Efficient Server Provisioning and Offloading Policies for Internet Datacenters with Dynamic Load-Demand
Dan Xu, Xin Liu, and Bin Fan

Abstract—In datacenters, traffic demand varies in both large and small time scales. A datacenter with dynamic traffic often needs to over-provision active servers to meet the peak demand, which incurs significant energy cost. In this paper, our goal is to reduce energy cost of a set of distributed Internet datacenters (IDCs) while maintaining the quality of service of the dynamic traffic. In particular, we consider the outage probability as the QoS metric, where outage is defined as service demand exceeding the available capacity. We require the outage probability at each IDC to be smaller than some predefined threshold. Our goal is thus to minimize total energy cost over all IDCs, subject to the outage probability constraint. We achieve the goal by dynamically adjusting server capacity and performing load shifting in different time scales. We propose three different load-shifting and joint capacity allocation schemes with different complexity and performance. Our schemes leverage both stochastic multiplexing gain and electricity-price diversity. Thus, improving over prior work, our schemes reduce energy consumption/cost even when all IDCs have the same electricity price. We use both simulated load traces and real traffic traces to evaluate the performance of the proposed schemes. Results show that our proposed schemes are efficient in reducing energy cost, and robust in QoS provisioning.

Index Terms—data center, energy efficiency, dynamic traffic, convex optimization, stochastic multiplexing, electricity price diversity

I. INTRODUCTION

Cloud computing is considered as a promising new paradigm of Internet services. Internet companies, such as Google, Microsoft, Yahoo!, and Amazon, are providing an expanding range of cloud services, including computing, storage, searching, content-delivery, online-shopping, and social networking. The popularity of cloud services results in the fast growth of data center development, both nationwide and worldwide. We refer to such distributed Internet-scale data centers as IDCs (Internet datacenters). A commercial IDC often farms hundreds of thousands servers and consumes an enormous amount of electricity. Thus, it is important for IDC operators to reduce IDC energy consumption and cost.

On one hand, reducing energy consumption often requires turning off as many servers as possible. On the other hand, providing satisfactory Internet services requires enough number of active servers. Thus, much research has been focused on designing load-aware server provisioning schemes, e.g., in [16][17][18][30]. The basic idea is to dynamically control the number of active servers based on the load. Designing efficient load-aware server provisioning schemes is challenging.

In datacenters, traffic, i.e., volume of service requests, varies in time, in both large and small temporal scales. In Fig. 1, we plot traffic, which is considered as the total received packet size from Hadoop Distributed File System (HDFS) log of a real datacenter\(^1\) in two time resolutions. Fig. 1(a) shows one-second traffic variation over three-minutes, and Fig. 1(b) shows ten-minute traffic variation in a day. In both time scales, we observe that traffic varies significantly. One can also refer to [1] and [17] for the large-time-scale and the small-time-scale traffic variations of Microsoft datacenters, respectively. Time-varying traffic variations result in dynamic load-demand, which challenges server provisioning.

The current load-aware server provisioning schemes only capture the large-time-scale variation of load-demand. For example, in [17], the authors use statistical models to predict the future load, in a time scale of half an hour. In [18], the authors consider a large time interval, and estimate the current load level at the beginning of each time interval. The small-time-scale load-demand variation is rarely considered for the following reasons. First, booting up a server to the active state needs a considerable amount of time, up to several minutes [41], which is slower than the small-time-scale load variation; second, frequently turning on/off servers is not desirable in practice, which incurs additional operation costs [18] and affects long term system reliability [16]. Thus, one cannot provision servers dynamically according to the small-time-scale load-demand variation. However, most cloud computing services, like search and web browsing, are resource demanding and require a small service delay. Failing in provisioning servers to meet service requirements in even a small time scale can result in poor user experience (This is probably one important reason that current operators are reluctant to turn off servers.). Consequently, an IDC has to pre-allocate enough active servers to meet the peak load-demand, which results in idle servers and energy waste if the demand is low. Thus, we are motivated to improve energy efficiency of IDCs with dynamic load-demand, with both the large-time-scale and small-time-scale load-demand variations considered.

In this paper, our goal is to reduce energy cost of a set of distributed IDCs while guaranteeing the the performance requirement of dynamic traffic. In particular, we consider the outage probability as the QoS metric, where outage is defined as service demand exceeding the available capacity of a datacenter. We require the outage probability at each IDC to be smaller than some predefined threshold. Our goal is thus to minimize total energy cost over all IDCs, subject to the

\(^1\)The datacenter is operated by a major cloud computing service provider in U.S.
QoS constraint. We achieve the goal by dynamically adjusting server capacity and performing load shifting in different time scales. First, since booting up or shutting down servers needs a considerable amount of time, we determine the number of active servers for each IDC in a relatively long time scale, i.e., on the order of tens of minutes, to capture traffic variation in the large time scale. This is referred to as server provisioning. Second, since current inter-IDC load shifting incurs negligible time, up to tens of milliseconds [33], we design intelligent load shifting schemes among IDCs in a short time scale, i.e., on the order of hundreds of milliseconds to seconds, to capture traffic variation in the small time scale. In summary, we make the following contributions:

- We investigate the problem of reducing energy consumption and electricity cost for distributed IDCs with dynamic load-demand. We address this problem through joint server provisioning and load shifting in both the large and the small time scales. In addition, dynamic speed scaling is also exploited to satisfy the instantaneous load-demand at each IDC. Our schemes leverage both stochastic multiplexing gain and electricity-price diversity. Thus, improving over prior work, our schemes reduce energy consumption/cost even when all IDCs have the same electricity price.

- We propose three distributed load-shifting schemes: ratio-based load swapping (RBLS), ratio-based offloading (RBO), and threshold-based offloading (TBO). The optimal configuration of each load shifting scheme coupled with server provisioning is determined in a large time scale by a convex optimization model. In the optimization models, we minimize the total energy cost subject to outage probability constraint at each IDC. We model outage probability with both heavy-tailed and non-heavy-tailed load distributions, including Gaussian, exponential, and heavy-tailed Weibull distributions. Each datacenter follows the optimal configuration to shift the load instantaneously and independently. The three load shifting schemes have different complexity and performance.

- We use both simulated load traces and real traffic traces to evaluate the performance of the proposed schemes. In particular, simulation results show that the proposed TBO scheme is efficient in reducing the energy cost and robust in QoS provisioning. When using historic load statistics, a simple local capacity adjustment scheme (i.e., DCP proposed in Section V-D) helps TBO achieve the desirable performance, which is close to that in the case of using accurate load statistics.

The rest of paper is organized as follows. In Section II, we survey related work. In Section III, we introduce the system models. In Section IV, we present ratio-based load swapping scheme. We further propose ratio-based offloading and threshold-based offloading, and discuss implementation issues in Section V. We evaluate our proposed schemes in Section VI, and discuss the efficiency of our schemes in section VII, followed by conclusions in Section VIII.

II. RELATED WORK

Datacenter energy efficiency has been extensively studied, with solutions ranging from processor-level to cross-datacenter level. We briefly review the most related ones here.

Dynamic speed/voltage scaling (DVS) reduces processor power consumption by adjusting the frequency based on the instantaneous load demand, e.g., in [2]-[13]. DVS works for the small-time-scale load variation, it takes only tens of microseconds to charge processor frequency [9][14]. But it may not be sufficient by itself for power management in a datacenter-level, where it is important to turn off as many servers as possible due to the significant power consumption of a server in the idle state.

Power management for a datacenter with dynamic load has been studied, e.g., in [15]-[28]. An important idea is to design load-aware power/server provisioning schemes [15]-[23], where the capacity is controlled dynamically according to load variation. For example, in [15], the authors present a datacenter architecture where loads are concentrated on a pool of active servers with dynamic scale. In [16][17], the authors use prediction methods for dynamic server provisioning. In [18], the authors consider a large time interval, and estimate the load at the beginning of each time interval. Capacity allocation is performed according to the current load, with server switching cost (the cost of turning on/off servers) considered. In [19], the authors design an online algorithm for dynamic server provisioning, where job queue information is leveraged and Lyapunov optimization is used to establish the performance of the algorithm. In [20], the authors develop an online algorithm that optimally exploits energy storage devices to minimize the average electricity cost, in dynamic load and price environments. Virtualization and server consolidation, e.g., in [24], [25], [26], [27], [28], can reduce the traffic dynamics by consolidating applications, by which the number of active servers can be reduced. Schemes in these studies either only work for the large-time-scale traffic variation, or can only be applied to a single datacenter operation.

Most recently, cross-IDC power management has received significant attention, e.g., in [29]-[40]. One central idea for cross-IDC power management is to leverage the diversity of geolocation and the corresponding difference in electricity price. One can route traffic to an IDC with a lower electricity price [29]. Server provisioning is often jointly performed with load shifting or dispatching. For example, in [30], the
authors consider a model where front-end portals allocate load to IDCs. They minimize the sum of IDCs energy cost as a function of number of active servers, subject to a response time requirement. They consider a fixed arriving rate of requests at each front-end. A similar delay constraint is considered in [32]. In [33], the authors propose a distributed load balancing and dynamic speed scaling algorithm to minimize the total energy cost of a number of distributed processor clusters. In [34], the authors minimize the energy and load shifting cost of geographically distributed IDCs by designing distributed capacity and load shifting algorithms. The authors extend their work to the case with renewable energy and limited energy storage devices [37]. In [35], the authors propose online job migration algorithms for delay-tolerant jobs among datacenters, which capture the fundamental tradeoff between the energy cost and load shifting cost. In all the above work, the small-time-scale traffic variation is not considered. In our work, both the large-time-scale and the small-time-scale traffic dynamics are explicitly considered. In [38], a joint design of ISP and content provider through load distribution is shown to reduce energy consumption subject to a user delay constraint. In [39], machine learning techniques are used for load distribution among IDCs that consider both utilization and weather forecast. In [40], the authors propose to develop ISP-controlled nano-scale data centers (i.e., home gateways) to reduce energy consumptions. There are also some other interesting studies on IDC power management, such as workload decomposition [41], optimal power allocation for servers with total power budget [42], and other techniques [43][44]. See [47][48] for discussions of related challenges and prior work.

Our schemes differ from prior work in that 1) we jointly design server capacity allocation and load shifting schemes; 2) we consider both large-time-scale and small-time-scale traffic variations; 3) we explicitly leverage stochastic multiplexing gain in a cross-datacenter level. A brief introduction to the work was presented in [54].

### III. System Models

In this section, we describe our system models, including IDC and server speed model, IDC load and load shifting model, and power consumption and cost model. We also introduce the outage probability and formulate a general optimization problem. We list the main notations used in this paper in Table I.

#### A. IDC and server model

Consider an operator with N IDCs located in different geographic areas. It is typical for an operator to have more than 10 IDCs. IDC i has Ki servers and ki of them are turned on, i.e., being in the active state. The remaining Ki-k_i servers are in the off/sleep state, which has negligible power consumption. Clearly, ki is a control variable that balances the tradeoff of service performance and energy consumption. Since we consider a service provider operating the N IDCs, we assume there is a decision-maker to determine ki for each IDC i.

Dynamic speed scaling is considered. To elaborate, an active server runs at a normalized CPU speed s, where 0 ≤ s ≤ 1, and 0 represents the idle state, and 1 means the maximum CPU frequency [7][9][33]. In this paper, we assume servers are homogenous for simplicity. That is, all servers have the same maximum speed and the same power consumption model as specified later. Note that in the idle state a server does have a considerable power consumption, as discussed in detail later. We will incorporate heterogeneous servers into our models in the future work. We further assume that the capacity of an IDC is the sum of the speed of all active servers. Then, if each server has the same speed s, the total capacity is k_is. The maximum capacity of ki active servers is k_i. Note that scaling up/down the speed s of an active server only takes several microseconds [9][14], which is negligible.

#### B. Load model

An IDC has different types of requests, coarsely classified as delay sensitive jobs and delay tolerant jobs. Search, web surfing, and email login are examples of the first class, which requires a small service latency. On the other hand, background or analytical jobs are often delay tolerant, such as data backup, MapReduce, and index generation for web searching, which can tolerate a relatively large delay. Traffic of different types of requests (i.e., the number of requests per second) is dynamic over time and thus results in a time-varying capacity demand. In this paper, we use Di to denote the total load or capacity demand at IDC i. Note that in this paper, we use capacity-demand and load-demand alternatively, which are different from the notion of traffic. Capacity or load demand has the same physical meaning of the processor speed s, while traffic is the number of requests per second as mentioned above. Di is a random variable, which changes in a small time scale (e.g., in subseconds). We further assume capacity demands of different datacenters are independent, although not necessarily identically distributed. Let \( \mu_i \) and \( \sigma_i^2 \) denote the mean and variance of Di, respectively. The variation of \( \mu_i \) and \( \sigma_i^2 \) describe the load dynamics in the large time scale. \( \mu_i \) and \( \sigma_i^2 \) are the main input parameters of our proposed schemes. \( \mu_i \) and \( \sigma_i^2 \) can be obtained by load prediction or estimation, or using historic information. In our simulations, we will evaluate the impact of using recent historic information

#### TABLE I: Main Notations

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>Number of IDCs</td>
</tr>
<tr>
<td>( k_i )</td>
<td>Number of active servers of IDC i (main control variable)</td>
</tr>
<tr>
<td>( K_i )</td>
<td>Total number of servers of IDC i</td>
</tr>
<tr>
<td>s</td>
<td>Normalized server speed, 0 ≤ s ≤ 1</td>
</tr>
<tr>
<td>( D_i )</td>
<td>Capacity demand of IDC i (before load shifting)</td>
</tr>
<tr>
<td>( \mu_i )</td>
<td>Mean of ( D_i ), i.e., ( \mu_i = E(D_i) ) (main input)</td>
</tr>
<tr>
<td>( \sigma_i )</td>
<td>Standard deviation of ( D_i ), i.e., ( \sigma_i = \sqrt{\text{Var}(D_i)} ) (main input)</td>
</tr>
<tr>
<td>( D_i^\ast )</td>
<td>Capacity demand of IDC i (after load shifting)</td>
</tr>
<tr>
<td>( \mu_i^\ast )</td>
<td>Mean of ( D_i^\ast ), i.e., ( \mu_i^\ast = E(D_i^\ast) )</td>
</tr>
<tr>
<td>( \delta_i )</td>
<td>Load shifted from IDC i to IDC j (random variable)</td>
</tr>
<tr>
<td>( \eta_i )</td>
<td>Load shifting constraint from IDC i to IDC j</td>
</tr>
<tr>
<td>( \nu_i )</td>
<td>Outage probability constraint at IDC i</td>
</tr>
<tr>
<td>P</td>
<td>Normalized power consumption of a server (a function of s)</td>
</tr>
<tr>
<td>( P_i )</td>
<td>Expected power consumption of IDC i (a function of ( k_i ) and ( \mu_i ))</td>
</tr>
<tr>
<td>( \alpha_i )</td>
<td>Electricity price at the location of IDC i</td>
</tr>
<tr>
<td>( r_{ij} )</td>
<td>Load splitting ratio at IDC i to IDC j (control variable)</td>
</tr>
</tbody>
</table>
as input. In our proposed offloading schemes, we assume that
\( D_i \) follows a certain distribution. Specifically, we consider
Gaussian distribution, exponential distribution, and Weibull
distribution, respectively.

The decision-maker receives updates of \( \mu_i \) and \( \sigma_i^2 \) from each
IDC \( i \), and does not need to know \( D_i \) in any specific time.
Hence the communication overhead between the decision-
maker and each IDC is negligible.

The decision-maker uses the load statistics to determine
the optimal configurations for server provisioning and loading
shifting. An IDC does not need to know the load information
of others. Each IDC shifts load in an independent and dis-
bursed manner, following a policy configured by the decision-
maker. Let \( D_{ij} \) denote the load shifted from IDC \( i \) to IDC \( j \),
which is also a random variable. We use \( D_i \) to denote the
load of IDC \( i \) after load shifting. The mean and variance of
\( D_i \) is denoted by \( \bar{\mu}_i \) and \( \sigma_i^2 \), respectively. To prevent excessive
load transfer delay or cost, we constrain the average shifted
load from IDC \( i \) to IDC \( j \), i.e., \( E(D_{ij}) \), by a threshold \( \eta_{ij} \). The value of \( \eta_{ij} \) is determined by the available bandwidth and
transfer speed between IDC \( i \) and \( j \). For certain IDC pairs, if the
distance is a concern (due to transmission delay), we can
simply set \( \eta_{ij} = 0 \) to prevent load shifting between the two. We
note that it is more desirable to shift the load of the delay-
tolerant jobs, since those jobs are less sensitive to the load
shifting delay.

Load shifting incurs costs. In practice, some major IDC op-
erators such as Google and Microsoft have their own backbone
networks to interconnect the IDCs [49]. Load shifting cost is
thus mainly incurred during the construction phase. Some
other IDC operators do not have their own backbone networks
to connect IDCs and need to pay ISPs for the traffic shifted.
In this case, bandwidth fees are typically charged proportionally
to the shifted traffic volume [35][46]. In addition, load shifting
incurs additional delay, which can result in revenue loss [34].
In this paper, since we consider load shifting constraints, we
do not formally formulate the shifting cost in our optimization
models. If desired, load shifting cost can be easily incorporated
into the problem formulation, e.g., by a linear or convex cost
function as in [35]. In our simulations, we will relax the load
shifting constraints and study the impact of load shifting costs
on our proposed schemes.

C. Power consumption model

According to [7][9], the normalized power consumption of
a server (processor) running at a speed \( s \in [0, 1] \) is

\[
P(s) = \nu + (1 - \nu)s^\epsilon,
\]
where the exponent \( \epsilon \geq 1 \), with a typical value of 2 [9], and
\( \nu \) is the power consumption ratio in the idle state, which is
around 0.6, and hardly lower than 0.5 [17]. This reflects the
high energy consumption of idle servers, which is the main
motivation for efficient server provisioning.

We next find the power consumption of \( k_i \) active servers.
When the load \( D_i \) is smaller than \( k_i \), assuming homogeneous
servers, each server should work with speed \( s = \frac{D_i}{k_i} \) to
minimize total power consumption. Note that here we make
an assumption that the load can be arbitrarily divided among
servers. When \( D_i \geq k_i \), each server works with speed \( s = 1 \).
Given the probability density function (PDF) of \( D_i \), denoted by
\( f_i(\cdot) \), the total expected power consumption of the \( k_i \)
servers is

\[
P_i = k_i \left\{ \int_0^{k_i} \left[ \nu + (1 - \nu) \left( \frac{d_i}{k_i} \right)^\epsilon \right] f_i(d_i)dd_i + \int_{k_i}^{\infty} f_i(d_i)dd_i \right\},
\]
where \( \int_{k_i}^{\infty} f_i(d_i)dd_i \) is the expected power consumption of a
server when \( d_i \geq k_i \).

To find an explicit and simple model of \( P_i \), we numerically
study the properties of \( P_i \). We choose different mean and
variance of \( D_i \), and different distributions, including Gaussian,
exponential, and Weibull distributions. From our numerical
experiments, it appears that \( P_i \) can be closely approximated
by a piecewise linear function:

\[
\tilde{P}_i = \bar{P}_i(\bar{\mu}_i, k_i) \approx \begin{cases} k_i, & \text{if } k_i \leq \bar{\mu}_i; \\
(1 - \nu)\bar{\mu}_i + \nu k_i, & \text{if } k_i > \bar{\mu}_i. 
\end{cases}
\]

The intuition of (3) is as follows. When \( k_i \) is less than \( \bar{\mu}_i \),
each server is likely to be fully loaded, then each server has a
power consumption approximated by 1. When \( k_i \) increases,
power consumption of all servers increases linearly with \( k_i \)
and \( \bar{\mu}_i \). When \( k_i \) is much larger than \( \bar{\mu}_i \), servers are almost
idle, i.e., \( s \) is small, by which the total power is approximated
by \( \nu k_i \).

In Fig. 2, we present results of the deviation of \( \tilde{P}_i \) from \( P_i \)
(i.e., by (2)) under the three distributions of \( D_i \) mentioned
above. Note that we consider Weibull distributions with a
shape factor smaller than 1, which are thus heavy-tailed. The
deviation is evaluated by a metric named deviation ratio,
which is defined as \( |\tilde{P}_i - P_i| / P_i \). For each type of distribution,
we randomly generate 1000 different pairs of \( \bar{\mu}_i \) and \( \bar{\sigma}_i \).
For each pair of \( \bar{\mu}_i \) and \( \bar{\sigma}_i \), we further randomly generated different
values of \( k_i \). We consider both the cases of \( k_i \leq \bar{\mu}_i \) and
\( k_i > \bar{\mu}_i \). For the case of \( k_i > \bar{\mu}_i \), we constrain \( k_i \) by \( 2\bar{\mu}_i \),
which is conservative since \( P_i \) is closer to \( P_i \) with a larger \( k_i \)
larger than \( 2\bar{\mu}_i \). In the case of \( k_i > \bar{\mu}_i \). By Fig. 2, we observe
that the deviation ratio is very small in the case of \( k_i \leq \bar{\mu}_i \),
especially for exponential distributions and heavy-tail Weibull
distributions. When \( k_i > \bar{\mu}_i \), the deviation ratio is roughly
10% for each type of distribution. Note that in this case, the
deviation ratio gets smaller if we choose \( k_i s \) larger than \( 2\bar{\mu}_i \).
Overall, the deviation ratio is around 5%. Results in Fig. 2
indicate that \( \tilde{P}_i \) is close to \( P_i \). In addition, in our numerical
study, we found \( P_i \) is larger than \( P_i \) in almost all cases, which
implies that \( P_i \) is an conservative upper bound.

Other equipments in an IDC, e.g., cooling systems, also
contribute to the total power consumption, which is roughly
proportional to that of the servers [50][51]. Thus the total
power consumption of an IDC can be obtained by scaling
up \( P_i \) by a constant factor. For notation brevity, we absorb
this constant factor into the electricity price \( \alpha_i \) at IDC \( i \).
Typically, the electricity price of a consumer (IDC operator
in this case) is determined hourly by a bilateral contract with
the Regional Transmission Organization (RTO). Electricity price exhibits significant diversity in both location and time. Typically, \( \alpha_i \) changes in the time scale of hours, which is in a large time scale compared to traffic dynamics. We consider the energy cost (per unit of time) as the product of total power consumption of an IDC and electricity price. Note that while we exploit price diversity, our schemes result in energy reduction in the case where \( \alpha_i \) is the same for all IDCs.

### D. Outage probability constraint

To provide a desirable quality of service (QoS), e.g., response time for requests, the maximum capacity \( k_i \) should exceed load-demand \( D_i \) most of the time. An outage happens when \( D_i > k_i \). Let \( \delta_i \) be the outage probability constraint at each IDC \( i \). We require \( \Pr\{D_i > k_i\} \leq \delta_i \), \( \forall i \). Note that when \( D_i \) is smaller than \( k_i \), there is no outage since each server can dynamically change its speed \( s \) to meet the load. When an outage happens, requests may not necessarily be dropped. They can be queued at the cost of an extra delay.

Note that since \( D_i \) varies in a small time scale, an outage event is measured in a small time granularity, e.g., every hundreds of milliseconds. An IDC monitors its outage events for a relatively long time duration, e.g., every ten minutes. Based on all the measurement samples, an IDC can calculate the outage probability of the time duration it monitors.

### E. Problem formulation

Our objective is to minimize the total energy cost per unit of time, under the outage probability constraint at each IDC and load shifting constraints. Formally, we have

\[
\text{Minimize } \sum_{i=1}^{N} \alpha_i \hat{P}_i(\hat{\mu}_i, k_i) \tag{4}
\]

Subject to

\[
\Pr\{\hat{D}_i > k_i\} \leq \delta_i, \tag{5}
\]

\[
E(\hat{D}_{ij}) \leq \eta_{ij}, \tag{6}
\]

\[
k_i \leq K_i, \tag{7}
\]

\[
i, j = 1, \ldots, N. \tag{8}
\]

Note that in (4), \( \hat{P}_i(\hat{\mu}_i, k_i) \), is the energy consumption per unit of time. This optimization is performed at the decision-maker in a relatively large time scale where the distributions of \( D_i \)'s remain unchanged. The capacity \( k_i \) and parameters of the load shifting policy remain static until being reconfigured.

At the same time, instantaneous load shifting will be executed at a much smaller time scale. Clearly, the two are closely coupled. Note that after instantaneous load shifting, dynamic speed/voltage (DVS) scaling is also performed to satisfy the instantaneous load-demand in a small time scale.

Designing load shifting schemes among IDCs is complicated, since how and how much load an IDC should shift may depend on other IDCs’ actions. In addition, it may depend on what information each IDC has. In this paper, we focus on schemes where an IDC only knows its own capacity demand \( D_i \) and executes load shifting independently. In the rest of the paper, we will consider three different load shifting schemes and the associated optimal server provisioning.

### IV. RATIO-BASED LOAD SWAPPING: A BENCHMARK SCHEME

In this section, we consider a load shifting scheme, where an IDC \( i \) can shift a portion of its load to all others. In particular, let \( D_{ij} = r_{ij} D_i \), where \( r_{ij} \) is a fixed ratio of load shifting from IDC \( i \) to IDC \( j \). We name the scheme ratio-based load swapping (RBLS), where the decision variables are the capacity vector \( \hat{k} = \{k_1, k_2, \ldots, k_N\} \), and the load shifting ratio matrix \( r = \{r_{ij}, i, j = 1, 2, \ldots, N\} \).

After load shifting, the load at an IDC \( i \) is \( \hat{D}_i = \sum_{j=1}^{N} r_{ji} D_j \), where \( \hat{D}_i \) is the sum of \( N \) independent random variables. Since it is typical that \( N \geq 10 \), one can apply the Central Limit Theorem (CLT), by which \( \hat{D}_i \) is approximated by a Gaussian distribution with mean \( \sum_{j=1}^{N} r_{ji} \mu_j \), and variance \( \sum_{j=1}^{N} r_{ji}^2 \sigma_j^2 \). We derive the representation of the outage probability constraint in (5) with a Gaussian distribution of \( \hat{D}_i \). Let’s introduce a random variable \( X \), which is equal to \( \left( \hat{D}_i - \sum_{j=1}^{N} r_{ji} \mu_j \right) / \sqrt{\sum_{j=1}^{N} r_{ji}^2 \sigma_j^2} \). Thus \( X \) follows the standard Gaussian distribution with mean of 0 and variance of 1. We can rewrite (5) as

\[
\Pr\left(X > \frac{k_i - \sum_{j=1}^{N} r_{ji} \mu_j}{\sqrt{\sum_{j=1}^{N} r_{ji}^2 \sigma_j^2}} \right) \leq \delta_i. \tag{9}
\]

Consider the Q-function, \( Q(x) = \frac{1}{\sqrt{2\pi}} \int_{x}^{\infty} \exp\left(-\frac{t^2}{2}\right)dt \), i.e., the tail probability of the standard Gaussian distribution. Since its reverse function, \( Q^{-1}(\cdot) \), is decreasing, Eq. (9) is equivalent to

\[
\frac{k_i - \sum_{j=1}^{N} r_{ji} \mu_j}{\sqrt{\sum_{j=1}^{N} r_{ji}^2 \sigma_j^2}} \geq Q^{-1}(\delta_i) \tag{10}
\]

Eq. (10) yields the representation of the outage probability constraint as

\[
-k_i + \sum_{j=1}^{N} r_{ji} \mu_j + Q^{-1}(\delta_i) \sqrt{\sum_{j=1}^{N} r_{ji}^2 \sigma_j^2} \leq 0 \tag{11}
\]

By (11), as long as \( \delta_i \) < 0.5, we have \( Q^{-1}(\delta_i) > 0 \) and thus \( k_i > \sum_{j=1}^{N} r_{ji} \mu_j = \hat{\mu}_i \). Since \( \delta_i \) typically takes a value that is much smaller than 0.5, we only need to consider the case of \( k_i > \hat{\mu}_i \), where the expected power consumption of an IDC \( i \), \( \hat{P}_i \), is \( u k_i + (1-u) \hat{\mu}_i \) based on (3). Then we have the following

\[
\begin{align*}
\hat{P}_i(\hat{\mu}_i, k_i) &= \left\{ \begin{array}{ll}
\alpha_i (u k_i + (1-u) \hat{\mu}_i) & \text{if } k_i > \hat{\mu}_i \\
\alpha_i \hat{\mu}_i & \text{if } k_i \leq \hat{\mu}_i
\end{array} \right.
\end{align*}
\]
The complexity of load shifting is therefore \( O(N) \), which also has its own load.

V. OFFLOADER-BASED LOAD SHIFTING

In this section, we consider offloader-based load shifting schemes. We focus on the case where there is one IDC as an offloader. Without losing generality, we assume that IDC 1 is selected as the offloader. IDC \( i, i = 2, \ldots, N \), shifts load to the offloader. Note that the offloader has its own load. We explore two offloading structures, i.e., ratio-based offloading and threshold-based offloading. We will then consider offloader selection and discuss the case of multiple offloaders. Last, we will consider implementation issues, in particular, we sketch a dynamic capacity provisioning scheme to handle the discrepancy between assumed load distribution and real IDC load pattern.

A. Ratio-based offloading

In ratio-based offloading (RBO), an IDC \( i \) transfers \( r_i \) portion of the load to the offloader. The control variable is \((\vec{k}, r)\), where \( \vec{k} = (k_1, \ldots, k_N) \), \( \vec{r} = (r_1, \ldots, r_N) \), and \( r_1 = 1 \).

At the offloader, the load from other IDCs and itself is aggregated and can be approximated by Gaussian distribution. The outage probability constraint is similar to (13). But at each IDC, the load left, i.e., \( D_i = (1 - r_i)D_i \), follows the same, but scaled down, distribution as \( D_i \). To explicitly obtain the outage probability for each IDC, we need to assume specific distributions for \( D_i \) in our optimization model. To overcome the discrepancy between the assumed distribution and real load pattern, we later propose a dynamic scheme that an IDC scales up or down the capacity allocated by the solutions derived from the optimization models. Generally speaking, if the assumed distribution is not heavy-tailed, and the real load distribution is heavy-tailed, the capacity probably needs to be tuned up, and vice versa.

We next consider three distributions: Gaussian, exponential, and Weibull. We choose these three because Gaussian distribution is not heavy-tailed, Weibull distribution with shape factor smaller than 1 is heavy-tailed, and exponential distribution is at the boundary of being heavy-tailed. In practice, other distributions can be chosen based on measurement. Similar to (12)-(17), we have the following optimization model for RBO.

a) Gaussian:

\[
\begin{align*}
\min_{(\vec{k}, \vec{r})} & \quad \sum_{i=1}^{N} \alpha_i \nu k_i + (1 - \nu) \left[ \sum_{j=1}^{N} r_{ij} \mu_j \right] \\
\text{s. t.} & \quad - k_i + \sum_{j=1}^{N} r_{ij} \mu_j + Q^{-1}(\delta_i) \sqrt{\sum_{j=1}^{N} r_{ij}^2 \sigma_j^2} \leq 0, \quad i = 1, \ldots, N \\
& \quad - k_i + (1 - r_i) \mu_i + Q^{-1}(\delta_i) \sigma_i \leq 0, \quad i = 1, \ldots, N \\
& \quad r_{ij} \mu_j \leq \eta_{ij}, \quad i \neq j, \quad i = 1, \ldots, N \\
& \quad k_i \leq K_i, \\
& \quad i = 1, 2, \ldots, N.
\end{align*}
\]
In (20), \( P_i = \nu k_i + (1 - \nu) \bar{\mu}_i \), because we have \( k_i > \bar{\mu}_i \) by (21) and (22), \( i = 0, 1, \ldots, N \). Similar to (12)-(17), (20)-(25) is also a convex optimization problem.

b) Exponential: We have \( D_i \sim \lambda_i e^{-\lambda_i (d_i - \tau_i)} \), where \( \tau_i \) is the minimum value of \( D_i \). The outage probability constraint for an IDC \( i \), which replaces (22), is

\[
-\lambda_i \left( \tau_i + \frac{\ln \frac{1}{\lambda_i}}{\lambda_i} \right) \leq 0. \tag{26}
\]

Clearly, the optimization problem for RBO is still convex if we replace (22) by either (26) or (27).

By RBO, the load left at each IDC is still random and follows the same distribution (although scaled down) as the original load. The role of the offloader is mainly to reduce the load burden for each IDC. Therefore, an IDC \( i \) may still need to reserve extra servers to handle the load dynamics, especially when the QoS constraint \( \delta_i \) is small, or when \( D_i \) follows a heavy-tailed distribution. To overcome this drawback, we consider a different offloading structure, named threshold-based offloading.

B. Threshold based offloading

In threshold-based offloading (TBO), an IDC \( i \) sets its capacity \( \kappa_i \) as the threshold. If the load \( D_i \) is larger than \( \kappa_i \), IDC \( i \) shifts the excessive load, \( D_i - \kappa_i \), to the offloader. If \( D_i \) is no larger than \( \kappa_i \), no load is shifted to the offloader. Therefore, in each IDC, the capacity always meets the demand with load shifting. Hence the outage probability is 0. The load shifted from IDC \( i \) to the offloader is \( D_i \geq \max\{D_i - \kappa_i, 0\} \), denoted by \( (D_i - \kappa_i)^+ \). The value of \( (D_i - \kappa_i)^+ \) depends on the distribution of \( D_i \). To study the properties of TBO and to obtain a closed-form representation, we assume that \( D_i \sim \lambda_i e^{-\lambda_i (d_i - \tau_i)} \), which replaces (22), is exponential. The value of \( (D_i - \kappa_i)^+ \) is the minimum value of \( D_i - \kappa_i \) when \( D_i > \kappa_i \), there is

\[
E[(D_i - \kappa_i)^+] = \int_{\tau_i}^{\infty} \lambda_i e^{-\lambda_i (d_i - \tau_i)} dd_i
\]

Further, there is

\[
\var[\max\{D_i - \kappa_i, 0\}] = \frac{2 e^{-\lambda_i (\kappa_i - \tau_i)} - e^{-3\lambda_i (\kappa_i - \tau_i)}}{\lambda_i^2}. \tag{29}
\]

At the offloader, the aggregated load is

\[
\tilde{D}_1 = D_1 + \sum_{i=2}^{N} (D_i - \kappa_i)^+. \tag{30}
\]

We still apply CLT to approximate the distribution of \( \tilde{D}_1 \) as Gaussian, with mean \( \frac{\lambda_i}{\lambda_i^{\prime}} \left( \tau_i + \frac{\ln \frac{1}{\lambda_i}}{\lambda_i} \right) \) and variance \( \frac{\lambda_i}{\lambda_i^{\prime}} \left( \tau_i + \frac{\ln \frac{1}{\lambda_i}}{\lambda_i} \right) \lambda_i^{\prime} \). Note that \( \mu_1 = \frac{1}{\lambda_i} + \tau_i \).

Our objective is to determine \( \tilde{k} = (k_2, \ldots, k_N) \) efficiently through convex optimization. The LHS of (31) is not necessarily a convex function of \( \tilde{k} \). To address this issue, we make an important observation here that \( e^{-3\lambda_i (k_i - \tau_i)} \ll 2 e^{-\lambda_i (k_i - \tau_i)} \).

Thus we can simplify (31) to

\[
-\tilde{k}_i + \frac{1}{\lambda_i} + \tau_i + \sum_{i=2}^{N} \frac{e^{-\lambda_i (k_i - \tau_i)}}{\lambda_i} + \frac{N}{\lambda_i} \leq 0. \tag{32}
\]

Clearly, \( -\tilde{k}_i + \frac{1}{\lambda_i} + \tau_i + \sum_{i=2}^{N} \frac{e^{-\lambda_i (k_i - \tau_i)}}{\lambda_i} \) is the sum of a set of convex functions of \( \tilde{k} \), and thus also a convex function of \( \tilde{k} \). The LHS of (32) is a convex function of \( \tilde{k} \). Meanwhile, a solution of \( \tilde{k} \) that satisfies (32) must also satisfy (31).

Now we consider the power consumption at the offloader and each IDC. First, by (32), we have \( k_i > \bar{\mu}_i \). Then the expected power consumption at the offloader is

\[
\tilde{P}_1 = \nu \tilde{k}_1 + (1 - \nu) \left( \frac{1}{\tilde{k}_1} + \tau_i + \sum_{i=2}^{N} \frac{e^{-\lambda_i (k_i - \tau_i)}}{\lambda_i} \right). \tag{33}
\]
We next determine the expected power consumption \( \bar{P}_i \) for each IDC \( i \), \( i \neq 1 \). By TBO, power consumption at an IDC \( i \) after load shifting is the same as that before load shifting, i.e., \( \bar{P}_i(k_i, \mu_i) \). This is because when there is load shifted, i.e., \( D_i > k_i \), speed \( s \) of each server is 1, \( \int_0^\infty f_i(d_i)dd_i \) in (2) still holds.

A key question here is whether \( k_i \) is larger or smaller than \( \mu_i \) in (3). Note the function (3) is concave on \( k_i \) over the entire range. To obtain a convex power function on \( k_i \), we first assume \( k_i \geq \mu_i \). Later we will propose a suboptimal algorithm that incorporates the case where \( k_i < \mu_i \). We then have \( \bar{P}_i = \nu k_i + (1 - \nu)\mu_i \), where \( \mu_i = \frac{1}{\lambda_i} + \tau_i \), \( i = 2, \ldots, N \).

We further have the following optimization problem for TBO

\[
\min_{\vec{k}} \sum_{i=1}^{N} \alpha_i \left[ \nu k_i + (1 - \nu) \left( \frac{1}{\lambda_i} + \tau_i \right) \right] + \\
\alpha_1(1 - \nu) \sum_{i=2}^{N} e^{-\lambda_i(k_i - \tau_i)} + \\
s. t. \quad -k_i + \frac{1}{\lambda_i} + \tau_i + \sum_{i=2}^{N} e^{-\lambda_i(k_i - \tau_i)} + \\
\frac{Q^{-1}(\delta_1)}{\lambda_i} + \sum_{i=2}^{N} 2e^{-\lambda_i(k_i - \tau_i)} \geq 0, \\
\frac{e^{-\lambda_i(k_i - \tau_i)}}{\lambda_i} \leq \eta_i, i \neq 1, \\
\mu_i \leq k_i \leq K_i, \\
i = 1, \ldots, N. 
\]

Clearly, (34)-(38) is a convex function of \( \vec{k} \), since both the objective function and all constraint functions are convex of \( \vec{k} \). In order to extend the result to the whole range of \( k_i \), we propose Algorithm 1 that relaxes the requirement of \( k_i \geq \mu_i \) for some IDCs. Our basic idea is to let IDCs with a higher price consider the option of \( k_i < \mu_i \). In Step 3 of Algorithm 1, we let \( j \) to index the IDC (currently with \( \mu_j \leq k_j \leq K_j \)) with the largest electricity price. We set its power consumption function as \( \bar{P}_j = k_j \) in (34), and replace the constraint of \( \mu_j \leq k_j \leq K_j \) in (37) by \( 0 \leq k_j < \mu_j \). If total energy cost reduces by this setting, we then consider the option of \( k_i < \mu_i \) for the IDC with second highest electricity price. We keep doing it until energy cost does not decrease anymore.

Intuitively, by TBO, for all non-offloader IDCs, load (after shifting) is bounded by the capacity. All “excess” load is aggregated at the offloader. A higher stochastic multiplexing gain is expected globally. In addition, the QoS constraint only affects the offloader, and thus renders TBO less sensitive to both the variation of QoS constraint and load distributions.

**Algorithm 1** Relax the constraint of \( k_i \geq \mu_i \) for TBO.

1: Computes \( \vec{k} \) by (34)-(38) with the constraint \( k_i \geq \mu_i \), \( \forall i \). Let \( \vec{k} = \vec{\tilde{k}} \).
2: Sort the price vector \( \{\alpha_i\} \).
3: Choose the IDC with the largest price. Index it by \( j \). Replace \( \nu k_j + (1 - \nu)(\frac{1}{\lambda_j} + \tau_j) \) in (34) by \( k_j \), and \( \mu_j \leq k_j \leq K_j \) in (37) by \( 0 \leq k_j < \mu_j \).
4: Recomputes \( \vec{k} \) by (34)-(38) with \( \bar{P}_i = k_i \) and \( 0 \leq k_i < \mu_i \).
5: if Total energy cost reduces then
6: Let \( \vec{k} = \vec{\tilde{k}} \). Remove \( \alpha_j \) from \( \{\alpha_i\} \). Goto 3.
7: else
8: Output \( \vec{k} \) as the capacity allocation solution.
9: end if

**C. Offloader selection**

Our offloading schemes assume a given offloader. In practice, it is also important to decide which IDCs should be the offloaders. As stated, a major IDC operator may have over 100 datacenters worldwide. Then IDCs can be clustered and each cluster can be served by a selected offloader. In practice, an offloader may need to copy the background data from the IDCs served. For example, to shift windows messenger login requests, the offloader first needs to copy the corresponding user profile information from the source IDCs. Moreover, it is desirable for the offloader to be near to the IDCs it served, to reduce the load shifting delay and other costs. Thus, mission-similar or geographic adjacent IDCs may be good candidates to form an offloading clustering. In this paper, we do not address how to cluster IDCs. We mainly consider two issues. The first is how many offloaders is proper given a number of IDCs. The second is how to select an offloader given a set of IDCs in an offloading cluster.

The more offloaders, the less IDCs each offloader servers. Thus, we study the first issue by considering how many IDCs an offloader should serve. Intuitively, if the offloader has a larger capacity limit, it can serve more IDCs to exploit a larger multiplexing gain. If the load shifting constraint between the offloader and IDCs is small, it may also be desirable for the offloader to serve a larger number of IDCs to fully utilize the capacity. Finding the optimal number of IDCs served by one IDC is difficult. We will study this issue by simulations in Section VI, under different offloading schemes.

Given a set of IDCs, one can search the optimal offloader by considering every IDC and selecting the one with the best performance, which is referred to as optimal offloader selection. However, when there are a large number of candidate IDCs, e.g., over 10, there might be an in negligible time overhead in selecting offloader by the scheme. We thus explore an alternative approach. There are several factors in choosing a good offloader. A candidate may have a lower electricity price, more available network bandwidth for load shifting from other IDCs, and a larger capacity limit. To comprehensively consider the factors, we rely on RBLs to choose the offloader. The optimization model (12)-(17) incorporates all these aspects. Its solution allows us to calculate the average amount of load each IDC absorbs from other IDCs, which is denoted by \( \Lambda_i \). We have \( \Lambda_i = \sum_{j \neq i} r_{ij}\mu_j \). The IDC with the largest \( \Lambda_i \) is chosen as the offloader. We refer to this scheme as **RBLs-based offloader selection**, which is suboptimal but has a smaller time overhead. We will compare this scheme to the optimal offloader selection scheme in the simulations. Note that different IDCs may be selected as the offloader at different time. The variation of electric price and load statistics can trigger the reselection of the offloader.

**D. Implementation considerations**

In all the proposed schemes, RBLs, RBO, and TBO, the decision is made in a relatively large time scale, based on statistics of load-demand. To apply them, there are several practical issues to be addressed.
To obtain load statistics, i.e., $\mu_i$ and $\sigma_i^2$, one can use instantaneous measurements, e.g., applying a sliding window, or use historic data. To update the load statistics with the decision-maker, an IDC can follow either predetermined schedule, or on-demand update, where an update is triggered if the current load statistics differ from the previous values by certain thresholds. Note that the communication overhead between each IDC and the decision-maker is low since updating the decision-maker is performed in a large time scale.

Since our analytical models are based on certain load distributions, there is potential discrepancy between the assumed load patterns (e.g., based on measurements) and the real load distribution. As a consequence, our scheme may result in over- or under-provisioning of capacity. To address this issue, we consider a dynamic capacity/server provisioning (DCP) scheme. DCP complements the optimization model based server provisioning by performing local adjustment based on the QoS measurement. In particular, each IDC monitors its own outage probability. It increases (decreases) its local active server number if the outage probability is higher (lower) than the desired value. Techniques from stochastic approximation and adaptive algorithms can be readily applied here. In this context, a good starting point is provided by the optimal solution determined by the decision-maker with assumptions on load distribution. Note the notion of DCP is different from dynamic speed/voltage scaling (DVS), which is to adjust server speed instead of the number of active servers. We will evaluate our proposed static schemes, i.e., RBO and TBO, with the DCP in Section VI.

VI. PERFORMANCE EVALUATION

In this section, we evaluate the performance of the proposed load shifting schemes. We first compare RRLS, RBO, and TBO based on synthetic load generated with a certain distribution. We then use real traffic traces to study the performance of RBO and TBO.

A. Comparing RRLS, RBO and TBO using simulated traces

1) Simulation setup: We consider $N = 15$ IDCs. Each IDC $i$ has a capacity limit $K_i$ of 100. We conduct 100 rounds of simulations. In each round, we generate load for each IDC independently following a certain distribution. We consider Gaussian, exponential, and Weibull distribution with a shape factor of 0.5 (heavy-tailed setting), respectively. We let the mean of load, $\mu_i$, and variance $\sigma_i^2$ for each IDC $i$, be 10 and 25, respectively. Electricity price $\alpha_i$ is randomly generated in the range from 1 to 5. Load shifting constraint $\eta_{ij}$ is set as the product of $\mu_i$ and a coefficient randomly chosen from $(0, 0.5)$. We consider each IDC has the same outage probability constraint $\delta$, which takes values in $[0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.15, 0.2]$. We consider the idle state power consumption ratio $\nu$ of 0.6. The performance metric is the total energy cost of IDCs. We use $P_i$ of (2) as power consumption of each server in simulation, and $P_t$ of (3) as power of an IDC in analysis. Note that among the 15 IDCs, there is an offloader selected by the RRLS-based offloader selection scheme in each simulation round, in both the cases of RBO and TBO.

2) Performance of RRLS, RBO and TBO: We compare the performance of RRLS, TBO, RBO, and no-load-shifting. Because of the approximation and assumptions on distributions, the proposed schemes (without capacity scaling) do not always satisfy the outage constraint. To remedy the discrepancy in $\delta$, we scale up/down the local server capacity for each scheme so that the outage probability constraint is satisfied with equality. Curves of the energy cost of different schemes without and with capacity scaling, are plotted in Fig. 3 and Fig. 4, respectively. Comparing Fig. 3 and Fig. 4, we observe that with Gaussian distribution, there is virtually no difference between capacity scaling and without scaling because the outage probability constraint is mostly satisfied in all schemes. A slightly larger difference is observed for exponential distribution, especially when $\delta$ is small. The largest difference occurs in Weibull distribution, and especially with a small $\delta$. Therefore, in practice, the capacity scaling or dynamic server provisioning as proposed in last section is needed when load is heavy-tailed and when the QoS constraint is tight. In other cases, i.e., the QoS constraint is loose and load is non heavy-tailed, our analytical results can be directly applied. Note that capacity scaling here only remedies the discrepancy of the assumed or approximated load distribution by our analytical results, while accurate load statistics, i.e., mean and variance, are the input in this simulation. In next part where simulation is based on real cluster traces, we use historic load statistics as the input instead of accurate load statistics. Therefore, we need dynamic capacity/server provisioning (DCP) proposed in Section V-D to remedy the gap between historic load statistics and accurate load statistics, and the discrepancy by the assumed or approximated load distribution as well.

We observe from both Fig. 3 and Fig. 4 that RRLS results in by far the lowest energy cost in all three load distributions, followed by TBO, No-load-shifting has the highest energy cost in most cases. The energy costs of RRLS and TBO increase slowly with $\delta$, and those of RBO and no-load-shifting increase much faster, in particular under Weibull distribution. This observation indicates that TBO is insensitive to the variation...
of the outage probability constraint. When $\delta$ is large, say 0.2, it is observed that TBO leads to a larger energy cost than that of RBO. This is because outage probability at each IDC is 0 in TBO, except the offloader. In summary, RBLs (when complexity allows) and TBO (a simpler architecture) are good choices for operation, in terms of energy efficiency and performance robustness.

We also observe our analytical results are more conservative than simulation results. Energy cost in analysis is about 5% larger than that in simulation for RBLs, 5-10% higher for RBO. The main contributor to the gap is the approximation in power consumption model, where the approximated expected energy cost and shifting cost, with different $\beta_{ij}$ values.

3) Impact of load shifting cost: In the paper, we do not consider load shifting cost. We numerically evaluate the impact of load shifting cost on our proposed schemes. Similar to [35], we consider a linear load shifting cost model. We use $\beta_{ij}$ to denote the load shifting cost per-unit of load from IDC $i$ to IDC $j$. Thus, the total load shifting costs for RBLs, RBO, and TBO are $\sum_{i=1}^{N} \sum_{j=1, j \neq i}^{N} \beta_{ij} r_{ij} \mu_i$, $\sum_{i=1}^{N} \beta_{i1} r_{1i} \mu_i$, and $\sum_{i=2}^{N} \beta_{1i} e^{-\lambda_i (t_{ij} - t_i)}$, respectively. We add the load shifting costs to the objective functions of the optimization models of RBLs, RBO, and TBO, respectively. We relax the load shifting constraints to focus on the impact of load shifting costs. We use the same simulation setting as in Fig. 3. We consider a homogenous cost per unit of load for all the IDC pairs, i.e., $\beta_{ij} = \beta$. We present the curves of total cost, including the energy cost and shifting cost, with different $\beta$ in Fig. 5. We observe that the total cost increases as $\beta$ increases, as expected.

When $\beta$ is large, say over 10, the total costs of RBLs and RBO do not increase anymore, because it is not desirable for RBLs and RBO to shift load in the case. In this case, the gain achieved by load shifting is less than the cost incurred by it. However, for TBO, the total cost increases much slower than

Fig. 3: Comparison among RBLs, RBO, TBO, and no load shifting with three load distributions. (The case without capacity scaling)

Fig. 4: Comparison among RBLs, RBO, TBO, and no load shifting with three load distributions (The case when $\delta$ is exactly satisfied at each IDC with capacity scaling. Note here not in all cases we need to scale up the capacity to make $\delta$ satisfied.)

Fig. 6: The impact on per-IDC energy cost and the optimal number of IDCs per offloader by different capacity limits and different load shifting constraints, respectively.

Fig. 7: Cost increment ratio by RBLs-based optimal offloader selection over optimal offloader selection.
that of RBLS and RBO when $\beta < 10$. When $\beta$ is between 3 or 70, the total cost of TBO is even smaller than that of RBLS. This is an interesting finding, which is different from Fig. 3 that shows RBLS has a smaller cost than TBO. The reason that RBLS achieves the smallest cost in the setting of Fig. 3 is because it has the largest amount of load shifted among IDCs. But in terms of cost reduction achieved by per unit of load shifted, RBLS is worse than TBO since it is a ratio-based scheme. In other words, RBLS is less efficient in reducing energy cost per unit of load shifted than TBO. Thus, when the shifting cost of per unit of load is large, i.e., 10, RBLS stops shifting load, while TBO still exploits load shifting to reduce energy cost. We also observe that the total cost of TBO is larger than RBLS and RBO when $\beta$ is larger than 60 and keeps increasing with $\beta$. This is because, when $\beta$ gets larger, TBO, each IDC $i$ has to tune a larger $k_i$ to reduce the load shifted. The larger $\beta$, the larger $k_i$ and thus the total (energy) cost. Results of Fig. 5 indicate that when there is a load shifting cost, RBLS, RBO and TBO need to be carefully compared. There is a threshold beyond which TBO is even better than RBLS due to its efficiency in reducing energy cost per unit of load. There is also a larger threshold beyond which TBO is even worse than RBO (i.e., no-load-shifting in this case). Note that Fig. 5 is based on Gaussian distributed load. We have the same conclusions with other load distributions.

4) Offloader selection: In Fig. 6, we study the impact of the number of IDCs per offloader by the offloader’s capacity limit and load shifting constraints, respectively. In Fig. 6(a), we evaluate the impact by different capacity limits of the offloader. We use the same simulation setting on mean and variance of each IDC’s load as Fig. 3. We assume homogenous electricity prices by setting $\alpha_1 = 10$ to focus on the stochastic multiplexing gain by different number of IDCs. We consider the number of IDCs from 5 to 15, among which there is an offloader selected. We consider a metric named per-IDC energy cost, which is simply calculated by dividing total cost by the number of IDCs. Obviously, a small per-IDC energy cost is preferred when choosing the number of IDCs an offloader serves. Note that Fig. 6(a) shows simulation results, not analytical results. We observe that under different capacity limits, and different offloading schemes, RBO and TBO, per-IDC energy cost does not change significantly under a different number of IDCs. We also observe that when the capacity limit of the offloader $K_1$ is small, i.e., 50, per-IDC energy cost keeps increasing as the number of IDCs increases. This is because a small number of IDCs can fully utilize the offloader’s capacity. Adding more IDCs to the offloader will lead to less load shifted, thus each IDC has a larger capacity. When $K_1 = 85$, per-IDC energy cost first decreases, and then increases after the number of IDCs is larger than a certain threshold, i.e., 11 for RBO and 9 for TBO. This is because, when the number of IDCs is smaller than the threshold, the offloader’s capacity is not fully utilized. Therefore, a larger number of IDCs achieves a higher stochastic multiplexing gain and results in a smaller cost. When the number of IDCs is larger than the threshold, the offloader’s capacity is fully utilized. More IDCs lead to a higher per-IDC cost for the same reason as the case of $K_1 = 50$. In this case, 11 and 9 are the optimal numbers of IDCs an offloader serves for RBO and TBO, respectively.

Fig. 6(b) shows the results under different load shifting constraints. We consider homogenous shifting constraints for different IDCs with the offloader, i.e., $\eta_{i1} = 1$ and 15, respectively. We observe that when $\eta_{i1} = 1$, per-IDC energy cost keeps decreasing as the the number of IDCs increases. In this case, only a large number of IDCs can fully utilize the offloader’s capacity. When $\eta_{i1} = 15$, per-IDC energy cost for RBO keeps increasing. Per-IDC energy cost for TBO first decreases, and then increases. The results indicate that load shifting constraints also have important effects on selecting the number of offloaders. In practice, the operator may fully consider the effect of capacity limit, and load shifting constraint in selecting the number of the offloader. There are also some other factors such as electricity price. When the price at the location of an offloader is low, it is desirable for it to serve a large number of IDCs.

Our simulation results for RBO and TBO in Fig. 3 and Fig. 4 are based on RBLS-based offloader selection, as introduced in Section V-C. RBLS-based offloader selection is sub-optimal. In Fig. 7, we study the cost increment by RBLS-based offloader selection over the optimal offloader selection scheme. We consider $N = 6$. The other simulation setting is the same as in Fig. 3. We perform 500 rounds of simulations. In each round, the cost-increment-ratio is calculated by dividing the cost increment of RBLS-based offloader selection by the cost of the optimal offloader selection scheme. We report the average cost-increment-ratio in Fig. 7 for both RBO and TBO under two different outage probability constraints, i.e., 0.001, and 0.0005. It is observed for RBO, the average cost-increment-ratio is around 1.5%, which is negligible. The average cost-increment-ratio in the case of TBO is larger, which is slightly larger than 3.5%. A smaller outage probability, i.e., 0.0005 leads to a slightly higher average cost-increment-ratio. Overall, it can be concluded that the performance of RBLS-based offloader selection is close to the optimal offloader selection scheme.

B. Real-trace based simulation

1) Simulation setup: We study the performance of our load shifting schemes using traffic traces of a real datacenter operated by a large cloud computing service provider. We only have the traces of the one cluster for 15 days. We use the one-cluster 15-day traces as the 15-cluster one-day traces. While this is not the ideal setting, it captures several important aspects of a real datacenter’s traffic traces. First, from the traffic traces, we observe time correlation. That is, there is likely low traffic volume for a consecutive time, followed by a high traffic period. Second, there is likely similarity in traffic volume at the same hours of different days, which enables us to evaluate our schemes in inter-dependent traffic.

The traces record the total incoming traffic volume in every 100ms interval for 15 days. We note that the traces we have are traffic traces, not CPU load traces. To approximate load traces, we assume a proportional relation between the traffic volume and load density. In the future, we hope to obtain load traces and compare load traces with traffic traces.
Our basic observation on the traces is that the traffic does not change smoothly. There are many idle periods followed by a busy period with large traffic volume in each time interval. Thus, our static schemes of RBO and TBO may not be sufficient. Thus we also combine them with the DCP scheme considered in Section V-D, and compare the performance of the combined schemes to RBO and TBO only. In DCP, each IDC monitors load density in every interval (a 0.1 sec time granularity) for 200 seconds and uses the total 2000 measurement samples to calculate the current outage probability. Each IDC then adjusts its capacity based on the ratio of recent measurement samples to calculate the current outage probability. The target outage probability for each IDC is set as $\delta = 0.05$. For RBO and TBO, we assume that load at each IDC follows an exponential distribution. We observe that DCP-only leads to the largest average energy cost per second. Meanwhile, it leads to a maximum outage probability that is roughly 2.7 times larger than the target outage probability. Thus, DCP alone is not efficient. We next summarize the results of various proposed schemes with accurate load statistics.

Another important problem is that RBO and TBO need load statistics, i.e., $\mu_i$ and $\sigma_i$ as the input. As the real traffic (assumed to be proportional to load) is not smooth, it is difficult to obtain accurate load statistics. We use the historic load statistics as the input. In the simulation, we determine the capacity of each IDC and load shifting parameters every half an hour from 8AM to 8PM. Load statistics of the past half hour is used as the input. For example, the capacity allocation by RBO and TBO at 9AM is based on the load statistics between 8:30AM and 9AM. Then, energy cost and outage probability from 9AM to 9:30AM (based on the capacity allocation at 9AM) is collected at 9:29:59:900AM (hour:minutes:seconds:milliseconds). We will also compare the performance of using historic load statistics to that in the ideal (not practical) setting where accurate load statistics information is available (i.e., load statistics between 9AM to 9:30AM in the example).

Our simulation is limited due to the data set available. Price and load shifting constraint are set the same as in the last subsection. We evaluate the performance in terms of both the energy cost and the outage probability. Energy cost is considered as the average value per second over the whole day. We consider the outage probability as the maximum one among all the 15 IDCs for every half an hour. We then obtain the average maximum outage probability over the daytime hours (8AM to 8PM) and the standard deviation of the maximum outage probability.

2) Results: We illustrate the energy cost per second and the average maximum outage probability in Fig. 8(a) and Fig. 8(b). Five schemes are compared: DCP-only (no-load-shifting), RBO-only, TBO-only, RBO+DCP, and TBO+DCP. For all the schemes except DCP-only, we consider the case of using accurate load statistics and using historic statistics. The target outage probability for each IDC is set as $\delta = 0.05$. For RBO and TBO, we assume that load at each IDC follows an exponential distribution. We observe that DCP-only leads to the largest average energy cost per second. Meanwhile, it leads to a maximum outage probability that is roughly 2.7 times larger than the target outage probability. Thus, DCP alone is not efficient. We next summarize the results of various proposed schemes with accurate load statistics. Compared to RBO, TBO can achieve a smaller outage probability at almost the same energy cost. RBO+DCP and TBO+DCP decrease the outage probability compared to RBO only and TBO only, respectively. When using accurate load statistics, TBO+DCP results in a much smaller energy cost than that of both RBO+DCP and DCP-only. In terms of the average maximum outage probability, we observe from Fig. 8(b) that TBO+DCP (with accurate load statistics) results in a smaller outage probability than RBO+DCP at a much smaller cost. This is because it is more difficult to use DCP to reduce the maximum outage probability in the case of RBO, where all IDCs have a nonzero outage probability. Further, the average maximum outage probability of DCP-only is significantly higher than that of RBO+DCP and TBO+DCP, which accounts for 220% and 330%, respectively. We also study the standard deviation of the outage probability of each scheme in Fig. 8(c), which indicates the robustness of each scheme. It shows that when using historic load statistics, outage probability of TBO+DCP only has a deviation of about 0.02, which is much smaller than other schemes in the case of using historic load statistics. The result shows that TBO+DCP is robust when using historic load statistics. We summarize the performance of various schemes with historic load statistics. First, it is observed that RBO-only
Our schemes include three phases: 1) inputs gathering, 2) computations, and 3) executions. The inputs of our schemes are simple. Load demand statistics are almost the only inputs needed in our schemes. Note that load statistics are widely considered as the input information in existing proposed schemes, e.g., in [18][34][45]. It is practical to estimate or predict load statistics since IDC traffic demand is usually similar for the same hour in different days. For example, traffic of requests is usually higher in daytime than in the night. In [17], it shows that traffic of a Microsoft datacenter fluctuates over day and night times and has the same patterns in the same hours during different days. Load estimation may have errors, which may affect the efficiency of our schemes. We will discuss the impact of estimation error in next subsection. As discussed in Section V-D, input gathering takes little time between each IDC and the decision-maker. Thus, different phases of our schemes are simple and practical. We next discuss the efficiency of our proposed schemes, especially on TBO+DCP.

B. Efficiency of the proposed schemes

Our schemes, especially TBO, are efficient since they leverage both electricity price diversity gain and load stochastic multiplexing gain. The former one has been well exhibited by previous literatures, e.g., up to 40% energy saving by the pioneering work of [29], 10%-40% cost reduction by the proposed distributed load balancing scheme in [33]. Our schemes bring additional cost reduction by exploiting stochastic multiplexing gain. As showed in Section IV, exploiting load stochastic multiplexing gain leads to $\Theta(N)$ reduction in number of active servers. Our extensive simulations in Section VI also indicates significant energy cost reduction by our proposed scheme, e.g., TBO vs no-load-shifting.

The efficiency of our schemes is mainly discounted by the deviation of real load statistics and distribution discrepancy. As discussed, the estimation of load statistics may have errors. Moreover, the assumption of load distribution, e.g., exponential distribution in TBO, may have discrepancy from real load distribution. It shows that using historic load statistics can lead to a higher energy cost and outage probability than using accurate load statistics, i.e. in Fig. 8(a) and Fig. 8(b). However, with DCP, our proposed TBO is shown to have a energy cost and outage probability that is close to the case of using accurate load statistics, and is much better than DCP only, i.e., near 40% energy cost reduction and 60% outage probability decrease at the same time. DCP is efficient for TBO to address load statistics and distribution discrepancy because only at the offloader needs to perform capacity adjustment. Outage probability of all other IDCs is always 0 and no capacity adjustment is needed.

Thus, our proposed TBO is efficient given the implementation of DCP or when load estimation is accurate. Moreover, our schemes are discounted by load shifting cost, which is mainly the revenue loss by extra delay as discussed in Section III.B. Thus the efficiency of our schemes can be improved when more delay-tolerant jobs instead of delay-sensitive jobs are offloaded.

VIII. CONCLUSIONS

In this paper, our objective is to reduce energy cost of distributed IDCs with dynamic load-demand. We design joint capacity allocation and load shifting schemes that minimize total cost under the outage probability constraint. We propose three load shifting schemes, i.e., ratio-based load swapping,
ratio-based offloading, and threshold-based offloading. We observe that ratio-based load swapping achieves the lowest cost, because it best exploits the multiplexing gain, electric price difference, and load balancing. For a simpler architecture with one offloader, we observe that TBO performs well — it achieves good energy efficiency and is robust to traffic variations and estimation errors. Simulations from both synthetic data and real traffic traces demonstrate significant performance improvement of the proposed schemes, in particular, RBLS and TBO, over the non-load-shifting scheme. Simulations also suggest that DCP plays an important role in satisfying the QoS constraint. With DCP, using recent historic load statistics as input leads to desirable performance compared to the case with accurate load statistics. Furthermore, load distributions affect performance. In general, heavy-tailed distributions result in higher energy cost. Therefore, better understanding of IDC load will improve the design of IDC operations.

IX. Acknowledgements

We thank the editor and all the reviewers for their comments and suggestions, which significantly improved the quality of the paper. We also thank Professor Charles U. Martel at University of California, Davis, for his help on the revision of the paper.

REFERENCES


