Cost Efficient Design of Fault Tolerant Geo-Distributed Data Centers

Rakesh Tripathi, S. Vignesh and Venkatesh Tamarapalli
Department of CSE, IIT Guwahati, Assam, India.
{t.rakesh, s.vignesh, t.venkat}@iitg.ernet.in
Deep Medhi
Department of CSEE, UMKC, Kansas City, USA.
dmedhi@umkc.edu

***Abstract*—Many critical e-commerce and financial services are deployed on geo-distributed data centers for scalability and availability. Recent market surveys show that failure of a data center is inevitable resulting in huge financial loss. Fault-tolerance in distributed data centers is typically handled by provisioning spare capacity to mask failure at a site. We argue that operating cost and data replication cost (for data availability) must be considered in spare capacity provisioning along with minimizing the number of servers. Since the operating cost and client demand varies across space and time, we propose cost-aware capacity provisioning to minimize the total cost of ownership (TCO) for fault-tolerant data centers.**

We formulate the problem of spare capacity provisioning in fault-tolerant distributed data centers using mixed integer linear programming (MILP), with an objective of minimizing the TCO. The model accounts for heterogeneous client demand, data replication strategies (single and multiple site), variation in electricity price and carbon tax, and delay constraints while computing the spare capacity. Solving the MILP using real-world data, we observed a saving in the TCO to the tune of 35% compared to a model that minimizes total number of servers and 43% compared to the model which minimizes the average response time. We demonstrate that our model is beneficial when electricity price, carbon tax and bandwidth price vary significantly across the locations, which seems to be the case with most of the operators.

***Index Terms*—Geo-distributed data center, capacity provisioning, fault tolerance, mixed integer linear programming**

I. Introduction

Recently a number of Internet services and applications are deployed over large scale geo-distributed data centers. A geo-distributed data center is an orchestrated collection of data centers, distributed across several locations and transparently interconnected with overlay links[1]. Geo-distributed data centers offer advantages such as increased availability, lower access time for users across the globe and horizontal scale out against capacity constraints (electricity, physical space etc.). Due to these advantages several cloud providers like Amazon and content distribution companies such as Akamai, invest in building geo-distributed data centers. For example, Google has data centers across 15 countries at more than 30 sites with an estimated 900,000 servers [2].

Critical e-commerce and financial services running on geo-distributed data centers (*henceforth simply referred data centers*) demand high availability because of huge loss of revenue associated with downtime. A survey by Gartner estimated that 60% companies incurred a loss to the tune of $250,000-$500,000 for an hour of downtime, and one sixth of the companies incurred a loss of $1 million or more [3]. Further, a latest survey by Ponemon institute showed that the frequency of data center outage (complete or partial) could be as high as once a month with an average duration of three hours. It was reported to cause a loss of $1,734,433 per organization with an average cost of $690,204 per incident. Instances of a data center failure at a site have been reported by many cloud service providers like Amazon, Facebook and Google [4], [5]. These failures are attributed to various reasons like power outages, cable cuts, software bugs, mis-configured routers, DDoS attacks, and natural disasters [6]. In this paper, by high availability we mean that the data center continues to deliver original service (may be with a degraded performance) after failure of a single site. This can be achieved by providing spare compute capacity across the data centers.

Along with service restoration, it is also important that the required data is available at an alternate location after failure. This is handled by replication of data according to a pre-determined policy. There are two options possible for data replication namely, single site replication and multiple site replication. In single site replication, the data is replicated to another nearby data center. In case of a failure, if the replicated site is overloaded, client requests are directed to any other data center meeting the latency requirement. In this case, the data would be pulled from the replica which results in greater latency and bandwidth cost (we call this post-failure penalty). In order to ensure co-location of data with the compute servers, the data is often replicated at multiple sites. However, multi-site replications involves large replication cost since the data center operators are typically charged for the number of bytes transfered [7] and/or the bandwidth cost between the replication sites [8]. Therefore, the replication cost should be considered while designing the data centers for high availability.

In summary, designing a fault-tolerant, highly available, distributed data center involves minimizing the spare capacity (number of servers) across the data centers considering the cost of power consumed and data replication, subject to a set of constraints related to client demand, delay bound, power and capacity available. We call this problem cost-aware

capacity provisioning (CACP) wherein, the main challenge is to minimize the total cost of ownership (TCO) for data center operators by leveraging the spatio-temporal variation in electricity price and user demand.

Motivation: The work in this paper is motivated by the following observations:

- *Electricity price variation:* In a deregulated electricity market, electricity price varies across space and time. Recent trends show that the operating cost exceeds the server cost at many data center locations. Assuming the server shelf life to be 4 years and its initial cost to be $\$2000$ [9], we define the energy to acquisition cost (EAC) as the ratio of the cost of powering a server to its acquisition cost as:

$$\begin{aligned}\text{Power cost} = 4 \text{ yrs} * (8760 \text{ hrs/yr}) * \\ (\text{electricity price}) * \text{server power} * \text{PUE}\end{aligned} \tag{1}$$

$$\text{EAC} = \frac{\text{power cost}}{\text{server cost}} * 100 \tag{2}$$

Using the electricity prices from [10]–[13], an average server power consumption of 300W, and power usage effectiveness (PUE) of 1.5 we compute the EAC values that are reported in Table I. The EAC values in Table I indicate that for most of the countries cost of power and cooling exceeds the cost of buying servers which suggests that a greater attention should be put on optimizing data center power consumption cost instead of only minimizing the servers while designing fault-tolerant data centers.

Country/Area	Electricity price($/kWh)	Cost(in $)	EAC
Canada	0.06	946	47
Oregon, USA	0.06	946	47
Virginia, USA	0.07	1104	55
Switzerland	0.07	1230	62
Netherlands	0.09	1419	71
Japan	0.10	17	84
California, USA	0.12	1971	99
Ireland	0.13	2050	103
UK	0.13	2050	103
Hongkong	0.17	2680	134

TABLE I: EAC for different countries

- *Replication Cost* Usually, cloud service providers connect their data centers with dedicated WAN links which are significantly expensive. Therefore, informed data replication must be carried out in order to minimize the operating cost involved. For example, AWS charges inter data center transfer for $0.12-0.2/GB across geographic regions and $0.01/GB in the same region [7]. Literature also suggests that replication cost may be charged based on distance between the replicating sites, like a cost of $1 to transfer 2.7 GB of data over 100km was reported in [8].

In this paper we give a MILP based solution for the CACP problem to optimize the TCO, while complying to the customer demand, latency requirements, and being cost effective while masking the failure of any one data center.

We summarize the main contributions as follows:

- We formulate the CACP problem as a mixed integer linear program with an objective of minimizing the TCO (includes cost of server acquisition, electricity, carbon tax and data replication) subject to latency, power, demand and availability constraints.
- We prove that the CACP problem for the design of a fault tolerant distributed data center is NP-hard.
- We collected traces from wikipedia.org [14] sites to create heterogeneous workload and used it to model the server utilization under heterogeneous demand.
- We modelled two strategies for data replication, single site and multiple site for data affinity. Evaluation of these models with our framework suggests that although multiple site model is costlier, it is preferable when post failure penalty is large in single site model.
- We used real-world data for price of electricity to evaluate the proposed model which shows that the CACP model results in significant savings in the TCO compared to the existing models.

The rest of the paper is organized as follows. Section II discusses the work related to capacity provisioning in distributed data centers. Section III presents the cost models used, formulation of the CACP problem and discusses the complexity of the formulation. We also illustrate the working of the model with a small example. Results demonstrating the advantages of the proposed model over the existing ones are reported in Section IV. Section V concludes the paper.

II. Related Work

There have been significant efforts to address the problems of server consolidation, server switching, VM migration, and load balancing to minimize the operating cost under the assumption that sufficient servers are already provisioned (for example [15], [16]. However, there is not much work done in planning data center deployment considering failures and the offline problem of capacity allocation to design fault-tolerant data centers. This section discusses the literature addressing server placement and capacity provisioning in geo-distributed data centers.

The authors of [8] proposed an optimization framework to provision servers across different locations using three different objectives: to minimize total carbon footprint, to minimize total cost and to minimize average service latency. The costs considered were electricity cost and bandwidth cost, while the constraints were related to client latency. In [9], the authors proposed a mechanism to select data center locations to minimize the total cost of ownership that includes capital as well as operating cost subject to delay, consistency and availability constraints. The capital cost factors included cost of land, data center construction, transmission line to power grid, OFC line to network backbone, cooling infrastructure, and internal network. The operating cost factors included cost of electricity, bandwidth, cooling the data center, carbon tax, and administration cost.

The work in [17], addressed the problem of maximizing profit by either building a new data center or by expanding the existing ones (increase the number of servers) to meet the increasing demand. Along with the cost of electricity, cooling, bandwidth and revenue generated, annual inflation rate over a period of time was also included in the profit generated.

The MILP optimization framework determines the best option to maximize revenue for a given data center location and its compute capacity . The work in [18] jointly handled the data center server placement, capacity provisioning and request routing while minimizing the total cost (adding/expanding data center), such that maximum average latency on every routing path is bounded.

A design for disaster-resilient data center using quality of resilience metrics like time to recovery and service availability was discussed in [19]. They also showed how these metrics affect the decision of selecting data center recovery mechanism, VM placement, site location, and backup strategies. Though a lot of literature addressed the problems of data center placement and capacity provisioning, provisioning to handle data center failures had not been adequately addressed. To the best of our knowledge, the only work that advocated the importance of fault tolerant capacity provisioning in distributed data center was [20]. The authors, proposed an optimization model to minimize the number of servers to be provisioned across different locations to handle the failure of an entire data center at a site. Though the basic problem is similar, we use minimization of the TCO as the objective apart from handling replication cost. As reported in Table I the electricity cost of powering a server is comparable to the server acquisition cost (and dominates it in some countries). Therefore, we use minimization of the TCO as an objective in spare capacity provisioning considering different models for data replication as well.

III. MILP Model Formulation

In this section, we first state the assumptions used in the model and present the models considered for various cost factors. Next, we present the MILP formulation of the CACP problem and also prove that the problem is NP hard.

A. Assumptions

The following assumptions were made in the model.

- We assume that the failure of the data center at a site is an independent process, *i.e.*, data centers are not susceptible to common disaster situation [19]. For example, power outage, building fire or any local disaster at one data center location will not effect the remaining data centers.
- Data replication happens with any popular geo-distributed data replication strategy.
- Failure detection and request re-routing is handled by the load balancer proxy.
- Data centers are connected using dedicated virtual links and the cost of data transfer is based on the actual usage.
- The demand from a client region is proportional to the population. Propagation delay within the client region is assumed to be negligible.
- All the servers have similar configuration and can serve requests for any service. However, the response sizes can be variable.

Variable	Meaning
Input Parameters	
S	set of data center locations
U	set of client locations
A	set of application types
H	total time horizon
s	index for data center location
u	index for client region
f	index for failed data center
h	index for time slot in time horizon
a	index for application request type
B	processing rate of server in bits per second
J_a	job size for request of type a in kB
P_s^{fh}	power consumed at data center s for application a during hour h with failed data center f
$P_s^{h\ max}$	maximum power available at data center s during hour h
γ_s^{fh}	average server utilization at data center s during hour h and failed data center f
γ^{max}	maximum value of γ to avoid waiting
L_u^{ah}	total number of requests generated for application A from user location u during hour h
D_{su}	propagation delay between client region u and data center s
D_{max}	the maximum tolerable latency
θ_s^h	electricity price per kWh at data center s at hour h
ρ_s	transmission loss of electricity at data center s
α	server acquisition cost
δ_s	carbon tax at data center s
M^{min}	minimum number of servers at any data center
M^{max}	maximum number of servers at any data center
ν_{si}	bandwidth cost for data center s to data center i
ξ	number of bytes required for data replication of single request
Decision Variables	
m_s	number of servers in data center s
λ_{su}^{afh}	number of requests for application a from user location u, served by data center s during hour h and failed data center f
y_{su}	binary variable that denotes whether client location u lies within the latency bound of data center s
Cost Components	
F	total cost of ownership, including server acquisition cost, operating cost and data replication cost
Φ	server acquisition cost
η	cost of data replication to nearest data center for durability
κ	cost of data replication, for the case of replication at multiple sites
Θ	power consumption cost
τ	carbon tax incurred

TABLE II: Summary of notation used in the paper

B. System Model

In this section, we define the variables and cost models used in the formulation. Table II lists all the input parameters, variables, and cost factors in the model.

Failures: Let S denote the set of data centers. The data centers are indexed between 1 and $|S|$. We use an index variable f to represent failure of a data center. f takes values from the set $\{0, |S|\}$, where $f = 0$ indicates the case of no failure and $f = s$ indicates that the data center indexed $s \in \{1, 2, \ldots, |S|\}$ has failed. We assume that the probability of single data center failure, i.e., $f \neq 0$, is very small.

Demand: Let λ_{su}^{afh} denote the number of requests for an application type a, from a client region u, served by the data center at site s, during hour h after the data center indexed $f \in \{1, |S|\}$ has failed. Let L_u^h be the total demand from client

region u at hour h.

Server Provisioning: Let m_s denote the number of servers required in a data center at s. We define M^{min} and M^{max} to be the minimum and maximum number of servers that can be provisioned at any data center based on the space and power availability.

Delay: Let D_{max} be the maximum latency for the service and Let D_{su} be the propagation delay between client region u and data center site s. A data center must be assigned to the client region such that even after the failure of a site, the latency continues to be lower than D_{max}.

Server Utilization: Let the processing rate of the server to be B bps and let J_a be the response size for an application type $a \in A$. The service rate for type a is defined by $\frac{B}{J_a}$ requests per second. There are three approaches to model the average utilization of servers as given below:

1) Mutually Exclusive (ME) approach: Each type of application is assigned to a pre-defined set of servers. Let m_{sa} be the number of servers allocated to serve the requests of type a. Requests for different services are queued in a single queue, from which a scheduler dispatches the requests of a type to the corresponding servers. The average utilization of servers serving the requests of type a can be defined as

$$\gamma_s^{fh} = \frac{\sum_u \lambda_{su}^{afh} J_a}{m_{sa} B}, \tag{3}$$

This approach of scheduling simplifies the resource provisioning but leads to under-utilization of servers.

2) Maximum (MAX) approach: Assuming all the requests to be homogeneous, the servers can be provisioned according to the highest processing rate required. In this case, the average utilization of any server can be defined as

$$\gamma_s^{fh} = \frac{\sum_{u,a} \lambda_{su}^{afh} J_{max}}{m_s B}, \tag{4}$$

where J_{max} is the maximum mean file size across different application types. This approach also suffers from resource under-utilization. On the other hand, provisioning based on the smallest processing rate required leads to under-provisioning of resources.

3) Multiplexed (MUX) approach: In a virtualized environment, any type of workload can be served by one of the free servers. All the requests are placed in a common queue and served by a set of identical servers. This model is followed in most of the recent papers [21]. The average utilization of a server in this case can be defined as

$$\gamma_s^{fh} = \frac{\sum_{u,a} \lambda_{su}^{afh} J_a}{m_s B} \tag{5}$$

In this paper, we consider this model for server utilization but study the implications of other models in Section IV-B8.

C. Cost Models

Server Acquisition: Let the cost of a server normalized over the duration considered for evaluation be denoted by α. The total cost of servers across all the data centers, denoted by Φ is simply given as

$$\Phi = \alpha \sum_s m_s \tag{6}$$

Data Replication: Let ν_{sg} be the bandwidth cost for data replication from data centers s to g. For every request served by a data center s, let ξ be the volume of data to be replicated. We consider two possible replication models.

1) Single site replication: In this case, the data from a primary data center is replicated to the nearest data center.
2) Multiple site replication: In this case, the data from a primary data center is replicated to all possible data centers where the client's request may be routed without exceeding the latency bound, denoted by PD_s.

We define the cost of replication R for these two options using the equation below:

$$R = \begin{cases} \sum_{a,f,u,s,h} \left(\lambda_{su}^{afh}\ \xi\ \nu_{sg}\right) & \text{Case 1} \\ \sum_{a,f,u,s,h} \left(\lambda_{su}^{afh}\ \xi \sum_{i \in PD_s} \nu_{si}\right) & \text{Case 2} \end{cases} \tag{7}$$

Power Consumption: Let θ_s^h denote the electricity price at data center location s in hour h of the day. Let P_{idle} be the average power consumed in idle condition and P_{peak} be the power consumed at peak utilization. Let E_s be the power usage effectiveness (PUE) of a data center. The total power consumed at s over an hour h can be expressed as [8]

$$P_s^{fh} = m_s(P_{idle} + (E_s - 1)P_{peak}) + m_s(P_{peak} - P_{idle})\gamma_s^{fh}. \tag{8}$$

The cost of power consumed at all data centers Θ can be expressed as

$$\Theta = \sum_{s,h,f} \theta_s^h P_s^{fh} \tag{9}$$

$$\tag{10}$$

Carbon tax: Let δ_s denote the carbon tax levied at data center location s and ρ_s denote the transmission loss incurred. The total cost due to carbon tax is

$$\tau = \sum_{s,f,h} \delta_s(\rho_s + 1)P_s^{fh} \tag{11}$$

D. CACP Model

Considering all the cost factors defined above, we can define the CACP problem as the problem of minimizing the TCO subject to set of constraints on latency and availability. The TCO, denoted by F is the sum of server cost Φ, data replication cost R, electricity cost Θ, and carbon tax τ. For notational simplicity, we define the following decision variables:

$\mathbf{m} \triangleq [m_s, \forall s \in S]$,

$\boldsymbol{\lambda} \triangleq [\lambda_{su}^{afu}, \forall s \in S,\ \forall u \in U,\ \forall a \in A,\ \forall h \in H, \forall f \in \{0, 1, 2.., S\}]$ and

$\boldsymbol{y} \triangleq [y_{su}, \forall s \in S,\ \forall u \in U]$

The CACP problem can be formally expressed as an optimization model given below:

$$\underset{\boldsymbol{m},\boldsymbol{\lambda},\boldsymbol{y}}{\textbf{minimize}} \quad \digamma = \Phi + R + \Theta + \tau \tag{12}$$

subject to,

$$\sum_{s\in S} \lambda_{su}^{afh} = L_u^{ah} \qquad \forall u,a,h,f \tag{13}$$

$$0 \leq \lambda_{su}^{afh} \leq y_{su} L_u^{ah} \qquad \forall s,u,a,h,f \tag{14}$$

$$M^{min} \leq m_s \leq M^{max} \qquad \forall s \tag{15}$$

$$P_s^{fh} \leq P_s^{h\ max} \qquad \forall s,h,f \tag{16}$$

$$2D_{su}\, y_{su} \leq D_{max} \qquad \forall s,u \tag{17}$$

$$\gamma_s^{fh} \leq \gamma^{max} \qquad \forall s,h,f \tag{18}$$

$$\lambda_{su}^{afh} = 0 \qquad \forall u,a,h,s=f \tag{19}$$

Among the constraints, Eq. (13) ensures that demand of all client regions in every hour is met. Eq. (14) ensures that all the client requests are served by data centers within the latency limit. Eq. (15) ensures that capacity limit of a data center (in terms of number of servers) is not exceeded. The constraint on the total power available at a data center is taken care in Eq. (16). Eq. (17) ensures that the delay experienced by a client lies within the maximum bound. Eq. (18) is used to limit the queuing delay at a data center by bounding the average server utilization to a constant value ($\gamma^{max} \in (0,1]$), similar to that in [8]. Eq. (19) ensures that no demand is served by the failed data center.

The inputs to the CACP problem are: the set of data center locations with the associated costs, maximum average utilization of servers, processing rate of the servers, maximum latency, demand distribution, maximum number of servers at each site, and maximum power available at each site. The model then gives the number of servers across the sites, request routing to the data centers and the data centers within the latency limit for each client location.

E. Complexity Analysis

The number of variables in the above formulation is $S + (S+1)SUAH$ and the number of constraints is $S + (S+1)\{UAH + SUAH + 2SH\} + SUAH + SU$. The asymptotic complexity of proposed CACP model is $\mathcal{O}(S^2UAH)$. With an increase in the number of data centers the complexity increases quadratically but, linearly with the number of client locations, time slots and application types. The following theorem states the complexity of the problem.

Theorem 1. *The feasibility problem of CACP in a distributed data center is NP-hard.*

Proof. The CACP problem in distributed data center (without fault tolerance) is NP-hard, even when resources are of unit size and unit operating cost. The reduction is from the set cover problem. Details of the proof are given in the Appendix.

Though the formulation is NP-hard, solving it is a one-time effort only at the time of design. We do not see the running time to be a matter of concern since the CACP problem is always solved offline. Currently, the number of data centers hosted by data center operators is small (15 for Google [22]). We solved all the models centrally using CPLEX with Matlab on a server with Intel Xeon processor, 64 GB of RAM and 64-bit OS. We could not solve the model for more than ten data centers on this server in a reasonable amount of time (few minutes) for an evaluation period of one day (24 hourly slots). We can solve the model optimally for capacity planning in large data centers with higher computational power. For much larger number of variables, we need to go for online heuristics or approximate algorithms.

F. Illustration for Working of the CACP Model

In this section, we give simple example to illustrate the impact of CACP model on TCO. The proposed CACP model mainly reduces the TCO by exploiting demand multiplexing and spatio-temporal variation in the demand and electricity price. For easier understanding on how this works, we show two examples for *(a)* the impact of demand multiplexing on capacity provisioning and *(b)* the impact of demand multiplexing and electricity price variation on the TCO.

Impact of demand multiplexing on capacity provisioning: Both the CACP and MS models take into account demand multiplexing while provisioning capacity while CDN trivially maps requests to nearest data center to minimize the latency. Consider a scenario with three data centers and three client regions with a maximum latency bound of 25 ms for the service. Fig. 1 shows the system used for illustration. Data centers DC_1,DC_2 and DC_3 serve the requests from client regions C_1,C_2 and C_3 given in Table III. Each edge between a data center and client region is weighted by the propagation delay. For simplicity, we considered a case where all the data centers lie within the latency bound (25ms) for all the client regions.

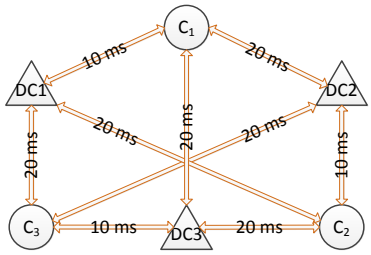
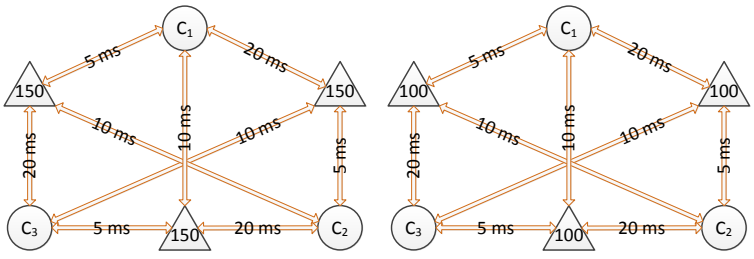


Fig. 1: System model

	Timeslot 1 (in hrs)	Timeslot 2 (in hrs)	Timeslot 3 (in hrs)
Client 1	100	50	50
Client 2	50	100	50
Client 3	50	50	100

TABLE III: Demand across different intervals

Considering the case of a data center failure, demand of 200 units generated from all the client regions need to be served by the remaining two data centers. In MS model we have equally distributed the workload among all the active data centers. This gives 100 servers at each data center and the total number of servers to tolerate any data center failure is



(a) (b)

Fig. 2: Capacity allocation using(a) CDN model, (b) MS model

300 units as shown in Fig. 2b. In case of CDN model, a client region is always served by the nearest data center after failure. For example, C_1 was served by DC_1 before failure whereas, it is served by DC_2 after failure. Therefore, DC_2 should be provisioned not only to satisfy C_2's demand but also with sufficient spare capacity to make up for failed data center DC_1. This gives rise to DC_2 being provisioned with 150 servers to meet the demand across any interval (when DC_1 might fail). Accounting for the possibility of any data center failure, the server distribution across all the data centers is obtained to be 150 units as shown in Fig. 2a. We can conclude that MS model exploits demand multiplexing while satisfying the latency bound of 25 ms and requires only 300 servers against 450 units with the CDN model. CACP model also gives the same result if we ignore the variation in the operating cost across the data centers.

Impact of demand multiplexing and electricity price variation on the TCO: Consider three data centers shown in Fig.1 with the electricity price variation as shown in Fig. 4. It may be noted that electricity price is highest at DC_2. The demand across the three regions C_1, C_2, and C_3, in shown in Fig. 3. For simplicity, we considered the processing rate as 100/sec, P_{peak} and P_{idle} as 400 W and 200 W, respectively and server cost as $2000 (17 cents/hr, assuming 4 years life). We assume that all the data centers are within the latency limit for any client region.

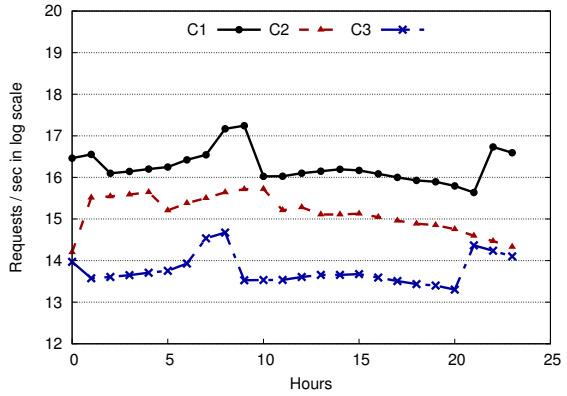


Fig. 3: Demand distribution at three chosen client regions

The server distribution obtained after solving the optimization model for CDN (minimize average latency), MS (minimize number of servers), CACP (minimize total cost) is showed

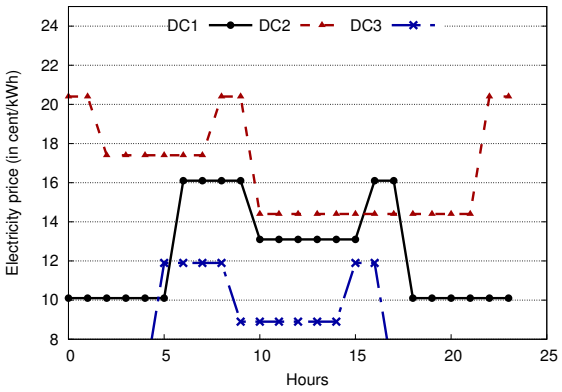


Fig. 4: Electricity price at three data center locations

in Fig. 5a, Fig. 5b and Fig. 5c, respectively. The number of servers allocated across all the data center locations is same with the MS model. However, the CACP model allocates fewer servers at DC_2, where the electricity price is higher. The CACP model always allocates more capacity at a site where electricity price is cheaper while satisfying the latency and other constraints.

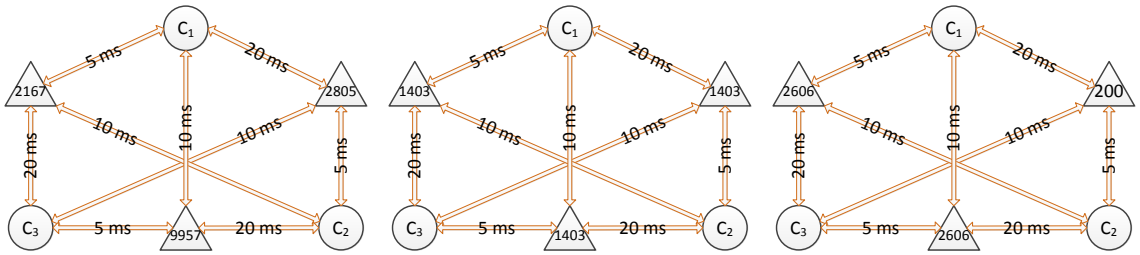
The normalized TCO obtained using the model is given in Table IV. Even though CACP model allocates larger number of servers than the MS model, the TCO is lowered by exploiting the spatio-temporal variation in the electricity prices for demand distribution. Though the MS model minimizes the number of servers provisioned at each location, it does not give minimum TCO being oblivious to operating cost.

Models	Normalized TCO
CDN	1
MS	0.89
CACP	0.62

TABLE IV: Normalized TCO

IV. Numerical Results

In this section, we solve the CACP model using real-world data and compare the TCO obtained with two other models from the literature. The MILP is solved using CPLEX (Interactive Optimizer 12.6.2.0.) with Matlab on a Ubuntu 14.04 server based on Intel Xeon processor with 64 GB of RAM. All the models were evaluated under identical constraints and



(a) (b) (c)

Fig. 5: Capacity allocation using(a) CDN model, (b) MS model, (c)CACP model

we used the same cost factors for all the models. The two other models considered were:

- MS model: A rudimentary version of this model was defined in [20]. The main objective of this model is to minimize the total number of servers deployed across all the data centers. The TCO for this model would be the cost of data center provisioned after minimizing the number of servers.
- CDN model: In this model, the objective is to balance the load across data centers such that average response time is minimized. The provisioning of servers in this model would be done such that the client latency is minimum [23].

We compare the TCO obtained using all the three models in the results. We also studied the advantages of the CACP model by varying the number of data centers, demand, request rate, and latency bound. We also studied the impact of server utilization models and replication models discussed earlier on the TCO. We first provide details on the scenarios used and the data set used for the evaluation.

A. Scenarios Used

Data center locations: The locations for the data centers were (10 of them): California, Oregon, Virginia, Switzerland, U.K, Ireland, Netherlands, Hong Kong, Japan and Canada. At each location, the number of servers could vary between 1000 and 100,000. This would help us evaluate smaller and mega data centers across the world.

Client locations: Based on the data collected for the number of Internet users from [24] we selected the following client regions (15 of them): Brazil, China, Egypt, France, Germany, India, Indonesia, Japan, Mexico, Nigeria, Russia, South Korea, UK, USA, and Vietnam. The propagation delay between the data center location and client location is assumed to vary linearly with geographical distance in the order of 10 ms for every 1000 km [8].

Electricity Prices: We used historical industrial electricity price data ($ per MWh) from publicly available government databases corresponding to various data center locations[10]–[13]. For the sake of brevity, we do not discuss regulated electricity market prices. Interested reader may see [10]. We use the electricity price model similar to the one in [17], where the price for each location varies during on-peak hours (7 A.M.-11 A.M. and 5 P.M.-7 P.M.), mid-peak hours (11 A.M-5 P.M.)and off-peak hours (7 P.M.-7 A.M.). The price varies across the periods by as much as 3 cents/kWh [17]. Some states in the USA like California and Colorado also add about $\$0.04$ to $\$0.6$/kWh as carbon tax for power consumed from brown energy sources. Though our model includes carbon tax, we ignore the same in results due to its small contribution in the TCO (less than 1%).

Traffic model: For the traffic we used the trace of requests to Wikipedia services downloaded from [14]. We downloaded the workload traces for the month of December 2015, containing the total number of requests and aggregate response size for different services of Wikipedia. The demand profile for a 24 hour period, averaged over a month, is plotted in Fig. 6. Since demand has diurnal pattern we use $H = 24$. This distribution of requests is used to derive hourly demand for different client regions. We upscaled the number of requests by a factor of 3000 to reflect traffic handled by larger service providers [16]. For each client region we divided the workload proportional to the number of Internet users in that region. Table V shows the split of workload across different client regions obtained from the number of Internet users. Fig. 7 shows the hourly demand for a few client regions. The demand during the on-peak period is kept as 1.4 times the mid-peak demand and demand during off-peak period is kept at 0.6 times that in the mid-peak period.

Inter data center communication cost: For inter data center communication cost we use pricing model similar to the one charged by AWS EC2 services [25]. For example, AWS charges $\$0.12 - \$0.2$/GB across geographical regions.

Other parameters: P_{idle} and P_{peak} are set to 200W and 400W, respectively [26]. Average PUE is set to 1.5 [27], [28]. P^{max} is taken as 100MW/hr for all the locations [17]. The default value for maximum latency is set to 300ms. The size of data to be replicated per request is assumed to be 10KB [29]. We set $\gamma^{max} = 0.8$ [30]. We use the probability of a single data center failure as 0.005 corresponding to 1.8 days of failure per year.

B. Results

In this section, we present the results from evaluating the models for the TCO by varying the number of data centers, demand, and latency bound. We also study the effect of different models for server utilization and data replication (single site and multi-site) on the TCO. In all the results, we show the normalized values of TCO, where the normalization is done using the maximum TCO in all the experiments.

Country	Brazil	China	Egypt	France	Germany	India	Indonesia	Japan	Mexico	Nigeria	Russia	S. Korea	UK	USA	Vietnam
% Demand	5.33%	31.76%	2.17%	2.78%	3.50%	15.43%	2.04%	5.64%	2.65%	3.37%	4.50%	2.13%	2.93%	13.69%	2.09%

TABLE V: Percentage of demand from different regions

Countries	**6 Data Centers**			**7 Data Centers**			**8 Data Centers**			**9 Data Centers**			**10 Data Centers**		
	CACP	**MS**	**CDN**	**CACP**	**MS**	**CDN**	**CACP**	**MS**	**CDN**	**CACP**	**MS**	**CDN**	**CACP**	**MS**	**CDN**
Japan	20000	15592	20000	20000	12994	20000	20000	11138	20000	20000	9745	20000	16072	8663	20000
Ireland	20000	15592	20000	16203	12994	14495	200	11138	13463	200	9745	13463	200	8663	13463
California, USA	20000	15592	20000	20000	12994	20000	17360	11138	20000	200	9745	20000	200	8663	20000
Hong Kong	200	15592	20000	200	12990	20000	200	11132	20000	200	9745	20000	200	8663	20000
Virginia, USA	20000	15592	20000	20000	12994	20000	20000	11138	20000	20000	9745	20000	20000	8663	12052
UK	17760	15592	20000	1558	12994	20000	200	11138	20000	200	9745	20000	200	8663	20000
Netherlands	-	-	-	20000	12994	20000	20000	11138	20000	17160	9745	20000	1089	8663	20000
Switzerland	-	-	-	-	-	-	20000	11138	20000	20000	9745	20000	20000	8663	20000
Canada	-	-	-	-	-	-	-	-	-	20000	9745	4104	20000	8656	15723
Oregon, USA	-	-	-	-	-	-	-	-	-	-	-	-	20000	8663	8035
No of servers	97960	93552	120000	97961	90954	134495	97960	89098	153463	97960	87705	157567	97961	86623	169273
Normalized TCO	0.69	0.89	0.90	0.65	0.89	0.91	0.63	0.89	0.95	0.60	0.88	0.98	0.57	0.88	1.00
% reduction in TCO (w.r.t CDN)	23.35	0.51		28.20	2.55		33.34	5.66		38.67	10.01		42.62	12.39	
% reduction in TCO (w.r.t MS)		**22.96**			**26.32**			**29.34**			**31.84**			**34.50**	

TABLE VI: Comparison of number of servers provisioned and TCO for all models

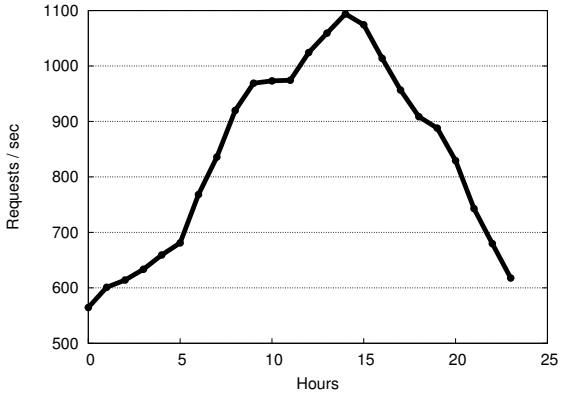


Fig. 6: Demand profile of wikipedia.org

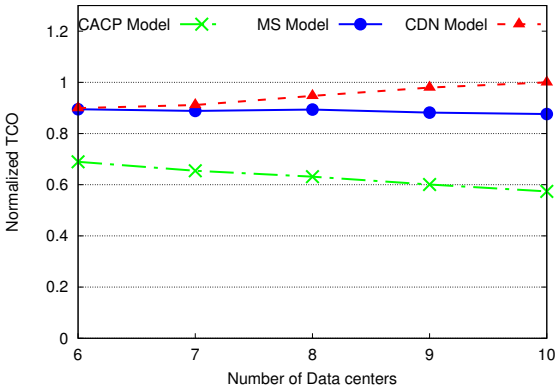


Fig. 8: Normalized TCO with varying number of data centers

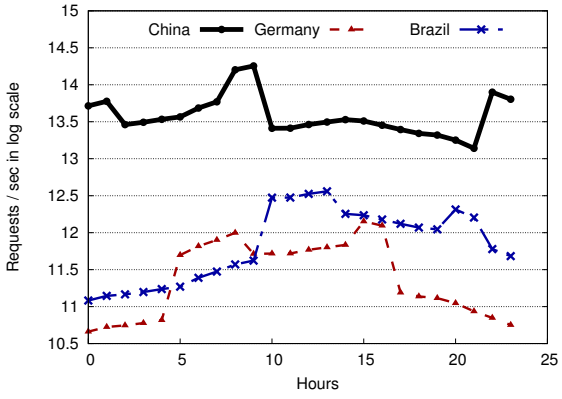


Fig. 7: Illustration of demand distribution from representative client regions

1) TCO comparison:
In this experiment, we increased the number of data centers between 6 and 10 serving the client requests as reported earlier within a maximum latency of 300 ms. Fig. 8 shows the normalized TCO for all the models with varying number of data centers. In this experiment we used a single site replication model.

Table VI reports the normalized TCO for different cases (third row from the bottom). Reduction in the TCO (percentage) with the CACP model (compared to MS model and CDN model) is shown in the last two rows. The fourth row from the bottom shows the total number of servers provisioned with each model across the data centers. The table also shows the locations chosen and the number of servers at each location as the desired number of locations increases. Since the CACP model exploits the spatio-temporal variation in the electricity prices, the TCO is lowest in the case of CACP model. Though MS model minimizes the number of servers provisioned at each location, it does not lead to minimum TCO due to being agnostic to operating cost.

From Table VI it can be observed that even with six data centers the benefit of CACP model is significant while, the other two models have similar TCO. This is due to the fact that with fewer data centers, there is not much scope for demand multiplexing. On the other hand, CACP model assigns larger workload at a data center location with lower electricity price. With addition of another location (Netherlands, with lower electricity price compared to U.K. and Ireland) the CACP model shifts the servers provisioned in U.K. and Ireland to Netherlands (see Table VI). This improves the TCO in CACP model by about 3.5%.While the CACP model suggests more servers, the TCO is minimized due to shifting them to locations with lower operating cost. This can be observed from the table which shows that MS model gives the same number of servers at each location. We can observe that the CACP model achieved a TCO reduction of upto 35% compared to the MS model, and upto 43% compared to the CDN model.

2) Impact of data center locations on the TCO:
We also studied how the choice of data center locations affects the TCO with the CACP model. We

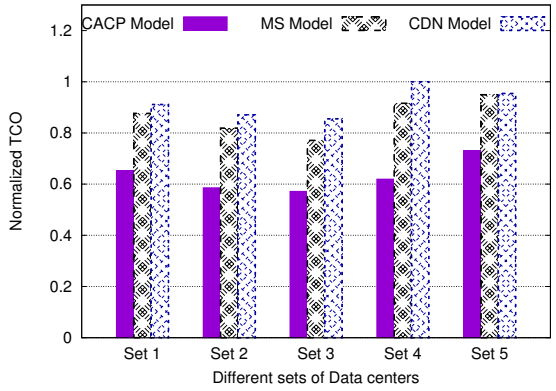


Fig. 9: Normalized TCO with different set of locations

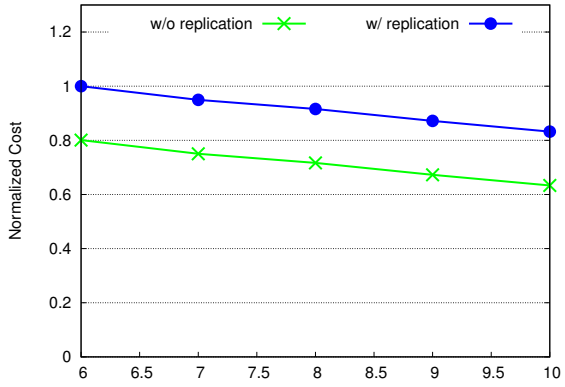


Fig. 11: Impact of Single Site Replication cost on TCO

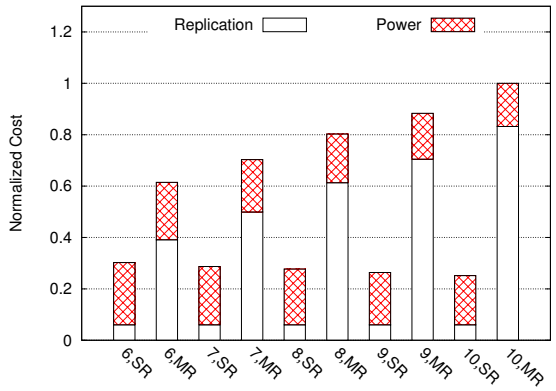


Fig. 10: Split up in TCO: Replication cost and electricity cost

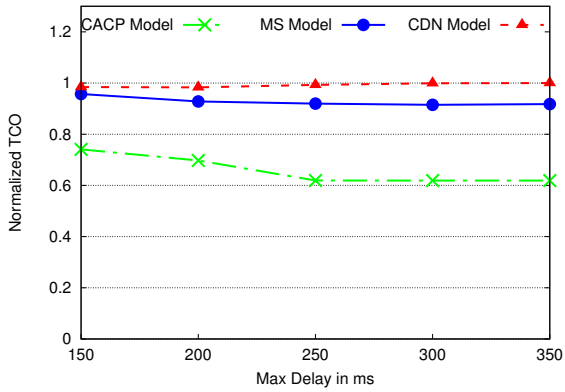


Fig. 12: Normalized TCO by varying maximum latency bound

evaluated our model for the following sets of locations:

Set 1: California, Japan, Hong Kong, Ireland, Switzerland, Virginia

Set 2: California, Japan, Hong Kong, Netherlands, Oregon, UK

Set 3: Japan, Hong Kong, Netherlands, Oregon, Switzerland, Virginia

Set 4: California, Japan, Hong Kong, Ireland, Netherlands, UK

Set 5: California, Japan, Hong Kong, Ireland, Netherlands, UK

The TCO obtained with the CACP model is shown in Fig. 9. Between *Set 2* and *Set 5* Oregon replaces Ireland, where electricity price is lower (refer Table I). Oregon being in the USA also meets the latency constraints for the largest number of users (from Americas as reported in Table V). Both these factors lead to a lower TCO for *Set 2* than *Set 5*.

3) Impact of Replication Cost:

To understand the contribution of replication cost to the TCO, we evaluated the CACP model considering the single site replication (SR) and multiple site replication (MR) models with varying number of data centers. The maximum latency was set to 300ms and the demand was generated as reported in Section IV-A. Fig. 10 shows the TCO split into replication cost and cost due to power consumed for both the replication models. It can be observed that in the SR model, contribution of the replication cost in minimal in the TCO. On the other hand, the MR model is costly for replication and the replication cost increases with the number of data centers as shown in Fig 10. Therefore, this approach may be preferred only when the post-failure penalty is very high.

Fig. 11 shows the TCO for the CACP model with and without replication cost being considered. It can be observed that single site replication cost alone accounts for 20% of the TCO. Therefore, CACP model without replication cost lowers the TCO by about 20% compared to the model with replication.

In all the subsequent experiments, we considered only a single site replication model while evaluating the TCO.

4) TCO vs Worst-case Latency:

Next, we studied the impact of maximum latency bound on the TCO. We evaluated the models for 8 data centers, 15 client regions, and the aggregate demand as mentioned in Section IV-A. The maximum latency was chosen in the range $150-350$ ms. Fig. 12 shows the normalized TCO for all the models with varying latency. We can observe that the CACP model results in a lower TCO by upto 38% and 32%, respectively compared to the CDN and MS models. In the CACP model, there is a choice in the number of data centers capable of serving the requests from a particular client region which leads to better multiplexing of resources and reduced TCO. Apart from this, CACP model also selects the data centers in regions with lower electricity prices while meeting the latency bound. Although the CDN model gives minimum latency, request routing is

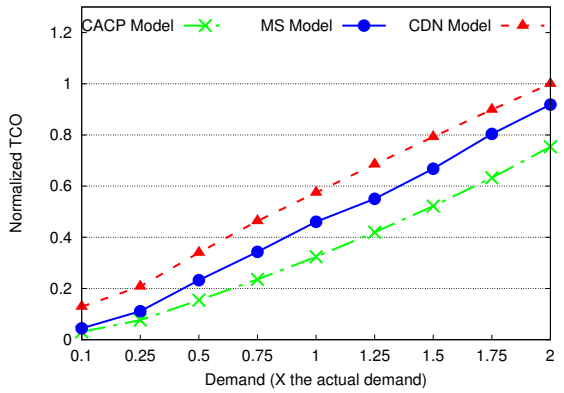


Fig. 13: Normalized TCO by varying demand

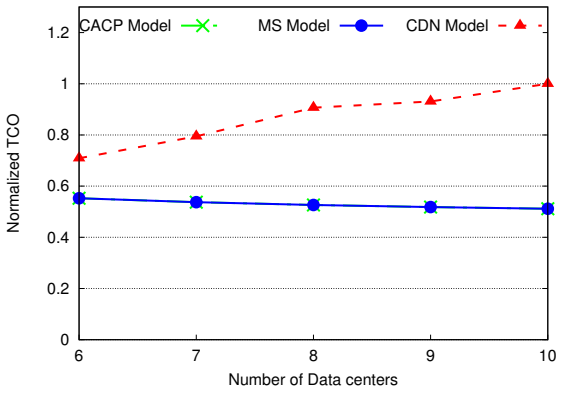


Fig. 14: Effect of Demand multiplexing

oblivious to the variation in the electricity price. Therefore, the TCO is higher for the CDN model particularly when the latency requirements are not very stringent. We conclude that CACP model is more advantageous for services without stringent latency requirement.

We also show how the worst-case latency increases when the CACP model targets the TCO reduction (as compared to the CDN model) in Table VII. At a worst-case latency of 150 ms, our model has about 25% lower TCO. When we target higher reduction in the TCO, the worst-case latency in the CACP model increases. For about 40% reduction in the TCO, our model leads to worst-case latency of 300 ms. The reduction in the TCO is achieved because, the CACP model exploits demand multiplexing and variation in electricity price when there is relaxation in the latency requirement.

Target Reduction in the TCO (%)	Worst-case Latency (ms)
25	**150**
30	**200**
35	**250**
40	**300**

TABLE VII: Worst-case latency with the CACP model corresponding to the TCO reduction (compared to CDN model)

5) Impact of Demand:
We evaluated all the models varying the total demand with 8 data centers and a maximum latency bound of 300ms. Results in Fig. 13 show that as the demand increases the TCO for CACP model is lower compared to other models. Due to the capacity limit of a data center, increased demand causes saturation of all the data centers in the regions with cheaper electricity. This reduces the choices available and leads to the selection of other locations with higher electricity price. The proposed model is advantageous only when the data center does not operate at peak utilization. Under heavy load, the CACP model can help the provider determine an optimal data center upgrade plan while minimizing the TCO.

6) Impact of demand multiplexing:
To study the impact of demand multiplexing on TCO, we evaluated the models by varying the number data centers from 6 to 10. Electricity price for all the data centers was fixed at 10 cents per kWh throughout the day and replication cost was fixed to $0.2/GB. Delay bound was set to 300ms. It can be observed from Fig.14 that CACP and MS models give lower TCO than CDN model, since CDN model does not allow demand multiplexing due to latency minimization objective. The CACP model reduces the TCO by almost 45% compared to the CDN model. We also noticed that CACP model eventually gives same TCO as the MS model because, cost reduction is only possible by demand multiplexing which minimizes the total number of servers (due to uniform electricity price). The TCO reduction of about 10% can be attributed to demand multiplexing as the number of data centers increases.

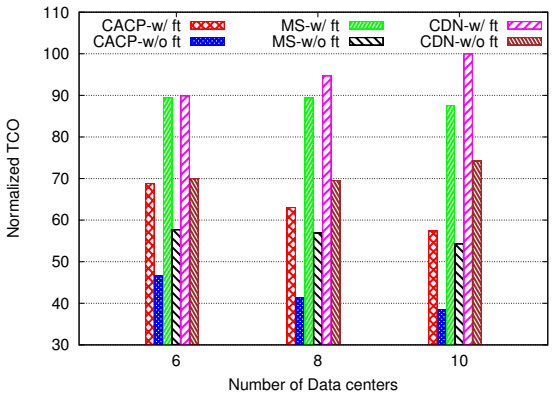


Fig. 15: Cost of provisioning w/ and w/o failure

7) Cost of over provisioning:
To study the cost of over-provisioning for fault tolerance, we evaluated all the models by varying the number of data centers. Fig. 15 shows the normalized TCO obtained with and without fault tolerance using each models (normalized with respect to largest TCO across all the cases). For example, CACP-w/ ft and CACP-w/o ft corresponds to the TCO achieved using CACP model with and without failure, respectively. Results show that when fault tolerance is not considered the TCO is always lower because, fault-tolerance demands over-provisioning of servers. This increases both CAPEX and OPEX and hence the TCO. For the case of 6 data centers, provisioning for fault-tolerance increases the TCO for CACP, MS, and CDN models by 47%, 55% and 28%, respectively. On the other hand when the number of data centers increases to 10, the cost of over-provisioning is 49%, 61% and 34% for CACP, MS and CDN models, respectively. We also notice that the CACP model with failure leads to a lower TCO than the

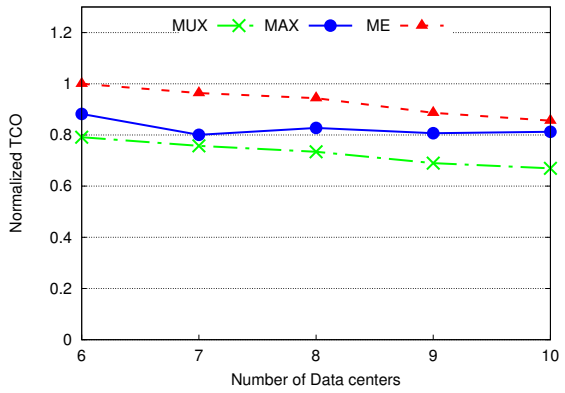


Fig. 16: Normalized TCO considering different approaches to address workload heterogeneity

CDN model without failure across all scenarios. CACP model provides resilience against single data center failure with no additional cost compared to the CDN model.

8) Server models for heterogeneous workload::
We evaluate the CACP model using each of the three models discussed in Section III-B to understand their impact on the TCO. We set the delay bound to 300ms and used the same demand as in other cases. Fig. 16 shows the TCO for different server utilization models. It can be seen that MUX approach results in a maximum reduction in the TCO (about 22% and 18% compared to ME and MAX, respectively). This is due to the fact that compute resources are effectively utilized in this approach. The ME and MAX approach both have a drawback of resource under-utilization (or over-provisioning of resources). The ME approach incurs the maximum cost because there is no scope for multiplexing demand across servers assigned for different types of services. In the MAX approach there is a scope for multiplexing of servers due to the use of a single server pool.

Key Observations:

- The CACP model provisions more servers but reduces the TCO upto 35% compared to the MS model, and upto 43% compared to the CDN model.
- The CACP model is cost effective when the latency requirement is not stringent and a data center does not operate at its peak utilization.
- It is possible to achieve availability against single data center failure with no additional cost using the CACP model compared to the CDN model. Choice of replication strategy (SR and MR) plays an important role in determining the TCO. Particularly, the contribution of replication cost to the TCO is significantly high when the number of data centers increases with the MR approach. Therefore, MR approach is good only when the post-failure migration penalty is high.
- MUX approach to handle heterogeneous workload at a data center results in maximum reduction in the TCO and this is a viable approach due to virtualization.

V. Conclusion

In this paper, we addressed the problem of cost-aware capacity provisioning for geo-distributed data centers capable of masking single data center failure. We prove that this problem in NP-Hard and proposed an MILP formulation to reduce the TCO. The proposed model is observed to be better than the MS and CDN models due to its ability to multiplex demand considering the spatio-temporal variation in electricity prices and the demand. We have also modeled different approaches to serve heterogeneous demand and data replication. Numerical results demonstrated that the approach of minimizing TCO is beneficial when electricity price varies significantly, which appears to be the case for most of the cloud providers operating geo-distributed data centers. The CACP model achieves a cost reduction of upto 34% and 50% when compared to MS and CDN models, respectively. Our model is also useful to study the effect of replication cost on the TCO for planning distributed data centers.

Appendix

In this section, we prove Theorem 1 stated in the paper. In a basic formulation, the cost aware capacity provisioning problem (without failure considerations) consists of a set of data center locations DC where cost of running servers at a data center i is given by $Cost_i$, and a set of client locations C generating demand to be serviced. Each client can be served by a data center lying within a given latency bound $Delay$. The goal is to provision a number of servers across data centers such that total cost incurred is minimum while satisfying client demand and latency bound.

In Lemma 1 we reduce the decision version of the set cover problem to the decision version of the CACP problem which is sufficient to show that Theorem 1 holds. Formally, the decision version of the CACP problem is defined as follows. Given a set of data centers and their server running costs, set of demand generating client regions and latency bound, does there exist a sub set of data centers which can satisfy the client demand with the total cost incurred being at most k.

Lemma 1. *The decision version of CACP problem is NP-hard.*

Proof: The decision version of the set cover problem is defined as follows. Given a set system $(\mathcal{U}, \mathcal{S})$ with $\bigcup_{S \in \mathcal{S}} S = \mathcal{U}$ and a positive integer k. The question is does there exist a collection of k or fewer sets of $\mathcal{S}$ that cover $\mathcal{U}$ [31]. This problem is known to be NP-complete and we give a reduction of this problem to the decision version of the CACP problem as follows.

Given an instance of the set cover problem $\mathcal{I}_S$, let us map it to an instance $\mathcal{I}_C$ of the decision version of CACP problem. For each $u \in \mathcal{U}$, we assign a client region c_u that generates a demand of unit compute capacity to meet its needs. For each $S \in \mathcal{S}$, we assign a data center d_S which is within the delay bound for the clients specified as its elements. For instance, if $S = \{u_1, u_2, \ldots, u_m\}$ then d_S has $c_{u_1}, c_{u_2}, \ldots, c_{u_m}$ within delay bound constraint. The cost associated with each data center is 1 unit and each of them have infinite capacity. This completes the reduction of instance $\mathcal{I}_S$ to $\mathcal{I}_C$. It is easy to observe that the reduction from $\mathcal{I}_S$ to $\mathcal{I}_C$ is polynomial time in the input size of instance $\mathcal{I}_S$. To complete the proof, we need to show that $\mathcal{I}_S$ admits a solution if and only if $\mathcal{I}_C$ has a solution which costs at most k units.

Suppose $\mathcal{I}_C$ has a solution with less than or equal to cost k units. Without loss of generality, let $d_{S_1}, d_{S_2}, \ldots, d_{S_l}$ be the solution to $\mathcal{I}_C$ that meets demands of all client regions. Note that $l \leq k$ as each data center consumes 1 unit of energy. Each of the client c_u is served by at least one data center in $d_{S_1}, d_{S_2}, \ldots, d_{S_l}$. Correspondingly the $S_1, S_2, \ldots, S_l$ covers each $u \in \mathcal{U}$ and thus it is a solution to $\mathcal{I}_S$ having size of $l \leq k$.

Conversely, if $\mathcal{I}_S$ admits a solution $S_1, S_2, \ldots, S_j$ with $j \leq k$ we can construct a solution to $\mathcal{I}_C$ which costs at most k units. The set of data centers $d_{S_1}, d_{S_2}, \ldots, d_{S_j}$ is able to meet the demand of all the client regions c_u as $\bigcup_{1 \leq i \leq j} S_i = \mathcal{U}$. Thus we have constructed a solution to $\mathcal{I}_C$ which costs $j \leq k$ units.

■

REFERENCES

[1] "Distributed virtual data center for enterprise and service provider cloud," http://www.cisco.com/c/en/us/products/collateral/routers/asr-9000-series-aggregation-services-routers/white_paper_c11-694882.html.

[2] P. X. Gao, A. R. Curtis, B. Wong, and S. Keshav, "It's not easy being green," in Proc. of the ACM SIGCOMM, 2012, pp. 211–222.

[3] L. G. Christy Pettey, "Gartner Reveals Top Predictions for IT Organizations and Users for 2011 and Beyond," http://www.gartner.com/newsroom/id/1480514, 2010.

[4] AWS, "Summary of the Amazon EC2 and Amazon RDS Service Disruption in the US East Region," http://aws.amazon.com/message/65648/, 2011.

[5] R. Johnson, "More Details on Today's Outage," https://www.facebook.com/notes/facebookengineering/more-details-on-todays-outage/431441338919, 2010.

[6] E. Nygren, R. K. Sitaraman, and J. Sun, "The akamai network: A platform for high-performance internet applications," SIGOPS Operating Systems Review, vol. 44, no. 3, pp. 2–19, Aug. 2010.

[7] AWS, "Amazon EC2 pricing," http://aws.amazon.com/ec2/pricing.

[8] A.-H. Mohsenian-Rad and A. Leon-Garcia, "Energy information transmission tradeoff in green cloud computing," Carbon, vol. 100, 2010.

[9] I. Goiri, K. Le, J. Guitart, J. Torres, and R. Bianchini, "Intelligent placement of datacenters for internet services," in Proc. of IEEE ICDCS, June 2011, pp. 131–142.

[10] "Electricity price in USA," http://www.eia.gov/.

[11] "Electricity price in european countries," http://ec.europa.eu/eurostat/statistics-explained/index.php/File:Half-yearly_electricity_and_gas_prices,_second_half_of_year,_2012\%E2\%80\%9314_(EUR_per_kWh)_YB15.png/.

[12] "Electricity price in Hong Kong," https://www.hkelectric.com/en/customer-services/billing-payment-electricity-tariffs/commercial-industrial-and-miscellaneous-tariff/commercial-industrial-and-miscellaneous-tariff-calculator.

[13] "International industrial energy prices," https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/487759/table_531.xls.

[14] "Page view statistics for wikimedia projects," http://dumps.wikimedia.org/other/pagecounts-raw/.

[15] M. Lin, A. Wierman, L. Andrew, and E. Thereska, "Dynamic right-sizing for power-proportional data centers," in Proc. of IEEE INFOCOM, April 2011, pp. 1098–1106.

[16] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs, "Cutting the electric bill for internet-scale systems," in Proc. of ACM SIGCOMM, 2009, pp. 123–134.

[17] M. Wardat, M. Al-Ayyoub, Y. Jararweh, and A. Khreishah, "To build or not to build? addressing the expansion strategies of cloud providers," in Proc. of International Conference on Future Internet of Things and Cloud (FiCloud), Aug 2014, pp. 477–482.

[18] S. Chen, Y. Wang, and M. Pedram, "Concurrent placement, capacity provisioning, and request flow control for a distributed cloud infrastructure," in Proc. of ACM Design, Automation and Test in Europe Conference and Exhibition (DATE), March 2014, pp. 1–6.

[19] R. Souza Couto, S. Secci, M. Mitre Campista, and L. Kosmalski Costa, "Network design requirements for disaster resilience in iaas clouds," IEEE Communications Magazine, vol. 52, no. 10, pp. 52–58, October 2014.

[20] I. Narayanan, A. Kansal, A. Sivasubramaniam, B. Urgaonkar, and S. Govindan, "Towards a leaner geo-distributed cloud infrastructure," in Proc. of USENIX Workshop on Hot Topics in Cloud Computing (HotCloud), Jun 2014.

[21] M. Ghamkhari and H. Mohsenian-Rad, "Energy and performance management of green data centers: A profit maximization approach," IEEE Transactions on Smart Grid, vol. 4, no. 2, pp. 1017–1025, 2013.

[22] "Google data center locations," https://www.google.com/about/datacenters/inside/locations/.

[23] B. Molina, C. E. Palau, and M. Esteve, "Modeling content delivery networks and their performance," Computer Communications, vol. 27, no. 15, pp. 1401–1411, 2004.

[24] "Internet world stats," http://www.internetworldstats.com/unitedstates.htm.

[25] "Amazon ec2 pricing," https://aws.amazon.com/ec2/pricing/.

[26] Y. Li, H. Wang, J. Dong, J. Li, and S. Cheng, "Operating cost reduction for distributed internet data centers," in Proc. of IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, May 2013, pp. 589–596.

[27] http://www.vertatique.com/no-one-can-agree-typical-pue.

[28] Y. Sverdlik, "Survey: Industry average data center pue stays nearly flat over four year," 2014, http://www.datacenterknowledge.com/archives/2014/06/02/survey-industry-average-data-center-pue-stays-nearly-flat-four-years/.

[29] Z. Wu, M. Butkiewicz, D. Perkins, E. Katz-Bassett, and H. V. Madhyastha, "Spanstore: Cost-effective geo-replicated storage spanning multiple cloud services," in Proc. of ACM Symposium on Operating Systems Principles. ACM, 2013, pp. 292–308.

[30] M. Al-Ayyoub, M. Wardat, Y. Jararweh, and A. A. Khreishah, "Optimizing expansion strategies for ultrascale cloud computing data centers," Simulation Modelling Practice and Theory, vol. 58, pp. 15–29, 2015.

[31] J. Kleinberg and É. Tardos, Algorithm design. Pearson Education India, 2006.