Cost Efficient Datacenter Selection for Cloud Services

Hong Xu, Baochun Li
henryxu, bli@eecg.toronto.edu
Department of Electrical and Computer Engineering
University of Toronto

***Abstract*—Many cloud services nowadays are running on top of geographically distributed infrastructures for better reliability and performance. They need an effective way to direct the user requests to a suitable datacenter, in a cost efficient manner. Previous work focused mostly on the electricity cost of datacenters. The approaches favor datacenters at locations with cheaper electricity prices. In this paper, we augment the picture by considering another significant cost contributor: network bandwidth. We propose to utilize statistical multiplexing to strategically bundle demands at different locations. The anti-correlation between demands effectively smooths out the aggregated bandwidth usage, thereby saving the bandwidth cost calculated by burstable billing methods that charge the peak bandwidth usage. We present an optimization framework that models the realistic environment and practical constraints a cloud faces. We develop an efficient distributed algorithm based on dual decomposition and the subgradient method, and evaluate its effectiveness and practicality using real-world traffic traces and electricity costs.**

I. Introduction

Internet-scale services are becoming essential to our everyday lives, with important applications including web search, video-on-demand, and file hosting. The emergence of cloud computing platforms, such as Amazon AWS [1], further enables rapid deployment of new services at scale. Almost all of these services are built atop geographically distributed infrastructures, i.e. datacenters located in different regions to provide better reliability and performance. They need an effective way to direct clients across the wide area to an appropriate datacenter. Usually, Internet-scale services handle datacenter selection by deploying mapping nodes, which are typically DNS servers as shown in Fig. 1, to customize the IP address(es) returned to different clients. Alternatively, they can also outsource datacenter selection to third-parties [2], [3] or the cloud provider [4].

An efficient datacenter selection algorithm is imperative to the operation of cloud services. Many previous works exist in this area. The problem can be cast as an optimization that maximizes the system-wide performance subject to certain cost constraints. Existing works usually consider the electricity cost of running datacenters. By taking advantage of the geographic diversity of electricity prices, requests are directed in favor of datacenters with lower electricity prices, and costs can be reduced [5], [6].

In this paper, we consider another significant cost contributor to datacenters: wide-area network bandwidth [7]. We propose to utilize statistical multiplexing to strategically bundle demands from different mapping nodes. The intuition is that, demands from different mapping nodes correlate with each other in a different fashion: some are positively correlated, i.e. they tend to peak at the same time, while some are negatively correlated, i.e. when demand from one node rises, demand from another node tends to decrease. By combining demands from negatively correlated nodes, the aggregated bandwidth required from a particular datacenter is smoothed out across time, thereby reducing the bandwidth cost which is determined by burstable billing methods, such as the 95-percentile billing that charges peak bandwidth usage [8]–[10].

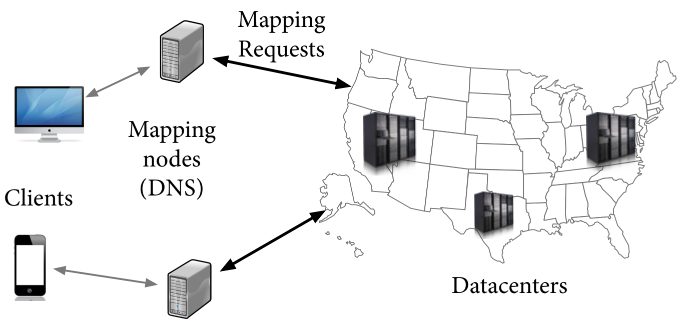

Fig. 1. An example of a cloud service running atop a geographically distributed cloud infrastructure.

To better illustrate the idea, Fig. 2 and 3 plot some sample demand data we collected from a major online multimedia company UUSee [11] in China. The red line corresponds to the 95-percentile bandwidth consumption, which amounts to 45 Mbps and 33 Mbps for node 1 and 2 respectively. If these two nodes were served by the same datacenter, the aggregated bandwidth consumption as shown in Fig. 4 is smoother than the individual curves. The 95-percentile of the aggregated bandwidth is around 70 Mbps, which is smaller than the sum of the individual 95-percentile values. This demonstrates the potential of multiplexing in terms of saving bandwidth consumption and cost.

Our main contribution in this paper is a general optimization framework for cost efficient datacenter selection that takes into account both electricity and bandwidth costs. Our framework is general in the sense that it models practical environments that a cloud operates in. The utility abstraction encompasses many performance considerations, including throughput, latency, as well as possible fairness criteria. The electricity and bandwidth

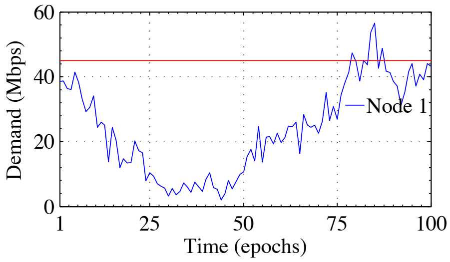

Fig. 2. Demand at node 1.

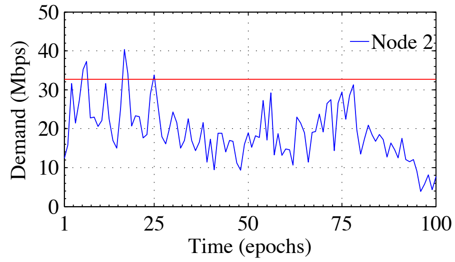

Fig. 3. Demand at node 2.

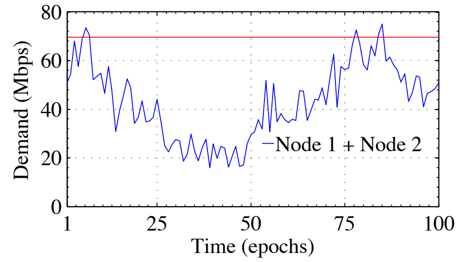

Fig. 4. Aggregated demand.

cost constraints capture the two most important ongoing costs associated with the operation of datacenters, i.e. operation expense (OPEX) [7]. Both the utility and electricity price are location dependent in order to realistically model the geographic diversity.

By using dual decomposition, our optimization formulation can be decentralized to the datacenter level. Specifically, the problem can be decomposed into subproblems, each solvable by an individual datacenter itself. This enables us to develop efficient distributed implementations of our datacenter selection algorithm to find the optimal node-datacenter assignment based on the subgradient method. Our algorithms remain relevant in and are applicable to other request direction scenarios, such as a content distribution network (CDN) [12].

We evaluate the effectiveness of our decentralized implementation using real-world traffic traces collected from UUSee [11], as well as real-world electricity prices [13]. Results demonstrate that our algorithm saves the overall operating cost of datacenters while offering a comparable performance compared to the vanilla bandwidth-agnostic solution.

II. An Optimization Framework for Cost Efficient Datacenter Selection

In this section, we present our optimization framework for cost-effective datacenter selection.

A. System Model

We start by introducing the system model. We consider a cloud infrastructure with M datacenters geographically distributed across the wide area. The cloud deploys N *mapping nodes* (e.g. DNS servers) at different locations to serve client requests. We use the term mapping nodes and nodes interchangeably in the sequel. The requests at a particular node are directed to a subset of all the datacenters determined by the datacenter selection algorithm. Since the request traffic fluctuates dynamically, the datacenter selection algorithm needs to be run periodically to optimize performance.

Let us introduce a few notations. We consider an individual time epoch without loss of generality, and thus we drop the time subscript t in our notations. The bandwidth demand of node i is a random variable D_i with mean μ_i and variance σ_i^2. The random demands $\boldsymbol{D} = [D_1, \dots, D_N]^\intercal$ may be correlated due to time difference and the natural correlation between viewer preferences, human behavior, etc. Let $\boldsymbol{\mu}$ denote the $N \times 1$ *mean demand matrix*, or *demand matrix* in short, and $\boldsymbol{\Sigma}$ be the $N \times N$ *covariance matrix*.

We assume that the cloud operator employs techniques such as those in our previous work [14], [15] to predict the demand matrix $\boldsymbol{\mu}$ with satisfactory accuracy. The covariance matrix $\boldsymbol{\Sigma}$ between demands at different nodes can also be predicted for the short-term future by using time series forecasting methods [14], [15].

We also assume that the electricity price at each datacenter p_d is available at the beginning of an epoch, and remains static throughout the entire epoch. This is a practical assumption in today's electricity market in the U.S. If the local electricity market of datacenter d is in a regulated utility region, the electricity price is fixed. If on the other hand the datacenter is in a deregulated market region, such as California and Texas, there is a forward market with settlements of various kinds, such as day-ahead and hour-ahead, for customers to lock in the price [6], [13]. The $M \times 1$ matrix $\boldsymbol{p} = [p_d]^\intercal$ is referred to as the *price matrix*.

We use an abstract *utility* notion u_{id} to capture the performance of the cloud service, when a request from node i is directed to datacenter d. This notion allows us a considerable amount of expressiveness. For example, if the cloud service is an interactive application and seeks minimal latency, u_{id} can be a decreasing function of the round trip time (RTT), directly measured or estimated by various means. If the cloud service is a bulk transfer application and seeks good throughput, U_{id} can be a decreasing function of the network congestion level or the link utilization. It can incorporate fairness considerations by making use of the canonical *alpha-fair* utility functions [16]. For more discussions of the generality of the utility notion one can refer to [3]. The $N \times M$ matrix $\boldsymbol{u} = [u_{id}]$ is the *performance matrix*, and the column vector $\boldsymbol{u}_d = [u_{id}]^\intercal$ is the performance vector of datacenter d.

Finally, we use $w_{id} \in [0, 1]$ to denote the proportion of traffic directed from node i to datacenter d, and $\boldsymbol{w}$ is a $N \times M$ *datacenter selection matrix*. w_{id} is the optimizing variable of our problem. Given $\boldsymbol{w}$, we observe that the vector $\boldsymbol{w}_d = [w_{1d}, \dots, w_{Nd}]^\intercal$ represents the workload portfolio of datacenter d.

B. Modeling the 95-percentile Bandwidth

Note that in the model, we choose not to take into account the bandwidth price, since in reality this is often a fixed price across different regions. It is equivalent to only considering the aggregated bandwidth usage at individual datacenters. The aggregated bandwidth consumption of d becomes a random variable

$$L_d = \boldsymbol{w}_d^{\intercal} \boldsymbol{D}.$$

whose mean and variance are $\boldsymbol{w}_d^{\intercal}\boldsymbol{\mu}$ and $\boldsymbol{w}_d^{\intercal}\boldsymbol{\Sigma}\boldsymbol{w}_d$, respectively, given the datacenter selection matrix $\boldsymbol{w}$. Suppose the peak bandwidth usage of datacenter d is A_d. This implies that, the probability that the bandwidth consumption of d exceeds A_d is equal to ϵ of the time, where ϵ is a small positive constant. That is,

$$\Pr(L_d > A_d) = \epsilon, \forall d.$$

For the 95-percentile charging model, $\epsilon = 0.05$. Note that this can also be interpreted as a QoS constraint, where the probability of bandwidth under-provisioning is bounded by ϵ.

Through reasonable aggregation, L_d follows a Gaussian distribution due to the law of large numbers. This has also been empirically verified using trace studies in previous work [14], [15]. Thus, the above constraint is equivalent to

$$\boldsymbol{w}_d^{\intercal}\boldsymbol{\mu} + \theta\sqrt{\boldsymbol{w}_d^{\intercal}\boldsymbol{\Sigma}\boldsymbol{w}_d} = A_d, \forall d$$

where $\theta = F^{-1}(1-\epsilon)$ and $F(\cdot)$ is the CDF of the Gaussian distribution $\mathcal{N}(0,1)$. For example when $\epsilon = 0.05$, $\theta = 1.96$. The total billable bandwidth usage of the cloud is

$$\sum_d^M \boldsymbol{w}_d^{\intercal}\boldsymbol{\mu} + \theta\sqrt{\boldsymbol{w}_d^{\intercal}\boldsymbol{\Sigma}\boldsymbol{w}_d}.$$

C. An Optimization Framework

Now we formally introduce our optimization framework. The datacenter selection problem at a particular epoch can be succinctly expressed as follows:

$$\text{DC-OPT:} \quad \max_{\boldsymbol{w}\succeq\boldsymbol{0}} \sum_d^M \boldsymbol{w}_d^{\intercal}\boldsymbol{u}_d \tag{1}$$

$$\text{s.t.} \quad \sum_d^M \boldsymbol{w}_d^{\intercal}\boldsymbol{\mu} + \theta\sqrt{\boldsymbol{w}_d^{\intercal}\boldsymbol{\Sigma}\boldsymbol{w}_d} \le A, \tag{2}$$

$$\sum_d^M p_d \cdot \alpha \boldsymbol{w}_d^{\intercal}\boldsymbol{\mu} \le B, \tag{3}$$

$$C_{\min} \le \boldsymbol{w}_d^{\intercal}\boldsymbol{\mu} + \theta\sqrt{\boldsymbol{w}_d^{\intercal}\boldsymbol{\Sigma}\boldsymbol{w}_d} \le C_d, \forall d, \tag{4}$$

$$\sum_d^M w_{id} = 1, \forall i \tag{5}$$

The decision variables are w_{id}, i.e. the proportion of requests directed to datacenter d from node i. The objective (1) calculates the system-wide utility given by the datacenter selection matrix $\boldsymbol{w}$ and the performance matrix $\boldsymbol{u}$. Constraint (2) is the total bandwidth usage constraint, where A is the bandwidth cap. (3) is the total electricity cost constraint. It enforces that the total electricity cost of serving all the requests should not exceed the budget B. α is a conversion factor that converts workload in Gbps into electricity consumption in KWh. Constraint (4) represents both the load balancing and capacity constraints at individual datacenters, where $C_{\min}$ is the minimum load that each datacenter must achieve, and C_d is the capacity of datacenter d. Constraint (5) corresponds to the simple fact that all the requests arriving at node i should be served, i.e. our algorithm is work-conserving.

III. A Decentralized Implementation

The optimization problem DC-OPT is essentially a second-order cone program, and can be solved in polynomial time. However, this requires a central coordinator which introduces a single point of failure and is vulnerable to attacks. Further, the computational complexity of solving the cone program also increases significantly when the problem size scales up. A centralized solution also makes it less adaptive to sudden changes in traffic demand in a flash crowd scenario. Thus, for reasons of reliability, security, scalability, and performance, we are motivated to develop distributed solutions in which the datacenters iteratively solve the optimization problem.

A. Dual Decomposition

Relax the constraints (2), (3), and (5), we can obtain the Lagrangian of DC-OPT:

$$L(\boldsymbol{w},\lambda,\delta,\boldsymbol{\nu}) = \sum_d \boldsymbol{w}_d^{\intercal}\boldsymbol{u}_d + \lambda\left(A - \sum_d\left(\boldsymbol{w}_d^{\intercal}\boldsymbol{\mu} + \theta\sqrt{\boldsymbol{w}_d^{\intercal}\boldsymbol{\Sigma}\boldsymbol{w}_d}\right)\right)$$
$$+\delta\left(B - \sum_d p_d \cdot \alpha\boldsymbol{w}_d^{\intercal}\boldsymbol{\mu}\right) + \sum_i \nu_i\left(1 - \sum_d w_{id}\right),$$

where λ, δ, and $\boldsymbol{\nu}$ are the Lagrange multipliers associated with the bandwidth usage, electricity cost, and work conservation constraints, respectively. The dual function is then

$$g(\lambda,\delta,\boldsymbol{\nu}) = \begin{cases} \max\limits_{\boldsymbol{w}\succeq\boldsymbol{0}} & L(\boldsymbol{w},\lambda,\delta,\boldsymbol{\nu}) \\ \text{s.t.} & \text{constraint (4)} \end{cases} \tag{6}$$

To solve $g(\lambda,\delta,\boldsymbol{\nu})$, it is equivalent to maximizing the following objective

$$\sum_d \boldsymbol{w}_d^{\intercal}\left(\boldsymbol{u}_d - \lambda\boldsymbol{\mu} - \delta\alpha p_d\boldsymbol{\mu} - \boldsymbol{\nu}\right) - \theta\lambda\sqrt{\boldsymbol{w}_d^{\intercal}\boldsymbol{\Sigma}\boldsymbol{w}_d}$$

where the constant terms in $L(\boldsymbol{w},\lambda,\delta,\boldsymbol{\nu})$ can be safely removed. The key observation here is that it can be decomposed into M per-datacenter maximization sub-problems

$$\max_{\boldsymbol{w}_d\succeq\boldsymbol{0}} \; \boldsymbol{w}_d^{\intercal}(\boldsymbol{u}_d - \lambda\boldsymbol{\mu} - \delta\alpha p_d\boldsymbol{\mu} - \boldsymbol{\nu}) - \theta\lambda\sqrt{\boldsymbol{w}_d^{\intercal}\boldsymbol{\Sigma}\boldsymbol{w}_d}$$
$$\text{s.t. constraint (4),} \tag{7}$$

The per-datacenter sub-problem naturally embodies an economic interpretation. Each datacenter d strives to maximize the total utility of serving the requests, discounted by the costs of violating the bandwidth, electricity cost, and work conservation constraints, as priced by the Lagrange multipliers. It is still a second-order cone program. However the problem size has been reduced. The per-datacenter sub-problem has only N variables and 2 constraints. In a typical production cloud, the number of mapping nodes N is on the order of hundreds, which can be solved efficiently by standard optimization solvers with the computing power of a datacenter.

B. A Distributed Algorithm

We have shown that the dual function of DC-OPT can be decomposed into M per-datacenter maximization problem, which is a smaller second-order cone program. Now we need to solve the dual problem

$$\begin{aligned} \min \;\; & g(\lambda, \delta, \boldsymbol{\nu}) \\ \text{s.t.} \;\; & \lambda \geq 0, \delta \geq 0. \end{aligned} \tag{8}$$

The subgradient method [17] can be used to solve the dual problem. The updating rules for the dual variables are as follows:

$$\lambda^{(l+1)} = \left[\lambda^{(l)} + \rho^{(l)} \left(\sum_d \left(\boldsymbol{w}_d^{\intercal}\boldsymbol{\mu} + \theta\sqrt{\boldsymbol{w}_d^{\intercal}\boldsymbol{\Sigma}\boldsymbol{w}_d}\right) - A\right)\right]^+, \tag{9}$$

$$\delta^{(l+1)} = \left[\delta^{(l)} + \eta^{(l)} \left(\sum_d p_d \alpha \boldsymbol{w}_d^{\intercal}\boldsymbol{\mu} - B\right)\right]^+, \tag{10}$$

$$\nu_i^{(l+1)} = \nu^{(l)} + \sigma_i^{(l)} \left(\sum_d w_{id} - 1\right), \forall i, \tag{11}$$

where $[x]^+$ represents $\max\{0, x\}$, and $\rho, \eta, \boldsymbol{\sigma}$ are the *step sizes*. According to [17], the above procedure is guaranteed to converge as long as the following condition is satisfied.

Proposition 1: The subgradient updates as in (9)–(11) converge to the optimal dual variables if a diminishing step size rule is followed for choosing $\rho, \eta, \boldsymbol{\sigma}$ [17].

The dual variables $\lambda, \delta, \boldsymbol{\nu}$ serve as price signals to coordinate the resource consumption and workload conservation. For example, when the 95-percentile bandwidth of all datacenters exceeds the bandwidth cap, i.e. $\sum_d \boldsymbol{w}_d^{\intercal}\boldsymbol{\mu} + \theta\sqrt{\boldsymbol{w}_d^{\intercal}\boldsymbol{\Sigma}\boldsymbol{w}_d} > A$, the cloud increases its price λ for the next iteration to suppress the excessive traffic. The process continues until it converges to the optimal resource allocation.

Dual optimization by the subgradient method can be done in a *distributed* fashion because of dual decomposition. First, in each iteration, the per-datacenter sub-problems (7) can be solved concurrently by individual datacenters. Second, subgradient updates can also be distributively performed by each datacenter and mapping node. Here λ and δ need to be updated with global information from all datacenters. This can be done in a distributed way as follows, using δ as an example. Initially, the previous $\delta^{(l)}$ is made common knowledge among the datacenters. First, a datacenter is randomly chosen and given a token with the total budget B. It calculates its own electricity cost of serving the requests $p_d \alpha \boldsymbol{w}_d^{\intercal}\boldsymbol{\mu}$, and deduct this amount from B. It puts a mark in the token, and pass it on to the next datacenter, who also updates the remaining budget, marks the token, and passes it further down. A datacenter determines it is the last one in the loop by examining that except itself, everyone else has marked the token. It thus updates the remaining budget, calculates the updated budget price $\delta^{(l+1)}$, and broadcasts to each datacenter. Finally, ν_i can be updated by each mapping node, with w_{id} received from each datacenter.

Algorithm 1 *Optimal Distributed DC-OPT Algorithm*

1. Initialize $\lambda^{(0)}$ and $\delta^{(0)}$ to 0. Each node initializes $\nu_i^{(0)}$.
2. Each datacenter collects $\boldsymbol{\nu}^{(l)}, \delta^{(l)}$, and independently solves the per-datacenter subproblem (7) using standard optimization solvers and obtain $\boldsymbol{w}_d$, which is broadcast to each node.
3. Each node performs a subgradient update for $\nu_i^{(l)}$ as in (11). The updated $\nu_i^{(l+1)}$ is broadcast to datacenters.
4. A datacenter is randomly chosen and given a token with the bandwidth cap A and budget B.
5. The datacenter deducts its bandwidth usage and electricity cost from the remaining bandwidth cap and budget respectively in the token, marks it, and passes it down.
6. Repeat step 5 until the last datacenter calculates the final remaining bandwidth cap and budget, updates $\lambda^{(l)}$ and $\delta^{(l)}$ as in (9) and (10), and broadcasts to every datacenter.
7. Return to step 2 until convergence.

The complete distributed algorithm is shown in Algorithm 1. Since it optimally solves the dual problem (8), it optimally solves the primal problem DC-OPT because the duality gap for convex optimization problems is zero.

Theorem 1: The distributed algorithm as shown in Algorithm 1 always converges, and when it converges its solution *optimally* solves the datacenter selection problem DC-OPT.

IV. Evaluation

We present our simulation studies in this section.

A. Setup

1) Demand matrix: To represent the request traffic for a cloud service, we use real-world traces collected from UUSee Inc. [11], a major online multimedia provider with servers deployed in different geographical regions in China. The dataset contains, among other information, the bandwidth demands for UUSee video programs sampled every 10 minutes, in a 12-day period during the 2008 Beijing Olympics. Although the scale of the UUSee infrastructure may not be as large as that of a cloud provider, we believe the traces faithfully reflect the demand distribution for a cloud service, and it is appropriate to use them for the purpose of benchmarking the performance of our datacenter selection algorithm.

We assume that the prediction of mean and covariance of traffic demands can be done accurately [14], [15], and in the simulation we simply adopt the predicted values for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. We use the traffic demands of distinct video channels to represent demands of distinct mapping nodes. We simulate a cloud with 100 mapping nodes. Since the data is collected every 10 minutes, the optimization epoch is also set to 10 minutes. The bandwidth cap A is set to 2000 Mbps. The minimum load of a datacenter $C_{\min}$ is 100 Mbps, and the capacity of datacenters C is randomly drawn.

2) Datacenter placement and price matrix: To capture the location diversity of the cloud infrastructure and electricity market, we assume the datacenters are deployed across the continental U.S. For the ease of exploration, we assume that there is one datacenter in a randomly chosen hub in each regional electricity market as shown in Fig. 5 [13]. We use the 2011 annual average day-ahead on peak price ($/MWh) at these regions provided by the Federal Energy Regulatory Commission (FERC) as the electricity price for each datacenter, i.e. p_d, as summarized in Table I [13].

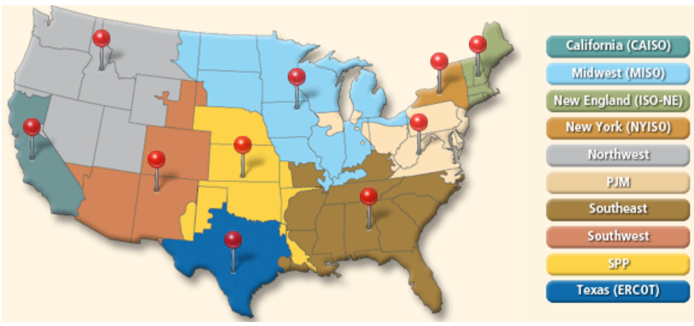

Fig. 5. The U.S. electricity market and our cloud datacenter map. Source: FERC [13].

TABLE I
2011 ANNUAL AVERAGE DAY AHEAD ON PEAK PRICE ($/MWH) IN DIFFERENT REGIONAL MARKETS. SOURCE: FERC [13].

Region	Hub	Price
California	NP15	$35.83
Midwest	Michigan Hub	$42.73
New England	Mass Hub	$52.64
New York	NY Zone J	$62.71
Northwest	California-Oregon Border (COB)	$32.57
PJM	PJM West	$51.99
Southeast	VACAR	$44.44
Southwest	Four Corners	$36.36
SPP	SPP North	$36.41
Texas	ERCOT North	$61.55

3) Performance matrix: We consider a utility function defined by the negative Euclidean distance between the mapping nodes and the datacenters. This definition instructs the algorithm to direct requests to datacenters in the geographical vicinity of a mapping node whenever possible, in an effort to minimize the transmission delay and optimize viewer experience. To calculate the performance matrix, we first obtain the longitude and latitude of ten counties near each of the ten hubs as the *exact* locations of our datacenters in the U.S. We then randomly choose another 100 counties as the locations of the 100 mapping nodes. All the location information is obtained from [18]. The Euclidean distance between any given pair of mapping node and datacenter then can be readily calculated, which constitutes the performance matrix $\boldsymbol{U}$. Without loss of generality, we assume that $\alpha = 0.01$, i.e. serving 1 Mbps per epoch of 10 minutes consumes 0.01 kWh electricity. The budget B is set to $4 per epoch.

4) Benchmark: Finally, we use a bandwidth-agnostic datacenter selection scheme that shares the same objective function (1) and constraints (3)–(5), except that it does not consider the bandwidth usage, i.e. constraint (2), as the benchmark for the performance of DC-OPT. This problem is also a second-order cone program and can be efficiently solved. This is referred to as *Benchmark* in the following.

B. Effectiveness

We evaluate the effectiveness of our distributed datacenter selection algorithm. Fig. 6 shows the 95-percentile bandwidth consumption of each datacenter for a 100-epoch period of time. We observe that, compared to the bandwidth-agnostic benchmark, DC-OPT reduces the bandwidth usage of most datacenters by 15%–20% by intelligently mixing negatively correlated demands. One may notice that datacenter 4, 6, 8, and 10 have the same bandwidth usage using both algorithms. This is due to the unattractive electricity price and performance at these locations. From Table I, we observe that datacenter 4, 6, and 10 have the highest electricity prices among all locations. Also from our performance matrix we observe that datacenter 8 is far away from many of the nodes. This prevents both DC-OPT and Benchmark from directing requests to these locations beyond the minimum load of 100 Mbps required by the load balancing constraint (4).

Fig. 7 demonstrates the average utility comparison between DC-OPT and Benchmark. We observe that DC-OPT has a slightly worse average utility across the time. The reason for the inferior performance is that in order to reduce the bandwidth usage, sometimes DC-OPT needs to direct requests to locations that are not necessarily the closest, but are more bandwidth efficient because these demands effectively smooth out the aggregated traffic.

By the same token, DC-OPT has to sacrifice the electricity cost in order to satisfy the bandwidth usage constraint. This is illustrated in Fig. 8. The electricity cost is on average around 5%-10% higher than Benchmark. Note that both DC-OPT and Benchmark violate the cost constraint during peak hours when demand rises to the point that this constraint becomes infeasible during epoch 70–85. The average performance at this period of time is also relatively worse as seen in Fig. 7.

The results show that there is an inherent trade-off between bandwidth usage and performance/electricity cost. Essentially, DC-OPT strives to be more bandwidth efficient, and achieves a different operating point on the trade-off curve. According to [7], both the electricity and wide-area bandwidth account for around 15% of the datacenter costs, respectively. Thus the overall cost of DC-OPT is reduced, while the performance is comparable to when bandwidth usage is not considered. DC-OPT represents a favorable solution than bandwidth-agnostic datacenter selection schemes for cloud operators.

V. RELATED WORK

The topic of datacenter selection and load direction for a geo-distributed cloud has started to gain attention in the research

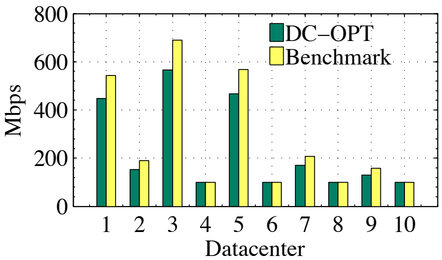

Fig. 6. 95-percentile bandwidth comparison.

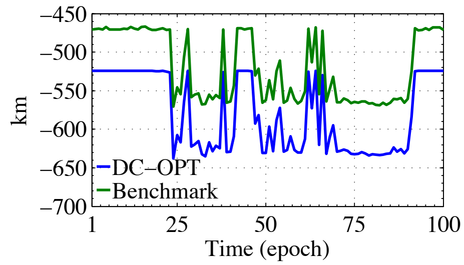

Fig. 7. Average utility comparison.

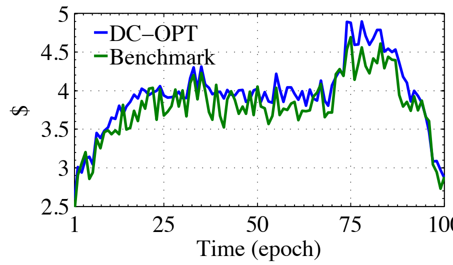

Fig. 8. Total cost comparison.

community. Qureshi et al. [5] introduced an intuitive idea of utilizing the location diversity of electricity spot price to intelligently direct requests to datacenters with lower prices. Wendell et al. [3] developed a decentralized datacenter selection algorithm for cloud services, and evaluated its performance using a prototype and realistic traffic traces. Rao et al. [6] considered a joint load balancing and power control problem for Internet datacenters to exploit the time and location diversity of electricity price. [19] specifically considered the effect of geographical load balancing on providing environmental gains by encouraging the use of green energy. [20] studied a complementary problem of data placement in a geo-distributed cloud, considering the data locality. These works, however, do not consider bandwidth usage in their problem formulations.

Our work relies on the idea of multiplexing demands with different degrees of correlation to reduce the peak aggregated demand of datacenters. Similar idea has been proposed in some recent works [14], [15], where a bandwidth reservation service in the cloud is envisioned for VoD applications, and multiplexing is utilized to reduce the total bandwidth reservation for a given level of QoS. Here we consider a more general setting where multiplexing is used for reducing the operating cost of the cloud. Another recent work [21] discusses correlation aware power optimization in datacenters. The focus is on local-area traffic in a datacenter network whose correlation statistics change frequently, while our approach deals with wide-area egress traffic of the datacenter. We also take into account the geographical diversity of electricity cost which is not considered in these works.

VI. Concluding Remarks

In this paper, we presented a general optimization framework that considers bandwidth usage and electricity costs to solve the datacenter selection problem for cloud services. Our idea is to exploit the different degrees of correlations between demands at different locations to reduce the peak demand of aggregated traffic at datacenters, thereby reducing the billing amount of wide-area bandwidth. We adopted a dual decomposition approach to solve the second-order cone program, and developed a distributed algorithm based on the subgradient method to iteratively achieve the optimal datacenter selection solution in a decentralized fashion. Simulation results with real-world traces and electricity prices show that our algorithm reduces the 95-percentile bandwidth usage by 20%, and offers comparable performance and electricity cost against bandwidth-agnostic solutions. Our work can be extended in many directions. One possible direction is to consider the online datacenter selection that makes decision on-the-fly with sequentially arriving requests, which is more difficult than the offline problem we solved in this paper.

References

[1] "Amazon web services (AWS)," http://aws.amazon.com/.
[2] "DynDNS," http://dyn.com/dns/.
[3] P. Wendell, J. W. Jiang, M. J. Freedman, and J. Rexford, "DONAR: Decentralized server selection for cloud services," in *Proc. ACM SIGCOMM*, 2010.
[4] "Amazon AWS elastic load balancing," http://aws.amazon.com/elasticloadbalancing/.
[5] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs, "Cutting the electricity bill for Internet-scale systems," in *Proc. ACM SIGCOMM*, 2009.
[6] L. Rao, X. Liu, L. Xie, and W. Liu, "Minimizing electricity cost: Optimization of distributed Internet data centers in a multi-electricity-market environment," in *Proc. IEEE INFOCOM*, 2010.
[7] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The Cost of a Cloud: Research Problems in Data Center Networks," *SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 1, pp. 68–73, 2009.
[8] X. Dimitropoulos, P. Hurley, A. Kind, and M. P. Stoecklin, "On the 95-percentile billing method," in *Proc. PAM*, 2009.
[9] N. Laoutaris, M. Sirivianos, X. Yang, and P. Rodriguez, "Inter-datacenter bulk transfers with netstitcher," in *Proc. ACM SIGCOMM*, 2011.
[10] D. Xu and X. Liu, "Geographic trough filling for Internet datacenters," in *Proc. IEEE INFOCOM*, 2012.
[11] "UUSee Inc." http://www.uusee.com/.
[12] H. Alzoubi, S. Lee, M. Rabinovich, O. Spatscheck, and J. van der Merwe, "A practical architecture for an anycast CDN," *ACM Trans. Web*, vol. 5, no. 4, October 2011.
[13] Federal Energy Regulatory Commission, "U.S. electric power markets," http://www.ferc.gov/market-oversight/mkt-electric/overview.asp, 2011.
[14] D. Niu, H. Xu, B. Li, and S. Zhao, "Risk management for video-on-demand servers leveraging demand forecast," in *Proc. ACM Multimedia*, 2011.
[15] ——, "Quality-assured cloud bandwidth auto-scaling for video-on-demand applications," in *Proc. IEEE INFOCOM*, 2012.
[16] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *IEEE/ACM Trans. Netw.*, vol. 8, no. 5, pp. 556–567, October 2000.
[17] S. Boyd and A. Mutapcic, "Subgradient methods," Lecture notes of EE364b, Stanford University, Winter Quarter 2006-2007. http://www.stanford.edu/class/ee364b/notes/subgrad_method_notes.pdf.
[18] "WebGIS," http://www.webgis.com/.
[19] Z. Liu, M. Lin, A. Wierman, S. H. Low, and L. L. Andrew, "Greening geographical load balancing," in *Proc. ACM Sigmetrics*, 2011.
[20] S. Agarwal, J. Dunagan, N. Jain, S. Saroiu, A. Wolman, and H. Bhogan, "Volley: Automated data placement for geo-distributed cloud services," in *Proc. USENIX NSDI*, 2010.
[21] X. Wang, Y. Yao, X. Wang, K. Lu, and Q. Cao, "CARPO: Correlation-aware power optimization in data center networks," in *Proc. IEEE INFOCOM*, 2012.