Service Deactivation Aware Placement and Defragmentation in Enterprise Clouds

Sourav Dutta
IBM Research, India.

Akshat Verma
IBM Research, India.

***Abstract*— On-demand service activation and deactivation has made private cloud a platform of choice for enterprises. Private clouds allow end users to request virtual machines along with a lease period, during which the virtual machine will be utilized. Automatic deactivation of virtual machines due to the expiry of the lease leads to fragmentation of the cloud; a large number of physical servers become operational with low utilization as virtual machines have been deactivated. To optimize the use of resources, clouds would need to periodically '*de-fragment*' the cloud, leading to performance implications. In this work, we propose a service deactivation aware placement methodology, $SDAP$, which places virtual machines in a way that minimizes the cost of de-fragmentation in the cloud. Coupled with $SDAP$, we also propose a low cost de-fragmentation methodology, $Defrag$, to periodically consolidate the cloud. Using request traces from a live production cloud, we show that $SDAP$ and $Defrag$ can ensure high utilization in the cloud without expensive de-fragmentation costs.**

I. Introduction

The illusion of infinite computing resources available on-demand coupled with the concept of pay-as-you-use through *Cloud Computing* [2], [9] has transformed the dream of computing as a utility into a commercial reality in the present IT world. With clouds, enterprise and other software business applications do not need a large initial capital outlay or worry about over and under-provisioning of resources. Clouds are witnessing favorable adoption from the software industry and major web application providers are increasingly shifting to this emerging concept.

One of the primary reasons for enterprises to move away from traditional data centers are the high facilities cost. The energy usage and environmental impact of data centers has become a matter of significant concern [12]. According to [18], 48% of the total IT electricity cost is used to keep the data centers running in USA alone. Traditional data centers are not energy proportional, i.e., the power drawn by data centers is not proportional to the amount of work done.

Servers are often characterized by the power model, which captures power as a function of their utilization. Typical servers draw a significant amount of power, even when they are not utilized (upto 70% of the peak power [1]). Hence, the power drawn by servers is not proportional to the work done if the servers are not running at 100% utilization. In data centers, server resources are allocated to workloads to cater to their peak resource usage. The ratio between the peak resource usage of an application and its average resource usage can be as high as 10 [21]. Hence, typical server utilization in data centers is around 10% making them very inefficient in their energy usage.

Dynamic consolidation or reconfiguration in virtualized data centers has been a popular area of research and has helped improve resource utilization and energy efficiency [13], [20], [22], [11]. In dynamic consolidation, the resource usage for the workloads are monitored and workloads consolidated periodically to the required number of servers. Dynamic consolidation is useful for long-running enterprise workloads. Enterprise clouds usually offer a more predictable lease-based model for resource usage. A request for a new virtual machine in an enterprise cloud may often specify the duration during which the virtual machine would be needed [10], [3], e.g., IBM SCE+, CSP2, Cloupia, OpenNebula, Amazon Labslice to name a few. Private clouds involve an approval for infrastructure requests and includes the request duration of the virtual machine. Hence, service deactivation is an important characteristic in enterprise clouds. Recent research in this domain [16], [19] establishes the importance of the lease time for reducing server sprawl.

Placement in clouds has often focused on improving the resource utilization of the servers. The placement problem can be modeled as the classical *bin-packing* problem [6] which is known to be NP-hard. The goal of the bin-packing problem is to minimize the total number of bins (or servers) used to pack a given number of balls (virtual machines in our case). Hence, clouds implement variants of bin-packing policies like first-fit or best-fit [8], [4] with the goal to reduce fragmentation and improve the resource utilization. Heuristics like the PCP [21] study the peak-requirement characteristics of the applications and place complementary workloads together in the goal to improve resource utilization. Improving the resource utilization reduces the per-VM cost of resource used. Further, since the power drawn by servers has a large component that is independent of the actual usage, reducing the number of servers reduces the overall power cost for the data center.

However, due to service deactivation, virtual machines may be deleted creating holes in the cloud infrastructure. State-of-the-art cloud placement techniques are oblivious to service deactivation and need to periodically consolidate or defragment the cloud infrastructure. Defragmentation involves consolidating the virtual machines to a smaller set of servers and improving the resource utilization. In order for the defragmentation to be transparent to cloud consumers, cloud providers use live migration to migrate virtual machines [7],

[14]. However, periodic defragmentation using live migration has associated performance costs [11], [20], [23].

In this work, we address the problem of placing virtual machines in a cloud-based data centers, where virtual machine requests have associated lease periods. We make the following contributions.

A. Contribution

We propose a detailed model to capture the various costs incurred for serving a virtual machine through its entire life-cycle in a cloud environment. We identify resource costs, skew costs, power-down costs, and defragmentation costs as the key components determining the facilities costs incurred by a virtual machine. We conduct an experimental study of cloud requests and characterize the request patterns. Using the insights drawn from our study, we propose a new placement-scheme *Service Deactivation Aware Placement (SDAP)* and a periodic defragmentation scheme to minimize the total facilities costs of running all the virtual machines in the cloud. *SDAP* uses the leases of the existing virtual machines in the cloud to identify the best candidate server for placing a new virtual machine. Using a real trace of cloud requests over a period of 1 year, we evaluate $SDAP$ with state of the art placement schemes in the cloud. We observe that $SDAP$ outperforms First-Fit based schemes and reduces the number of migrations by more than 30%.

The rest of the paper is organized as follows. In Section II, we provides a model and formulation of the placement problem in cloud. We study cloud requests from a live cloud and identify some salient characteristics in Section. III. We present the $SDAP$ and defragmentation algorithm in Section IV. In Section V, we present an evaluation study using a 1 year long trace of cloud requests to an operational cloud. We compare our work with the related work and conclude with our key observations in Section. VI.

II. Model and Preliminaries

In this section, we present the optimization problem addressed in this work.

A. Problem Formulation

We consider an enterprise cloud (extensible to public clouds), where customers request for virtual machines (VM). The request captures the resource requirement and a lease period. A cloud placement algorithm creates a VM on a target server to maximize the efficiency of the data center and minimize the additional power and operational costs. After the expiration of a VM's lease, it is deallocated from the hosting server. The deallocation actions lead to fragmentation of the datacenter. The server thereby run at lower utilization levels until additional VMs are allocated to it.

In order to increase the utilization, clouds may periodically defragment the cloud. During the defragmentation, VMs are migrated from low utilization servers to high utilization servers. If all VMs hosted on a server are deallocated or migrated, the server can be shut down to reduce power. Hence, defragmentation improves the overall resource utilization in the data center and reduces power costs.

However, a VM migration incurs a cost and in real applications, such live migration costs are a major deterrent. In this paper, we propose a novel placement algorithm such that the total number of migrations during the 'Defrag' phase is reduced. We also show that this intelligent placement procedure, in addition, also reduces the average power consumed with a higher utilization of the datacenter.

B. Server Characteristics

Each server of the datacenter is characterized by the total amount of CPU cores and memory capacity, represented as C_{max} and M_{max} respectively. The power drawn by the compute elements of a server are characterized by CPU *power intercept* P_i^c and the CPU *power slope* P_s^c. P_i^c captures the power consumed by the server cores at idle state (or 0 utilization), while P_s^c captures the rate at which the power usage increases per unit CPU capacity used in the server. Similarly, we define P_i^m and P_s^m to capture the memory power intercept and the memory power slope respectively.

The CPU power characteristics are determined by the model and family of the server, e.g., a blade server may consume lesser static power than a rack server and have a lower P_i^c. However, it may have a higher power-proportionate and thus a higher P_s^c. The memory power parameters are likewise determined by the type of RAM used by the server, i.e., DDR2 or DDR3 etc. These parameters characterize the overall power consumed by a server, while it is operational. Since servers may also be switched off, we also consider the duration ON for which a server is powered on. The ON period of a server includes the time during which at least one active VM is placed on it.

C. VM Characteristics

A request for a virtual machine R_i is characterized by its CPU core and memory demands, denoted by C_i and M_i respectively. The request also contains information about the lease period for the VM using its start time T_s and the end time T_e. In this work, we propose that the placement algorithm should take into account T_e of the VM and existing VMs in addition to the resource requirements. Placement algorithms often consider over-commitment of the resources by sizing each VM (workload) to an off-peak value (e.g., 95 percentile CPU/Memory). We are oblivious to a sizing algorithm and use any provided VM sizes as input to $SDAP$. Hence, both $SDAP$ and $Defrag$ work independent of whether the resources are over-committed or not.

We also use a live migration cost function to capture the cost of migrating the VM. The live migration cost of a VM is typically captured by a function of the current CPU utilization of the host server and the active memory used by the VM [23], [24].

The overall goal of the placement problem is to minimize the total data center cost, which includes the operational cost of the data center resources as well as the migration cost.

III. UNDERSTANDING CLOUD REQUESTS

Clouds provide a new compute model to end-users. In this section, we study the nature of virtual machine requests made in clouds.

For the study, we used the request log from IBM Research Compute Cloud (RC2) [3]. IBM RC2 is a cloud computing platform for use by the worldwide IBM Research community. It allows users to request for virtual machines hosted on KVM, Xen or $pHyp$ hypervisors hosted on IBM X-Series and P-Series blade servers. The storage for the hosts is provisioned from a SAN. The cloud supports a large number of users and VM instances (many hundreds). We obtained a log of all requests made to RC2 from October, 2009 to October, 2010. Each request entry consisted of the following fields: (i) VM ID (ii) creation date for the VM (iii) expiry date for the VM (iv) number of cores needed by the VM and (v) the amount of memory needed by the VM. The total number of VM requests made over this one year period exceeded 10000. Hence, we believe the request log is fairly representative of typical cloud request patterns.

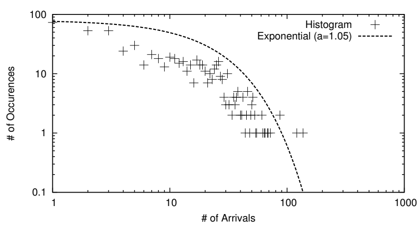

Fig. 1. Histogram of the number of arrivals

We first studied the arrival pattern of the requests. One of the most important things we wanted to study was whether requests to a cloud came in large bursts. Large bursts would imply that the total resources used the cloud would suddenly increase. Hence, when the requests went away, the resources would be freed up. Hence, bursty arrival rate would require fairly frequent defragmentation to reduce facilities cost. We divided the entire trace into 12 hour periods and computed the number of requests made in each 12 hour period. Fig. 1 captures the histogram of the number of arrivals. We observed that the probability that D requests would arrive in a period falls exponentially with D. Hence, *we conjectured that the arrival distribution is exponential in nature*. To confirm this conjecture, we also plotted an exponential distribution and observed that an exponential distribution with $\alpha = 1.05$ fitted the distribution very well.

We next studied the distribution of the lease period for the virtual machine requests. If the lease periods are stable, then defragmentation is not a big issue. This is because most requests would expire after similar times. Hence, a first-fit mechanism can work well by always placing new virtual machines on the first available servers. We observed that the lease period distribution is very heavy-tailed. In fact, we initially conjectured that, similar to many web workloads, lease durations would fit a lognormal distribution. However,

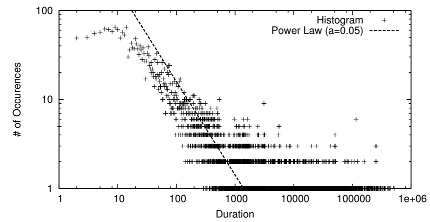

Fig. 2. Histogram of Lease Periods

we observed that lease periods were even more bursty than lognormal. Hence, we fitted a power law distribution with a high burst factor ($a = 0.05$) and found it to fit the distribution well (Fig. 2). Hence, lease periods in cloud are fairly bursty.

Our study helped us to draw two very important insights. We noted that very frequent defragmentation may not be required since the arrivals are not bursty. However, due to high variation in lease periods, it is important to include the service deactivation, while coming up with a placement decision. We next propose a placement scheme that is designed based on these insights.

IV. ALGORITHM

In this section, we discuss the *Service Deactivation Aware Placement* $SDAP$ algorithm. The algorithm is inspired from our workload characterization study and incorporate service deactivation during a VM's placement. We next present a novel break-up of the overall data center operational resource costs and defragmentation costs, which helps us to solve the problem.

A. Cost Parameters

We present an alternate representation of the total data center costs, which captures the impact of service deactivation. The alternate representation still accurately captures total data center costs but allow a more natural interpretation. Our placement algorithm uses this cost model to come up with an efficient placement. We divide the total data center cost into four cost components.

- Operational Resource Cost (RC): The operational resource cost captures the operational cost of resources used by a request VM to a server containing enough free resources to accommodate the VM. RC depends on the amount of CPU cores C and memory M requested by the VM and is given as $CP_s^c + MP_s^m$. If the server did not have any VMs already placed on the server, we also add the static power intercepts P_i^c and P_i^m to RC. This is multiplied by the lease duration of the server to estimate the cost during the lifetime of the VM. In this work, we only use power to capture the resource cost. However, the intercept-slope model can be used to capture other costs such as space and labour.
- Skew Cost (SC): Applications may be CPU-intensive or memory-intensive. Skew Cost is used to capture wastage of one resource because the server was bottle-necked by

the other resource. To reduce the wastage of resources of the datacenter, ideally we would like each server to have a perfect mixture of CPU and memory-intensive applications. We define the application skew of a server as the ratio of memory to CPU core usage. Closer the skew is to the ratio of total memory to total number of cores, lesser is the resource wastage. SC computes the resource wastage cost incurred by placing the VM on a server and equals the cost of wasted resources.

- Power-down Cost (PC): Placing a VM on a server may prevent the server from being switched off later. Hence, placing a VM R_i on a server leads to a loss of potential shutdown in the future and an increase in ON period. We capture this effect by the *Power-down* cost for a server. To estimate PC, we find an existing VM on the server with the largest end time T_e^*. If the ending time of the new VM's lease T_e^i is greater than T_e^*, the server loses the opportunity to shut down in the period $T_e^i - T_e^*$. Hence, the power-down cost is computed as $(P_c^i + P_m^i)(T_e^i - T_e^*)$. If $T_e^i \leq T_e^*$, then the ON period for the server does not change and PC equals 0.
- Defrag Cost (DC): When a server is 'defragmented', we incur the migration costs of the VMs of the server minus the power cost used by the VM. So, if the current VM is allocated to a server that is a candidate for the 'Defrag' procedure in the next period T_D, then we experience an extra cost for migrating the request VM also. If the lease time of the VM ends at time T_e^i before the next defrag period T_D, then there exists no such costs. Hence, DC computes the *potential extra migration cost* by placing the VM on a particular server. Formally, we estimate DC as $p_D MC_i$ if $T_e^i > T_D$ and 0 otherwise, where p_D is the probability that the server will be 'defragmented'. p_D is calculated empirically using a distribution of the aggregate resource requirement.

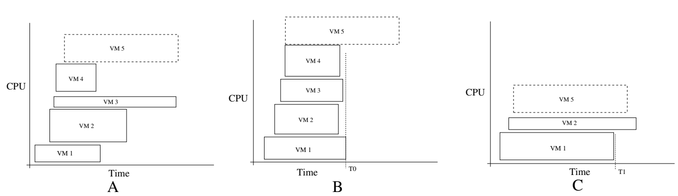

Fig. 3. Cost parameters.

We elaborate the 2 new cost parameters PC and DC using an example in Figure 3. In this example, we have three separate running candidate servers (A, B, and C) with a set of existing VMs. Consider a request for a new $VM5$ to be placed. The lease time of $VM5$ is longer than any of the VMs allocated to Server B. Thus, if $VM5$ is placed on Server B, then the server will be denied a chance of shutting down after time $T0$. Hence, by placing $VM5$ in Server B, we incur the extra cost of running the server longer than it may be required. This is incorporated as the Power-down cost (PC).

Similarly, we find that Server C has low utilization, and hence forms a good candidate for defrag (high p_D). Further, $VM1$ in Server C expires before the defragmentation is started at $T1$, and hence its utilization further decreases. The defragmentation procedure would then try to migrate $VM2$ from Server C, and shut it down. This would improve the efficiency of the datacenter and reduce the cost. However, if we were to place $VM5$ in Server C, which has a lease period beyond $T1$, then during the defragmentation period we will have to migrate the extra VM placed. This cost is represented by the Defrag Cost (DC).

Server A has high utilization and hence the probability that it will be defragmented is low. Similarly, it incurs no shut-down cost as the ending time for $VM5$ is less than an existing VM, $VM3$. Hence, Server A has no shut-down cost or Defrag cost. Hence, a placement algorithm should prefer Server A over both server B and server C, while placing $VM5$.

We next present an algorithm that uses the above cost parameters to place VMs on a server that minimizes the sum of all these cost parameters.

B. $SDAP$ Algorithm

The *Service Deactivation Aware Placement* ($SDAP$) algorithm helps to intelligently place application VMs into the datacenter guided by their request and cost parameters to minimize of the number of live migration during 'Defragmentation' procedure, while ensuring that the power cost of the data center is minimized.

Given a request virtual machine characterized by its resource requirement and also the period of its lease, $SDAP$ initially finds all the servers in the datacenter that are candidates to host the VM. That is, it recognizes the servers that at least have the free resources to meet the demands of the VM.

For each of the servers, $SDAP$ then calculates the total cost that would be incurred if the VM is to be placed in it. The total cost is obtained by the sum of the cost parameters described earlier, i.e., $RC + SC + PC + DC$. The VM is then placed on the server which offers the minimum total cost.

It is interesting to observe that RC will be the least in the most power efficient servers, which also have a lower idle power consumption. Hence it ensures that the efficient servers run at higher utilization levels. Also, unless the running servers do not have enough resources to host the VM, new servers will not be started, as it would additionally incur the startup cost. This keeps the number of running servers to the minimum. SC will be least if the CPU and memory intensive applications are equally mixed in the target server. Minimizing PC ensures that we minimize the ON time for the servers in the data center, reducing the total power cost. Minimizing DC ensures that we place the VM on a server that is less likely to be defragmented. Hence, we are able to minimize the number of migrations.

The cost parameters for the candidate servers is determined by the placement of existing VMs on the server. Hence, these parameters change with the activation or deactivation of any VM on any server. Hence, we recompute these parameters for each placement request.

C. Defrag Algorithm

On-demand service activation and deactivation of the virtual machines is characterized by their lease period. After the expiry of the lease period, the allocated resources become free. This leads to fragmentation of resources in the servers of the datacenter. To optimize the utilization of the servers, the datacenter thus has to be periodically "defragmented".

In the 'Defrag' procedure we first compute the defrag cost DC of the servers that are running at the current time T_D. Let us assume that M servers are operational at time T_D. DC for a server is computed as the sum of the migration cost of all running VMs on the server at time T_D. The servers are sorted in descending order of their DC. We then scan the sorted list and find the minimum number of servers S_{min} that have enough CPU and memory resources to host all the VMs active at time T_D. The first S_{min} servers are tagged as Receivers. We then pick the $M - S_{min}$ lowest utilization servers, tag them as Donors, and start defragmentation (with the lowest utilized server first).

The VMs on the candidate Donor server are then assumed to be removed. We place the VM using $SDAP$ with the candidate target servers restricted to the S_{min} highest utilization servers. If the Donor servers are not able to accommodate all the VMs due to fragmentation, the limit S_{min} is increased by 1 and the highest utilized Donor server is added to the Receiver set. Once all VMs from the Donor servers are placed on $Receiver$ server, the corresponding placement is selected. We then migrate the required number of VMs to achieve the selected placement.

We conduct experiments with real datacenter production traces, and report in later sections that the $SDAP$ algorithm help to reduce the migrations encountered during the 'Defrag' process. We also show that $SDAP$ also leads to reduced operational costs and increased efficiency of the datacenter.

V. Experiments

In this section we discuss the experiments performed to benchmark the performance of *Service Deactivation Aware Placement* (SDAP) and $Defrag$ procedure. We perform a trace-driven evaluation of the proposed algorithms. We first describe our experimental setup.

A. Experimental Setup

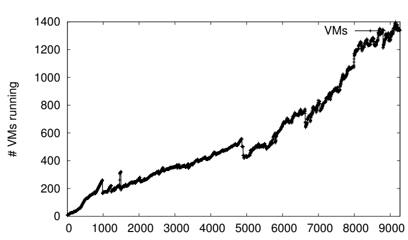

Fig. 4. Number of active VMs with Time

We use request traces from a live production cloud, IBM Research Compute Cloud ($RC2$). For each cloud request, the trace captures the resource demands, start time, end time and its lease period. The trace captures all the requests made to the cloud over a period of 1 year. We show all the active VMs with time in the cloud in Figure 4, depicting that the cloud is in a growing phase currently. However, there are intervals (e.g., $time > 8000$ in Fig. 4), when the number of VMs are mostly steady. We note that a growing cloud is an adversarial scenario and $SDAP$ would perform same or better when the cloud is in steady state.

Family	CPU Cores	Memory (GB)
HS20 Blade	8	24
IBM x3200	8	24
IBM x3400	12	36
HS21 Blade	16	48
IBM x3550	32	96
IBM x3950M2	64	192

TABLE I

Server Models used in the Cloud

We simulate a datacenter scenario with 6 different server models. We use servers across 3 generations to capture diversity. Further, we consider both rack and blade servers in our evaluation setup. The various server models and their configuration used in our evaluation is listed in Table I. Our experimental setting also consisted of various cost parameters, which simulate the actual costs incurred in the cloud. We place the requests from the trace and use $SDAP$ to place each VM on candidate servers in the cloud. We also invoke $Defrag$ periodically and use the parameter T_D to capture the defragmentation period. We list out the baseline setting in Table II.

Parameter	Base Value
Power cost per unit ($\$/KwH$)	0.12
Per CPU core cost ($\$/perhour$)	0.3
Per GB memory cost ($\$/perhour$)	0.2
Migration cost per core	0.5
Migration cost per GB	0.4
Defrag Period T_D	2hrs

TABLE II

Baseline Experimental Setting

To understand the relative performance of our algorithm, we also implemented $First - Fit$ algorithm. First-Fit is the most common algorithm used in public clouds and captures the state-of-the-art implementation techniques, hence we benchmark the proposed procedure against it. For the First-Fit procedure, the servers are sorted in order of their power wastage cost and the first server in the sorted list having enough free resources (both CPU and memory) to accommodate the request VM is selected as the host server. Further, the 'Defrag' procedure is run on both $SDAP$ and First-Fit to provide a fair comparison between the two.

Both the algorithms were implemented in Java and evaluated on an Intel Core 2 Duo processor with 3 GB memory. We also varied the parameters from the baseline setting to explore the performance of the algorithms under a wide variety of settings. We report the cost parameters over each defragmentation period T_D and report our findings.

B. Baseline Evaluation

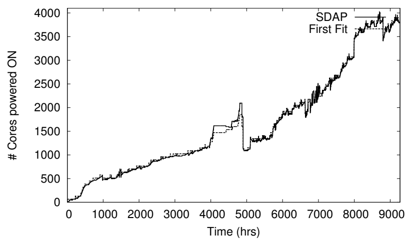

Fig. 5. Average number of cores powered on.

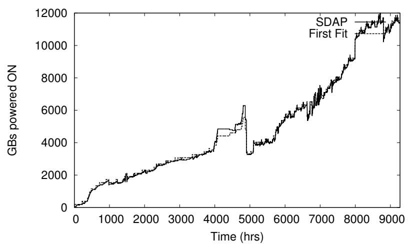

Fig. 6. Average amount of memory (in GB) used.

Figure 5 shows the average number of cores that are powered on per defrag period. The numbers take into account both the cores that are being used and also those that are idle. So, better the consolidation, lesser is the number of servers used and hence lower is the number of cores using idle power. $SDAP$ and $First-Fit$ roughly exhibits the same behaviour. Hence, defrag reduces the power wastage to minimum in both the procedures, eliminating fragmentation by VM deallocation due to lease period termination. The same scenario is observed in terms of memory usage, as shown in Figure 6.

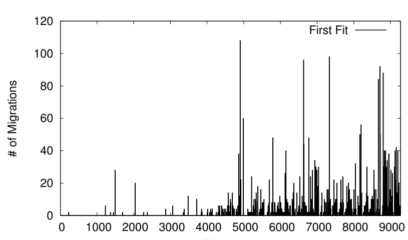

Fig. 7. Migrations encountered in first-fit procedure.

The salient feature of the proposed method is that it encounters lesser number of migrations during the defragmentation phase. This is due to the initial intelligent placement of the requests taking into account the various cost factors. These also include the service deactivation parameter. In Figure 8 we

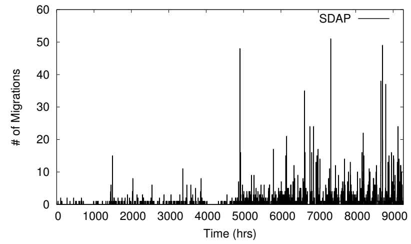

Fig. 8. Migrations encountered in SDAP.

observe that we indeed witness far lesser number of migrations as compared to the first-fit protocol, shown in Figure 7.

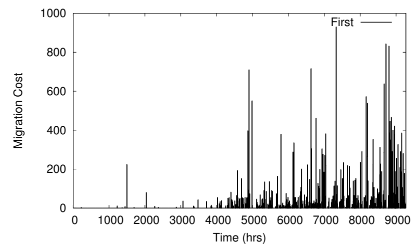

Fig. 9. The cost of migrations in first-fit procedure.

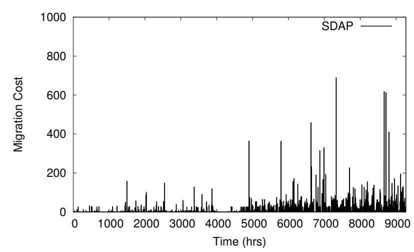

Fig. 10. The cost of migrations in SDAP.

We also observed that the migrations in $SDAP$ and $Defrag$ mostly took place from the servers which were underutilized. As the migration cost of a VM is also dependent on the current CPU utilization of the server, a low utilized server which experience a lower live migration cost compared to a heavily loaded one. Based on fewer number of migrations, smart initial placement and efficient choice of servers to defrag we incur lesser overall migration cost when compared to that of the first-fit procedure. This can be observed from Figures 9 and 10.

Figure 11 shows that we consume lesser average power cost compared to the competing algorithm. $SDAP$ uses lesser number of servers to allocate all the request VM, than the first-fit method. Hence, we observe a lesser operational cost of the datacenter.

C. Sensitivity Analysis

In order to experiment the robustness of our proposed algorithm, in this section we vary the parameters from the baseline

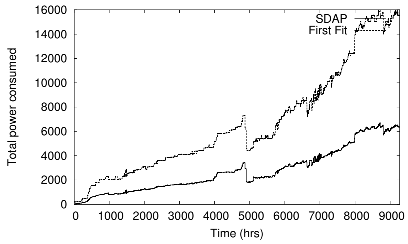

Fig. 11. Power cost incurred.

settings and compute the different comparison metrics.

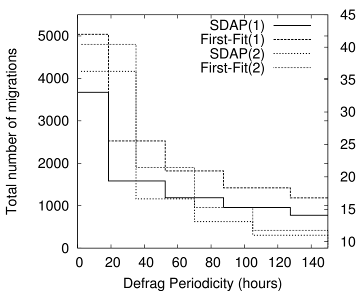

Fig. 12. (1) Number and (2) cost of migrations incurred.

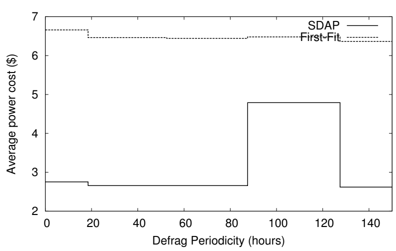

Fig. 13. Power cost incurred.

Consolidation or ‘defragmentation’ of a datacenter is costly in terms of computations, time as well as live migrations taking place. Further, this procedure needs to be fired periodically to keep the datacenter running efficiently. If ‘Defrag’ is performed too often, it may lead to a large number of migrations without a corresponding increase in efficiency. This is due to the fact that the datacenter scenario has not changed significantly from the time when defragmentation was last performed. On the other hand, if consolidation is far and few, mostly the datacenter will be running a large number of servers and will also be underutilized. This leads to inefficiency and surging operational cost.

Hence, we vary the periodicity of the ‘Defrag’ procedure to observe the trade-off between the two situations. Figure 12 shows the variation of the number of migrations and the migration costs incurred with the periodicity for both the procedures. Figure 13 shows the power drawn for the same experiment. We observe that the total migration cost reduces as the defragmentation period T_D is increased. However, the number of migrations stabilize at approximately 80 hours. Similarly, the power drawn by the data center may increase if defragmentation is infrequent. However, we observe that the power cost is stable until T_D is increased beyond 90 hours. Hence, for the above workload, a defrag periodicity of approximately 80 - 90 hours achieves the right trade-off between migration cost and power cost. We also note that $SDAP$ outperforms the first-fit algorithm for the entire range of the experiment.

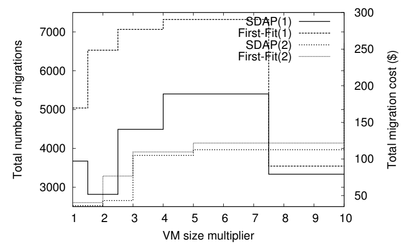

Fig. 14. (1) Number and (2) cost of migrations incurred.

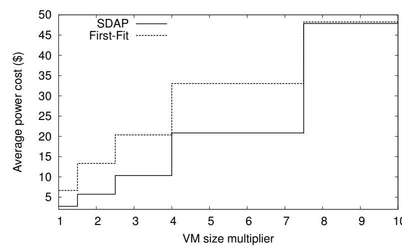

Fig. 15. Power cost incurred.

Next, we vary the request VM sizes and observe the migrations and power cost incurred. The results are reported in Figures 14 and 15. We observe that as the VM sizes increases the number of migrations increases. Since, for larger VMs the fragmentation due to deactivation is significant and migrations of other VM become necessary for running the datacenter efficiently. However, as the VM request sizes further increase, the number of migrations drop as now each server can accommodate only a single VM on average. So, even deactivation of a large VM may completely free up the server which can be powered down. Hence the migrations drop and become constant after that.

We also observe that the migration cost encountered exhibits similar behaviour, increasing upto a certain limit with increase in VM request sizes and then becomes constant. $SDAP$ is seen to observe lesser number of migrations and hence lower migration costs than the first-fit procedure.

As the VM resource request size increases, a server can accommodate lesser number of VMs. Hence to allocate all the VMs, higher number of servers should be running in the datacenter. This leads to an increase in the power cost of the datacenter, as shown in Figure 15. Again, $SDAP$ outperforms First-Fit for the entire range of the experiment.

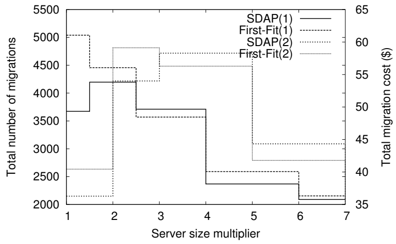

Fig. 16. (1) Number and (2) cost of migration experienced.

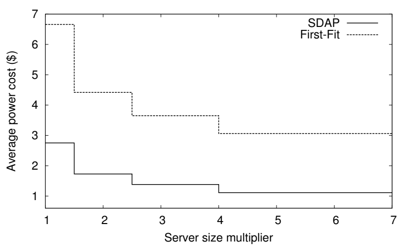

Fig. 17. Power cost incurred.

We now vary the capacity of the servers in the datacenter. As the size of the server increases, more VMs get packed into each server reducing the number of servers running in the datacenter. This leads to a decreased in the power consumption cost in the datacenter, as shown in Figure 17.

Figure 16 shows that the migration cost decreases with increase in server size. The number of migrations decreases with increase in server size because we get fewer servers to consolidate as the number of active servers are reduced. We again observe that $SDAP$ encounters lesser migrations, lower migration and power cost as compared to first-fit procedure.

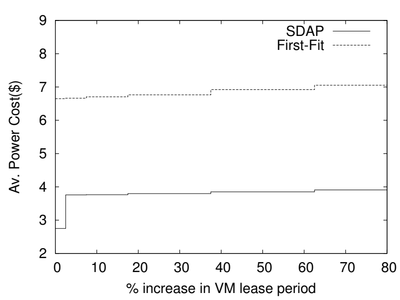

Fig. 18. Effect of lease time extension on power cost incurred.

We study the effect of lease period extension of the VMs on the total power cost in Fig. 18. Here, we extend the lease duration for 50% of the VMs selected uniformly and randomly. As $SDAP$ and $Defrag$ rely on the lease period of the VMs for the various cost factor affecting the VM placement, we observe an increase in the total power cost incurred by $SDAP$. However, the amount of extension of the lease duration does not seem to have any significant effect. We observe a similar behaviour with varying number of VMs extending their lease time. The more the number of such VMs, the larger the initial increase in the total power consumed. However, $SDAP$ still performs better than First-Fit.

VI. Related Work and Discussion

Automated provisioning and placement is a defining feature of clouds today. Placement techniques in publicly available cloud platforms focus on reducing fragmentation and use bin packing heuristics like first-fit or best-fit. Placement in virtualized data centers take a more rigorous approach to placement and use monitored data to come up with optimized sizing and placement [5]. Verma *et al.* [21] propose Peak Clustering-based Placement (PCP) that improves resource utilization by placing complementary workloads together. However, all these techniques implicitly assume that the workloads are permanent and do not take the finite lease period of the virtual machines into account.

Dynamic consolidation has been a popular technique to improve resource utilization of virtualized data centers. A dynamic consolidation approach monitors the resource usage of all the VMs, periodically re-sizes and places them on a minimal set of servers [20], [13], [15]. In order to transparently reconfigure the data center, these techniques use live VM resizing and live VM migration. Such an approach ensures that the servers in the data centers always run at high utilization and adapt to variation in workload demands.

Frequent adaptation of workloads in virtualized data centers has associated overheads. Some of the dynamic consolidation techniques model the impact of live migration on the performance of applications and minimize the number of migrations [20], [11]. Verma *et al.* model the impact of live migration and present recommendations to minimize the impact of live migration during consolidation [23].

Improving the power efficiency of data centers has been another popular area of research. Most dynamic consolidation techniques share the goal of reducing the power consumption in data centers. Ranganathan *et al.* present a technique using DVFS to budget an ensemble of blades, allowing the individual blades to operate at higher power efficiency [17]. The Brownmap system uses both DVFS and live migration to ensure that a data center operate within a power budget, while maximizing the performance of the applications [22]. However, none of the earlier work in cloud placement or consolidation take into account the lease periods, which can be leverage to minimize the impact of reconfiguration.

In this paper, we present the first characterization of cloud virtual machine requests. We propose the novel $SDAP$ algorithm to place applications in a public cloud setting. $SDAP$ takes into account service deactivation to come up with a placement that minimizes the number of migrations, while reducing the power cost of the data center. We also propose a low-cost defragmentation procedure that uses live migration to consolidate a cloud with the minimum number of migrations. We perform extensive experiments to evaluate the performance of the algorithms with state-of-the-art placement algorithms and establish their effectiveness.

REFERENCES

[1] D. Alvarez. Improving data center (server room) energy efficiency at Caltech. Technical report, SURF Project, 2008.

[2] M. Ambrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. Above the Clouds: A Berkeley View of Cloud Computing. UC Berkeley RAD Laboratory, 2009.

[3] G. Ammons and *et al*. Rc2 - a living lab for cloud computing. Technical report, IBM Research Technical Report, 2010.

[4] N. Bansal, A. Caprara, and M. Sviridenko. Improved approximation algorithms for multidimensional bin packing problems. In *IEEE Symp. on Foundations of Computer Science*, 2006.

[5] N. Bobroff, A. Kochut, and K. Beaty. Dynamic placement of virtual machines for managing sla violations. In *IEEE Conf. Integrated Network Management*, 2007.

[6] R. E. Brown. Optimum packing and depletion. *American Elsevier*, 1971.

[7] C. Clark, K. Fraser, S. Hand, J. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield. Live migration of virtual machines. In *Proc. Usenix NSDI*, 2005.

[8] E. G. Coffman, D. S. Johnson, P. W. Shor, and R. R. Weber. *Average Case Analysis Of On-Line Bin Packing Algorithms*. Michaelmas, 1996.

[9] D. Hilley. Cloud computing: A taxonomy of platform and infrastructure-level offerings. Technical report, Georgia Institute of Technology, 2009.

[10] A. Inc. Amazon elastic compute cloud. Technical report, Amazon Inc.

[11] G. Jung, K. Joshi, M. Hiltunen, R.Schlichting, and C. Pu. A cost-sensitive adaptation engine for server consolidation of multitier applications. In *Proc. Middleware*, 2009.

[12] J. M. Kaplan, W. Forrest, and N. Kindler. Revolutionizing data center energy efficiency. Technical report, McKinsey & Company, 2008.

[13] R. Nathuji and K. Schwan. Virtualpower: coordinated power management in virtualized enterprise systems. In *Proc. ACM SOSP*, 2007.

[14] M. Nelson, B.-H. Lim, and G. Hutchins. Fast transparent migration for virtual machines. In *Usenix ATC*, 2005.

[15] P. Padala, K.-Y. Hou, K. Shin, X. Zhu, M. Usyal, Z. Wang, S. Singhal, and A. Merchant. Adaptive control of multiple virtualized resources. In *Eurosys*, 2009.

[16] M. Paterson. Evaluation of nagios for real-time cloud virtual machine monitoring. Technical report, 2010.

[17] P. Ranganathan, P. Leech, D. Irwin, and J. Chase. Ensemble-level power management for dense blade servers. In *Proc. ISCA*, 2006.

[18] J. Rutkowski, M. Anderson, and M. Hazelton. *Empowering Energy Efficiency In Federal Data Centers: Data Center Consolidation Initiatives*, 2010.

[19] B. Sotomayor, K. Keahey, and I. Foster. Combining batch execution and leasing using virtual machines. In *Proc. of HPDC*, 2008.

[20] A. Verma, P. Ahuja, and A. Neogi. pmapper: Power and migration cost aware application placement in virtualized systems. In *Proc. Middleware*, 2008.

[21] A. Verma, G. Dasgupta, T. Nayak, P. De, and R. Kothari. Server workload analysis for power minimization using consolidation. In *Proc. Usenix ATC*, 2009.

[22] A. Verma, P. De, V. Mann, T. Nayak, A. Purohit, G. Dasgupta, and R. Kothari. Brownmap: Enforcing power budget in shared data centers. In *Proc. Middleware*, 2010.

[23] A. Verma, G. Kumar, and R. Koller. The cost of reconfiguration in a cloud. In *Middleware (Industrial Track)*, 2010.

[24] A. Verma, G. Kumar, R. Koller, and A. Sen. CosMig: Modelling the impact of reconfiguration in a cloud. In *ACM/IEEE MASCOTS*, 2011.