A few algorithmic issues in data centers

Adam Wierman
Caltech

A significant theory literature on “green” computing has emerged over the last decade…

BUT…

theory has yet to have significant impact in practice.

WHY?
IS IT A PROBLEM?
WHAT IS THE ROLE OF THEORY IN THE FIELD?

Case study: Data Centers

Big improvements in energy efficiency in last few years…

PUE 3+ → < 1.2 now!

This has happened through improvements in
distribution efficiency (DC only, power supplies)
cooling efficiency (run hotter, liquid cooling)

All fall into "engineering better devices",
…not "algorithmic improvements"…why?

Engineering had all the low hanging fruit?

Lack of communication between areas?

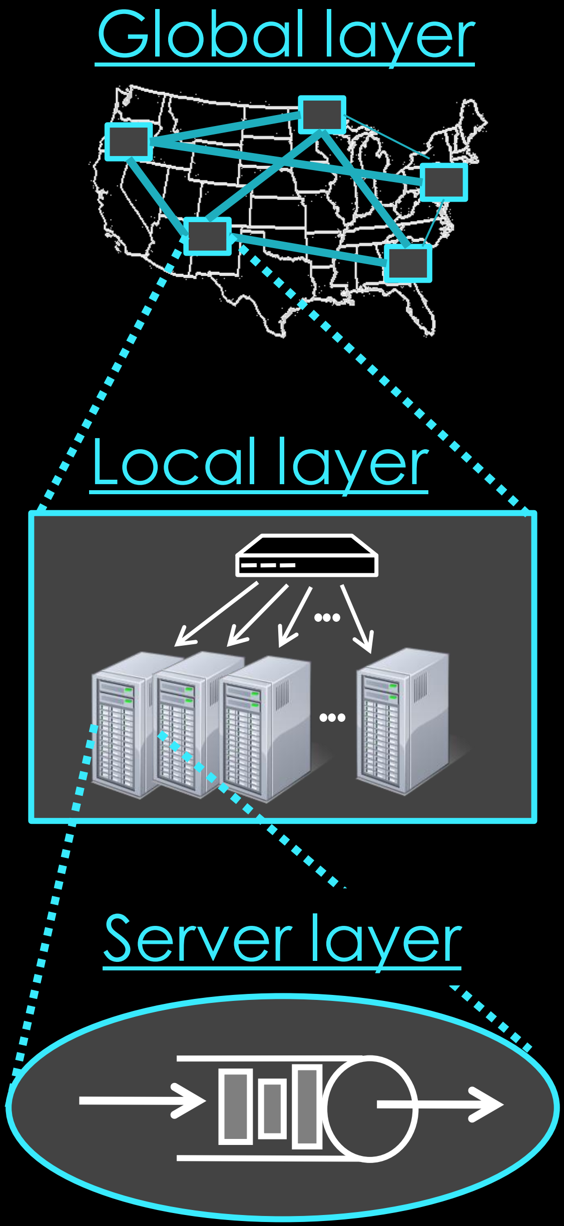
Lack of trust in emerging models
and workload assumptions?

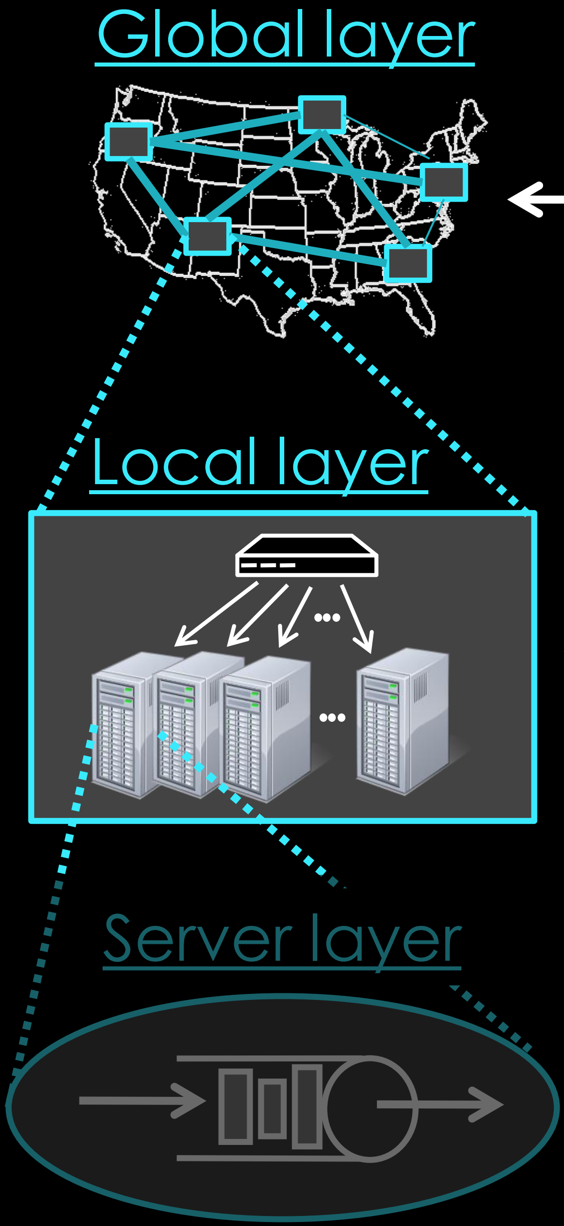
All fall into "engineering better devices",
...not "algorithmic improvements"...why?

Where can algorithmic advances help?

Is there low hanging fruit where we can build trust?

The data center hierarchy

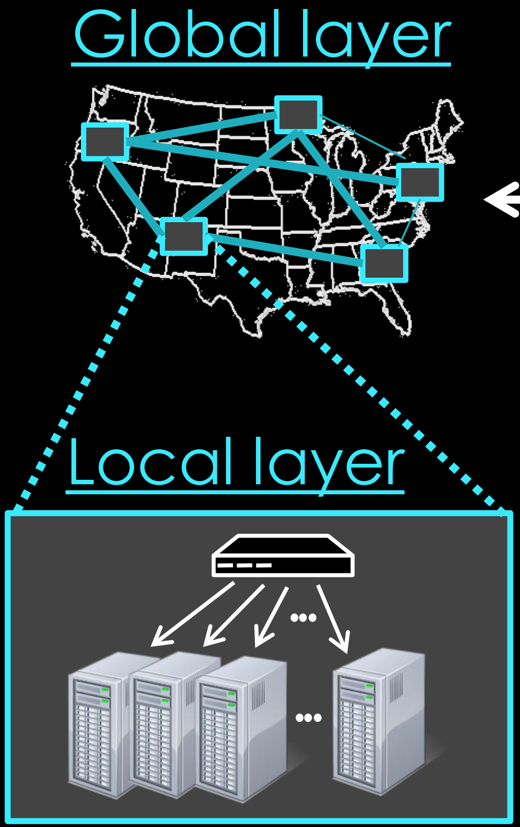




Global load balancing
-- Energy price aware routing
-- Incentives for using renewable sources
-- data placement
-- dynamic capacity provisioning

Local load balancing
-- dynamic capacity provisioning
-- thermal-aware dispatching
-- data placement
-- valley filling & workload mixing

Scheduling & speed scaling
We've done lots of work here!!



Global load balancing

-- Energy price aware routing

-- Incentives for using renewable sources

-- data placement

-- dynamic capacity provisioning

Local load balancing

-- dynamic capacity provisioning

-- thermal-aware dispatching

-- data placement

-- valley filling & workload mixing

Very similar problems…but global must be decentralized

Dynamic capacity provisioning

Goal:

How many servers should stay on at each time,
and how should jobs be dispatched among them?

Issues:

-- Significant costs for turning on/off servers
-- Decisions must be online (or with limited prediction)

Dynamic capacity provisioning

(starter formulation)

# of servers at time t

$$\text{minimize} \quad \sum_{t=1}^{T}\sum_{i=1}^{x_t} f(\boxed{\lambda_{i,t}}) + \beta \sum_{t=1}^{T} (\boxed{x_t} - x_{t-1})^+$$

$$\text{subject to} \quad 0 \le \lambda_{i,t} \le 1 \text{ and } \sum_{i=1}^{x_t} \lambda_{i,t} = \lambda_t,$$

Arrival rate at time t

Operating cost for arrival rate λ (assume convex)

Switching cost

Dynamic capacity provisioning

(starter formulation)

$$\begin{aligned}
\text{minimize} \quad & \sum_{t=1}^{T} \sum_{i=1}^{x_t} f(\lambda_{i,t}) + \beta \sum_{t=1}^{T} (x_t - x_{t-1})^+ \\
\text{subject to} \quad & 0 \le \lambda_{i,t} \le 1 \text{ and } \sum_{i=1}^{x_t} \lambda_{i,t} = \lambda_t,
\end{aligned}$$

Key Features:

-- Flow based not packet based

-- homogeneous servers

-- Ignoring availability, reliability, thermal, …

-- General "cost" model allowed

Dynamic capacity provisioning

(starter formulation)

$$\begin{aligned} &\text{minimize} \quad \sum_{t=1}^{T} \sum_{i=1}^{x_t} f(\lambda_{i,t}) + \beta \sum_{t=1}^{T} (x_t - x_{t-1})^+ \\ &\text{subject to} \quad 0 \le \lambda_{i,t} \le 1 \text{ and } \sum_{i=1}^{x_t} \lambda_{i,t} = \lambda_t, \end{aligned}$$

…we just finished a first cut at this formulation and can give a 3-competitive algorithm

BUT…

Dynamic capacity provisioning

(starter formulation)

$$\text{minimize} \quad \sum_{t=1}^{T}\sum_{i=1}^{x_t} f(\lambda_{i,t}) + \beta \sum_{t=1}^{T} (x_t - x_{t-1})^+$$

$$\text{subject to} \quad 0 \le \lambda_{i,t} \le 1 \text{ and } \sum_{i=1}^{x_t} \lambda_{i,t} = \lambda_t,$$

What is a good way to model the “predicted” arrival rates?
…and then incorporate them into the algorithm?

Dynamic capacity provisioning

(starter formulation)

$$\begin{aligned} &\text{minimize} \quad \sum_{t=1}^{T}\sum_{i=1}^{x_t} f(\lambda_{i,t}) + \beta \sum_{t=1}^{T} (x_t - x_{t-1})^+ \\ &\text{subject to} \quad 0 \le \lambda_{i,t} \le 1 \text{ and } \sum_{i=1}^{x_t} \lambda_{i,t} = \lambda_t, \end{aligned}$$

Add user classes and user specific operating costs and this becomes the global load balancing problem.

How does this change the achievable competitive ratio?

Dynamic capacity provisioning

(starter formulation)

$$\begin{aligned}
&\text{minimize} \quad \sum_{t=1}^{T}\sum_{i=1}^{x_t} f(\lambda_{i,t}) + \beta \sum_{t=1}^{T} (x_t - x_{t-1})^+ \\
&\text{subject to} \quad 0 \le \lambda_{i,t} \le 1 \text{ and } \sum_{i=1}^{x_t} \lambda_{i,t} = \lambda_t ,
\end{aligned}$$

How should reliability and availability constraints be added?

Dynamic capacity provisioning

(starter formulation)

$$\begin{aligned} \text{minimize} \quad & \sum_{t=1}^{T} \sum_{i=1}^{x_t} f(\lambda_{i,t}) + \beta \sum_{t=1}^{T} (x_t - x_{t-1})^+ \\ \text{subject to} \quad & 0 \le \lambda_{i,t} \le 1 \text{ and } \sum_{i=1}^{x_t} \lambda_{i,t} = \lambda_t, \end{aligned}$$

In practice, "valley filling" can be used to adjust λ

-- how can background work (with deadlines) be used for valley-filling (in an online way)?

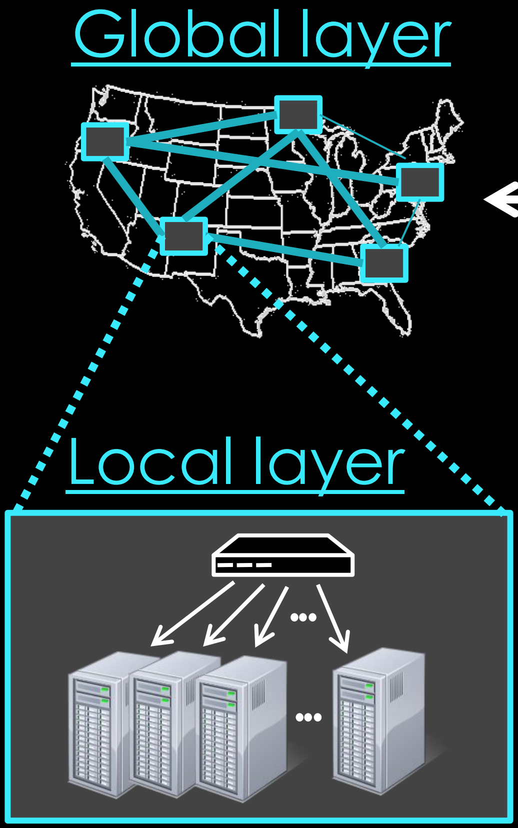
-- how should extra capacity be priced (online)?

Dynamic capacity provisioning
(starter formulation)

$$\text{minimize} \quad \sum_{t=1}^{T} \sum_{i=1}^{x_t} f(\lambda_{i,t}) + \beta \sum_{t=1}^{T} (x_t - x_{t-1})^+$$

$$\text{subject to} \quad 0 \le \lambda_{i,t} \le 1 \text{ and } \sum_{i=1}^{x_t} \lambda_{i,t} = \lambda_t,$$

What about considering cap & trade or other mechanisms for incentivizing green energy?



Global load balancing

-- Energy price aware routing
-- Incentives for using renewable sources
-- data placement
--dynamic capacity provisioning

Local load balancing

-- dynamic capacity provisioning
-- thermal-aware dispatching
-- data placement
-- valley filling & workload mixing

Very similar problems…but global must be decentralized

A few algorithmic issues in data centers

Adam Wierman
Caltech