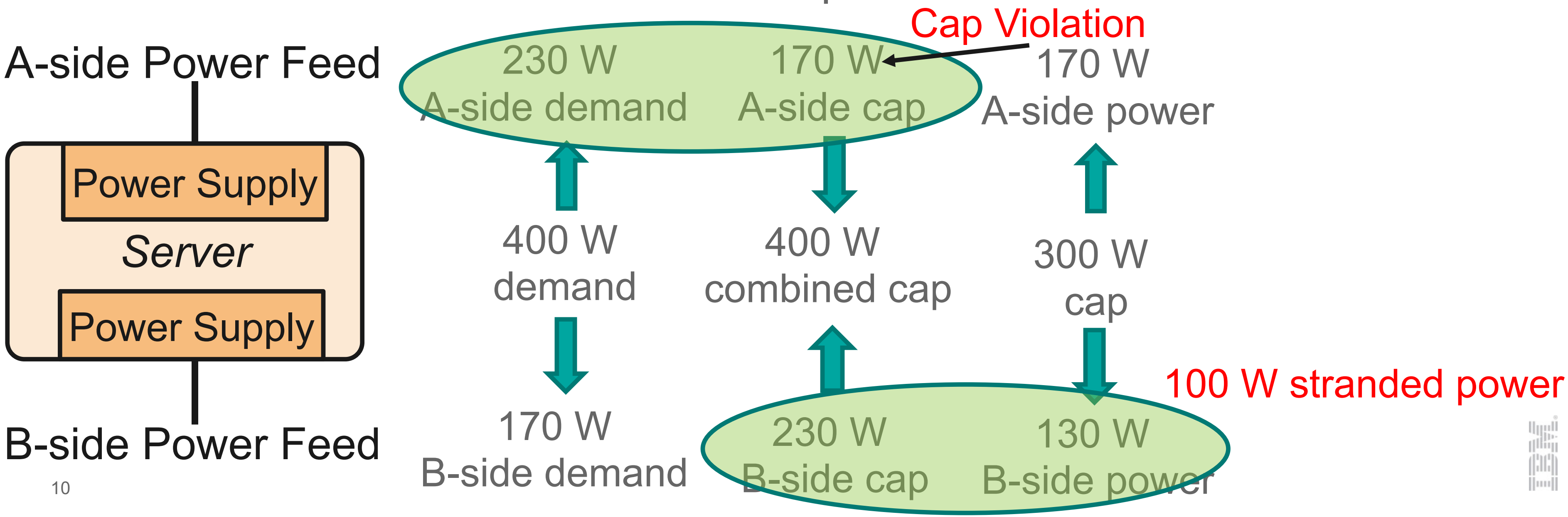
- Servers do not split power ***equally***
- Need to enforce different power caps for different supplies





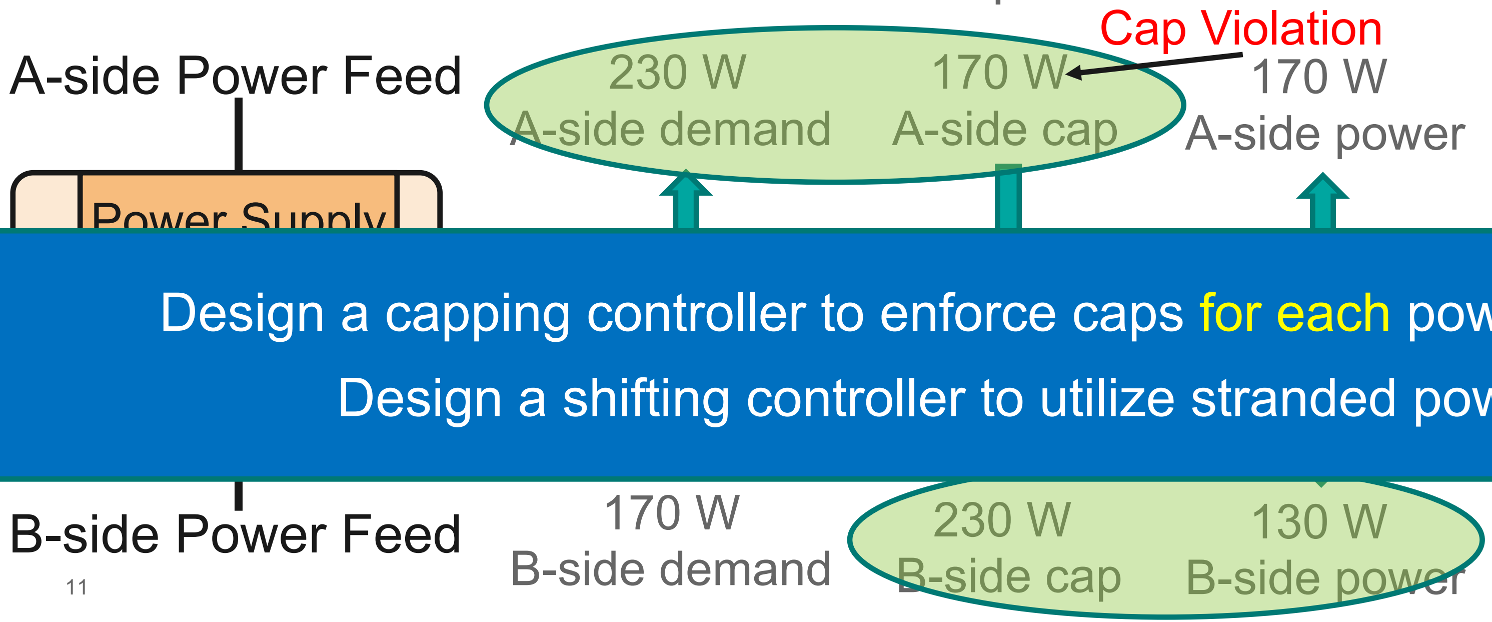
Design Challenges #1: Consider Redundant Connections

- Servers do not split power ***equally***
- Need to enforce different power caps for different supplies
- Prior works can only enforce a single ***combined*** power cap
- It is desirable to utilize the ***stranded*** power



Design Challenges #1: Consider Redundant Connections

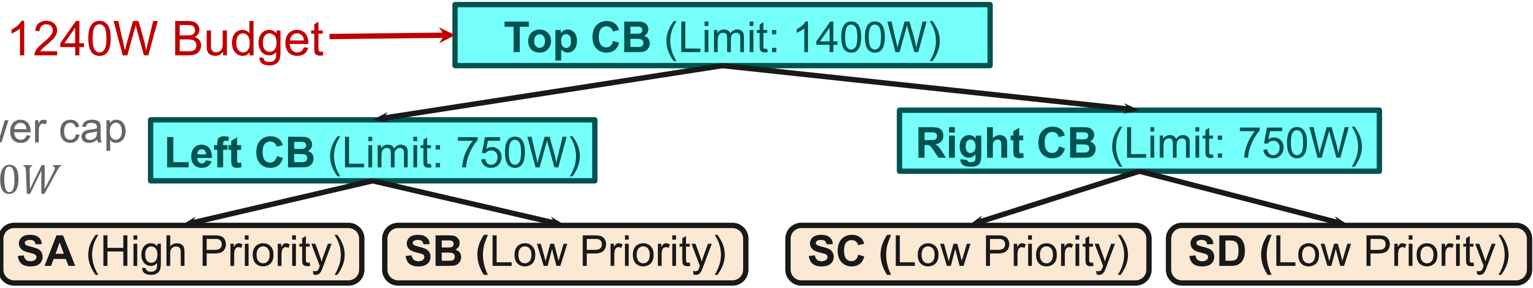
- Servers do not split power ***equally***
- Need to enforce different power caps for different supplies
- Prior works can only enforce a single ***combined*** power cap
- It is desirable to utilize the ***stranded*** power



Design a capping controller to enforce caps for each power supply;
Design a shifting controller to utilize stranded power

Design Challenges #2: Enforce Priority Globally

- Global Priority-Aware: Always cap low-priority servers before high-priority servers **across the entire data center**
- Prior works can only enforce priority ***locally*** within rack

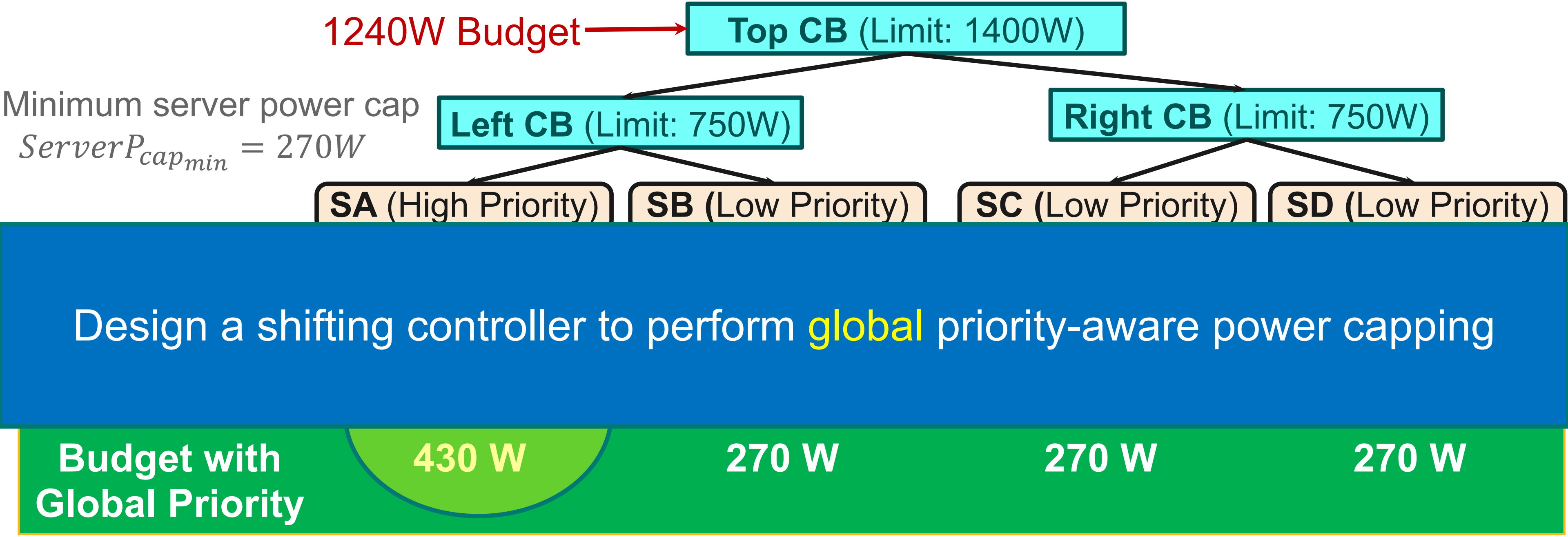


Minimum server power cap
$ServerP_{cap_{min}} = 270W$

Demand	**430 W**	**430 W**	**430 W**	**430 W**
Budget with Local Priority	**350 W**	**270 W**	**310 W**	**310 W**
Budget with Global Priority	**430 W**	**270 W**	**270 W**	**270 W**

Design Challenges #2: Enforce Priority Globally

- Global Priority-Aware: Always cap low-priority servers before high-priority servers **across the entire data center**
- Prior works can only enforce priority ***locally*** within rack

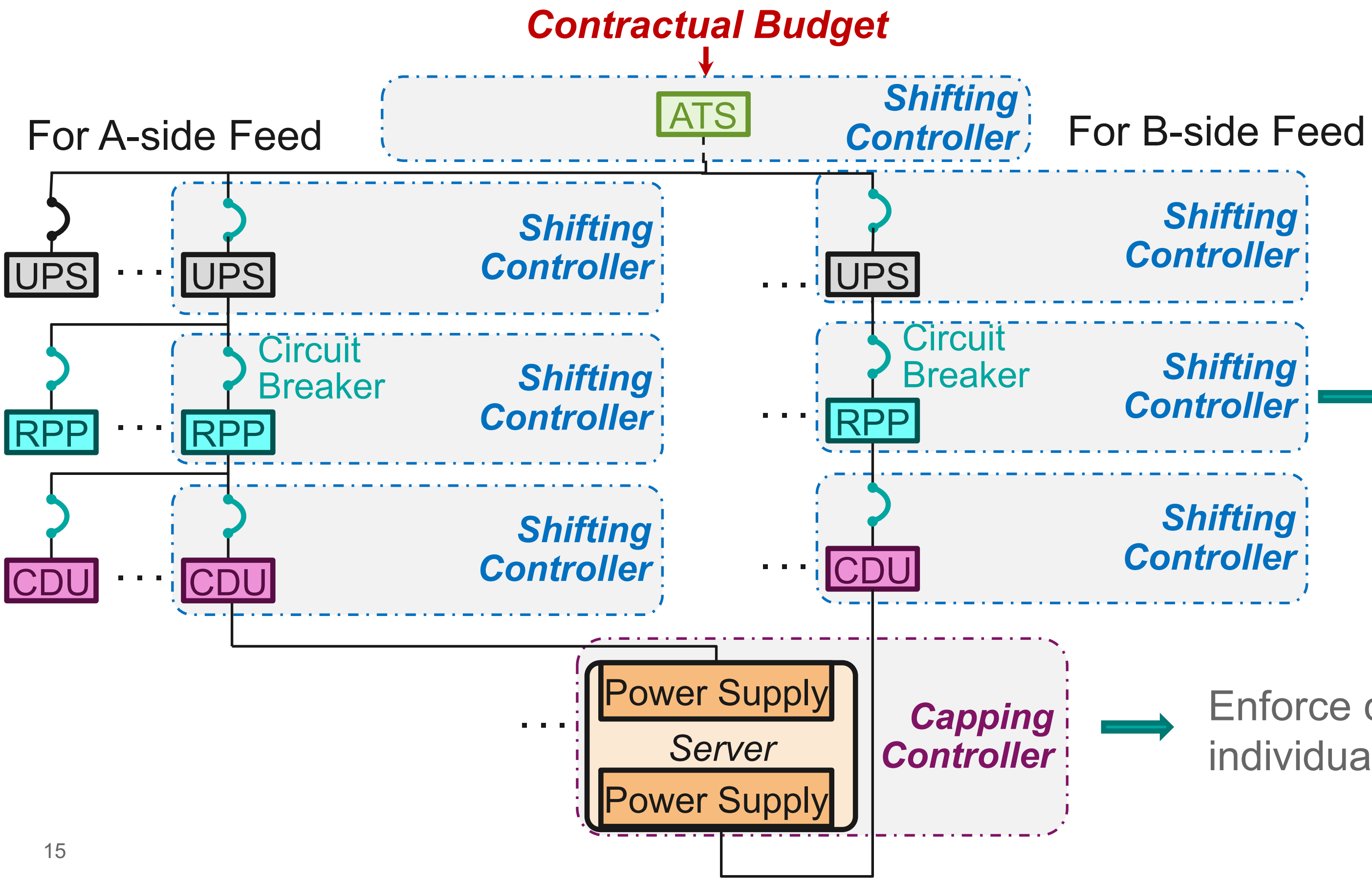


Design a shifting controller to perform global priority-aware power capping

Outline

- Problem
- Background
- Current Practice & Design Challenges
- **Key Solutions**
- Implementations
- Evaluations
- Open Challenges
- Conclusions

Overview of CapMaestro

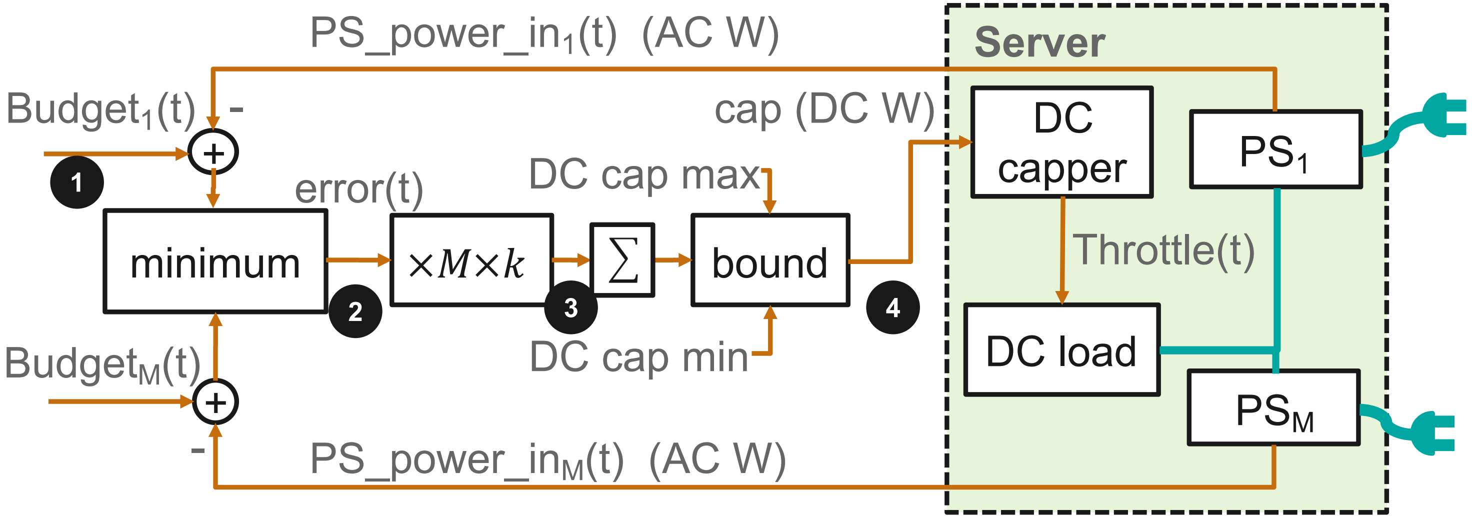


Enforce global priority aware power capping

Enforce cap for individual supplies

Capping Controller for Servers with Multiple Power Supplies

- Goal: Ensure all individual power supply caps are respected, while allowing as much power consumption as possible
- Key Idea:
  - Monitor the minimum error between power caps and power consumption
  - Employ a Proportional-Integral (PI) controller



Global Priority-Aware Power Capping

- Key Idea: Power Metric Summary
  – Summarize metrics for servers by priority under each shifting controller
  – Use metrics to budget from high priority to low priority

- Power metrics (at each shifting controller):

  For each priority j,

  – $P_{cap\_min}(j)$: The minimum total power budget
  – $P_{demand}(j)$: The total power demand → May not be fulfillable
  – $P_{request}(j)$: The requested power budget → Fulfillable power with respect to other priorities

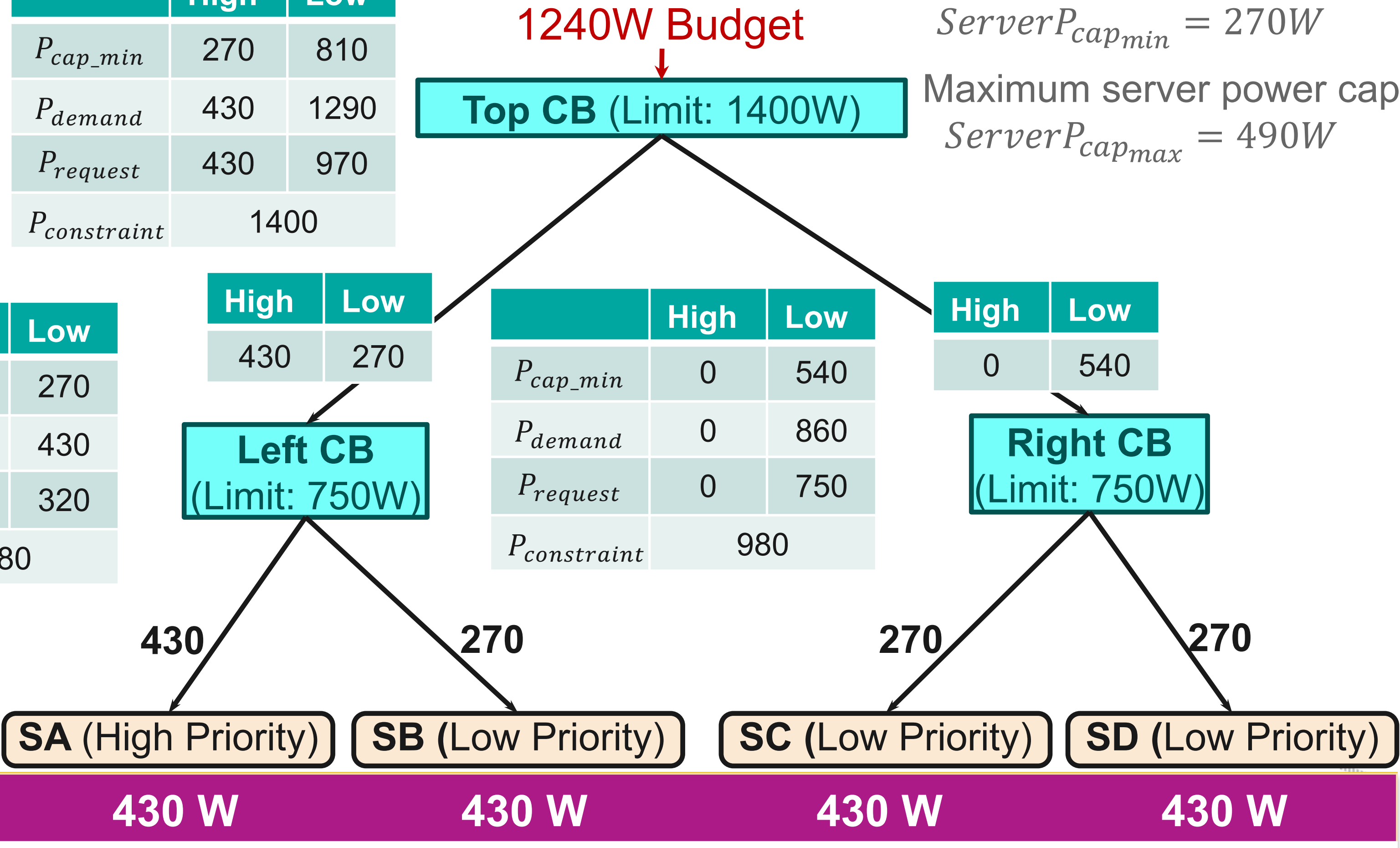
  $P_{constraint}$: The maximum power budget for all the servers (regardless of priority)

Global Priority-Aware Power Capping

Minimum server power cap $ServerP_{cap_{min}} = 270W$

Maximum server power cap $ServerP_{cap_{max}} = 490W$

	High	Low
P_{cap_min}	270	810
P_{demand}	430	1290
$P_{request}$	430	970
$P_{constraint}$	1400	



High	Low
430	270

High	Low
0	540

	High	Low
P_{cap_min}	270	270
P_{demand}	430	430
$P_{request}$	430	320
$P_{constraint}$	980	

	High	Low
P_{cap_min}	0	540
P_{demand}	0	860
$P_{request}$	0	750
$P_{constraint}$	980	

Global Priority-Aware Power Capping

- Rigorous theoretical proof:
  - Servers with high priorities are always throttled after servers with lower priorities, as long as the circuit breaker limits allow
  - See our IBM technical report

- Good scalability:
  - Linear algorithm complexity at each controller
  - Fixed ratio of overhead to # servers

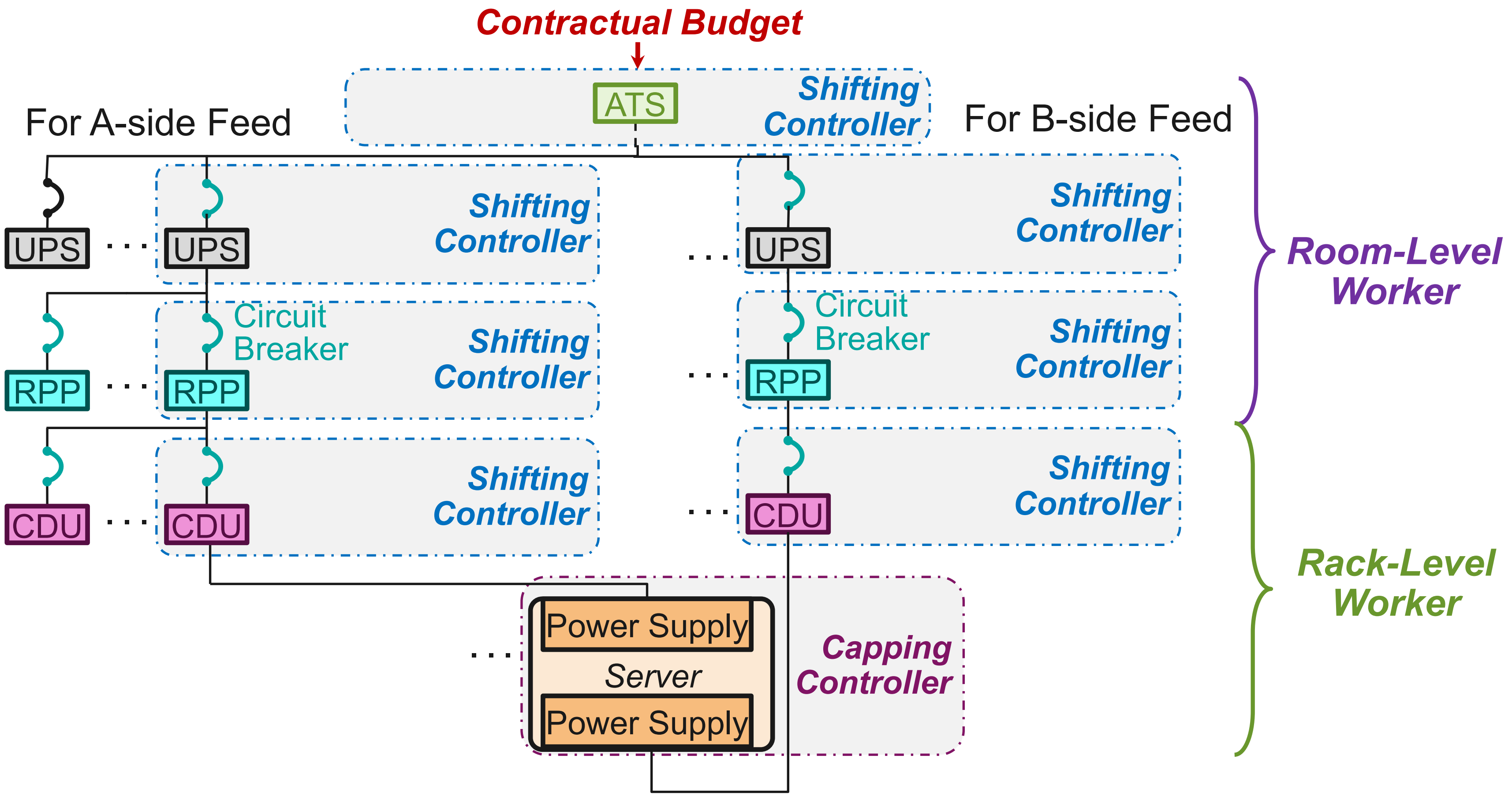
Outline

- Problem
- Background
- Current Practice & Design Challenges
- Key Solutions
- **Implementations**
- Evaluations
- Open Challenges
- Conclusions

Implementations

- Implemented as a first-of-a-kind cloud service
- Controllers are packaged into containers

Implementations



Implementations

- Implemented as a first-of-a-kind cloud service
- Controllers are packaged into Worker containers
- Power controller measures and controls power at 8 second intervals
- Employ Intel® Node Manager as the underlining server throttling engine
  - Cap power within 1 second
  - Measure power supplies every 1 second
- Our power control framework supports dynamic server priorities

Outline

- Problem
- Background
- Current Practice & Design Challenges
- Key Solutions
- Implementations
- **Evaluations**
- Open Challenges
- Conclusions

Evaluations

- Performed several **real-system experiments** demonstrating that:
  - Our capping controllers successfully enforce power caps for individual supplies
  - Our shifting controllers ensure servers with high priority are throttled after servers with lower priorities
  - Our shifting controllers reallocate stranded power to the underpowered servers

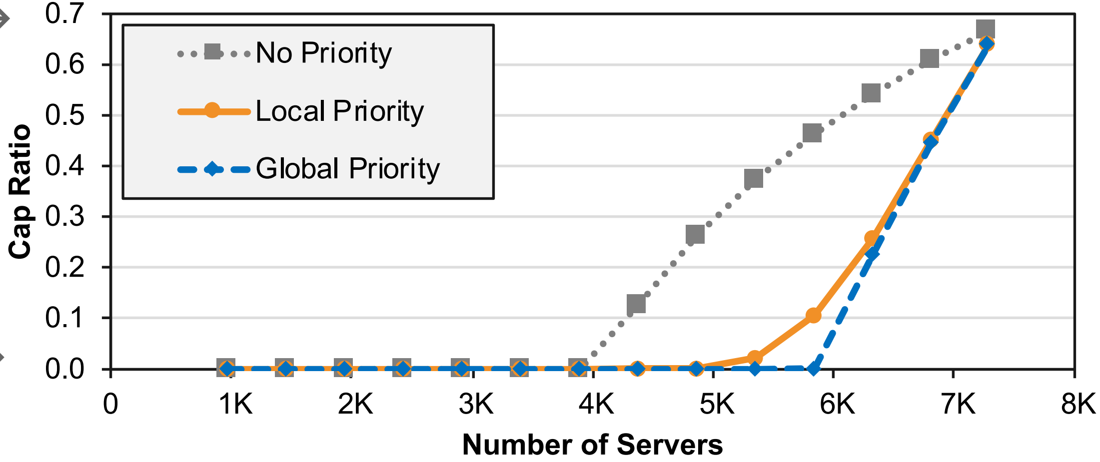
- Performed a large-scale data center simulation demonstrating that:
  - Compared to the case of no power management system, our system enables the data center to deploy 50% more servers

Key Results

- Simulate a data center of 162 racks; 30% of servers are assigned high priority
- Cap Ratio for high-priority servers under a power emergency:

Throttled performance →

Full performance →



- Using 1% cap ratio as threshold, our system supports 5832 servers, while Local Priority only supports 4860 servers

Outline

- Problem
- Background
- Current Practice & Design Challenges
- Key Solutions
- Implementations
- Evaluations
- **Open Challenges**
- Conclusions

Open Challenges

- Broadening the target of power management:
  - Comprehensive power capping scheme for more system components, such as GPU, FPGA, storage, and networking
- Coordination of job scheduling with power management:
  - Controlling server power by controlling the number of jobs scheduled
  - Fine-grained job-level power control for jobs collocated on the same server
- Crossing provider-user boundaries for energy savings:
  - Cloud providers need to make the benefits of energy savings visible to users
  - Need to overcome the issue of per-user power metering

Outline

- Problem
- Background
- Current Practice & Design Challenges
- Key Solutions
- Implementations
- Evaluations
- Open Challenges
- **Conclusions**

Conclusions

- Data center power capacity is heavily underutilized
- Utilize this spare capacity by adding additional servers
- Employ a power management system to deal with power emergencies or faults
  - Deal with power feeds
  - Enforce priority-aware power capping globally across the entire data center
  - Utilize the stranded power
- Our power management system boosts data center server capacity by 50%
- Highlight other open challenges in data center power management

Thank you!