Increasing Large-Scale Data Center Capacity by Statistical Power Control

Guosai Wang, Shuhao Wang, Bing Luo, Weisong Shi, Yinghang Zhu, Wenjun Yang, Dianming Hu, Longbo Huang, Xin Jin, Wei Xu
Data Centers

Expensive to build and operate

Building cost (large DCs): $9,000–$13,000/KW*
High power consumption: 10–20 MW

Goal: Fully utilize the capacity of data centers to reduce the TCO.

Our Result:

• +17% servers → +15% throughput
• Power violations effectively avoided.
• No performance disturbance to existing jobs.

[*LA Barroso, etc. The datacenter as a computer: An introduction to the design of warehouse-scale machines. 2013*]
Underutilized Capacity in DCs

Observation: Avg power utilization < 72% at DC level

Reason: Conservative power provisioning
  Provision according with rated power
  Running power < Rated power
Underutilized Capacity in DCs

Observation: Avg power utilization < 72% at DC level

Reason: Conservative power provisioning
   Provision according with rated power
   Running power < Rated power

Over-provisioning of the facility power?
   Increase the number of servers on each rack.
Why People Under-provision?

[Fan X, etc. Power provisioning for a warehouse-sized computer. ISCA 2007]
Why People Under-provision?

[Fan X, etc. Power provisioning for a warehouse-sized computer. ISCA 2007]
Why People Under-provision?

[Fan X, etc. Power provisioning for a warehouse-sized computer. ISCA 2007]
Why People Under-provision?

Servers on the row level

Power violation!

Row Power

Time

Over-provisioning

[Fan X, etc. Power provisioning for a warehouse-sized computer. ISCA 2007]
Power Capping Degrades Performance

Traditional approach: Power capping
Dynamic Voltage and Frequent Scaling (DVFS)
Power ≈ C \cdot V^2 \cdot F

Degrade the performance of running jobs!
Violate the SLA of the latency-sensitive jobs.
Power Capping Degrades Performance

Traditional approach: Power capping
Dynamic Voltage and Frequent Scaling (DVFS)
Power $\approx C \cdot V^2 \cdot F$

Degrade the performance of running jobs!
Violate the SLA of the latency-sensitive jobs.
Power Control Method

Can we control the power without affecting the performance of existing jobs?
Key Observation

Large variations on power utilization at row level
Temporal (over time) and spatial (across different rows).

Idea: Dynamically move workload out of the heavily used rows.
Key Observation

Large variations on power utilization at row level
Temporal (over time) and spatial (across different rows).

Idea: Dynamically move workload out of the heavily used rows.
Key Observation

Large variations on power utilization at row level
Temporal (over time) and spatial (across different rows).

Idea: Dynamically move workload out of the heavily used rows.
Our Solution: Statistical Power Control

- Minimize interface with the scheduler.
  - Two simple APIs: Freeze/unfreeze. Decoupled with the over-complicated scheduler.

- Statistically influence new job placement.
  - Indirect workload balancing. Running jobs unaffected. Does not necessarily work perfectly.

- Dynamic system control
  - Tolerate noises. System identification in a production environment.
Example: Statistical Power Control

Light workload
No control action.
Example: Statistical Power Control

Light workload
No control action.

New jobs
Scheduler

Running Jobs

Aggregated real-time power
Power Controller
Example: Statistical Power Control

Light workload
No control action.

Scheduler

Running Jobs

New jobs

Aggregated real-time power

Power Controller
Example: Statistical Power Control

Light workload
No control action.

New jobs

Scheduler

Running Jobs

Aggregated real-time power

Power Controller
Example: Statistical Power Control

Light workload
No control action.

New jobs

Running Jobs

Aggregated real-time power

Scheduler

Power Controller
Example: Statistical Power Control

Light workload
No control action.

Scheduler

Running Jobs

New jobs

Aggregated real-time power

Power Controller
Example: Statistical Power Control

Heavy workload.
High row power.

Running Jobs

Aggregated real-time power

Scheduler

Power Controller
Example: Statistical Power Control

Heavy workload.
High row power.

Scheduler

Running Jobs

Power Controller

Aggregated real-time power

Freeze
Example: Statistical Power Control

Heavy workload.
High row power.

Scheduler
New jobs
Running Jobs

Power Controller
Aggregated real-time power
Freeze
Example: Statistical Power Control

Heavy workload.
High row power.

New jobs

Scheduler

Running Jobs

Aggregated real-time power

Power Controller

Freeze
Example: Statistical Power Control

Heavy workload.
High row power.

Scheduler

Running Jobs

New jobs

Aggregated real-time power

Power Controller

Freeze
Example: Statistical Power Control

Heavy workload.
High row power.

Scheduler

Running Jobs

New jobs

Aggregated real-time power

Power Controller

Freeze
Example: Statistical Power Control

Heavy workload.
High row power.

New jobs

Running Jobs

Aggregated real-time power

Freeze

Scheduler

Power Controller
Example: Statistical Power Control

Heavy workload.
High row power.

New jobs

Scheduler

Unused power

Running Jobs

Aggregated real-time power

Power Controller

Freeze

Jobs
Example: Statistical Power Control

Some jobs finished.

Scheduler

Running Jobs

Aggregated real-time power

Power Controller

Freeze
Example: Statistical Power Control

Some jobs finished.
Some jobs finished.

Example: Statistical Power Control
Power Control Model Blueprint

- Dynamic control at each minute.
- No control needed when the power is low.
- Freeze more/fewer servers when power is high/low.
Power Control Model Blueprint

- Dynamic control at each minute.
- No control needed when the power is low.
- Freeze more/fewer servers when power is high/low.
Power Control Model Blueprint

- Dynamic control at each minute.
- No control needed when the power is low.
- Freeze more/fewer servers when power is high/low.
Effect of Freezing Servers

Two effects jointly impact on the row-level power.

- Existing jobs will finish
- Statistically fewer jobs scheduled to the row

![Graph: Average normalized power of about 80 servers after they are frozen.](image)
Effect of Freezing Servers

Two effects jointly impact on the row-level power.

- Existing jobs will finish
- Statistically fewer jobs scheduled to the row

How to **quantify** these effects?

System identification in a production environment?

Designed a controlled experiment.
Controlled Experiment Design

Controlled experiment in production environment.

Idea: A/B testing
Controlled Experiment Design

Controlled experiment in production environment.

Idea: A/B testing
Controlled Experiment Design

Controlled experiment in production environment.

Idea: A/B testing
Controlled Experiment Design

Controlled experiment in production environment.

Idea: A/B testing

Row 1
Row 2
Row n

Power Controller
Control Actions

Experiment Group
Control Group

Correlation coefficient of the group power is 0.946
Dynamic Control Model

How many servers do we need to freeze in a row?
   Freeze too few: Risk of Power violations!
   Freeze too many: Reduce the throughput!

Optimization problem:
   Maximize: TPW (Throughput per Provisioned Watt)
   s.t.   No power violation

Key idea:
   Use simple system model and tolerate inaccuracy with dynamic control.
Dynamic Control Model

Use heuristics to derive a simple control model.
Take control actions at each minute.
Details in the paper.
Dynamic Control Model

Use heuristics to derive a simple control model.
Take control actions at each minute.
Details in the paper.
Dynamic Control Model

Use heuristics to derive a simple control model.

Take control actions at each minute.
Details in the paper.
Dynamic Control Model

Use heuristics to derive a simple control model.
Take control actions at each minute.
Details in the paper.

![Diagram showing Freezing Ratio as a function of Realtime Row Power]

- Threshold Ratio
- $E_t$
- Power Limit
- $r_{threshold}$
- $P_M = 1.0$
Use heuristics to derive a simple control model.
Take control actions at each minute.
Details in the paper.
Dynamic Control Model

Use heuristics to derive a simple control model.
Take control actions at each minute.
Details in the paper.
How to Emulate Over-provisioning?

- Safety: Unacceptable to truly trigger power violations in production environment.
- Flexibility: How to test various over-provisioning ratio?

Solution: Emulating power violations by virtually scaling down the power budget of the row.
How to Emulate Over-provisioning?

- Safety: Unacceptable to truly trigger power violations in production environment.
- Flexibility: How to test various over-provisioning ratio?

Solution: Emulating power violations by virtually scaling down the power budget of the row.

\[ \text{Over-provisioning ratio: } \frac{P - P'}{P'} \]

- **Actual** row power budget: \( P \)
- **Assumed** row power budget: \( P' \)
How to Emulate Over-provisioning?

- Safety: Unacceptable to truly trigger power violations in production environment.
- Flexibility: How to test various over-provisioning ratio?

Solution: Emulating power violations by virtually scaling down the power budget of the row.

Actual row power budget: $P$

Assumed row power budget: $P'$

Over-provisioning ratio: $(P-P')/P'$
Effectiveness

Controlled experiments on production environment.
Over-provisioning ratio = 0.25
Effectiveness

Controlled experiments on production environment.
Over-provisioning ratio = 0.25
Effectiveness

Controlled experiments on production environment.
Over-provisioning ratio = 0.25
Effectiveness

Controlled experiments on production environment.
Over-provisioning ratio = 0.25
Effectiveness

Controlled experiments on production environment.
Over-provisioning ratio = 0.25
Effectiveness

Controlled experiments on production environment.
Over-provisioning ratio = 0.25
How to Decide Over-provisioning Ratio?

Throughput per Provisioned Watt (TPW):

\[ TPW = \frac{\text{Throughput during time interval } T}{P \cdot T} \]

Gain in TPW:

\[ G_{TPW} = r_T \cdot (1 + r_O) - 1 \]

- \( P \) Provisioned power
- \( r_T \) Throughput ratio (\( \leq 1 \))
- \( r_O \) Over-provisioning ratio (\( \geq 1 \))
How to Decide Over-provisioning Ratio?

Throughput per Provisioned Watt (TPW):

\[
TPW = \frac{\text{Throughput during time interval } T}{P \cdot T}
\]

Gain in TPW:

\[
G_{TPW} = r_T \cdot (1 + r_O) - 1
\]

- \(P\) Provisioned power
- \(r_T\) Throughput ratio (\(\leq 1\))
- \(r_O\) Over-provisioning ratio (\(\geq 1\))
How to Decide Over-provisioning Ratio?

Throughput per Provisioned Watt (TPW):

\[ TPW = \frac{\text{Throughput during time interval } T}{P \cdot T} \]

Gain in TPW:

\[ G_{TPW} = r_T \cdot (1 + r_O) - 1 \]

By emulations we found \( G_{TPW} = 0.149 \) when \( r_O = 0.17 \).

\( P \) Provisioned power

\( r_T \) Throughput ratio (≤1)

\( r_O \) Over-provisioning ratio (≥1)
Conclusion

- Admission control to statistically influencing new job placement
- Minimal APIs (freeze/unfreeze)
- Simple dynamic system control
- Controlled experiment

- Avoid performance degradation.
- Decouple the power control module and the complicated scheduler.
- Tolerate inaccuracy.
- Build and evaluate system model in production environment without disturbing it too much.
Conclusion

- Admission control to statistically influencing new job placement
  - Avoid performance degradation.
- Minimal APIs (freeze/unfreeze)
  - Decouple the power control module and the complicated scheduler.
- Simple dynamic system control
  - Tolerate inaccuracy.
- Controlled experiment
  - Build and evaluate system model in production environment without disturbing it too much.
Q&A

Outline:

- Power over-provisioning motivation
- Ideas of statistical power control
- Dynamic Control model
- Controlled experiment design
- Effectiveness
- Deciding over-provisioning ratio
- Conclusion
Q&A

Outline:
- Power over-provisioning motivation
- Ideas of statistical power control
- Dynamic Control model
- Controlled experiment design
- Effectiveness
- Deciding over-provisioning ratio
- Conclusion
Q&A

Outline:

Power over-provisioning motivation
Ideas of statistical power control
Dynamic Control model
Controlled experiment design
Effectiveness
Deciding over-provisioning ratio
Conclusion
Q&A

Outline:

Power over-provisioning motivation
Ideas of statistical power control
Dynamic Control model
Controlled experiment design
Effectiveness
Deciding over-provisioning ratio
Conclusion
Q&A

Outline:
- Power over-provisioning motivation
- Ideas of statistical power control
- Dynamic Control model
- Controlled experiment design
- Effectiveness
- Deciding over-provisioning ratio
- Conclusion
Backup Slides
Ampere Architecture

Scheduler

Row #1  Row #2  ...  Row #n

Power Monitor

Controller

scheduling actions

freeze/unfreeze

aggregated power
Job Durations
$G_{TPW}$ under Different $r_O$

<table>
<thead>
<tr>
<th>#</th>
<th>$r_O$</th>
<th>$P_{mean}$</th>
<th>$P_{max}$</th>
<th>$u_{mean}$</th>
<th>$r_T$</th>
<th>$G_{TPW}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.25</td>
<td>0.903</td>
<td>1.028</td>
<td>0.019</td>
<td>0.953</td>
<td>19.70%</td>
</tr>
<tr>
<td>2</td>
<td>0.25</td>
<td>0.931</td>
<td>1.062</td>
<td>0.134</td>
<td>0.941</td>
<td>17.60%</td>
</tr>
<tr>
<td>3</td>
<td>0.25</td>
<td><strong>0.936</strong></td>
<td><strong>1.062</strong></td>
<td><strong>0.152</strong></td>
<td><strong>0.885</strong></td>
<td><strong>10.60%</strong></td>
</tr>
<tr>
<td>4</td>
<td>0.25</td>
<td>0.927</td>
<td>1.061</td>
<td>0.196</td>
<td>0.835</td>
<td>4.30%</td>
</tr>
<tr>
<td>5</td>
<td>0.21</td>
<td>0.786</td>
<td>0.913</td>
<td>0</td>
<td>1.0</td>
<td>20.70%</td>
</tr>
<tr>
<td>6</td>
<td>0.21</td>
<td>0.835</td>
<td>0.982</td>
<td>0.0016</td>
<td>1.0</td>
<td>20.70%</td>
</tr>
<tr>
<td>7</td>
<td>0.21</td>
<td>0.894</td>
<td>1.000</td>
<td>0.009</td>
<td>0.979</td>
<td>18.20%</td>
</tr>
<tr>
<td>8</td>
<td>0.21</td>
<td><strong>0.903</strong></td>
<td><strong>1.036</strong></td>
<td><strong>0.11</strong></td>
<td><strong>0.88</strong></td>
<td><strong>6.20%</strong></td>
</tr>
<tr>
<td>9</td>
<td>0.17</td>
<td>0.836</td>
<td>0.931</td>
<td>0</td>
<td>1.0</td>
<td>17%</td>
</tr>
<tr>
<td>10</td>
<td>0.17</td>
<td>0.839</td>
<td>0.926</td>
<td>0</td>
<td>1.0</td>
<td>17%</td>
</tr>
<tr>
<td>11</td>
<td>0.17</td>
<td><strong>0.908</strong></td>
<td><strong>0.992</strong></td>
<td><strong>0.07</strong></td>
<td><strong>0.984</strong></td>
<td><strong>14.90%</strong></td>
</tr>
<tr>
<td>12</td>
<td>0.17</td>
<td>0.938</td>
<td>1.004</td>
<td>0.12</td>
<td>0.904</td>
<td>5.50%</td>
</tr>
<tr>
<td>13</td>
<td>0.13</td>
<td>0.847</td>
<td>0.969</td>
<td>0</td>
<td>1.0</td>
<td>13%</td>
</tr>
</tbody>
</table>

\[ \text{Fix } r_O, P_{mean}, u_{mean}, r_T \Rightarrow G_{TPW} \downarrow \]
\[ r_O \uparrow \Rightarrow u_{mean} \uparrow \Rightarrow r_T \downarrow \Rightarrow G_{TPW} \downarrow \]
\[ G_{TPW} < r_O \]
Quantify the Effect of Freezing Ratio

The effects of freezing ratio $u$ on the power change $f(u)$. 

![Graph showing the effect of freezing ratio on power change]
Limitations and Discussion

What if the workload increases in the future?

What if the jobs are locality-aware scheduled?

What if the amount of jobs is small and they are long-lived?

How to jointly optimize the control among all rows?

Experiments needed before deployment?