# Cooling-Aware Green Geographical Load Balancing Visualized

Michael Hirshleifer  Yizhen Wang
Zhenhua Liu  Adam Wierman

California Institute of Technology
1200 E California Blvd
Pasadena, CA 91106

2012-10-20

# Acknowledgements

We are deeply grateful to:

- Professor Adam Wierman for insightful and patient guidance.
- The NSF, Elliot family, and Rose Hills Foundation for financial support.

# Outline

# Table of Contents

# Motivation

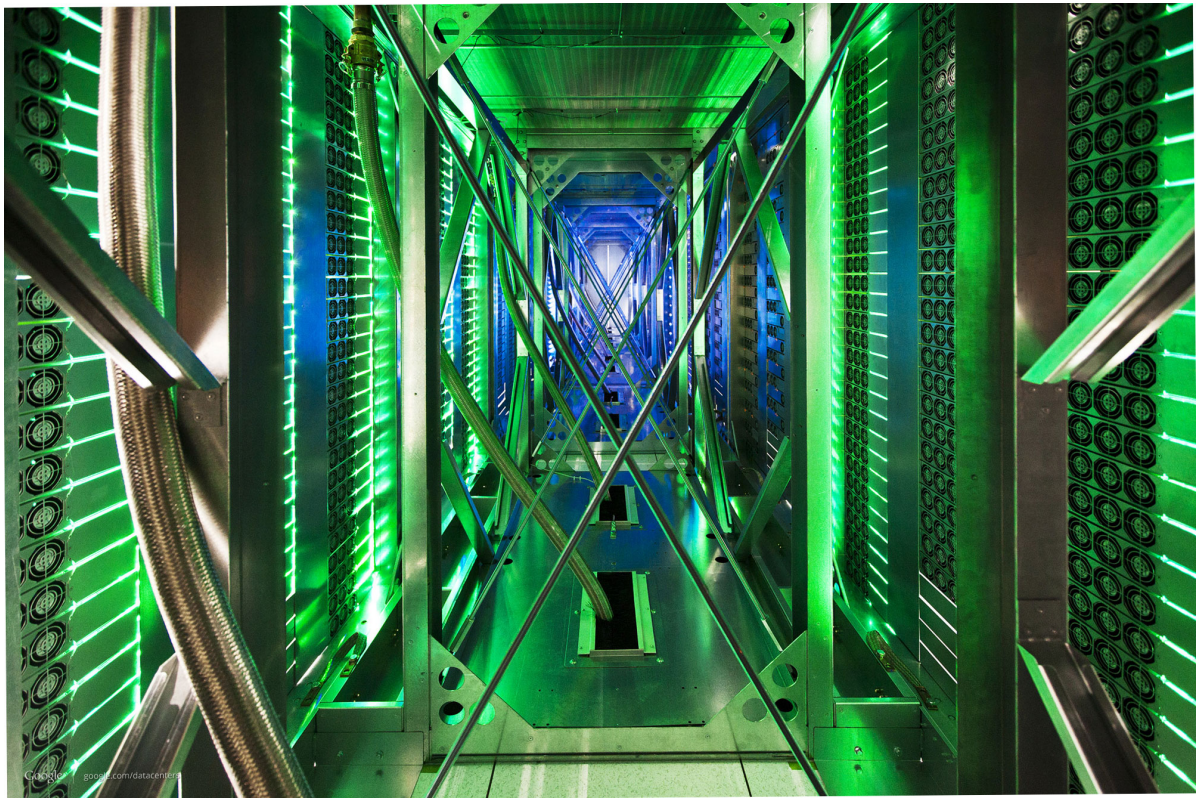
## Vast, high-growth energy use by data centers.

"Data Centers Waste Vast Amounts of Energy, Belying Industry Image"

—*New York Times*, 2012-09-22

# Motivation

## How can we make them ‘green’?



“The Green Data Center Market Will Grow from $17.6 Billion to $45 Billion by 2016, Forecasts Pike Research”

—*Wall Street Journal*, 2012-09-14

# Motivation, contd.

## DCs use a *lot* of power

Substantial fraction of world non-green energy production.

- Expensive
- Dirty (brown energy)

## DC servers' energy demand

- To operate and **cool**
- Variable
  - Over time (e.g. cooling more costly in summer)
  - Cooling costs vary geographically.
- Some DCs make use of green energy (solar, wind).

# Motivation, contd.

## Use renewable energy!

Figure: Solar and wind power

# Motivation, contd.

## Renewable generation already being built

- Companies like Apple, HP, Facebook have already installed renewables
  - HP: 100% wind-cooled DC in England (2010)
  - Apple: North Carolina DC, 60% solar (2012)
  - Facebook: Open Compute Project
- DCs have dedicated solar or wind facilities

# Motivation, contd.

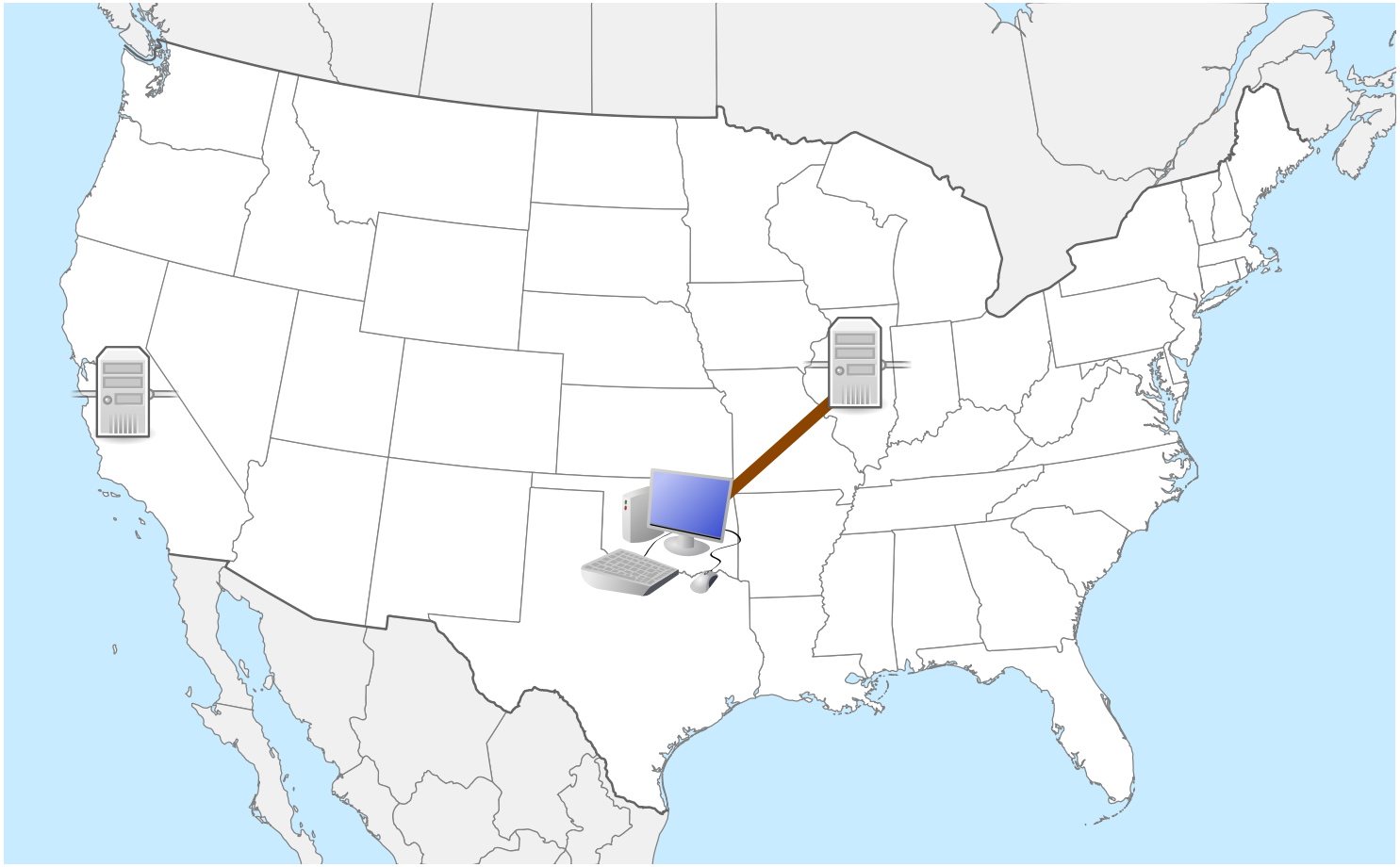
## Problem: Green energy highly variable

- Time of day, cloud cover, wind speed.
- Does not match user demand.
- Very expensive to store.

Must still buy brown energy off grid
$\Rightarrow$ expensive; pollutes.

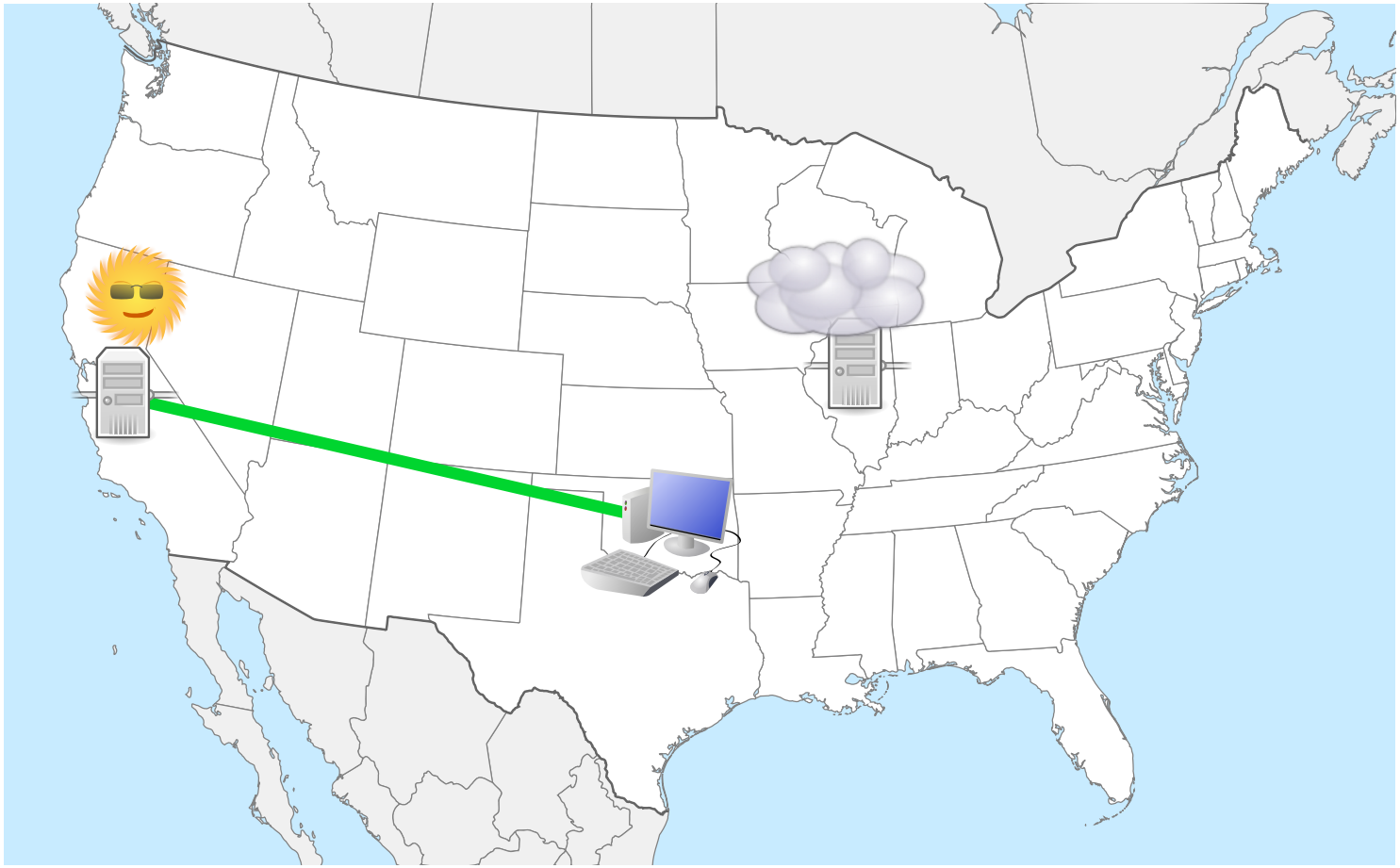
# Our approach: Geographical Load Balancing

Figure: **Traditional routing**: route request to nearest DC, which minimizes latency



# Our approach: Geographical Load Balancing

Figure: **GLB** (Liu et al., 2011): prefer routing to DCs that currently have renewables available



# Summary of our project

## Research goal

Develop software to show how geographical load balancing (GLB) improves data center (DC) efficient use of renewable energy.

## Issue

Minimize DC electricity cost over time by convex optimization over real workload inputs - temp, solar, & wind traces.

## Our contributions

- Visualization
  Animation of 10 DCs' demand and usage of energy (grid & renewable energy generation) in 48 contiguous U.S. states
- Test effectiveness of, and refine, routing algorithms using visualization
- Cooling-aware GLB model
  Uses locally available renewable energy more effectively
  Reduces electricity grid usage despite weather shifts
  Potentially uses renewables almost exclusively

# The cooling-aware GLB model

- Introduces cooling costs into GLB algorithm
  - Better-informed decisions
- Use actual data center workload, weather, renewable energy (wind, solar) availability and grid electricity prices
  - 10 Data Center locations serving population centers in 48 contiguous U.S. states
- Numerically compute the optimal solution.
  - Determine economic, environmental impact
- Implement visualization of simulation solution
  1. Easier to see what each optimization algorithm is doing.
  2. Can test effectiveness of, and refine, routing algorithms.
  3. Facilitate development of an implementable algorithm.
- Compare performance using GLB vs. energy storage

# The cooling-aware GLB model, contd.

## Key insights and results

- Significantly lower total energy cost, $\mathrm{CO}_2$ emission halved.
  - Distributes requests to locations with cheap energy, favorable weather for cooling
- Cost of running data center fairly constant across seasons, weather, diurnal temp. variation
- Performance of GLB comparable to using storage, while cost is much lower

# Table of Contents

# Model setup

Modify Liu et al.'s (2011) model to include cooling costs.

## Key factors

### IT Load

### Supply of renewables at different DCs

### Storage

### Costs

- Total Cost = Energy Cost + Propagation & Queuing Delay Costs + Switch Cost
- **New: Cooling costs**
- 2 Cooling options - Air, Chilled-Water

# Assumptions

## IT Load $L$ in each state

- Obtain base load from real-world traces (HP Labs).
- Scale by state population.
- Shift by time zone.

## Data centers

- 1 Google data center in each of 10 states.
- Capacity limited by finite # servers in each DC.

## Renewable availability

- DCs use combination of green & brown energy sources.
  - Use avg mix of 80% of wind and 20% solar (Liu et al. 2011).
- Proportional to wind speed & solar irradiance; from http://wind.nrel.gov/.
- Fluctuates with season, sunrise/sunset time
- Capacity $= c\times$ national aggregate demand under full load

# Cost factors

Optimization problem: Choose routing plan and # active servers to minimize total cost (= delay cost + switching cost + energy cost).

## Delay cost

Business cost of request latency (distance traveled, queuing delay at DC)

- Calculated using a sharing queue model.

## Switching cost

Delay, wear-and-tear cost from powering servers on/off.

## Energy cost

# Cost factors

**Optimization problem:** Choose routing plan and # active servers to minimize total cost (= delay cost + switching cost + energy cost).

## Delay cost

## Switching cost

## Energy cost

Cost of the energy used to power the DC.

- Demand
  - **IT cost** running active servers (CPU, disk, ...)
  - cooling
- Supply
  - **renewable** free
  - grid depends on market price in each state, amount used

# Cooling costs

- Part of energy cost (demand)
- Air cooling cheaper when outside air is cold
- Water cooling (relatively) cheaper when outside air hot.
- Cost of chilled-water cooling proportional to processing load (active servers) cooled with water.
- Air cooling cost rises nonlinearly with processing load.
- So DC chooses amount of air- vs. water-cooling, as a function of total processing load, to minimize cost (convex optimization).

# Minimize total costs using convex optimization

$$\underset{\mathbf{x(t)},\lambda(\mathbf{t})}{\text{minimize}} \quad \sum_{i\in\mathcal{N}} p_i \cdot (l(x_i(t)) + c(x_i(t)) - r_i(t) - e_i(t))^+ + \sum_{j\in\mathcal{J}}\sum_{i\in\mathcal{N}} \lambda_{ij}(t)\left(\frac{1}{\mu_i - \lambda_i(t)/x_i(t)} + d_{ij}\right) + \beta \cdot (x_i(t+1) - x_i(t))^+$$

subject to

$$\sum_{i\in\mathcal{N}} \lambda_{ij}(t) = L_j(t), \qquad \forall j \in J$$

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

$$0 \leq x_i(t) \leq X_i, \qquad \forall i \in N$$

$$\lambda_i(t) \leq x_i(t) \cdot \mu_i \qquad \forall i \in N$$

$$0 \leq es_i(t) \leq ES_i \qquad \forall i \in N$$

$$e_i(t) = es_i(t) - es_i(t+1) \qquad \forall i \in N$$

# Minimize total costs using convex optimization

$$\underset{\mathbf{x(t)},\lambda(\mathbf{t})}{\text{minimize}} \quad \overbrace{\sum_{i\in\mathcal{N}} p_i \cdot (l(x_i(t)) + c(x_i(t)) - r_i(t) - e_i(t))^+}^{\textbf{energy cost}}$$

$$+ \overbrace{\sum_{j\in\mathcal{J}}\sum_{i\in\mathcal{N}} \lambda_{ij}(t)\left(\frac{1}{\mu_i - \lambda_i(t)/x_i(t)} + d_{ij}\right)}^{\textbf{delay cost}}$$

$$+ \overbrace{\beta \cdot (x_i(t+1) - x_i(t))^+}^{\textbf{switching cost}}$$

Boundary conditions model real-world constraints:

- GLB system processes all requests. Each DC gets $> 0$ work.
- Each DC has finite server capacity.
- Each DC's energy storage capacity bounded.

# Modeling the carbon emission

After we find the optimal solution based on cost, we want to know how much the $\mathrm{CO}_2$ emission is. This can be represented as

$$\sum_{i \in \mathcal{N}} \eta_i \cdot (x_i(t) + c(x_i(t)) - r_i(t) - e_i(t))^+$$

where $\eta_i$ be the rate of carbon emission per kW/h at data center $i$.

# Table of Contents

# Motivation for visualization

Numerical simulation result is very large time series describing routing plans and data center activity.

Hard to interpret large quantity of data in matrix form.

Visualization:

- Key characteristics of the optimal solution salient.
- Acts as simple check for validity of solution
- An obvious way to spot abnormal behaviors in the output.

# Technical description

## Scalable Vector Graphics (SVG) format

High-quality XML-based vector graphics format supports fluid animations displaying time-series data.
Animation can be embedded into a web page.

## Wrapper XHTML web page

Implemented using Javascript
User-friendly, can control animation (play, pause, or seek) using web interface
Possible enhancements could include zooming, panning, and showing/hiding types of map elements.

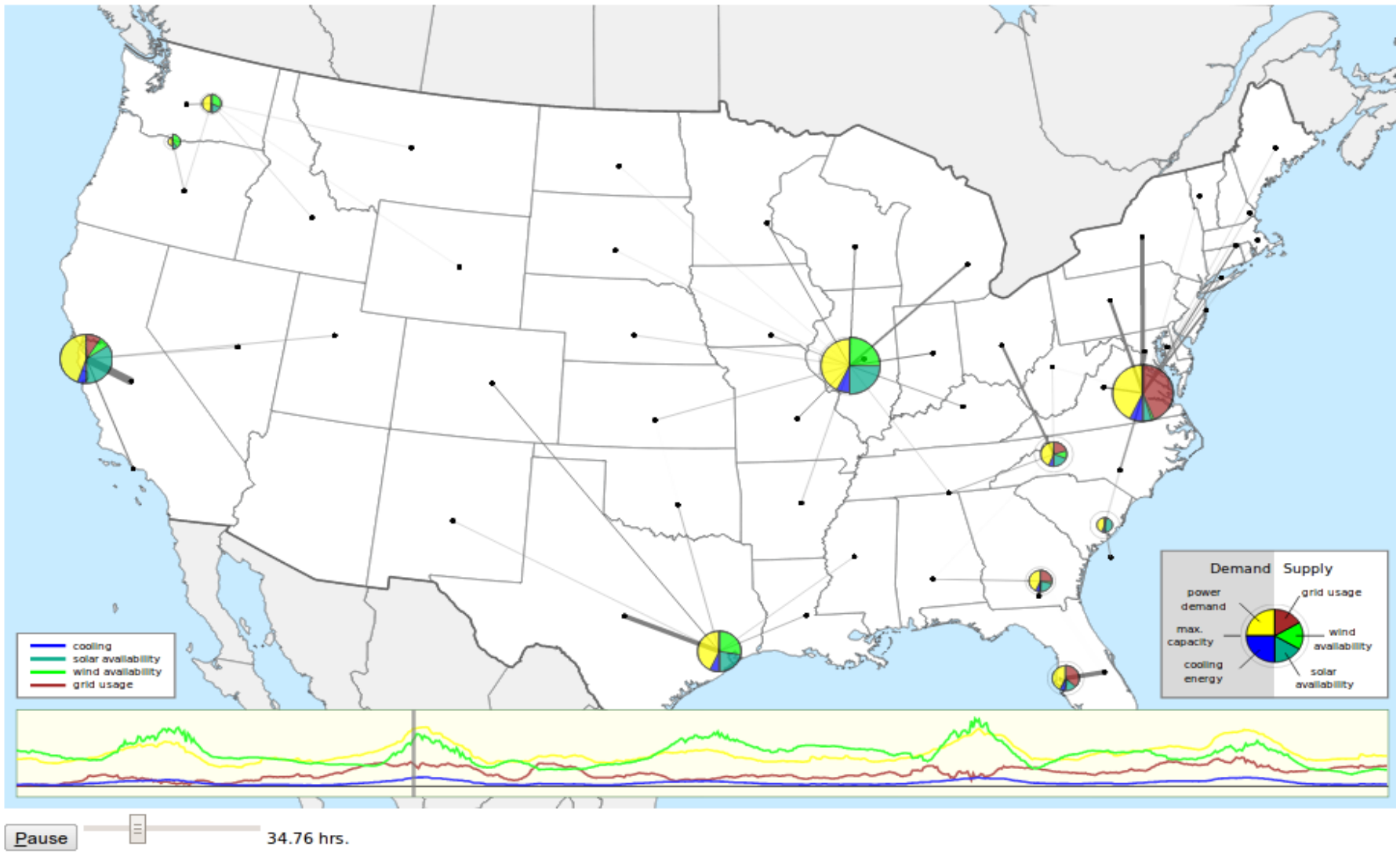
## Both components work in the Chromium web browser and pass the W3C validators

# Technical description, contd.

## Back-end script generating SVG map from input data

Written in Scala programming language.
Script reads in and displays DC center locations, client (request source) locations, real solar- and wind-generation traces.
Routings optimized using various algorithms (from Matlab optimization outputs).
Raw input data in CSV format.
Matlab outputs exported to NetCDF format for reading into Scala.

# Data center visualization



# Table of Contents

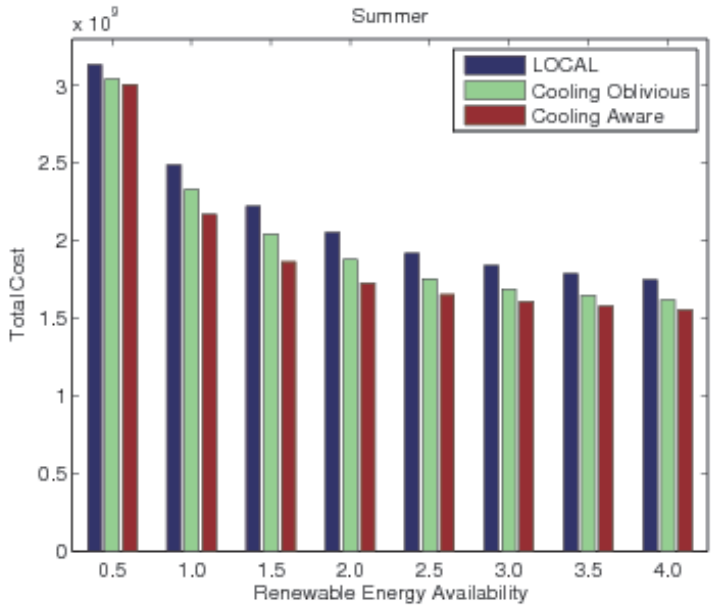
# Experiment Result

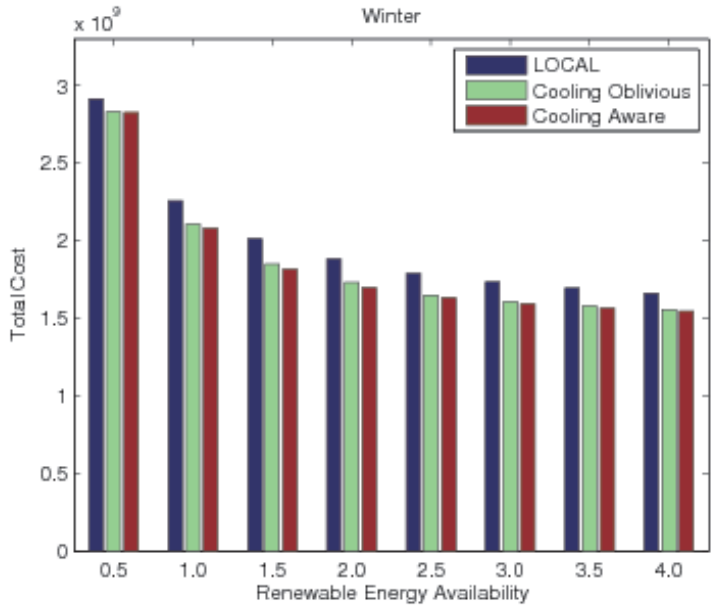
## We want to explore

- How much is the cost saving using cooling aware GLB?
- Does this result hold in all seasons?
- How much carbon emission are we cutting?
- Is our model better than its alternative, using storage for electricity?

# Experiment Result

## Cost saving



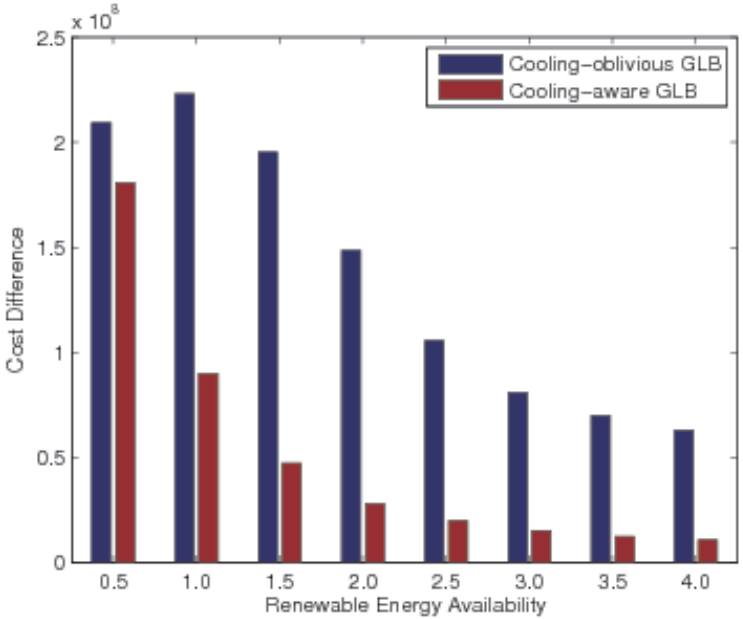




## Observation

- Using cooling aware GLB model causes less cost than using both the old GLB and the LOCAL.
- The cost edge compared to the old GLB is evident when aggregate renewable supply is in the reasonable range.

# Experiment Result

## The effect of seasonality

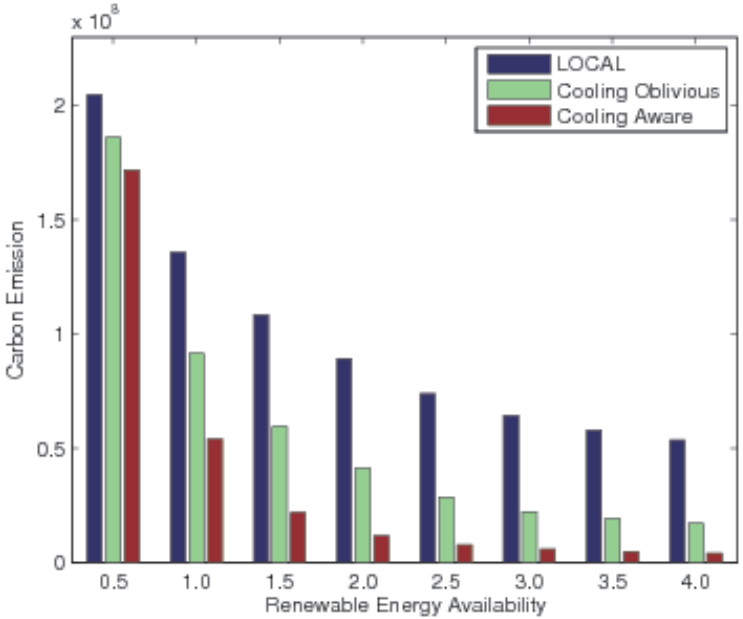



## Observation

- The cost difference between in winter and in summer is significantly reduced if we have enough renewable supply. The renewable supply doesn't have to be in vast surplus.
- The cost saving effect is robust against seasonality.

# Experiment Result

## Carbon Emission




## Observation

- The carbon emission saving is much more significant than the cost saving
- Achieving such effect also only requires reasonable amount of renewables.

# Experiment Result

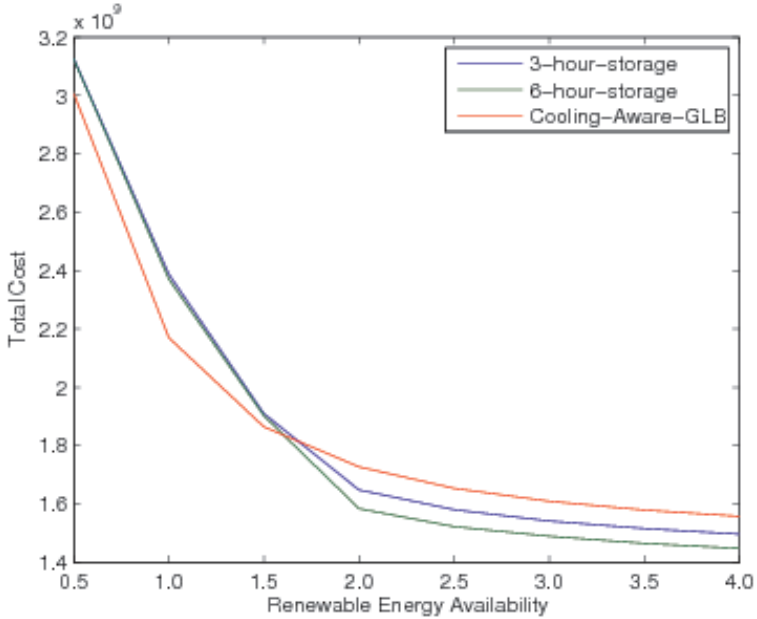
## GLB vs Storage

We want to compare the performance of cooling aware GLB and storage model in both economic and environmental aspect.

- We compare both the *cost* and the *brown energy* usage.
- A set of benchmarks of different storage capacity is chosen to fully represent the power of storage model.

# Experiment Result

## GLB vs Storage on cost

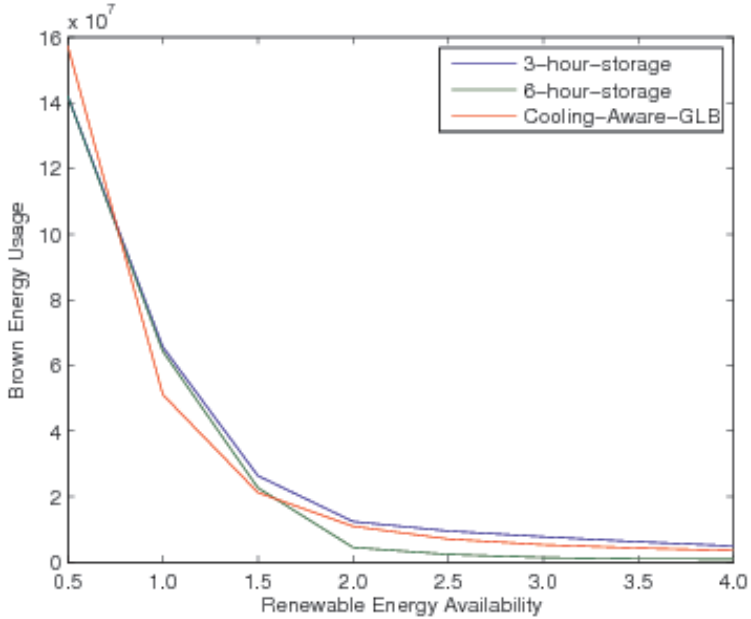



## Observation

- GLB initially has cost advantage when renewable supply is not in vast surplus.
- Storage model catches up when the renewable energy supply is large.

# Experiment Result

## GLB vs Storage on brown energy usage




## Observation

- GLB is better than the storage when the renewable supply is in $[1, 1.5]$ times the demand.
- Achieves similar result to the storage model with much less prior investment.

# Table of Contents

# Summary of our project

## Research goal

Develop software to show how geographical load balancing (GLB) improves data center (DC) efficient use of renewable energy.

## Issue

Minimize DC electricity cost over time by convex optimization over real workload inputs - temp, solar, & wind traces.

## Our contributions

- Visualization
  Animation of 10 DCs' demand and usage of energy (grid & renewable energy generation) in 48 contiguous U.S. states
- Test effectiveness of, and refine, routing algorithms using visualization
- Cooling-aware GLB model
  Uses locally available renewable energy more effectively
  Reduces electricity grid usage despite weather shifts
  Potentially uses renewables almost exclusively

# Future work

Extend simulation and visualization software to:

- Incorporate other kinds of renewables - tailor to local renewables & storage systems.
- Endogenize data centers - Build new ones?
- Optimize timing of switching routes - currently assume algorithm switches routing at fixed time intervals.
- Examine dynamic grid price.
- Allow DCs to sell excess renewable energy back to grid.

# Play with the visualization

- `http://geographicalloadbalancing.github.com/`
- Working visualization (use Google Chromium)
- Source code for visualization, draft paper also linked