Power Minimization Techniques for Networked Datacenters

PI:	Professor Steven Low CMS, EE, Caltech
Co-PI:	Professor Kevin Tang EE, Cornell

Period:	Jan 2010 – June 2011
Type:	Concept definition

Acknowledgments

Caltech

- Professor Adam Wierman
- Grad students: Minghong Lin, Zhenhua Liu

Cornell

- Grad student: Chiunlin Lim

Outline

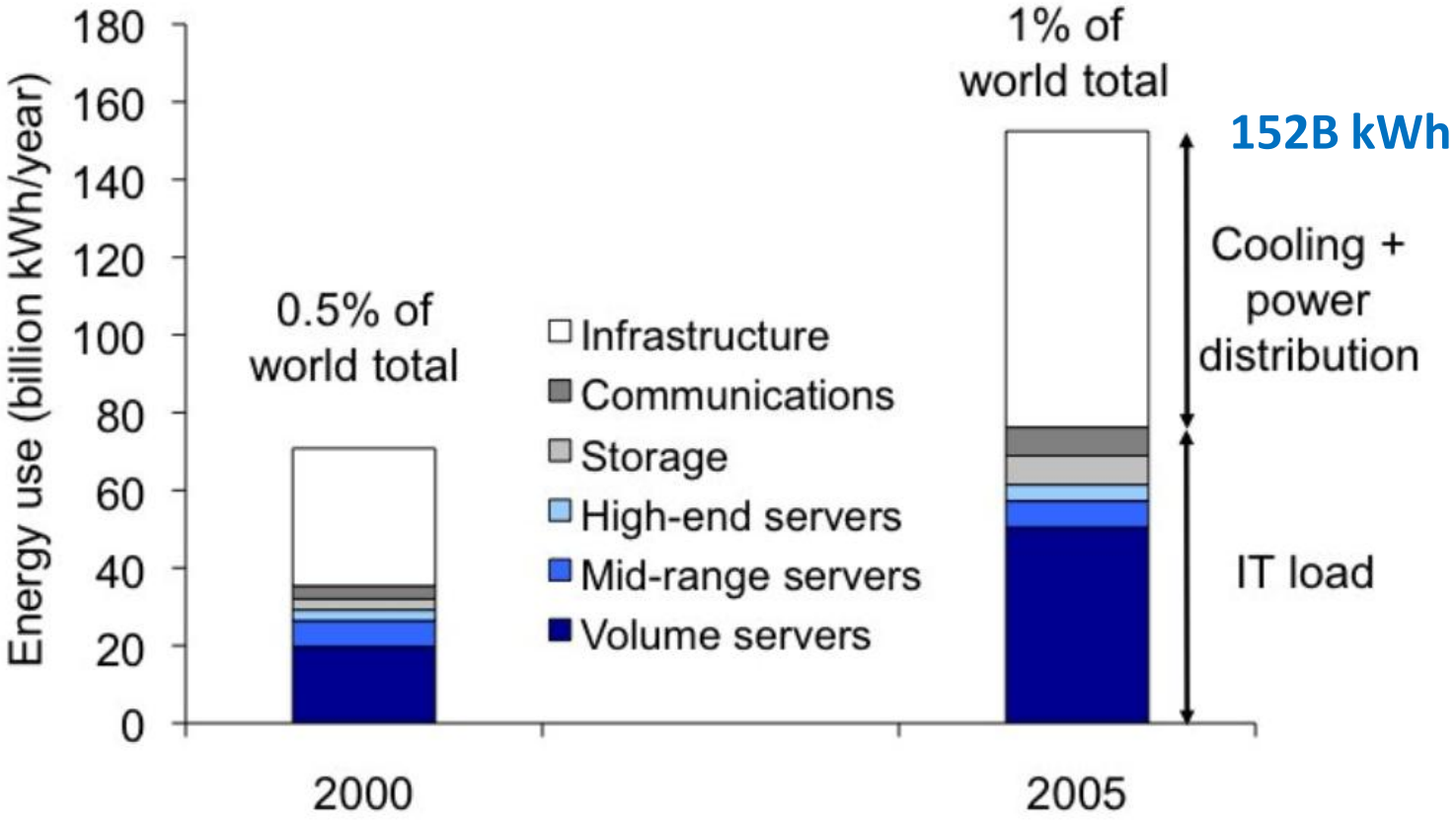
Motivation and objectives

Our approach and expected outcomes

Key results and implications

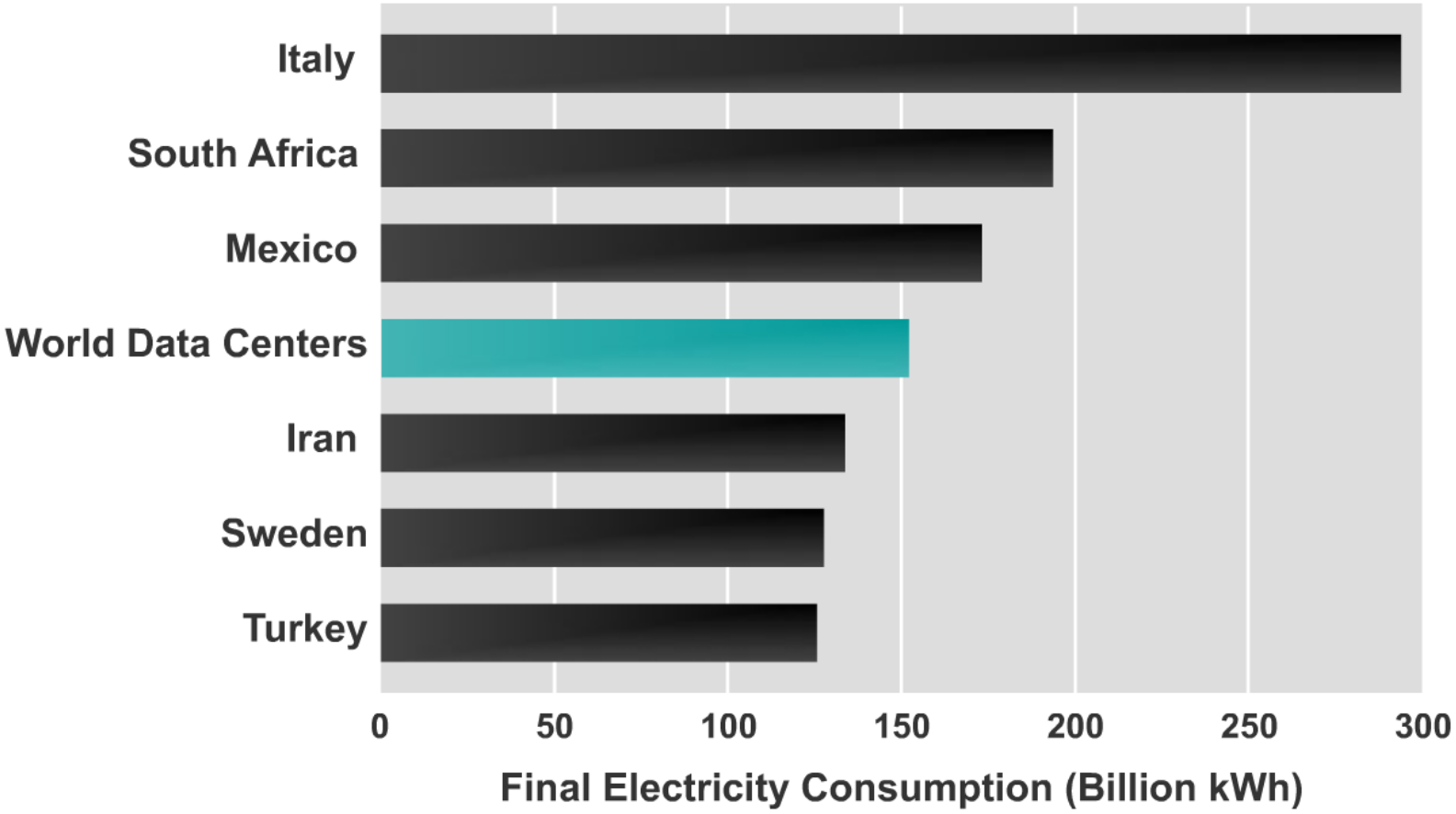
Examples

World data center electricity use, 2000 and 2005



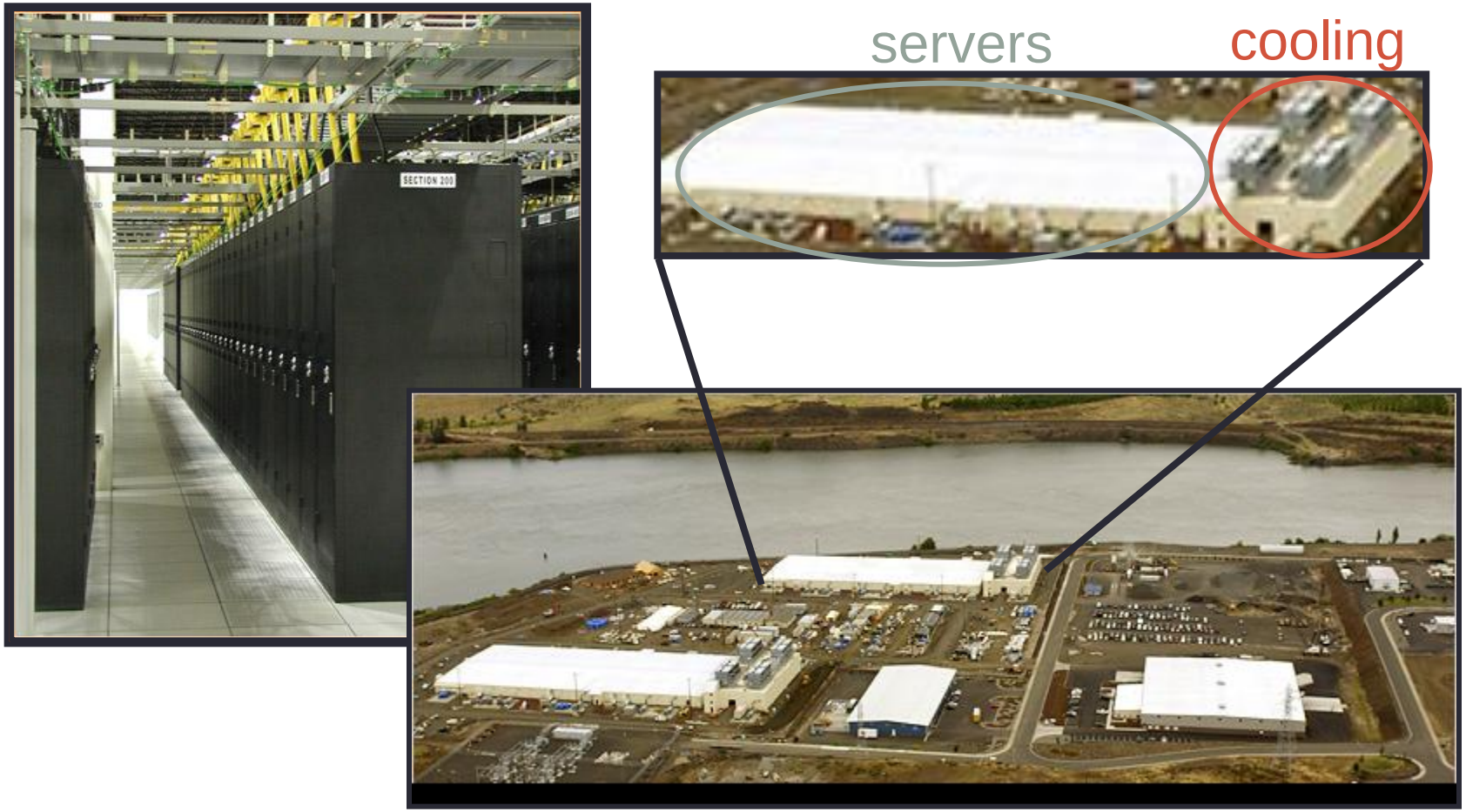
Source: Koomey 2008

How much is 152B kWh?



Source for country data in 2005: International Energy Agency, *World Energy Balances* (2007 edition)

Data Centers



“a 1.3 million core warehouse-sized computer”

DoE ITP Review

Datacenters: 3% of energy use

- ~50% hardware/software ← this project
- ~25% power supply chain
- ~25% cooling

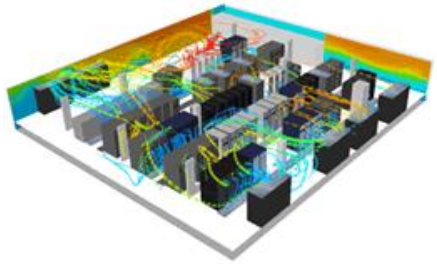
Growing 12% annually

Exceeds server cost over its 3-5yr lifetime

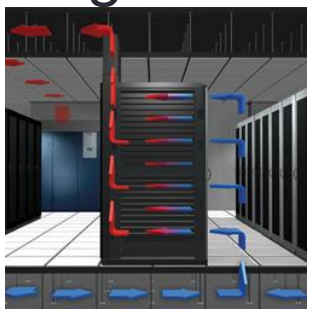
From Gideon Varga at ITP Review, 2011

Significant engineering improvements

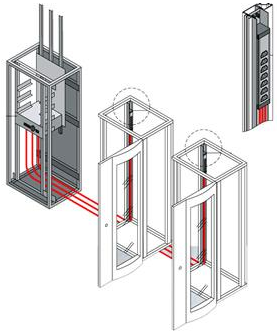
From measurement & monitoring



…to cooling technologies



…to power distribution



…to generation

Now PUE ➔ 1.2

So it's time to "green" the algorithms

Optimizing computation

Dynamic voltage scaling

Spin down or park the disk

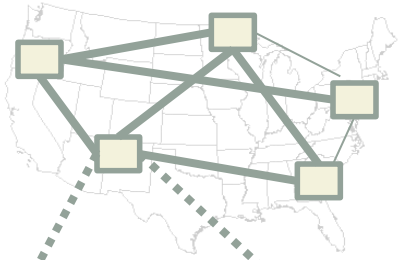
Dynamic transmission power

Application level power manager

Dynamically turn on/off servers

Source: Adam Wierman, Caltech

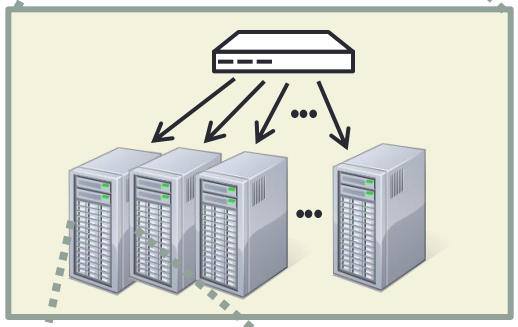
Global layer



backbone networks
(bandwidth vs. energy cost) [Caltech, Bell Labs]

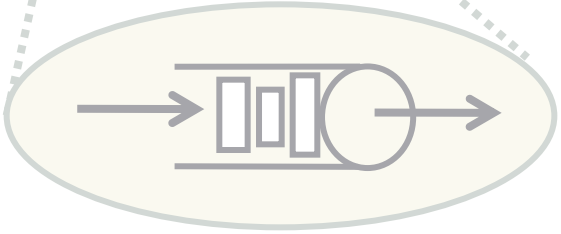
geographic load balancing
(latency vs. utilization vs. energy cost)
[this research, Ling, Wierman, Andrew, Thereska 2011]

Local layer



local load balancing and provision
(performance vs. utilization vs. energy)

Server layer

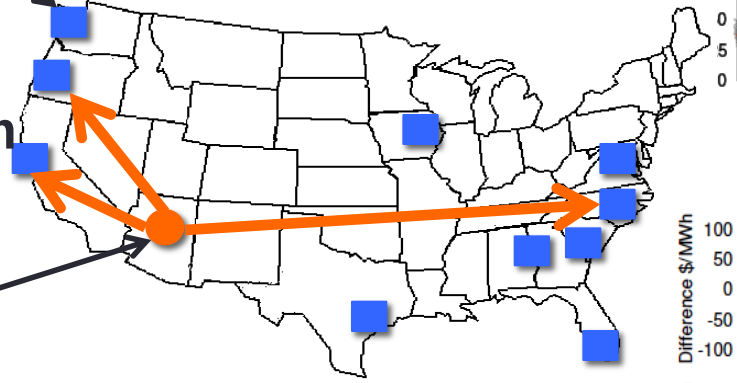


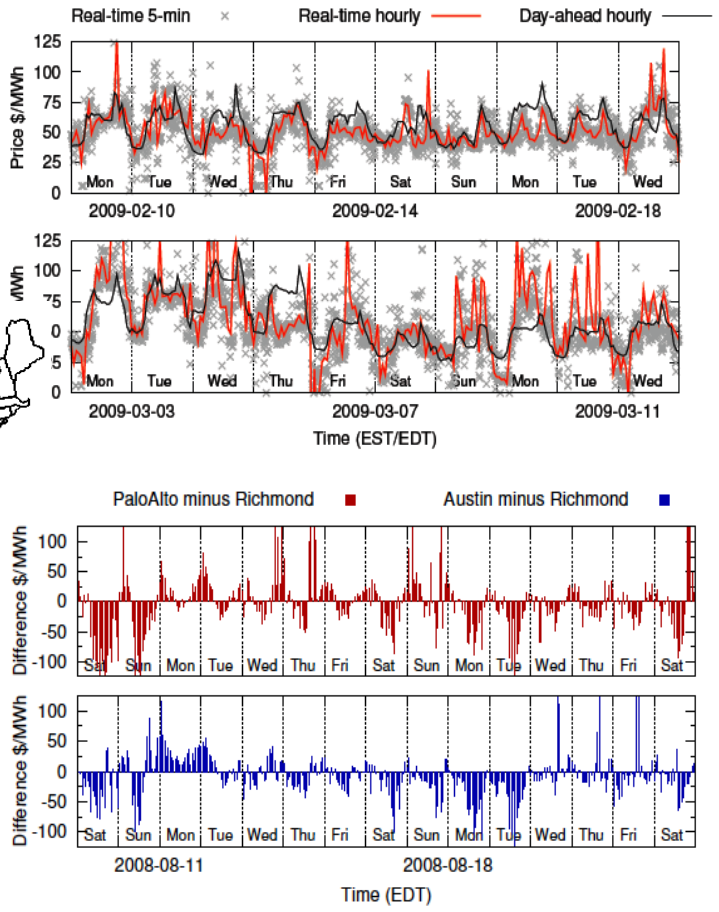
scheduling & speed scaling
(performance vs. energy)
[Nair, Wierman, Zwart 2010]

Geographic Load Balancing (GLB)

Data centers dynamically choose capacity based on routing

Distributed algorithm





Routing depends on

- ➔Data center loads
- ➔Network latency
- ➔**Energy prices**

Question: Can we give optimal, distributed algorithms?

Challenge: distributed control of routing and capacity simultaneously

A. Qureshi, et al. Cutting the electric bill for internet-scale systems. SIGCOMM 2009.

Objectives and deliverables

Objectives

- Develop models for energy optimization
- Develop algorithms
- Evaluate algorithms

Outline

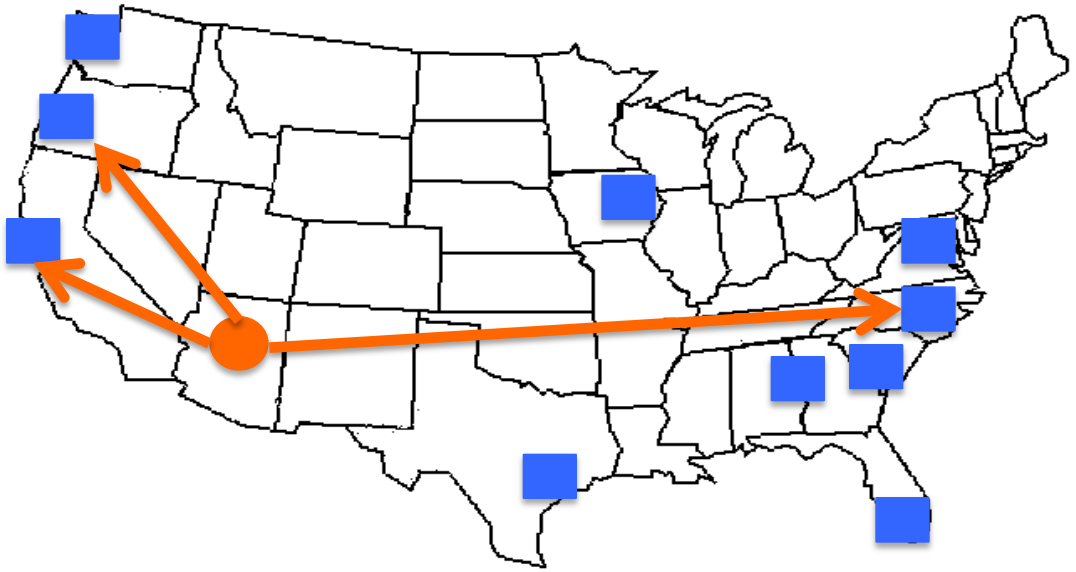
Motivation and objectives

Our approach and expected outcomes

Key results and implications

Examples

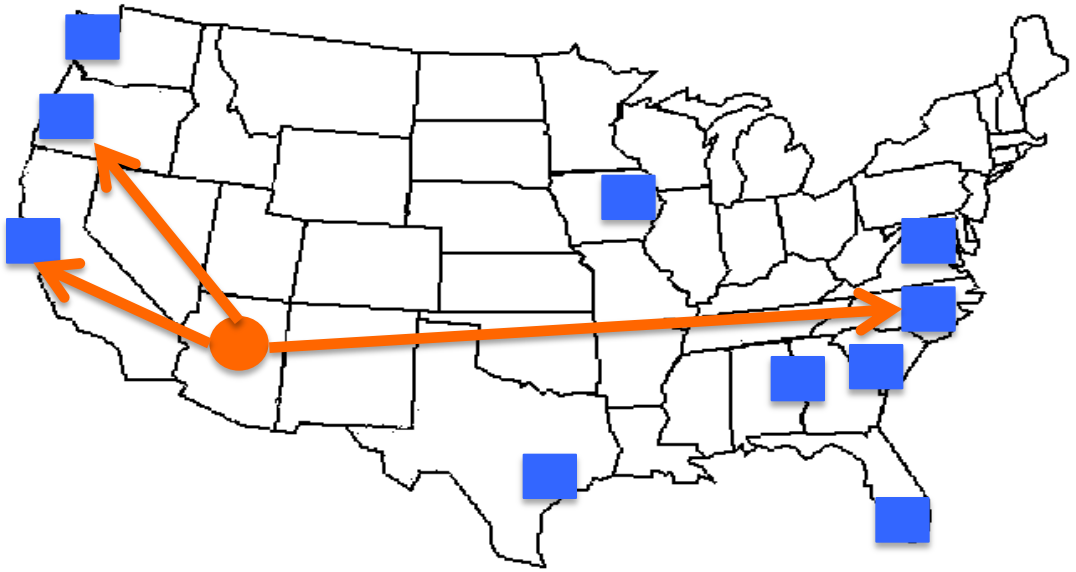
Basic approach



Fundamentally a tradeoff between energy consumption & delay performance

- Optimize over request direction and server capacity
- Balance
  - propagation delay (which DC?)
  - queueing delay (how much server capacity at DC?)
  - energy consumption (electricity cost)

Challenges



Distributed control

- implementable, but how to coordinate

Feedback control

- careful about stability

Theoretical guarantee: optimality, stability

Sanity check: examples

To make this possible …

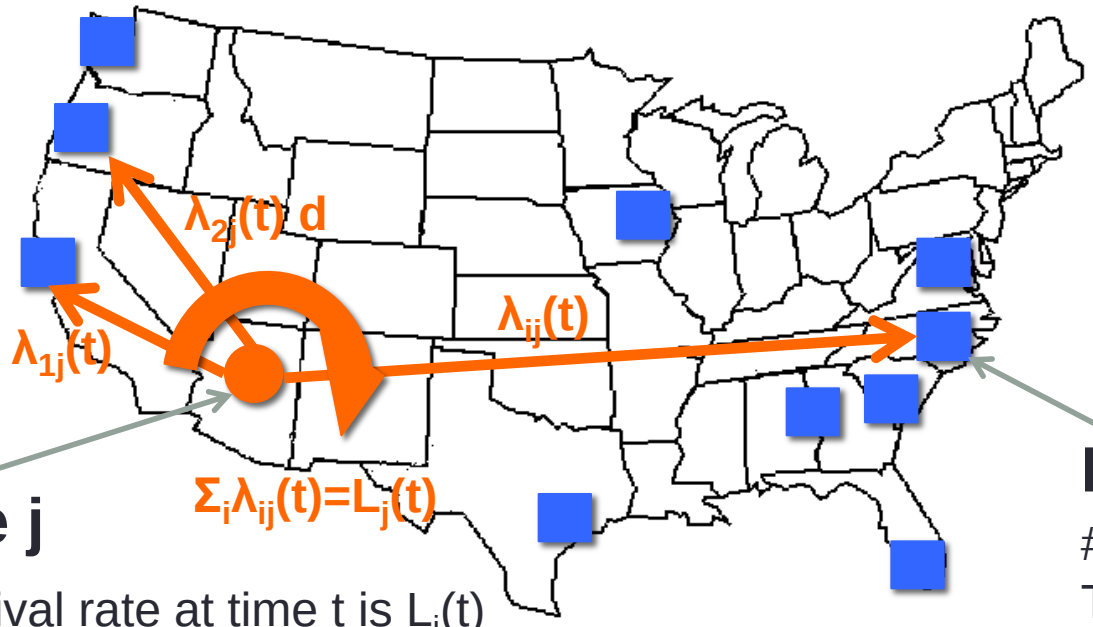
Dynamic capacity provisioning

- ➔ 3-competitive online algorithm
- ➔ Research at HP Labs on feedback and feed-forward, e.g. AppRAISE, Resource pool management

Distributed routing by DNS/proxy

- ➔ DNS still preferred by many industry
- ➔ HTTP redirection and tunneling

Algorithms for real-time distributed load balancing



Source j

Mean arrival rate at time t is $L_j(t)$

Data Center i

# of active servers is $m_i(t)$
Total # of servers is M_i

Total cost $\Sigma_t \Sigma_i \left(\mathcal{E}_i(t) + \mathcal{D}_i(t)\right)$

In this talk $p_i m_i$ $\beta \sum_{j \in J} \lambda_{ij} \left(\frac{1}{\mu_i - \lambda_i / m_i} + d_{ij} \right)$ M/GI/1/PS

Goal

$$\min_{m,\lambda} \sum_{i \in N} p_i m_i + \beta \sum_{j \in J} \lambda_{ij} \left(\frac{1}{\mu_i - \lambda_i / m_i} + d_{ij} \right)$$

Dynamic capacity provision

$$\text{s.t.} \sum_{i \in N} \lambda_{ij} = L_j, \qquad \forall j \in J$$

$$\lambda_{ij} \geq 0, \qquad \forall i \in N, \forall j \in J$$

$$0 \leq m_i \leq M_i, \qquad \forall i \in N$$

Note:

➔ each time step is solved independently, i.e., we ignore the switching cost

➔ this is a convex optimization

Lin, Tang, 2011;
Lin, Wierman, Andrew, Thereska 2011
(Best Paper, Infocom 2011)

Goal

$$\min_{m,\lambda} \sum_{i \in N} p_i m_i + \beta \sum_{j \in J} \lambda_{ij} \left(\frac{1}{\mu_i - \lambda_i / m_i} + d_{ij} \right)$$

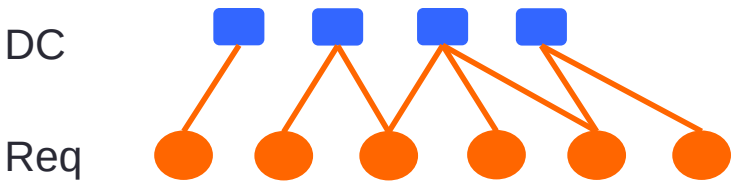
$$\text{s.t.} \quad \sum_{i \in N} \lambda_{ij} = L_j, \qquad \forall j \in J$$

$$\lambda_{ij} \geq 0, \qquad \forall i \in N, \forall j \in J$$

$$0 \leq m_i \leq M_i, \qquad \forall i \in N$$

Structural implications of optimality condition

- Sources only choose data centers with the same, lowest marginal cost
- All datacenter utilizations are equalized
- Simplest (sparse) routing structure



Gauss-Seidel iteration

→ **Sources update in a round-robin manner**

→ **Simple, but no concurrent updates**

Distributed Gradient Projection

→ **Sources concurrently do the gradient projection**

→ **Scaled to ensure constraints and Lipschitz continuity**

→ **Works, but needs projection computation**

Distributed Gradient Descent

→ **Sources reassign traffic according to gradient**

→ **Carefully adjust a time-varying stepsize**

→ **Faster convergence and suitable for protocol design**

Outline

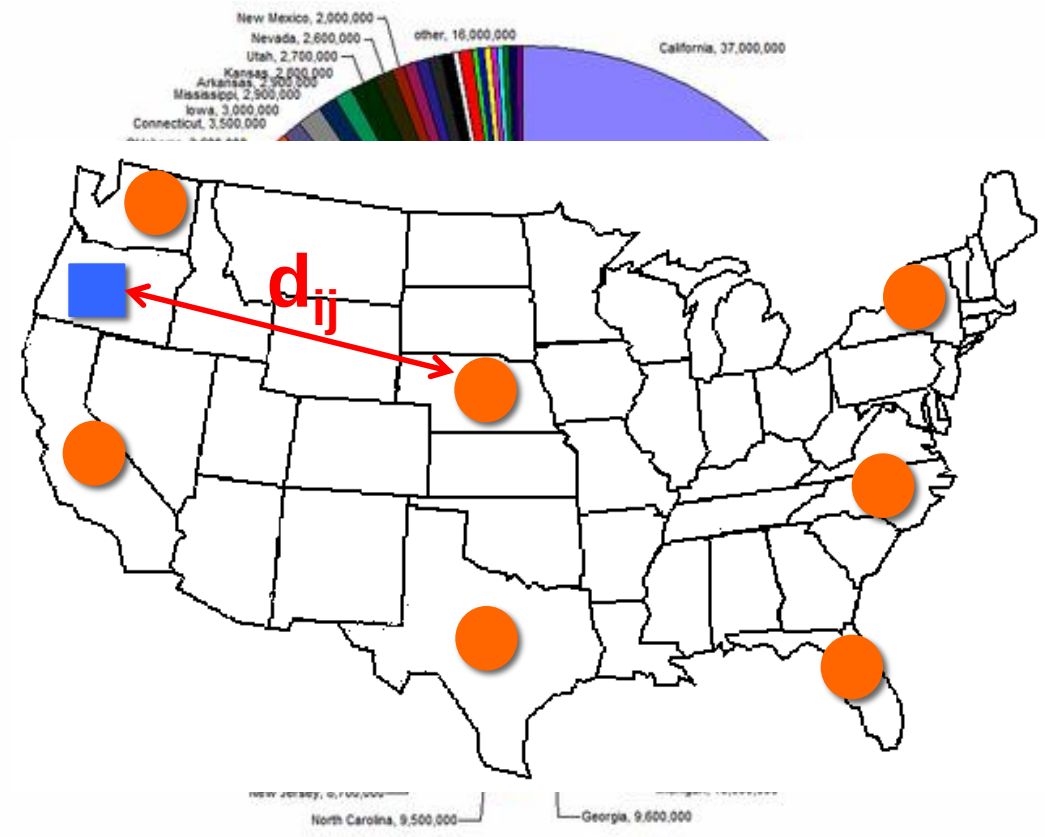
Motivation and objectives

Our approach and expected outcomes

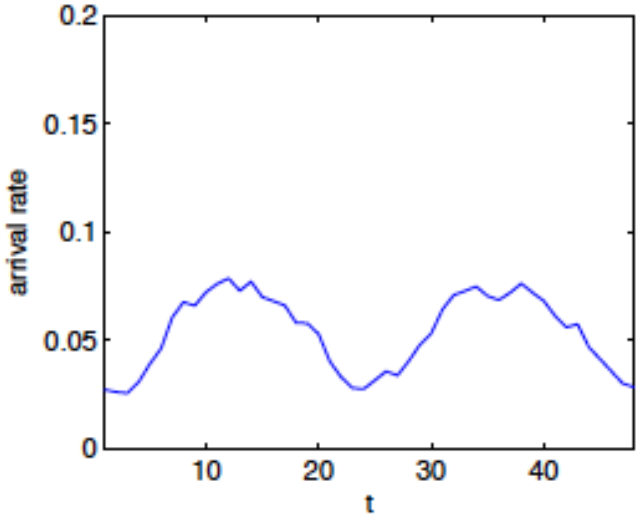
Key results and implications

Examples

Setup: Workload



Workload: I/O activity @ Microsoft Cambridge from 8 servers over a 48-hour period, starting at midnight PDT on Monday August 4, 2008.



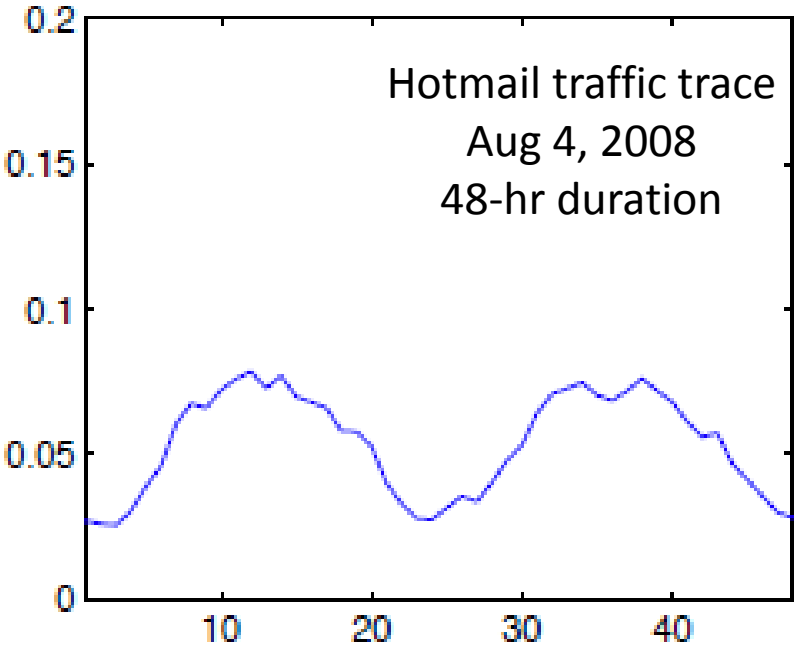
→ one source per state

→ workload scaled by population with Internet accesses

→ shift by time zones

→ propagation delay proportional to distance

Setup



48-hr of traffic traces for Hotmail

20 data centers (Google-like)

Industrial electricity price of each state in May 2010

Compare optimal strategy and min-delay strategy relative to min-load strategy

Optimizing delivery

- Global load balancing traditionally optimizes for server load or distance, not energy
- Route jobs to data centers with
  - min load
  - min distance from client

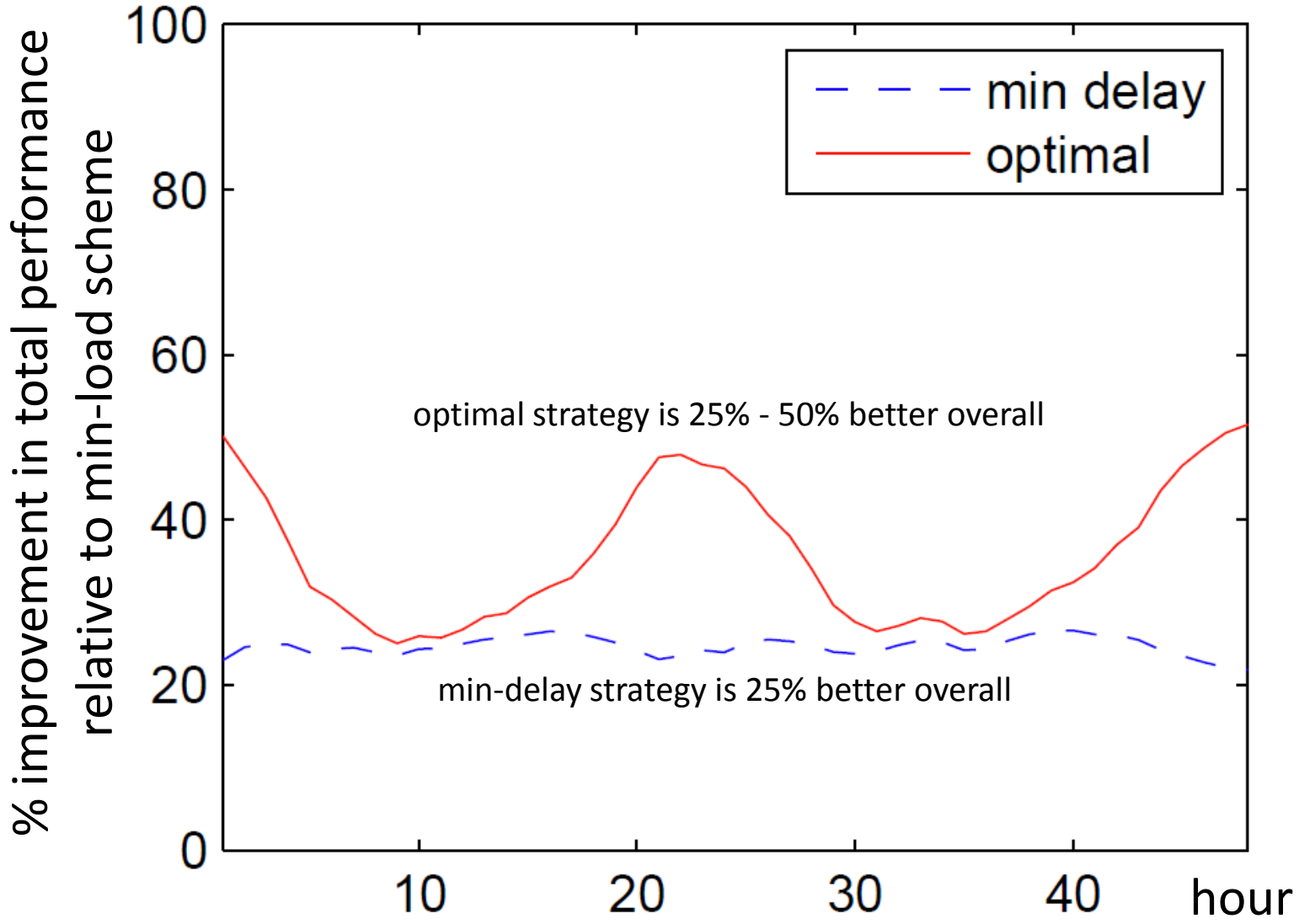
Optimizing delivery

- Our proposal: jointly minimize delay and energy consumption
  - must optimally balance load, distance, electricity cost
- Control actions available
  - Routing: which data center to serve a job/request
  - Server: how many servers to activate (vs in sleep mode)
- Exploit fluctuations in traffic & electricity prices
  - Both fluctuate across time and space
  - Optimally match traffic, requirements, prices

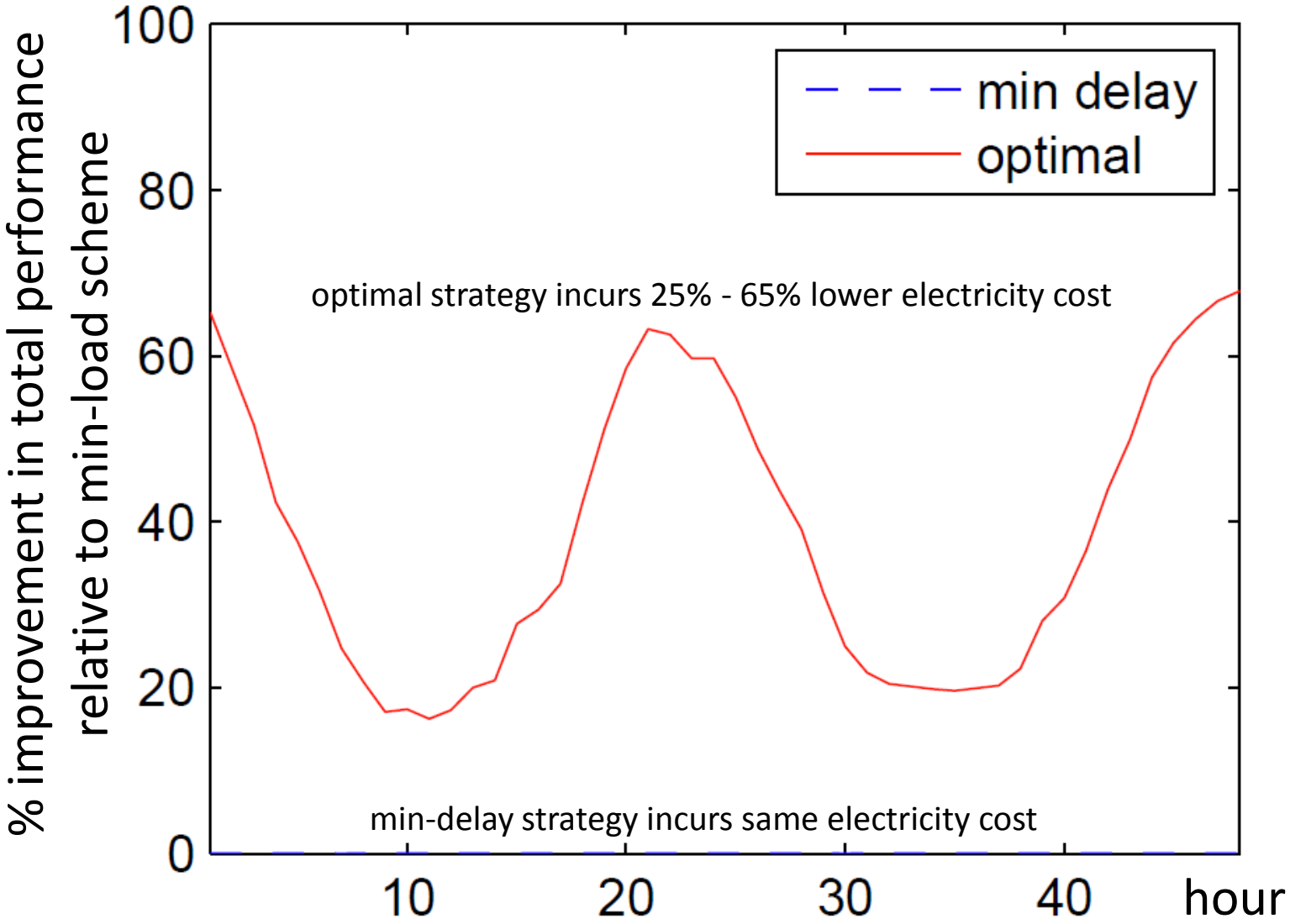
Example: optimizing delivery

Conclusion	Optimal strategy	Min-delay strategy	Min-load strategy
Overall performance	Best	Good	Poor
Electricity cost	Best	Poor	Poor
Delay	Good	Best	Poor

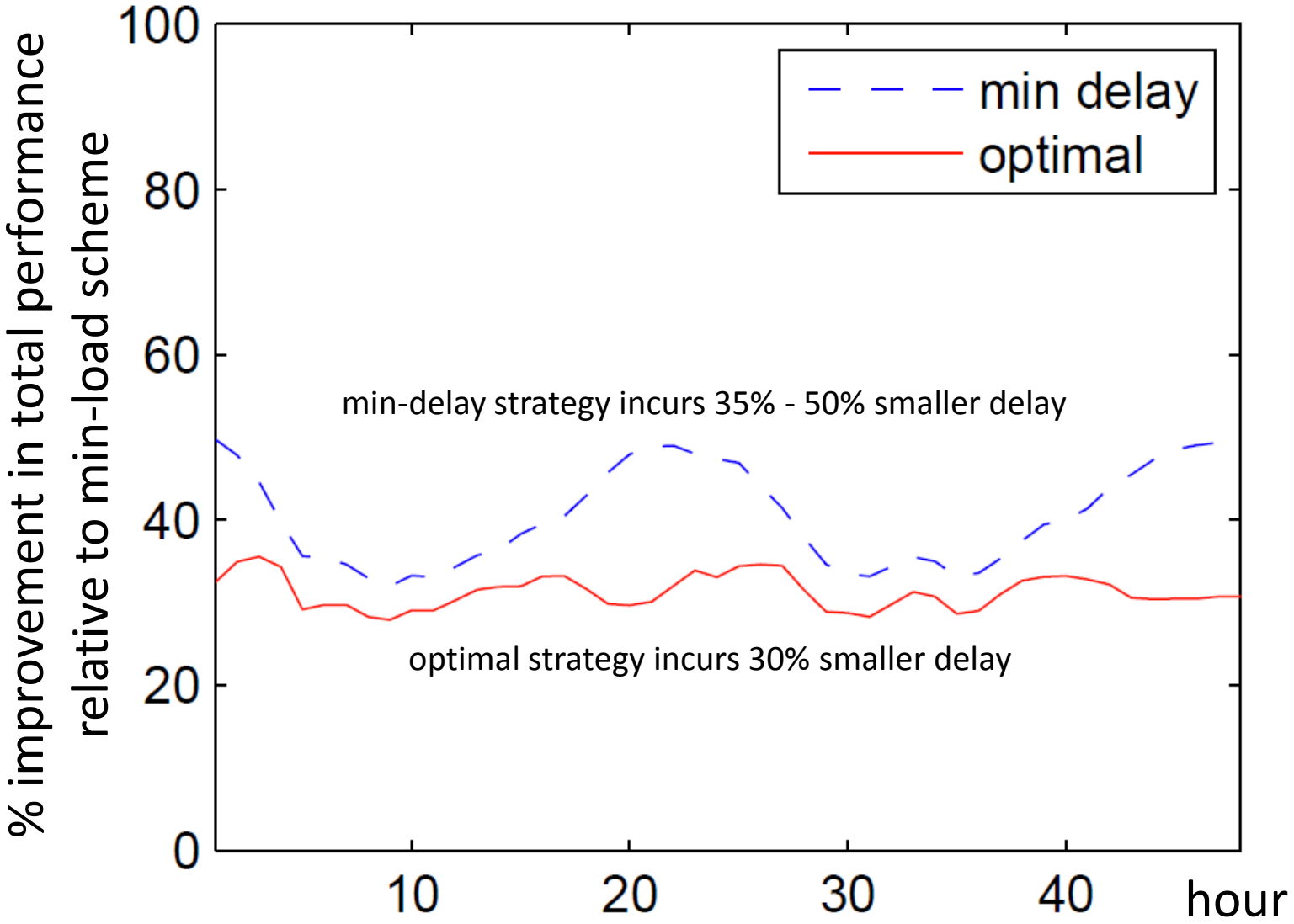
Optimal strategy is 25% to 50% better overall relative to min-load strategy



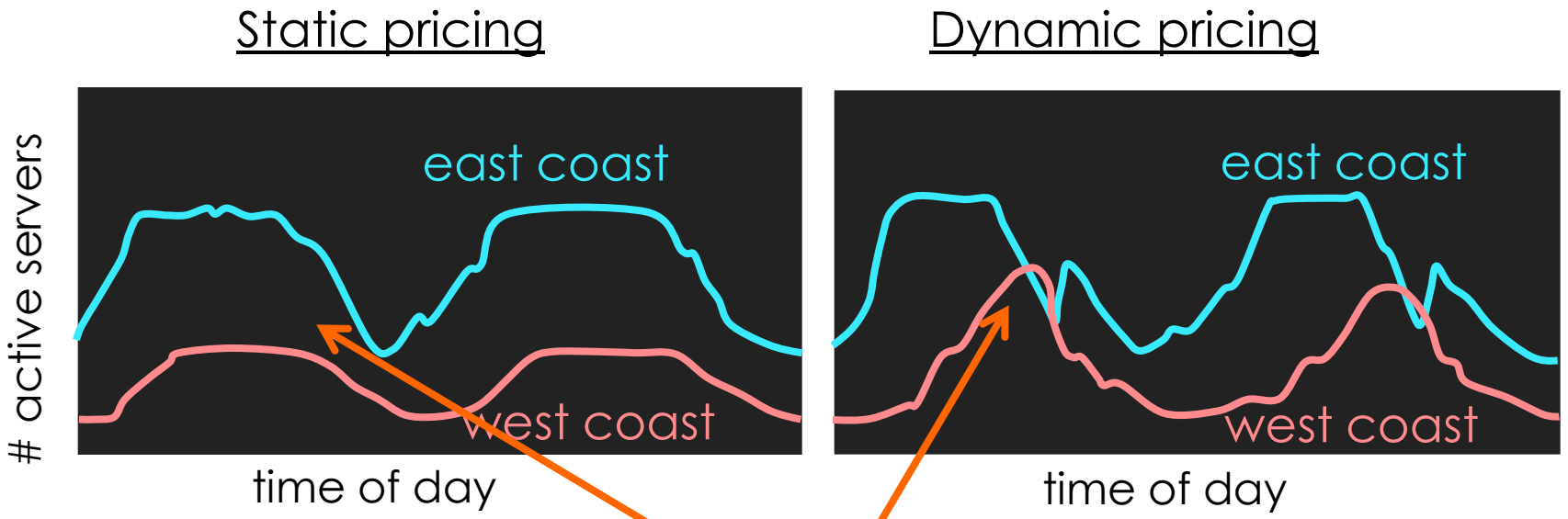
Optimal strategy incurs 25% - 65% lower cost relative to min-load strategy



Optimal strategy incurs 30% smaller delay relative to min-load strategy



"Follow the renewables" routing



Solar available on the west coast

To make this possible requires:

Carefully designed demand-response markets