Data Centers Power Reduction: A Two Time Scale Approach for Delay Tolerant Workloads

Yuan Yao, Longbo Huang, Abhihshek Sharma, Leana Golubchik and Michael Neely

University of Southern California

May 3, 2012

Recall last talk

Figure: Data center

Expenses on electricity bill:

- One 15MW data center $\rightarrow$ \$1M per month;
- $30 - 50\%$ of all operational expenses.

Electricity supplier: power grid

Figure: Power grid

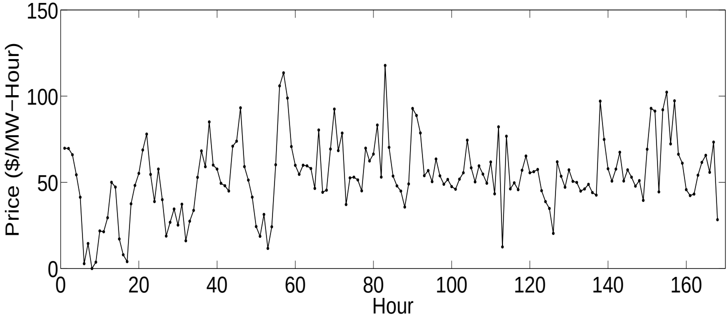

Figure: Avg. hourly spot market price during the week of $01/01/2005 - 01/07/2005$ for LA1 Zone

Figure: Uninterrupted Power Supply (UPS), *e.g.*, Battery

Motivation: store energy within the UPS when prices are low and discharge it when prices are high.

- Contribution of that paper: online control algorithm to optimally exploit these UPS devices to minimize the time average cost.

- Contribution of that paper: online control algorithm to optimally exploit these UPS devices to minimize the time average cost.
- Merit: without any knowledge of the statistics of the workload or electricity cost processes.

- Contribution of that paper: online control algorithm to optimally exploit these UPS devices to minimize the time average cost.
- Merit: without any knowledge of the statistics of the workload or electricity cost processes.
- Tradeoff: optimality gap reduces as the storage capacity is increased.

- Contribution of that paper: online control algorithm to optimally exploit these UPS devices to minimize the time average cost.
- Merit: without any knowledge of the statistics of the workload or electricity cost processes.
- Tradeoff: optimality gap reduces as the storage capacity is increased.

Only exploit the temporal diversity of power price at single data center.

What's new in this talk

- Exploit both **spatial** and **temporal** variations in the **workload arrival process** and the **power prices**;

What's new in this talk

- Exploit both **spatial** and **temporal** variations in the **workload arrival process** and the **power prices**;
- Cost vs. delay tradeoff: reduce power cost at the expense of increase service delay;

What's new in this talk

- Exploit both **spatial** and **temporal** variations in the **workload arrival process** and the **power prices**;
- Cost vs. delay tradeoff: reduce power cost at the expense of increase service delay;
- Two time scale control algorithm;

What's new in this talk

- Exploit both **spatial** and **temporal** variations in the **workload arrival process** and the **power prices**;
- Cost vs. delay tradeoff: reduce power cost at the expense of increase service delay;
- Two time scale control algorithm;
- Environmentally friendly: reduction in both power cost and power usage.

Multiple geographically distributed data centers, each with multiple servers.

Three levels of power reduction:

Multiple geographically distributed data centers, each with multiple servers.

Three levels of power reduction:

- Server level: save power usage by adjusting the *CPU speed* of a single server;

Multiple geographically distributed data centers, each with multiple servers.

Three levels of power reduction:

- Server level: save power usage by adjusting the *CPU speed* of a single server;
- Data center level: dynamically control the *number of activated servers* in a data center;

Multiple geographically distributed data centers, each with multiple servers.

Three levels of power reduction:

- Server level: save power usage by adjusting the *CPU speed* of a single server;
- Data center level: dynamically control the *number of activated servers* in a data center;
- Inter-data center level: exploit the price diversity of geographically distributed data centers and *route* more workload to *places with lower power prices*.

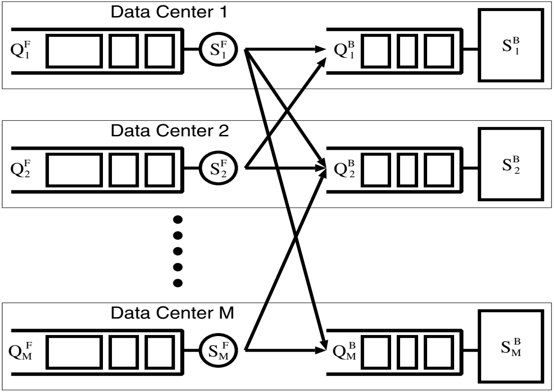

Figure: A model of M geographically distributed data centers.

Workload model

Workload arrival rate at D_i in slot t: $A_i(t)$,

- $0 \leq A_i(t) \leq A_{max}$;
- i.i.d. every time slot.

Job distribution and server operation model

Two time scale control:

- Every T time slots, *i.e.*, $t = kT$, with $k = 1, 2, \ldots$:

Job distribution and server operation model

Two time scale control:

- Every T time slots, *i.e.*, $t = kT$, with $k = 1, 2, \ldots$:
  - $N_i(t)$: number of active serves on data center D_i; $N^i_{min} \leq N_i(t) \leq N_i$;
    - Activating servers $\Rightarrow$ non-negligible time and power;
    - Frequently switching between active and sleep $\Rightarrow$ reliability problems.

Job distribution and server operation model

Two time scale control:

- Every T time slots, *i.e.*, $t = kT$, with $k = 1, 2, \ldots$:
  - $N_i(t)$: number of active serves on data center D_i; $N^i_{min} \leq N_i(t) \leq N_i$;
    - Activating servers $\Rightarrow$ non-negligible time and power;
    - Frequently switching between active and sleep $\Rightarrow$ reliability problems.
- Every time slot $t = 1, 2, \ldots$:

Job distribution and server operation model

Two time scale control:

- Every T time slots, *i.e.*, $t = kT$, with $k = 1, 2, \ldots$:
  - $N_i(t)$: number of active serves on data center D_i; $N^i_{min} \leq N_i(t) \leq N_i$;
    - Activating servers $\Rightarrow$ non-negligible time and power;
    - Frequently switching between active and sleep $\Rightarrow$ reliability problems.
- Every time slot $t = 1, 2, \ldots$:
  - $\mu_{ij}(t)$: number of jobs routed from Q_i^F to Q_j^B; $\mu_i(t) = (\mu_{i1}(t), \ldots, \mu_{iM}(t)) \in \mathcal{R}_i$.

Job distribution and server operation model

Two time scale control:

- Every T time slots, *i.e.*, $t = kT$, with $k = 1, 2, \ldots$:
  - $N_i(t)$: number of active serves on data center D_i; $N^i_{min} \leq N_i(t) \leq N_i$;
    - Activating servers $\Rightarrow$ non-negligible time and power;
    - Frequently switching between active and sleep $\Rightarrow$ reliability problems.
- Every time slot $t = 1, 2, \ldots$:
  - $\mu_{ij}(t)$: number of jobs routed from Q^F_i to Q^B_j; $\mu_i(t) = (\mu_{i1}(t), \ldots, \mu_{iM}(t)) \in \mathcal{R}_i$.
  - $b_i(t)$: CPU rate on each serve at D_i; $0 \leq b_i(t) \leq b_{max}$;
    - All servers in data center i operate at same rate;
    - Provable optimal choice with convex power consumption function.

Cost model

- Power usage function: $P_i(N_i(\lfloor \frac{t}{T} \rfloor T), b_i(t)) \leq P_{max}$;
- Power price: $p_i(t) \leq p_{max}$; changes every T_1 slots; $T = cT_1$;
- Power cost function: $f_i(t) = P_i(N_i(\lfloor \frac{t}{T} \rfloor T), b_i(t))p_i(t)$;
- $\sum_i f_i(t) \leq f_{max} \triangleq MP_{max}p_{max}$.

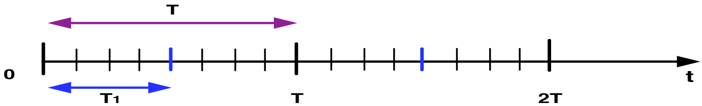

Figure: An example of different time scales T and T_1. In this example, $T = 8$, $T_1 = 4$, and $T = 2T_1$.

Queues

- Front end servers:

$$Q_i^F(t+1) = \max\{Q_i^F(t) - \sum_j \mu_{ij}(t), 0\} + A_i(t); \qquad (1)$$

- Back end clusters:

$$Q_i^B(t+1) \leq \max\{Q_i^B(t) - N_i(t)b_i(t), 0\} + \sum_j \mu_{ji}(t). \qquad (2)$$

Feasible policy $\prod$

- Every T slots: $N_{min}^{i} \leq N_i(t) \leq N_i$;
- Every time slot: $\mu_{\mathbf{i}}(t) \in \mathcal{R}_i$ and $0 \leq b_i(t) \leq b_{max}$;

such that

$$\bar{Q} \triangleq \lim_{t\to\infty} \sup \frac{1}{t} \sum_{\tau=0}^{t-1} \sum_{i=1}^{M} \mathbb{E}\{Q_i^F(\tau) + Q_i^B(\tau)\} \leq \infty.$$

Power cost minimization problem

$$\min_{\Pi} \quad f_{av}^{\Pi} \triangleq \lim_{t \to \infty} \sup \frac{1}{t} \sum_{\tau=0}^{t-1} \sum_{i=1}^{M} \mathbb{E}\{f_i^{\Pi}(\tau)\} \tag{3}$$

Front end routing

In every time $t = kT$, $k = 0, 1, \ldots$, each D_i solves $\mu_{ij}(t)$ to maximize:

$$\sum_{i=1}^{M} \mu_{ij}(t)[Q_i^F(t) - Q_j^F(t)]. \tag{4}$$

In every time slot $\tau \in [t, t+T-1]$, $\mu_{ij}(\tau) \leq \mu_{ij}(t)$.

Back end server management

In every time $t = kT$, $k = 0, 1, \ldots$, each D_i solves $N_i(t)$ to minimize:

$$\mathbb{E}\{\sum_{\tau=t}^{t+T-1}\sum_{j}[Vf_j(\tau) - Q_j^B(t)N_j(t)b_j(\tau)]|\mathbf{Q}(t)\}. \qquad (5)$$

Need statistical information on workload arrival rates $A_i(t)$ and power prices $p_i(t)$.

Back end server management (Cont.)

In every time $\tau = 1, 2, \ldots$, with solved $N_i(t)$, each D_i solves $b_i(\tau)$ to minimize:

$$V f_j(\tau) - Q_j^B(t) N_j(t) b_j(\tau). \tag{6}$$

Performance of SAVE

Suppose there exists an $\epsilon > 0$ such that $\lambda + 2\epsilon\mathbf{1} \in \mathbf{\Lambda}$, then under the SAVE algorithm, we have:

$$\bar{Q} \triangleq \lim_{K\to\infty} \sup \frac{1}{K} \sum_{k=0}^{K-1} \sum_{i=1}^{M} \mathbb{E}\{Q_i^F(kT) + Q_i^B(kT)\} \leq \frac{B_2 + V f_{max}}{\epsilon},$$

$$f_{av}^{SAVE} \triangleq \lim_{t\to\infty} \sup \frac{1}{t} \sum_{\tau=0}^{t-1} \sum_{i=1}^{M} \mathbb{E}\{f(\tau)\} \leq f_{av}^* + \frac{B_2}{V}.$$

Here, $B_2 \triangleq B_1 + (T-1)\sum_j [N_j^2 b_{max}^2 + (M^2+1)\mu_{max}^2]/2$ with $B_1 \triangleq MA_{max}^2 + \sum_i N_i^2 b_{max}^2 + (M^2+M)\mu_{max}^2$.

Robustness of SAVE

Suppose there exists an $\epsilon > 0$ such that $\lambda + 2\epsilon\mathbf{1} \in \mathbf{\Lambda}$. Also suppose there exists a constant c_e such that at all time t, the estimated backlog sizes $\hat{Q}_i^F(t)$, $\hat{Q}_i^B(t)$ and the actual backlog sizes $Q_i^F(t)$, $Q_i^B(t)$ satisfy $|\hat{Q}_i^F(t) - \hat{Q}_i^F(t)| \leq c_e$ and $|\hat{Q}_i^B(t) - \hat{Q}_i^B(t)| \leq c_e$ then under the SAVE algorithm, we have:

$$\bar{Q} \triangleq \lim_{K\to\infty} \sup \frac{1}{K} \sum_{k=0}^{K-1} \sum_{i=1}^{M} \mathbb{E}\{Q_i^F(kT) + Q_i^B(kT)\} \leq \frac{B_3 + V f_{max}}{\epsilon},$$

$$f_{av}^{SAVE} \triangleq \lim_{t\to\infty} \sup \frac{1}{t} \sum_{\tau=0}^{t-1} \sum_{i=1}^{M} \mathbb{E}\{f(\tau)\} \leq f_{av}^* + \frac{B_3}{V}.$$

Here, $B_3 \triangleq B_2 + 2Tc_e(\mu_{max} + A_{max} + N_{max}b_{max} + M\mu_{max})$.

Schemes for comparison

- **Local computation**: No routing, *i.e.*, $\mu_{ii} = A_i$ and $\mu_{ij} = 0$ if $j \neq i$;
- **Load balancing**: $\mu_{ij}(t)$ proportional to service capacity of D_j;
- **Low price**: Heuristic protocol routing more jobs to data centers with lower power prices;
- **Instant on/off**: Idealized protocol, assuming no delay/cost to activate/put to sleep any server; the same routing scheme with *Load balancing* scheme.

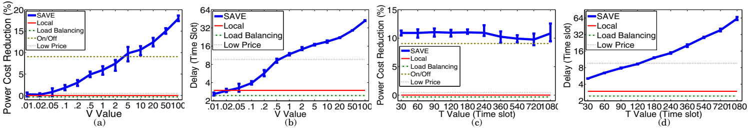

Figure: Average power cost and delay of all schemes under different V and T values.

- Impact of V: adjust the tradeoff between power cost reduction and service delay;
- Impact of T: little influence on power cost while proportional to service delay.

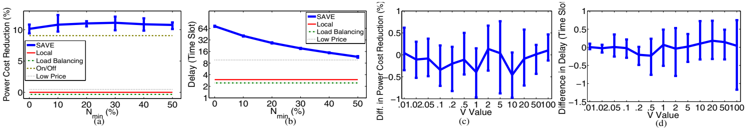

Figure: Average power cost and delay of all schemes under different N_{min} values and robustness test results.

- Impact of N_{min}: little influence on power cost while inverse proportional to service delay;
- Impact of estimation error on workloads: robust performance to errors.

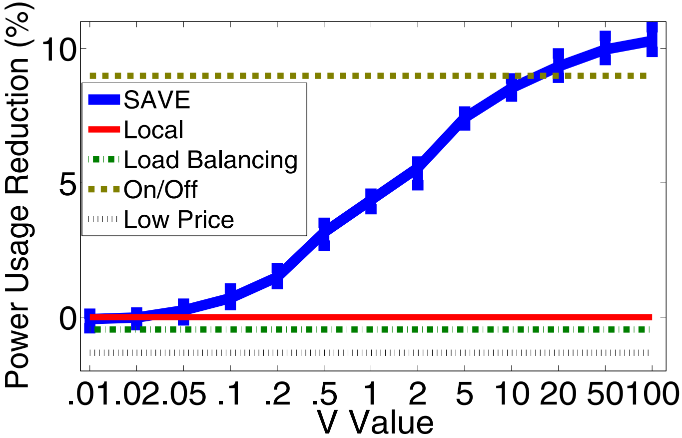

Figure: Differences in average power usage reduction for different V values.

Environmentally friendly: reduction in actual power usage.

Contributions

- Technical contribution:
  - Traditional single time-scale Lyapunov optimization $\Rightarrow$ two time-scale;
  - Traditional Lyapunov optimization based on accurate information $\Rightarrow$ Expectation-based & error-tolerant;

Contributions

- Technical contribution:
  - Traditional single time-scale Lyapunov optimization $\Rightarrow$ two time-scale;
  - Traditional Lyapunov optimization based on accurate information $\Rightarrow$ Expectation-based & error-tolerant;

Neither novel nor solid; but, good illustration with fine story.

Contributions

- Technical contribution:
  - Traditional single time-scale Lyapunov optimization $\Rightarrow$ two time-scale;
  - Traditional Lyapunov optimization based on accurate information $\Rightarrow$ Expectation-based & error-tolerant;

  Neither novel nor solid; but, good illustration with fine story.
- Utilize both spatial and temporal diversities in both workload arrival processes and power prices.

Remarks

- Lyapunov optimization has met its bottleneck on technical improvement;

Remarks

- Lyapunov optimization has met its bottleneck on technical improvement;
- Simple application of Lyapunov is not enough for good publication;

Remarks

- Lyapunov optimization has met its bottleneck on technical improvement;
- Simple application of Lyapunov is not enough for good publication;
- Good story, neat application, and practical insights are necessary;

Remarks

- Lyapunov optimization has met its bottleneck on technical improvement;
- Simple application of Lyapunov is not enough for good publication;
- Good story, neat application, and practical insights are necessary;
- New trend: trace-based empirical study.
  - Seems practical and applicable;
  - In fact, trace data has patterns!

Thank You!

Q&A