

Intelligent Placement of Datacenters for Internet Services

Íñigo Goiri^{†‡}, Kien Le[‡], Jordi Guitart[†], Jordi Torres[†], and Ricardo Bianchini[‡]

[†]Universitat Politècnica de Catalunya
Barcelona Supercomputing Center
Barcelona, Spain
{igoiri,jguitart,torres}@ac.upc.edu

[‡]Department of Computer Science
Rutgers University
Piscataway, NJ USA
{lekien,ricardob}@cs.rutgers.edu

Abstract—Popular Internet services are hosted by multiple geographically distributed datacenters. The location of the datacenters has a direct impact on the services' response times, capital and operational costs, and (indirect) carbon dioxide emissions. Selecting a location involves many important considerations, including its proximity to population centers, power plants, and network backbones; the source of the electricity in the region; the electricity, land, and water prices at the location; and the average temperatures at the location. As there can be many potential locations and many issues to consider for each of them, the selection process can be extremely involved and time-consuming.

In this paper, we focus on the selection process and its automation. Specifically, we propose a framework that formalizes the process as a non-linear cost optimization problem, and approaches for solving the problem. Based on the framework, we characterize areas across the United States as potential locations for datacenters, and delve deeper into seven interesting locations. Using the framework and our solution approaches, we illustrate the selection tradeoffs by quantifying the minimum cost of (1) achieving different response times, availability levels, and consistency times, and (2) restricting services to green energy and chiller-less datacenters. Among other interesting results, we demonstrate that the intelligent placement of datacenters can save millions of dollars under a variety of conditions. We also demonstrate that the selection process is most efficient and accurate when it uses a novel combination of linear programming and simulated annealing.

I. INTRODUCTION

Today, popular Internet companies, such as Google, Yahoo, and Microsoft offer a range of services to millions of users every day. These services are hosted in datacenters that contain thousands of servers, as well as power delivery (and backup), cooling, and networking infrastructures.

Because users demand high availability and low response times, each service is mirrored by multiple datacenters that are geographically distributed. Each datacenter is supposed to serve the requests of the users that are closest (in terms of network latency) to them. If this datacenter becomes unavailable or unreachable, these requests are forwarded to a mirror datacenter.

Although the users' experience is an overriding concern, Internet companies must also consider the enormous cost of provisioning and operating such a network of datacenters. The capital costs include land acquisition, datacenter construction, and bringing enough network bandwidth and

electricity to the datacenter. The operational costs include electricity, water (for the cooling system), and system administration staff. Moreover, companies are starting to consider the environmental impact of their infrastructures. In particular, a datacenter may have a significant impact, depending on the type of electricity (renewable vs. CO_2 -intensive) it consumes.

An interesting aspect of these response time, availability, cost, and environmental concerns is that many of them depend heavily on the specific locations of datacenters. For example, land, electricity, and water prices, as well as network latency, depend directly on location. Even more interestingly, the location-dependent costs are not always directly related. For example, land is cheap in a desert area but cooling is expensive.

Given the pervasive impact of the datacenters' locations, companies must intelligently select them. As there can be many potential locations for each datacenter and many issues to consider in evaluating them, the selection process can be involved and time-consuming. Unfortunately, because competition between services is fierce, no information is available in the public domain about how Internet companies actually select locations for their datacenters.

Thus, in this paper we study the selection process, while fully characterizing the different parts of the US as potential locations for datacenters. First, we propose a framework for selection that includes parameters representing all aspects of datacenter costs, response times, data consistency, and availability. The framework allows us to define the selection process as a non-linear cost minimization problem with response time, consistency, and availability as constraints.

Second, we propose approaches for solving the problem efficiently. Some of our approaches transform the problem into a collection of linear problems, whereas others use linear programming only in an auxiliary role.

Third, we characterize areas across the entire US according to the parameters of our framework. The characterization includes each area's population, average temperature, electricity prices, average datacenter Power Usage Efficiency (or simply PUE), and proximity to network backbones and power sources (plants or transmission lines), and the CO_2 intensity of the area's power generation. We then delve deeper into seven interesting areas as potential locations

for a datacenter: Austin (Texas), Bismarck (North Dakota), Los Angeles (California), New York (New York), Orlando (Florida), Seattle (Washington), and St. Louis (Missouri).

Based on our framework, solution approaches, and characterization data, we built a tool for selecting datacenter locations. The tool is extensible so that new parameters and constraints can be added. The tool can be used to select locations for an entire network of datacenters or to extend an existing network with one or more datacenters.¹

Our evaluation uses the tool to compare the efficiency and accuracy of approaches for solving the optimization problem. We also identify the most important optimization criteria, and illustrate the selection tradeoffs by quantifying the minimum cost of (1) achieving different response times, availability levels, and consistency times, and (2) restricting services to green energy and chiller-less datacenters.

Our results demonstrate that the intelligent placement of datacenters can save millions of dollars under a variety of conditions. They also demonstrate that the selection process is most efficient and accurate when it uses a novel combination of linear programming and simulated annealing. Finally, among other interesting results, we show that the cost of a network of datacenters almost doubles when the maximum acceptable response time is decreased from 50ms to 35ms.

In summary, our contributions are:

- We propose a framework and optimization problem for selecting datacenter locations;
- We propose solution approaches for the problem;
- Using extensive real data, we characterize areas across the US as potential locations for datacenters; and
- We answer many interesting quantification and tradeoff questions about datacenter networks.

As far as we know, no previous work has considered the selection of locations for the datacenters of an Internet service in detail.

The remainder of this paper is organized as follows. The next section describes our framework, optimization problem, and solution approaches. Section III describes our tool, highlighting how we instantiate the components of the framework using real cost and location data. Section IV characterizes the US using our real data and optimization machinery. Section V evaluates our solution approaches and assess the importance of the different optimization criteria. Section VI answers questions about datacenter network design and costs. Section VII presents the related work. Finally, Section VIII draws our conclusions.

II. FRAMEWORK FOR PLACEMENT

Our main goal is to efficiently select locations for one or more datacenters, so that the overall cost is minimized and the service's response time, consistency, and availability

requirements are respected. To that end, the next subsection defines the most important parameters in the selection process. Based on these parameters, we then formulate a cost model and optimization problem. The last subsection proposes approaches for solving the problem.

A. Parameters

Table I lists the entire set of parameters in our framework. They range from inputs provided by the user (MaxS, ratioServerUser, MAXLAT, MAXDELAY, and MINAVAIL) to parameters that we seek to instantiate via optimization (S_d , P_c^d , SB_d , and PB_c^d). Among the more interesting parameters are those related to costs, temperatures, power and energy consumptions. Next, we discuss those related to costs in detail.

Costs. The overall cost of a network of datacenters can be broken down into capital (CAPEX) and operational (OPEX) components. The CAPEX costs are those investments made upfront and depreciated over the lifetime of the datacenters. CAPEX can be further divided into capital costs that are independent of the number of servers to be hosted (CAP_ind), those that do depend on the *maximum* number of servers that can be hosted (CAP_max), and those that depend on the *actual* number of hosted servers (CAP_act). (The maximum and actual number of servers may differ, because a datacenter may not be fully populated right from the start of its operation or because some servers are kept off during operation.)

The CAP_ind costs relate to bringing electricity and external networking to the datacenters. (Although the amount of electricity and external bandwidth depends on the number of servers, the base cost of laying out any transmission line or optical fiber dominates.) These costs vary according to location. They can be estimated from the distance between the location and (1) the closest transmission line or power plant; and (2) the closest network backbone.

The CAP_max costs relate to land acquisition, datacenter construction, and purchasing and installing the power delivery, backup, and cooling infrastructures. The land price varies according to location, whereas the other costs do not to a first approximation. All of these costs depend on the level of redundancy that will be built into each datacenter. The construction cost is typically estimated as a function of the maximum power to be consumed by the datacenter. This maximum power is that required by the maximum number of servers (and networking gear) when running at 100% utilization times the maximum expected PUE of the datacenter. The PUE is computed by dividing the overall power consumption by the power consumption of the computational equipment. The PUE is higher when temperature and/or humidity are high, since cooling consumes more energy under those conditions.

CAP_act accounts for the expenses with purchasing the

¹More information about our tool, including a short video illustrating its use, can be found at <http://www.darklab.rutgers.edu>.

Symbol	Meaning	Unit	Symbol	Meaning	Unit
S_d	Max #servers in DC d	serv	SB_d	Is a DC placed in location d ?	bool
P_c^d	Actual #servers of DC d providing c	serv	PB_c^d	Is a DC d providing c ?	bool
$MaxS$	Max #servers across all DCs	serv	$MAXLAT$	Max network latency	ms
$MINAVAIL$	Min availability	%	$MAXDELAY$	Max consistency delay	ms
$latency(p_1, p_2)$	Latency between p_1 and p_2	ms	$distance(p_1, p_2)$	Distance between p_1 and p_2	mile
$closerPower(d)$	Closer power source (grid or plant)	coord	$closerNetwork(d)$	Closer network backbone	coord
$emissions(d)$	CO_2 emissions at location d	g/KWh	$population(c)$	Population of center c	user
$priceLand(d)$	Land price at location d	\$/sf	$priceWater(d)$	Water price at location d	\$/gal
$priceEnergy(d)$	Energy price at location d	\$/Wh	$maxTemp(d)$	Max temperature at location d	$^{\circ}C$
$avgTemp(d)$	Avg temperature at location d	$^{\circ}C$	$powerServer(u)$	Power at u util. (incl. network)	W/serv
$priceServerNet$	Price including internal networking	\$/serv	$servers(c)$	$population(c) \cdot ratioServerUser$	serv
$ratioServerUser$	Servers required per user	serv/user	$priceLinePower$	Price to layout power line	\$/mile
$priceLineNet$	Price to layout optical fiber	\$/mile	$pueDC(t)$	PUE of DC at temperature t	
$priceBuild(power)$	Price of building a DC of $power$ W	\$/W	$waterDC$	Water required	gal/Wh
$landDC$	Land required	sf/W	$adminSalary$	Administrator salary	\$/adm
$maintDC$	Cost to maintain DC	\$/W	$priceNetwork$	Price of net bandwidth	\$/Mbps
$adminServer$	Administrators required	adm/serv			
$networkServer$	External net bandwidth required	Mbps/serv			
T	Operation period	h			
$maxPower(d)$	$powerServer(1) \cdot pue(maxTemp(d))$		$energy(u, d)$	$T \cdot powerServer(u) \cdot pue(avgTemp(d))$	
$buildCost(d)$	$maxPower(d) \cdot priceBuild(maxPower(d) \cdot S_d)$		$landCost(d)$	$maxPower(d) \cdot landDC \cdot priceLand(d)$	
CAP_act	$priceServerNet$				
$CAP_max(d)$	$landCost(d) + buildCost(d)$				
$CAP_ind(d)$	$priceLinePower \cdot distance(closerPower(d), d) + priceLineNet \cdot distance(closerNetwork(d), d)$				
$OP_act(d)$	$maxPower(d) \cdot maintDC + adminServer \cdot adminSalary + networkServer \cdot priceNetwork$				
$OP_utl(u, d)$	$energy(u, d) \cdot priceEnergy(d) + energy(u, d) \cdot waterDC \cdot priceWater(d)$				

Table I

SUMMARY OF FRAMEWORK PARAMETERS. EACH LOCATION d BELONGS TO THE SET D OF POSSIBLE LOCATIONS, WHEREAS EACH POPULATION CENTER c BELONGS TO THE SET C OF POPULATION CENTERS.

servers and (internal) networking gear that will be hosted by the datacenters. These costs are proportional to the actual number of hosted servers and do not depend on location.

The OPEX costs are those incurred during the operation of the datacenters. OPEX can be divided into costs relating to the actual number of hosted servers (OP_act) and those that also depend on the utilization of those servers (OP_utl).

OP_act relates to the maintenance and administration of the equipment, as well as their external network bandwidth use. Maintenance and administration costs are dominated by staff compensation. OP_act costs depend on location.

OP_utl represents the electricity and water costs involved in running the servers at a certain level of utilization. These costs can be computed based on the servers' energy proportionality [1] and average utilization (typically around 30% [1]), the average PUE, and the electricity and water prices. These costs vary with location.

Finally, *lower taxes and one-time incentives* are another important component of the cost of a datacenter. For example, some states lower taxes on datacenters, as they generate employment and wealth around them. This component depends on the nature of the savings and applies to each cost component in a different way. Although we do not consider this component further, it is easy to include it in our framework.

Response time. As mentioned above, Internet services must exhibit low response times to remain competitive. Thus, in selecting locations, it is critical to model the network latency between the service's potential users and the potential locations. We define $latency(c, d)$ to be the one-

way network latency between a population center c and a potential location d . In addition, we define P_c^d to be the number of servers at location d that serve requests from center c , and $servers(c)$ to be the number of servers required by center c . In more detail, $servers(c)$ is the average number of users of the service in population center c (e.g., 30% of the population of c) divided by the average number of users that can be accommodated by a server.

Consistency delay. Because services must keep working seamlessly even when a datacenter becomes unreachable or unavailable, the datacenters that provide a service must be mirrors of each other. This implies that changes to persistent state must be dynamically propagated to all the mirrors. The consistency delay refers to the time required for state changes to reach all mirrors. To model consistency, we define $latency(d_1, d_2)$ to be the one-way network latency between potential locations d_1 and d_2 . We model the consistency delay as the maximum network latency between any two mirror locations.

Availability. The availability of a service depends on the availability of the network of datacenters that host the service. We model the availability of such a network as:

$$Availability = \sum_{i=0}^n \binom{n}{i} \cdot a_i^{n-i} \cdot (1 - a_i)^i \quad (1)$$

where n is the number of datacenters and a_i is the availability of each datacenter. This model has been used in multiple previous reliability and availability works, e.g. [2], [3]. The availability of each datacenter depends on

the level of redundancy in its design. Industry commonly classifies datacenters into tiers [4], where each tier implies a different redundancy level and expected availability. At one extreme, Tier I datacenters have a single path for power and cooling distribution. At the other extreme, Tier IV datacenters have two active power and cooling distribution paths, with redundant components in each path. Existing Tier I datacenters have been found to achieve an availability of 99.67%, whereas Tier II datacenters achieve 99.74%, Tier III datacenters achieve 99.98%, and Tier IV datacenters achieve 99.995% [4].

CO₂ emissions. The type of electricity consumed at each potential location determines the resulting carbon emissions of the datacenter. (Obviously, the construction and operation of datacenters involves other sources of carbon emissions. For simplicity, we do not model them.) We define $\text{emissions}(d)$ as the carbon emissions in grams/KWh of the electricity generation at location d .

B. Formulating the optimization problem

Using the parameters of our framework, we now formally define the cost model and optimization problem. The problem setup consists of an Internet company that seeks to select locations for a set of datacenters out of a set of potential locations for them (D). The company also seeks to offer its services to a set of population centers (C), so each datacenter needs to be sized accordingly. The optimization goal is to minimize the overall cost of the datacenter network, while respecting network latency, consistency delay, and/or availability constraints.

The inputs to the optimization are (1) the maximum number of servers to be hosted across the network of datacenters (MaxS); (2) the expected average utilization for the servers in use (Util); (3) the number of users that each server can accommodate; (4) the amount of redundancy that will be built into each datacenter; (5) the maximum network latency (MAXLAT), maximum consistency delay (MAXDELAY), and/or minimum availability (MINAVAIL) required by the network of datacenters; (6) the CAPEX and OPEX costs for each location $d \in D$; (7) the one-way network latency between any population center $c \in C$ and each location d ; and (8) the one-way network latency between any two locations d_1 and d_2 .

The outputs of the optimization are the optimal cost, the maximum number of servers at each location d (S_d), and the number of servers that service population center c at location d (P_c^d).

Equation 2 in Figure 1 defines the cost we seek to minimize (TotalCost), where SB_d and PB_c^d are booleans. SB_d represents the placement of a datacenter at location d , i.e. it is defined as 1 if $S_d > 0$, and 0 if $S_d \leq 0$. Similarly, PB_c^d represents the use of servers at location d to serve users from a population center c , i.e. it is defined as 1 if $P_c^d > 0$,

and 0 if $P_c^d \leq 0$. Recall that CAP_ind is the CAPEX cost that is independent of the number of servers; CAP_max is the CAPEX cost per server assuming the maximum number of servers; CAP_act is the CAPEX cost per hosted server; OP_act is the OPEX cost per hosted server; and OP_utl is the utilization-dependent OPEX cost per hosted server. All costs assume 12 years of lifetime for datacenters and 4 years for servers. In addition, our modeling assumes that the CAPEX costs already embody the depreciation costs and any interest payments they may incur.

TotalCost should be minimized under the constraints that follow the equation in the figure. The constraints include the network latency, consistency delay, and availability requirements of the service. In reality, the availability constraint is more complex than in the figure. In particular, in a network with at least 2 datacenters and S servers, we also ensure that the failure of 1 datacenter will leave $S/2$ servers available to handle the load. Similarly, in a network with at least 3 datacenters, we ensure that the failure of 2 datacenters will leave $S/3$ servers available.

Existing datacenters. We can use a small variation of the problem in Figure 1 for adding datacenters to an existing network of datacenters. In fact, this type of network extension is likely to be the most common scenario for real Internet companies. Companies may want to extend their networks to provide more services, serve more users, reduce latency, reduce consistency delay, or increase availability.

The variation adds new constraints specifying the number of servers at each datacenter that has already been allocated. With these constraints, we can generate the location of the new datacenters and their size.

C. Solution approaches

The problem described in Figure 1 can be time-consuming to solve for two reasons: (1) it is non-linear, since SB , PB , and CAP_max are non-linear functions (CAP_max has different values per MW depending on the size of the datacenter); and (2) there may be a large number of potential locations to evaluate. Unfortunately, the fact that the problem is non-linear means that it is not directly solvable by linear programming (LP) solvers, which are very fast. If an LP solver is to be used at all, we must eliminate the non-linearities (perhaps at the cost of accuracy). Next, we discuss approaches that use LP to different extents.

Simple linear programming (LP0). The first solution approach reformulates the optimization problem to remove variables S_d and P_c^d , and use a linear version of CAP_max (called LCAP_max). The removal of S_d and P_c^d requires stricter constraints on the placement of servers for each population center (constraints 5 and 6). The reformulated problem appears in Figure 2.

We solve the reformulated problem to find SB_d and PB_c^d directly. The actual number of servers at each datacenter is

$$TotalCost = \sum_{d \in D} SB_d \cdot CAP_ind(d) + \sum_{d \in D} S_d \cdot CAP_max(d) + \sum_{c \in C} \sum_{d \in D} P_c^d \cdot (CAP_act + OP_act(d) + OP_utl(Util, d)) \quad (2)$$

1. $\forall d \forall c PB_c^d \cdot latency(c, d) \leq MAXLAT \Rightarrow$ i.e., no user should experience higher latency than MAXLAT.
2. $\forall d_1, d_2 SB_{d_1} \cdot SB_{d_2} \cdot latency(d_1, d_2) \leq MAXDELAY \Rightarrow$ i.e., consistency should take no longer than MAXDELAY.
3. $Availability \geq MINAVAIL \Rightarrow$ i.e., availability (defined in Equation 1) must be at least MINAVAIL.
4. $\sum_c servers(c) \leq MaxS \Rightarrow$ i.e., the total number of servers is no greater than MaxS.
5. $\forall c \sum_d P_c^d = servers(c) \Rightarrow$ i.e., must provision enough servers for every population center.
6. $\forall d \sum_c P_c^d \leq S_d \Rightarrow$ i.e., no datacenter can host more servers than its max capacity.

Figure 1. We solve the optimization problem to find S_d and P_c^d (and consequently SB_d and PB_c^d) for every population center c and potential datacenter location d . CAP_ind = CAPEX independent of the number of servers; CAP_max = CAPEX per server assuming maximum configuration; CAP_act = CAPEX per hosted server; OP_act = OPEX per hosted server; OP_utl = utilization-dependent OPEX per server. All costs assume a 12-year lifetime for datacenters and a 4-year lifetime for servers.

$$ApproxCost = \sum_{d \in D} SB_d \cdot CAP_ind(d) + \sum_{c \in C} \sum_{d \in D} PB_c^d \cdot servers(c) \cdot (LCAP_max(d) + CAP_act + OP_act(d) + OP_utl(Util, d)) \quad (3)$$

1-4. Same as in Figure 1.

5. $\forall c \sum_{d \in D} PB_c^d = 1 \Rightarrow$ i.e., each population center is served by a single datacenter.
6. $\forall c \forall d PB_c^d \leq SB_d \Rightarrow$ i.e., a datacenter must exist to serve a population center.

Figure 2. LP0 solves the optimization problem to find SB_d and PB_c^d for every population center c and potential datacenter location d . LCAP_max is a linear version of CAP_max. The first 4 constraints are the same as those in Figure 1. Again, all costs assume a 12-year lifetime for datacenters and a 4-year lifetime for servers.

then $\sum_c \sum_d PB_c^d \cdot server(c)$. To retain linearity, this value is also assumed to be the maximum number of servers of each datacenter. In addition, LCAP_max assumes the same cost per MW for large datacenters as for small datacenters.

After the problem is solved, we proportionally set the maximum number of servers at each datacenter, so that the sum of all maximum numbers of servers is equal to MaxS. With the maximum number of servers defined for each datacenter, we compute its CAP_max using the proper costs per MW. Finally, we use the original function (Equation 2) to compute the cost of the network of datacenters.

Due to its simplifications and restrictiveness, this approach may produce higher total cost for a datacenter network than the other approaches.

Pre-set linear programming (LP1). Another approach for linearizing the problem is to pre-set SB_d and remove PB_c^d . Pre-setting is effected by creating additional equality constraints to the problem. Instead of using SB_d as a boolean, we pre-set it to 0 (meaning no datacenter at location d), 1 (small datacenter at location d), or 2 (large datacenter at location d). Using these values, we can apply the proper CAP_max. The removal of PB_c^d only affects the maximum network latency constraint. Instead of using that constraint, we check the latency between every datacenter d and population center c . If the latency is larger than MAXLAT, we pre-set P_c^d to 0. This approach allows us to solve the problem to find S_d and the values for P_c^d that were not pre-set.

As it requires a previously selected set of datacenters, this approach cannot be used by itself. Nevertheless, it will be

used by the next solution approaches to calculate the server distribution and the cost for a given set of datacenters.

Brute force (Brute). This approach generates all the possible combinations of datacenters in every tentative location and tests each of them using the LP1 approach. It ranks each of the placements and returns the best one. Since this solution approach is exhaustive, it obtains the optimal result at the cost of extremely long execution times.

Heuristic based on LP (Heuristic). This approach comprises three steps. The first step uses LP0 to generate a set of $M1$ ($M1 = 10$ by default) datacenter networks for each number of datacenters ranging from 1 to $|D|$. We limit the number of datacenters using an additional constraint ($\sum_d SB_d = NumDC$) that is added to the problem after each set of $M1$ configurations is produced. As we generate each set, every time a configuration is produced, a new constraint ($\sum_{SB_d=1 \in PreviousConfig} SB_d < NumDC$) is added to the problem that prevents the next execution of LP0 from selecting the same datacenter locations again.

The first step leads to $M1 \cdot |D|$ configurations, each of which includes the datacenter locations, the maximum numbers of servers, and the actual numbers of servers. Since these LP0-derived configurations may be sub-optimal, the second step uses their SB_d and PB_c^d values to drive LP1. From the $M1 \cdot |D|$ LP1 results, we create a ranking of configurations in increasing order of cost. Out of this ranking, we take the $M2$ ($M2 = 10$ by default) configurations with the best total costs.

As some important combinations of locations may not have been considered by LP0 (and consequently LP1), the

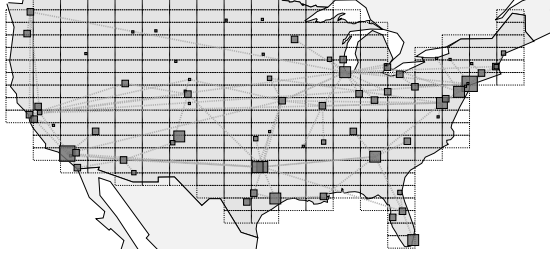


Figure 3. Potential datacenter locations in the US using a 20×20 grid. Population centers and network backbones are also shown.

third step selects the most popular locations from the shorter ranked list of configurations and runs brute force on them.

Simulated annealing plus LP1 (SA+LP1). This approach applies Simulated Annealing (SA) [5] and LP1 to solve the problem in Figure 1.

SA is a generic probabilistic meta-heuristic for non-linear optimization problems. In our implementation, each candidate solution is the SB_d and PB_c^d values for each potential location d and population center c . The optimization starts out with a configuration that fulfills the constraints and has one datacenter in each available location. From this point, in each iteration of the algorithm, SA evaluates one of the neighboring configurations of the current candidate solution using LP1. For example, a neighbor configuration may not include one of the datacenters in the current solution or it may include one in a new location. In each iteration, SA tries to find a lower cost neighbor to use as the next candidate solution, but may also select a higher cost neighbor (with a decreasing probability). If there are no cost reductions for a given number of iterations, the algorithm returns the best candidate solution considered during the optimization.

Optimized SA+LP1 (OSA+LP1). SA+LP1 can be optimized by adjusting the results produced by LP1. Specifically, when LP1 assigns no servers to a datacenter that is part of a configuration, the datacenter can be removed from it. OSA+LP1 then re-evaluates the modified configuration to get a new (lower) total cost. This approach speeds up the optimization process because it drives toward the lowest cost configuration faster.

III. PLACEMENT TOOL AND INPUT DATA

We created a datacenter placement tool based on the framework and optimization machinery described above. Our tool takes as input the parameters that define the user's desired datacenter network: $MaxS$, $1/ratioServerUser$, $MAXLAT$, $MAXDELAY$, and $MINAVAIL$. In addition, the tool allows the user to define the area of interest, the granularity of the potential datacenter locations, and the location of the company's existing datacenters (if any).

The default area of interest is the entire US, but any geographical area can be used as long as we can instantiate

the framework parameters for it. The granularity is specified as an $n \times n$ grid of tiles. For example, Figure 3 shows the US split into a 20×20 grid. Note that some of the tiles are located in areas where a datacenter cannot be placed, such as an ocean, a lake, Canada, or Mexico. Overall, we get 253 potential locations with this grid size. These locations form the D set.

The set of users to be served is assumed to be a (configurable) fraction of the population of the main metropolitan centers within the area of interest. Figure 3 illustrates the 66 metropolitan centers in the US using squares of sizes proportional to their populations. These centers form the C set. Overall, the total population of these centers is 174 million people.

We obtained the information about each location in D from Internet services that provide the coordinates (latitude and longitude) of different places and the name of each place². With the coordinate information, we instantiate framework parameters $distance(p_1, p_2)$. We obtained the population information about the centers in C from the US census³. With the population information, we instantiate parameters $population(c)$. Next, we describe how we instantiate the other location-dependent and datacenter parameters of our framework and tool.

A. Location-dependent data

Network backbones. We extracted the backbone information from the topology of the different ISP backbones⁴. Figure 3 illustrates them with straight lines. In addition, we obtained the shortest path between any two points in the network using Dijkstra's algorithm [6]. Following a model based on the number of hops and the distance [7], [8], we estimate the latency for every path. Specifically, the model assumes an average delay of 8 milliseconds per hop and a propagation speed equivalent to the speed of light on fiber (around 193,121km/s). This model fits the latency data available from the ISPs⁵. With these information, we instantiate parameters $latency(p_1, p_2)$ and $closestNetwork(d)$.

Power plants, transmission lines, and CO_2 emissions. We extracted the power plant location and type information from the Department of Energy (DOE) [9], whereas information about the main power transmission lines came from National Public Radio⁶. Figure 4 illustrates the locations of the major power plants in the US, as well as the major transmission lines. We obtained information about the CO_2 emissions resulting from electricity generation from [10]. Using these data, we instantiate $closestPower(d)$ and $emissions(d)$.

²<http://maps.google.com>, <http://www.geonames.org>

³<http://www.census.gov>, <http://www.city-data.com>

⁴<http://www.ipservices.att.com/backbone>

⁵http://ipnetwork.bgtmo.ip.att.net/pws/network_delay.html

⁶<http://www.npr.org>

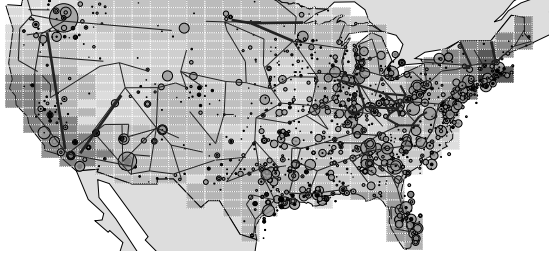


Figure 4. Energy prices (darker is more expensive), power plants, and transmission lines across the US.

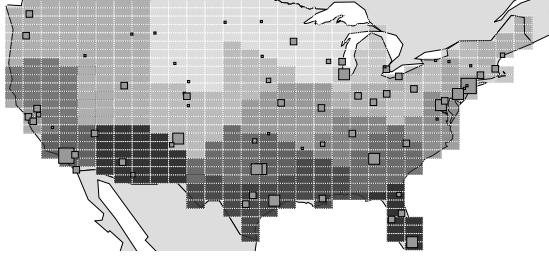


Figure 5. Average temperatures across the US (darker is warmer), as well as population centers.

Electricity, land, water, and temperature. We obtained the price of electricity from the DOE⁷. For example, the electricity price is low (around \$0.04 per KWh) in Washington State, whereas it is high (around \$0.14 per KWh) in Rhode Island. Figure 4 plots these data. With electricity price information, we instantiate parameters *priceEnergy(d)*.

We obtained the price of the land using the methodology presented in [11], which gets its data from a real-state portal⁸ and calculates an average price for each location. We use data about industrial land with more than 10K square feet (SF), which is suitable for a datacenter. These data instantiate parameters *priceLand(d)*.

The amount of water that is used for cooling a datacenter depends on the energy consumption and indirectly on the average temperature at the location. To obtain the temperature information, we used a weather portal with historical data⁹. Figure 5 illustrates the average temperatures in the US. These information allow us to instantiate *avgTemp(d)* and *maxTemp(d)*. We obtained the price of water from a study of water rates and pricing in different cities [12].

Missing data. Depending on the grid size, our tool may not have all the important data for every potential location. In these cases, the tool uses information from the closest neighboring location for which it has the needed data. If there is a tie in the distance, it chooses one of the neighboring locations randomly.

B. Datacenter characteristics

Datacenter size, cooling, and PUEs. Current datacenters are commonly measured by the maximum power they are

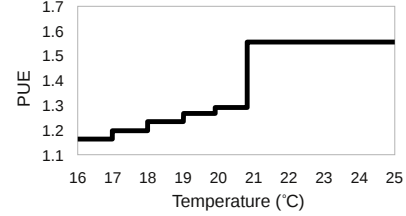


Figure 6. Datacenter PUE, as a function of outside temperature.

designed to consume. The maximum power is a function of the maximum PUE that can be expected, which in turn depends on the type of cooling, the power delivery losses, and the maximum temperature at the location.

Regarding the cooling system, we assume that the datacenters are cooled by Computer Room Air Conditioners (CRACs) and water chillers. The CRACs are always running, but the chiller is only activated when a threshold outside temperature is crossed (20°C). Thus, energy consumption increases with outside temperature. To complete the computation of the PUEs, we assume that the power delivery losses are 8% by default. Taking all of these factors into account, Figure 6 illustrates our modeling of the PUE as a function of outside temperature.

Connection costs. Since each datacenter will not necessarily be located near a power plant or existing transmission line, there may be costs associated with laying out lines to bring enough electricity to it. The same can be said about laying out optical fiber to reach an Internet backbone. Our default values for these costs are \$500K per mile of transmission line and \$480K per mile of fiber laid out [13]. The amortization period for this infrastructure is 12 years.

We assume that the amount of external network bandwidth required by a datacenter depends linearly on the number of servers it hosts. Our default amount is 1Mbps per server. A high bandwidth link to a backbone typically costs between \$1 and \$2 per Mbps¹⁰; our default cost is \$1 per Mbps.

Building costs. The cost of building a datacenter, including purchasing and installing its cooling and power delivery infrastructures is typically computed as a function of its maximum power [14]. In particular, small datacenters ($\leq 10\text{MW}$) are more costly to build than large ones ($> 10\text{MW}$). The specific cost per Watt depends on the level of availability desired for the datacenter [14]. For example, building a small Tier I datacenter costs \$10 per Watt, whereas a large one costs \$8 per Watt. The numbers for the other tiers are \$11 and \$8.8 for Tier II, \$20 and \$16 for Tier III, and \$22 and \$17.6 for Tier IV. This cost is amortized for 12 years.

In all our experiments below, we assume new datacenters close to Tier III with \$15 (small) and \$12 (large) per Watt

⁷ <http://www.eia.doe.gov>

⁸ <http://www.showcase.com>

⁹ <http://www.weatherbase.com>

¹⁰ <http://www.internetbandwidth.us/>

and a typical availability of 99.827%, as suggested by [1].

Land cost. The amount of real-estate required by a datacenter is also typically computed based on its maximum power. Current datacenters occupy around 6K SF per Megawatt [14]. Using this size ratio and the price of land, we compute the total land cost.

Water cost. According to [15], a datacenter of 15 Megawatts with the type of cooling we model uses around 360K gallons of water per day. Using the same ratio, we assume that a datacenter consumes 24K gallons of water per MW per day. From this water consumption and the water price, we compute the total water cost.

Servers and internal networking costs. Our default servers are Dell PowerEdge R610 with 4 processor cores running at 2.66GHz with 6GBytes of RAM. These servers consume a maximum of 260 Watts and an average of 200 Watts at a 30% utilization (a common average utilization in Internet service datacenters [1]). The cost of each of these servers is around \$2000, which is amortized over 4 years (i.e., each of them costs \$42 per month).

Our default interconnect is the 40-port Cisco Nexus 5020, which consumes 480 Watts. In our datacenter designs, each switch connects 32 servers; the switches also connect to each other hierarchically to reach all servers in the datacenter. The cost of each switch is \$20K, which is amortized over 4 years.

Staff costs. Any datacenter needs personnel to operate and maintain it. Following [1], our default for maintenance costs is \$0.05 per Watt per month. In addition, our default operation costs assume that each administrator can manage 1K servers for an average salary of \$100K per year.

IV. EXPLORING THE TOOL

In this section, we use our tool and input data to characterize seven interesting locations in the United States. Then, we present a case study in which we place a network of datacenters across the country.

A. Characterizing locations

To study the influence of the location on the cost of a datacenter, we have chosen seven representative locations with different characteristics: Austin, Bismarck, Los Angeles, New York, Orlando, Seattle, and St. Louis. Our study consists of placing a datacenter with 60K servers in each location, and assessing the components of the overall cost. The placement area is a square of 200 miles around these cities, which is divided in 9 tentative placement areas. Our optimization finds the most cost-effective of these areas.

Figure 7 shows the contribution of each component to the overall cost of the datacenter per month. As it does not depend on the location, the cost with servers and internal networking is always the same: \$3.3M per month. The other components are affected by location-dependent parameters,

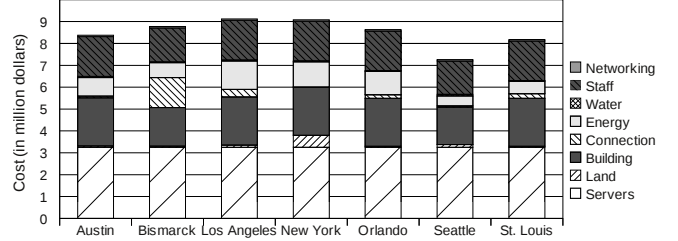


Figure 7. Datacenter cost depending on location.

Location	PUE	Land (\$/SF)	Energy (\$/kWh)	Water (¢/gal)	CO ₂ (g/kWh)
Austin	1.39	0.394	0.066	0.395	569
Bismarck	1.20	0.434	0.062	0.324	869
Los Angeles	1.41	0.638	0.099	0.332	286
New York	1.29	3.460	0.096	0.348	960
Orlando	1.42	0.272	0.081	0.225	541
Seattle	1.19	0.987	0.041	0.652	120
St. Louis	1.32	0.264	0.047	0.205	806

Table II
LOCATION CHARACTERISTICS.

such as electricity price, water price, land price, distance to infrastructure, and weather. Table II summarizes some of the main characteristics of the seven locations.

From this figure and table, we make several observations:

1. *Austin.* Despite the fact that water is expensive in Texas, it does not represent a significant cost for a datacenter. Nevertheless, the arid weather in the region produces a relatively high average PUE of 1.39.
2. *Bismarck.* The Midwestern region is far from the existing network infrastructure. As a result, the cost of providing network connectivity to a datacenter in Bismarck is high. Moreover, the CO₂ emissions in that region are relatively high, as most of its electricity is coal-generated.
3. *Los Angeles.* Temperatures in Southern California are high, which implies a high average PUE. Land prices in this region are also high. These two factors make this location the most expensive to place a datacenter.
4. *New York.* Land in this region is the most expensive around the country. Electricity is also expensive. Again, these factors make the New York area an inappropriate location for a datacenter.
5. *Orlando.* Florida is the warmest state in the country, producing the highest average PUE of 1.42. The PUE affects many costs, including building and land costs. Moreover, such a high PUE implies an electricity consumption that is unnecessarily high.
6. *Seattle.* Energy is inexpensive in Washington State, leading to one of the lowest electricity costs in the country. As most of the electricity in this region is produced by hydro-electric plants, the CO₂ emissions are low.

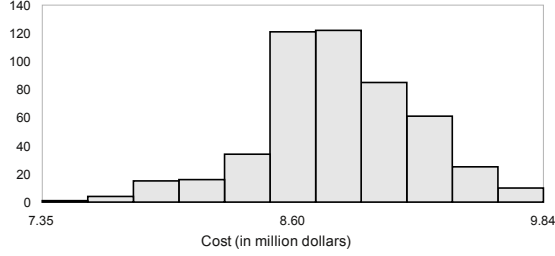


Figure 8. Cost per month distribution in 500 locations of the US.

7. *St. Louis*. This location is in the center of the country; users in any part of the country can reach it in less than 75ms. In terms of costs, the St. Louis region exhibits low electricity, land, and water prices. However, its PUE and CO_2 emissions are relatively high.

To get a deeper understanding of this datacenter’s cost, we detail the most relevant aspects of the placement in St. Louis. We start with those aspects that affect the maximum power consumed by the datacenter. The average and maximum temperatures are around $13^\circ C$ and $26^\circ C$, respectively. These temperatures lead to average and maximum PUEs of 1.32 and 1.6, respectively. As this datacenter hosts 60K servers, operating them at maximum utilization during the summer would produce a maximum power of 26.4 MW. For this maximum power, the building cost is \$2.2M per month and the space required is around 158K SF (more than 2 and a half football fields). To connect the datacenter to the power and the network infrastructures, 60 miles of transmission lines and optical fiber would be required. These connections would cost \$208K per month.

In terms of operational costs, a datacenter in this region would use an average of 12253 MWh of energy per month. This consumption implies a cost of \$573K per month and translates into the emission of 9876 tones of CO_2 per month (80% of the energy generated in Missouri comes from coal and 6% from natural gas). In addition, the cooling system of a datacenter of 26.4 MW consumes more than 12M gallons of water per month, translating into a cost of \$25K per month. Finally, the labor costs of the datacenter would amount to \$1.8M per month.

The final cost of this data center is \$8.22M per month, i.e. more than \$98M per year or \$1183M during the whole datacenter lifetime of 12 years.

Broadening the scope. Still placing a single datacenter with 60K servers, we now consider the corresponding costs across the country. Figure 8 presents the distribution of the cost assuming 500 potential locations (a 30×30 grid). The distribution shows that there are a few locations where the datacenter can be cheaper (around \$7.35M), whereas others lead to much higher costs (around \$9.84M). This observation highlights the importance of properly selecting locations for datacenter networks.

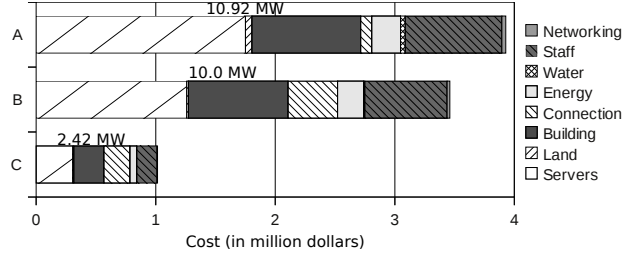


Figure 9. Cost of a network of datacenters for 60K servers.

Grid	D	SA				
		LP0	Heur.	+LP1	OSA	Brute
3×3	4	10.34	9.27	9.14	9.07	9.07
4×4	9	9.79	9.79	9.67	9.63	9.63
5×5	16	8.91	8.91	8.78	8.78	8.78
6×6	18	9.26	8.98	8.89	8.89	8.89
7×7	30	8.40	8.38	8.54	8.31	8.31
8×8	33	8.89	8.51	8.52	8.41	8.41
10×10	62	8.45	8.22	8.16	8.13	8.13
15×15	124	8.81	8.61	8.81	8.13	8.13
20×20	253	8.85	8.35	8.37	8.05	8.05
30×30	500	13.08	8.10	8.44	7.85	7.85

Table III
OVERALL COSTS (IN \$M PER MONTH) OF SOLUTION APPROACHES.

B. Case study: Placing a datacenter network

As a complete example of the use of our tool, we compute the total cost of placing 60K servers, so that any user can reach the closest datacenter in less than 60ms, the maximum consistency delay is 85ms, and the minimum availability is 5 nines. Figure 9 presents the results. Our tool placed 3 datacenters (A, B, and C) close to Seattle, St. Louis, and Oklahoma City. These datacenters host 31789, 22712, and 5501 servers, respectively. The figure shows the total costs per month for the three datacenters.

V. EVALUATING OUR SOLUTION APPROACHES

To evaluate our solution approaches, we assess the running time and overall cost they produce, as a function of the number of potential locations. Our study assumes that the goal is to place 60K servers with a maximum latency of 60ms, a maximum consistency delay of 85ms, and a minimum required availability of 5 nines. Each experiment is performed on an 8-core 2.4GHz machine with 8GB of memory. Our tool fully exploits the 8 cores by running multiple solution threads in parallel.

Table III shows the overall cost achieved by each solution approach. To assess the quality of the results, we compare them to the *Brute* approach, which computes the optimal cost. However, running *Brute* to completion is infeasible, so we stopped each of its executions after 3 days – enough time to explore all possible placements of up to 5 datacenters. Beyond this point, better solutions are highly unlikely to exist. The colors in the figure represent how close each result is to *Brute*; darker colors mean worse solutions. Recall that

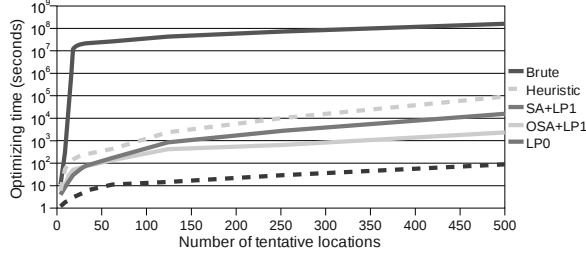


Figure 10. Running times of solution approaches.

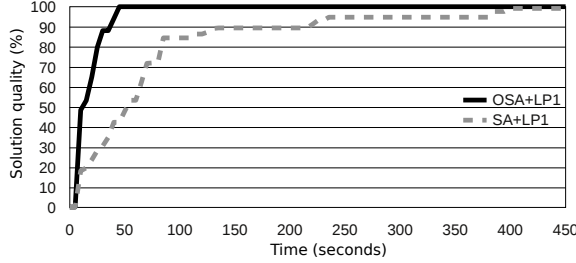


Figure 11. Solution quality over time using *SA+LP1* and *OSA+LP1*.

we cannot use *LP1* in isolation, so we do not include it in this comparison.

We observe that *OSA+LP1* finds the optimal solution in all cases, whereas *SA+LP1* only does so in two cases. The reason is that Simulated Annealing is more likely to get stuck in local minima when the search space is not pruned. We also see that *Heuristic* exhibits good behavior in most cases. However, this approach does not find the optimal results, as it leverages *LP0* heavily. Finally, we find that *LP0* exhibits the worst behavior. Due to its simplifications, *LP0* is inflexible in terms of the assignment of user load to datacenters, and does not consider the non-linearity of building costs.

Figure 10 shows the approaches' running times. The times for *Brute* were extrapolated from the 3-day runs mentioned above, whereas the times for the other approaches were measured from complete executions. Note that the Y-axis is in log scale. From this figure, we can observe that all approaches scale linearly with the number of locations (quadratically with the dimensions of the grid), but with different slopes. *LP0* exhibits the lowest running times, as it relies solely on linear programming with a small number of variables. At the other extreme, *Brute* exhibits very high running times. In fact, this approach takes multiple hours for as few locations as 18. As the other approaches explore much smaller spaces, they execute at least 3 orders of magnitude faster than *Brute*. *Heuristic* is the slowest of these approaches, whereas *OSA+LP1* is the fastest. Overall, *OSA+LP1* is 1-2 orders of magnitude faster than *Heuristic* and at least 5 orders of magnitude faster than *Brute* for non-trivial problem sizes.

Figure 11 illustrates how quickly *OSA+LP1* converges to its results compared to *SA+LP1*, for a scenario in which both approaches reached the same solution. It is clear from the

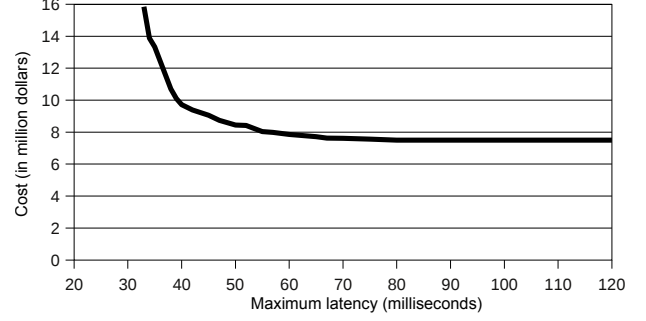


Figure 12. Cost of a datacenter network, as a function of max latency.

Availability	#9s	Tier I	Tier II	Tier III	Tier IV
99%	2	1	1	1	1
99.9%	3	2	2	1	1
99.99%	4	2	2	2	1
99.999%	5	3	2	2	2
99.9999%	6	3	3	2	2

Table IV

NUMBER OF DATACENTERS REQUIRED FOR A GIVEN AVAILABILITY.

figure that pruning the search spaces allows *OSA+LP1* much faster convergence. *Heuristic* converges more slowly than these two approaches, as it has to wait until *LP0* generates all the tentative locations to finally evaluate them using *LP1*. In contrast, the quality of the *Brute* solution over time does not exhibit a monotonic behavior, as it tries every possible solution in turn without regard for how good they are.

Overall, it is clear that *OSA+LP1* provides the best tradeoff between running time and search quality; it achieves optimal results with the second lowest execution times.

VI. EXPLORING DATACENTER PLACEMENT TRADEOFFS

We now use our tool (*OSA+LP1*) to answer the following interesting questions about datacenter placement and design:

- How much does lower latency cost?
- How much does higher availability cost?
- How much does faster consistency cost?
- How much does a green datacenter network cost?
- How much does a chiller-less datacenter network cost?

Throughout the section, unless otherwise stated, we place a network of datacenters for 60K servers using a 20×20 grid (253 potential locations). The default constraints are a maximum latency of 60ms, a maximum consistency delay of 85ms, and a minimum availability of 5 nines.

A. Latency

Figure 12 presents the total cost per month, as a function of the maximum latency constraint. We can see that required maximum latencies greater than 70ms all lead to the same cost, around \$7.8M per month. Lower latencies require more datacenters and higher costs. A latency of 50ms strikes the best compromise between latency and cost; cost increases are steep for lower latencies. For 33ms, the minimum latency that can be achieved with this grid size, the total cost is \$15.8M per month (almost twice the cost for 50ms).

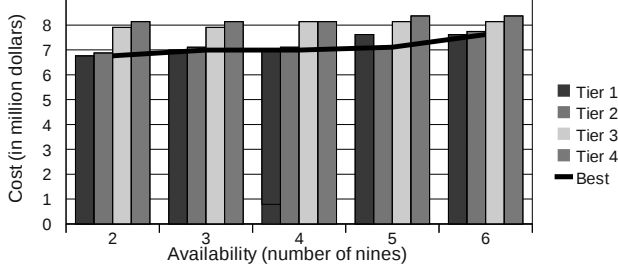


Figure 13. Cost of a datacenter network, as a function of min availability.

Consistency	Latency	Datacenters	#9s	Cost
120 ms	47 ms	7	>6	\$10.70M
110 ms	50 ms	5	>6	\$10.78M
80 ms	50 ms	5	>6	\$10.78M
70 ms	62 ms	3	>6	\$8.66M
60 ms	65 ms	3	>6	\$8.66M
50 ms	75 ms	2	5	\$8.06M

Table V

IMPACT OF MAX CONSISTENCY DELAY ON MAX LATENCY AND COST.

B. Availability

Based on Equation 1 and the availability data in Section II-A, Table IV presents the number of datacenters of each tier that are required to achieve different network-wide availabilities. Using these numbers of datacenters as constraints, Figure 13 presents the total cost per month of achieving different availabilities. The figure shows that *it is usually cheaper to build networks out of less redundant datacenters*. In fact, the figure shows that this is usually the case even if one needs to build more datacenters. For example, it is cheaper to build 3 Tier II datacenters than 2 Tier IV datacenters to achieve an availability of 6 nines. Overall, *Tier II datacenters are the best option*.

C. Consistency delay

Table V presents the maximum one-way network latency to any user (second column from the left), as a function of many maximum consistency delays (leftmost column). Due to time constraints, we used a grid of 5×5 locations for the results in the table. This is why the total costs per month (rightmost column) are higher in this table.

Intuitively, note that *low latency and low consistency delay are conflicting goals*, since the former tends to spread the datacenters geographically and the latter tends to do the opposite. For example, having a relaxed consistency delay requirement of 120ms leads to a maximum latency of only 47ms and a high cost of \$10.7M per month. In contrast, a strict consistency delay requirement of 50ms leads to a maximum latency of 75ms and a lower cost of \$7.79M per month. Lower maximum delay than 50ms cannot be achieved with this grid size.

D. Green datacenters

Today, the CO_2 emissions of datacenters from electricity consumption are huge. For example, an optimal-cost net-

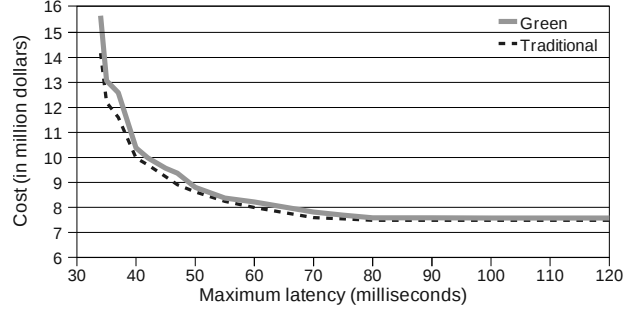


Figure 14. Cost of a green network, as a function of max latency.

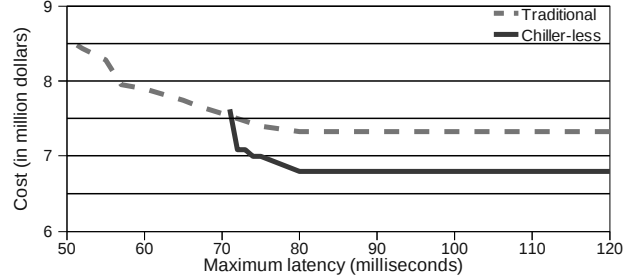


Figure 15. Cost of a chiller-less network, as a function of max latency.

work of 8 datacenters with 60K servers generates around 8K tonnes of CO_2 due to its energy consumption every month. This is equivalent to the direct emissions of 50 round-trip flights between New York and Los Angeles. These emissions would be avoided by using green electricity only. Figure 14 compares the total cost per month of a datacenter network using green sources, as well as the cost-optimal network, as a function of the maximum latency. The figure demonstrates that *a green network is less than \$100K more expensive per month than the cost-optimal network when the maximum latency can be relatively high (> 70 ms)*. However, for very low maximum latency, this additional cost can reach \$1.6M per month.

E. Chiller-less datacenters

One of the largest individual overheads in a datacenter is the water chiller. The chiller can increase energy consumption by 20% and building costs by 30% [16]. Unfortunately, any location in which the outside temperature can exceed $20^\circ C$ requires a chiller. Figure 15 compares a chiller-less datacenter network against the cost-optimal one, as a function of the maximum latency. As we can see, *avoiding chillers reduces costs by 8% for maximum latencies higher than 70ms*. However, *maximum latencies lower than 70ms cannot be accommodated*, since the outside temperature constraint limits the set of locations that can be used.

VII. RELATED WORK

To the best of our knowledge, no other works have considered the intelligent placement of datacenters for Internet services. The closest related work is [17], which

identifies the main costs in datacenters and states that their placement and provisioning can have a significant impact on service profitability. Unfortunately, they did not quantify the actual costs or their relationship to geographical locations. Moreover, they did not formulate an optimization approach to solve the placement problem.

A few white papers and electronic articles have ranked a small number of specific locations in the country in terms of operational costs or the likelihood of natural disasters [18], [19], [20]. However, these works did not consider automation, optimization, or all the relevant costs and constraints. For example, none of them consider network latencies, consistency delays, or availability requirements.

Obviously, large Internet companies, such as Microsoft and Google, have their own processes for selecting locations for their datacenters. (We have discussed our approach with Microsoft and received positive feedback [21].) Unfortunately, their processes are highly confidential for commercial reasons. One of the goals of our work is to shed light on this problem and enable the research community to start tackling it in the public domain.

Although previous works have not studied the placement of datacenters comprehensively or in detail, there have been some rigorous studies of the placement of other distributed IT resources. For example, [22] uses optimization techniques for placing caches in Content Distribution Networks (CDN), achieving significant improvements in performance. Our work differs from these efforts in terms of our focus on datacenters, Internet services, and the physical and cost characteristics of the geographical locations.

Finally, facility location is a well-known theoretical problem [23]. However, treatments of this problem have focused on its algorithmic aspects and theoretical implications, in the absence of real location data.

VIII. CONCLUSIONS

In this paper, we considered the automatic placement of datacenters for Internet services. We proposed an optimization framework for the problem and many solutions approaches for it. We also characterized different US regions as potential locations for datacenters. Our evaluation compared the solution approaches and illustrated many tradeoffs. Among other interesting results, we showed that the intelligent placement of datacenters can save millions of dollars.

Our effort brings attention to an important problem that has not been tackled in the scientific literature. Although a strong first step, we believe that our work can be extended by creating ways to collect even more fine grained input data automatically, studying the frequency with which the input data should be updated, and covering other parts of the world. These advances will enable companies that do not have the financial and human resources of established Internet giants to intelligently place their datacenters.

Acknowledgments

This work has been partially supported by Spain's Ministry of Science and Technology and the European Union under contract TIN2007-60625 and grant AP2008-0264, the Generalitat de Catalunya grant 2009-SGR-980, NSF grant CSR-0916518, and the Rutgers Green Computing Initiative.

REFERENCES

- [1] L. A. Barroso and U. Hölzle, *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines*. Synthesis Lectures on Computer Architecture, Morgan & Claypool Publishers, 2009.
- [2] E. Pinheiro, R. Bianchini, and C. Dubnicki, "Exploiting Redundancy to Conserve Energy in Storage Systems," in *Proceedings of the International Conference on Measurement and Modeling of Computer Systems*, June 2006.
- [3] D. Siewiorek and R. Swarz, *Reliable Computer Systems Design and Evaluation*, 3rd ed. A. K. Peters, 1998.
- [4] W. P. Turner, J. H. Seader, and K. G. Brill, "Tier Classifications Define Site Infrastructure Performance," 2006, Uptime Institute.
- [5] S. Kirkpatrick, "Optimization by Simulated Annealing: Quantitative Studies," *Journal of Statistical Physics*, vol. 34, no. 5, 1984.
- [6] E. W. Dijkstra, "A Note on Two Problems in Connection With Graphs," *Numerische mathematik*, vol. 1, no. 1, 1959.
- [7] S. Sivasubramanian, M. Szymaniak, G. Pierre, and M. Steen, "Replication for Web Hosting Systems," *ACM Computing Surveys*, vol. 36, no. 3, 2004.
- [8] R. Abolian, Y. Sun, and G. J. Koehler, "A Location-Allocation Problem for a Web Services Provider in a Competitive Market," *European Journal of Operational Research*, vol. 194, no. 1, 2009.
- [9] US Department of Energy, "1990–2008 Net Generation by State by Type of Producer by Energy Source," Energy Information Administration, Office of Energy Markets and End Use, 2010.
- [10] B. Sovacool, "Valuing the Greenhouse Gas Emissions from Nuclear Power: A Critical Survey," *Energy Policy*, vol. 36, no. 8, 2008.
- [11] J. B. Nichols, S. D. Oliner, and M. R. Mulhall, "Commercial and Residential Land Prices Across the United States," *Finance and Economics Discussion Series*, 2010.
- [12] B. Walton, "The Price of Water: A Comparison of Water Rates, Usage in 30 U.S. Cities," 2010, Circle of Blue.
- [13] T. Seppa, "Improving Asset Utilization of Transmission Lines by Real-Time Rating," July 1999, T&D Committee Meeting, IEEE/PES Summer Power Meeting.
- [14] W. P. Turner and J. H. Seader, "Dollars per KW Plus Dollars per Square Foot are a Better Data Center Cost Model than Dollars per Square Foot Alone," 2006, Uptime Institute.
- [15] R. Miller, "Data Centers Move to Cut Water Waste," 2009, Data Center Knowledge.
- [16] V. Anagnostopoulou, H. Saadeldien, and F. T. Chong, "Quantifying the Environmental Advantages of Large-Scale Computing," in *Proceedings of the International Green Computing Conference*, August 2010.
- [17] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The Cost of a Cloud: Research Problems in Data Center Networks," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 1, 2008.
- [18] M. Stansberr, "Data Center Locations Ranked by Operating Cost," 2006, SearchDatacenter.com.
- [19] D. Alger, "Choosing an Optimal Location for Your Data Center," 2005, InformIT.
- [20] B. Oley, "Where's the Best Place to Build a Data Center?" 2010, Data Center Journal.
- [21] C. Belady, "Personal communication," August 2010.
- [22] A. Wierzbicki, "Internet Cache Location and Design of Content Delivery Networks," *Lecture Notes in Computer Science*, 2002.
- [23] K. Jain, M. Mahdian, E. Markakis, A. Saberi, and V. V. Vazirani, "Greedy Facility Location Algorithms Analyzed Using Dual Fitting With Factor-Revealing LP," *Journal of the ACM*, vol. 50, no. 6, 2003.