Virtual Machine Consolidation in the Wild

Akshat Verma
IBM Research - India

Juhi Bagrodia
IIT Guwahati

Vimmi Jaiswal
JIMS Delhi

ABSTRACT

Dynamic Virtual Machine (VM) consolidation dynamically adapts VM resource allocation to resource demands promising significant cost benefits for highly variable enterprise workloads. In this work, we analyze large enterprise workloads with the goal of understanding how effective are the VM consolidation variants in real world. We observe that burstiness in memory demand is much lower than the burstiness in CPU demand. Further, memory is the more constrained resource in virtualized servers, significantly reducing the potential gains due to dynamic consolidation. We study consolidation planning in four very large data centers and observe that the savings in facilities cost due to dynamic consolidation over static consolidation is not as large as estimated by past studies. Further, the savings over intelligent semi-static consolidation are surprisingly modest in most cases, putting a question mark over the applicability of dynamic consolidation in real world.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Design studies

General Terms

Performance, Experimentation

1. INTRODUCTION

Virtual Machine (VM) Consolidation is traditionally classified into (i) static consolidation (ii) semi-static consolidation and (iii) dynamic consolidation. Static consolidation is a simple application of virtualization to run multiple virtual machines on one physical server. The sizing and placement of virtual servers on physical servers is static and does not change over a period of time. Static consolidation drives up data center utilization by consolidating unutilized servers as virtual machines on a shared physical server. Each virtual machine is sized to the expected peak usage for its workload and virtual machines are placed on physical servers using simple bin-packing approaches.

Enterprise workloads often exhibit bursty behavior with the peak resource demand often higher than average demand by a factor of $five$. Static consolidation, which sizes workloads based on peak demand, leads to prolonged periods of low utilization in each virtual machine. Semi-static consolidation attempts to take advantage of medium to long term workload variations by periodically re-sizing workloads and relocating them on target physical servers [27]. Semi-static consolidation typically takes advantage of intra-week variations (weekends for some workload may exhibit resource demand) or intra-month variations (e.g., payroll workloads need peak resource demand on the first and last day of a month). Semi-static consolidation is performed by re-sizing and relocating workloads once a week or once a month.

Dynamic VM consolidation extend the idea of semi-static consolidation even further by consolidating workloads daily or multiple times on the same day (e.g., every 2 hours). Consolidations on such a frequent basis can not be performed using VM relocation due to downtime required for VM relocation. Instead, dynamic VM consolidation needs live VM resizing and live VM migration as pre-requisites in the data center. Workloads often exhibit burstiness at short time scales. Hence, dynamic VM consolidation and placement, which leverages short-term burstiness, has been a very popular area of research.

1.1 The Case for Dynamic VM Consolidation

The potential of dynamic VM consolidation to reduce infrastructure costs is fairly easy to establish. We looked at the CPU utilization of two physical servers (picked completely at random) from a production data center of one of the largest banks in the world (Fig. 1). Both servers had an average utilization of less than 5%. However, the peak utilization exceeds 50%. Static or semi-static consolidation needs to provision for the peak demand (50%) of individual workload. One can leverage dynamic consolidation optimally to provision only 5% CPU and provide additional resources on demand. This has the potential to reduce infrastructure costs by a factor of 10 over static consolidation. Dynamic VM consolidation has thus been proposed with focus on data center power minimization, network bandwidth minimization, storage performance and SLA management. The popularity of dynamic consolidation can be gauged from the fact that google scholar returns more than 200 papers that deal with dynamic VM placement. Similarly, VMWare's enterprise virtualization products come with generic consolidation tools like VMWare DRS and specific tools like DPM [12] that use dynamic consolidation for power minimization.

1.2 VM Consolidation in the Wild

Our research team has developed consolidation planning and runtime management tools to aid in static consolidation, semi-static consolidation [27] and dynamic VM consolidation [25, 28]. Our tools have been used in more than 30 server consolidation engagements over a period of $four$ years. During this period, we got a

Middleware '14, December 08 - 12 2014 Bordeaux, France
Copyright 2014 ACM 978-1-4503-2785-5/14/12 ...$15.00
http://dx.doi.org/10.1145/2663165.2663316.

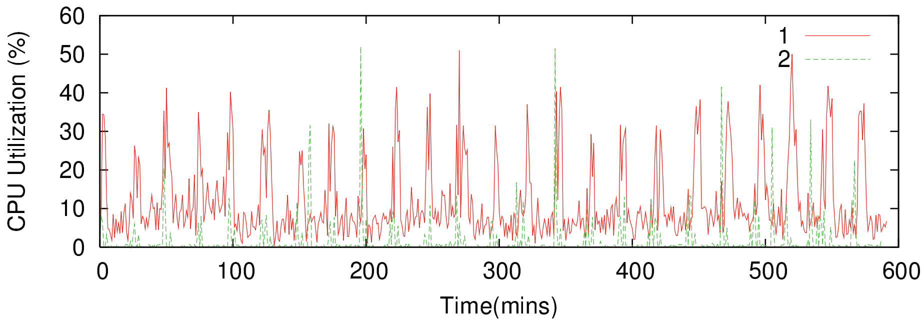


Figure 1: Burstiness in Server Workloads

chance to take a close look at resource usage of these data centers. Given the obvious potential of dynamic VM consolidation and deployment of automated consolidation tools (e.g., VMWare DPM [12]), we were surprised to discover that real data centers rarely employ dynamic VM consolidation. VM live migration is often employed for high availability and server maintenance but not for dynamic VM consolidation. Dynamic VM consolidation in a real data center has associated implications. Dynamic VM consolidation leverages live migration (e.g., VMWare DPM uses VMotion), which is a resource-intensive activity. Live migration consumes significant network and CPU resources, which vary with the memory load on the host system [29, 18]. Right sizing a VM may lead to high server load and live migration to alleviate the high load introduces additional resource contention. This may lead to prolonged or failed live migrations, which is unacceptable in production data centers. The uncertainty in the duration and impact of dynamic consolidation induced migration has impacted the adoption of dynamic consolidation in real data centers.

1.3 Our Contributions

We perform the first large-scale study of enterprise workloads from the perspective of VM consolidation. Even though we only report numbers for $four$ data centers ($>$ 3000 servers) spanning across different industries, our key observations hold true in general for all consolidation engagements, where our tool was deployed. We confirm earlier studies that CPU usage is highly bursty for enterprise workloads. However, we observe that variability in memory usage is much lower than variability in CPU usage. Further, we also observe that increasing VM density has led to memory becoming a more constrained resource than CPU. These two observations combined reduce the potential of dynamic VM consolidation to reduce infrastructure costs from $10X$ to a much modest $1.5X$. Finally, we show that enabling dynamic VM consolidation in a data center often requires allocating 20% or more resources for live migration to ensure the reliability of the consolidation process. We study the effectiveness of simple semi-static consolidation, stochastic semi-static consolidation and dynamic VM consolidation in reducing infrastructure costs for the $four$ selected data centers. We observe that dynamic VM consolidation reduces infrastructure costs over and above simple semi-static consolidation. However, intelligent semi-static consolidation (e.g., [27]) often performs as well as dynamic VM consolidation without introducing the risk associated with dynamic consolidation. We discuss potential improvements in live VM migration to improve the efficacy of dynamic VM consolidation.

1.4 Overview

The rest of this paper is structured in the following manner. In Section. 2, we provide a brief overview of VM consolidation, key concepts and constraint framework. We describe the workloads used in this work in Section. 3. Section 4 presents a trace analysis of the workloads to highlight some of the key workload attributes that dictate VM consolidation in the real world. We present an experimental comparison of various consolidation approaches in Section. 5. We present a study of the related work in Section 6 and discuss the implications of our study in Sec. 7.

2. VM CONSOLIDATION: BACKGROUND

In this section, we provide a brief overview of virtual machine consolidation and its various variants.

2.1 VM Consolidation Flow

Virtual machine consolidation consists of the following steps.

- *Monitoring*: VM consolidation starts with monitoring the resource consumption for all the virtual machines in a data center. Monitoring metrics typically include CPU usage metrics, memory usage metrics, paging metrics, and network metrics.

- *Prediction*: The monitored data captures the resource usage history of a virtual machine. The *Prediction* step uses the historical resource usage data and estimates the resource usage for the future. Prediction may be short-term or long-term in nature.

- *Size Estimation*: The third step in consolidation is estimating the resource demand for each virtual machine based on the *Prediction* step. The estimate is usually a scalar number for each resource under consideration. Since a demand estimate is made for a period with potentially multiple predicted data points (e.g., a 2 hour window with monitored data for every 5 min interval), a sizing function is used to convert multiple predicted values to a single demand value. The most common sizing function used is $\max$. Specific algorithms use other sizing functions like $90percentile$ [27].

- Placement: The placement step decided the actual placement of virtual machine on physical servers. There are many popular placement schemes dealing with energy minimization, space and infrastructure reduction, network bandwidth, migration cost reduction and different metrics. The placement step typically solves an optimization problem and the objective function used by different placement schemes may include different dimensions.

- Execution: The last step in consolidation is the execution of the recommended placement using VM resizing, VM relocation or VM live migration.

2.2 VM Consolidation Variants

VM consolidation can typically be classified into three variants.

2.2.1 Static VM Consolidation

Static VM consolidation is possibly the most popular variant of server consolidation in the real world. Static consolidation, as the name implies, is one-time consolidation of virtual servers on physical machines. This one time placement is done in such a way that there will not be resource contention on the physical host, where virtual machines are co-located. Static consolidation improves CPU and memory utilization by allowing multiple applications to share a server. The crux of static VM consolidation is a sizing step, which estimates the resource reservation for each virtual machine. Based on the resource demands, a bin-packing heuristic is applied to pack all the virtual machines on minimum number of physical servers, while ensuring that resource demand for all virtual machines are met. In this work, we use the maximum expected resource demand for sizing and First Fit Decreasing algorithm for bin packing [26] as representative of *Static* consolidation.

2.2.2 Semi-Static VM Consolidation

Static VM consolidation uses an estimate of the maximum resource demand over the lifetime of a workload. Semi-static consolidation allows higher resource utilization by allowing consolidation to be performed at coarse-grained intervals (e.g., once a month or once a week in the weekend). Semi-static consolidation allows downtime and leverages VM relocation to move virtual machines from one physical host to another. The key advantage of semi-static consolidation is that it allows less conservative resource demand estimation, taking into account a smaller time period. Vanilla Semi-static consolidation in this work refers to usage of maximum expected resource demand for sizing and First Fit Decreasing algorithm for bin packing.

Semi-static consolidation can also leverage stochastic properties of the workload. Consolidation engagements often analyse workloads and identify workloads with negative correlation. Ensuring that positively correlated workloads are not placed together allows more aggresive sizing (e.g., using average resource demand as opposed to max). Verma *et al.* present few stochastic semi-static algorithms in [27]. In this work, we use a variant of the PCP algorithm described in [27] as representative of *Stochastic* consolidation.

2.2.3 Dynamic VM consolidation

Dynamic VM consolidation is the most popular consolidation technique studied by researchers. The most important concept in dynamic consolidation is that of a consolidation interval. Consolidation actions are performed in the context of a consolidation interval, which can be very short (e.g., 2 hours).

DEFINITION 1. **Consolidation Interval***: Consolidation interval is defined as the time period of dynamic consolidation. The consolidation actions are repeated at the start of every consolidation interval, allowing fine-grained consolidation actions to be performed.*

Dynamic consolidation leverages VM live resizing and VM live migration to consolidate workloads at fine-grained intervals, without incurring application downtime. Multiple techniques have been proposed for dynamic consolidation that focus on reducing the power consumption as well as reduce other infrastructure costs like space. Fine-grained consolidation allows resources to be quickly moved from one server to another, allowing all applications to run from a much smaller server footprint as well. One of the key challenge in dynamic VM consolidation is the performance impact of live migration. Researchers have focused on taking the impact of migration into account during consolidation [15, 26]. In this work, we use a dynamic consolidation algorithm that captures the salient features of [26] and [15].

2.2.4 Consolidation constraints in Real World

Popular consolidation schemes in research have primarily looked at resource efficiency. Enterprise applications often have deployment constraints, which consolidation algorithms need to take into account. Constraints are broadly classified into inclusion and exclusion constraints. Inclusion constraints capture affinity between two entities. Affinity may include affinity between two virtual machines, affinity between a VM and a host, affinity between a VM and a subnet. These may require constraints that place two VMs on the same host/subnet/rack or pin a VM to a specific host/subnet/rack. In our work, we have extended popular consolidation algorithms to also support deployment constraints.

3. WORKLOADS USED

In this work, we select workloads from $four$ different industries to ensure that our findings are as general as possible. We next describe characteristics of each workload used.

3.1 Workload Characteristics and Collection

VM consolidation is performed based on resource usage and configuration data. Resource usage data was collected using a popular agent-based monitoring tool (details withheld for confidentiality). Each source server (physical or virtual) periodically collects system usage data and sends it to a central server. The central server acts as a data warehouse for the monitored data and maintains data with policies on retention and expiration. We get monitored data for consolidation planning from the data warehouse hosted by the central server.

Metric	Description
% Total Processor Time	Total Processor Time
% Priv	Percent time spent in System mode
% User	Percent time spent in User mode
Proc Queue Length	Processor Queue Length
Pages Per Sec	Pages In Per Second
Memory Committed	Memory Committed in Bytes (MB)
Memory Average	% of Memory Committed Used
DASD % Free	% time DAS Device is free
# Log Vol Red	
TCP/IP Conn	Number of TCP/IP Packets transferred
TCP/IP Conn v6	Number of IPv6 Packets transferred

Table 1: List of monitored metrics

The monitoring agent collects a wide variety of metrics every minute for each operating system instance in the data center. These metrics include CPU metrics, memory metrics as well as disk and network metrics. Table. 1 captures all metrics collected by the system. The data warehouse uses the monitored data to collect aggregates and stores the aggregate data at different granularity. In our work, we use hourly averages of the monitored data for the most recent 30 days. Using data for an entire month helps capture diurnal, weekly as well as monthly variations in resource usage. This data is analyzed to plan the consolidation for the data center. Enterprise data centers often use SAN for storage. Hence, CPU and memory are the only resources owned by a VM. Consolidation planning optimizes CPU and memory, while using network and disk throughput as constraints to identify hosts with sufficient link bandwidth.

3.2 Selected Workloads

Name	Industry	# of Servers	CPU Util (%)
A	Banking	816	5
B	Airlines	445	1
C	Natural Resources	1390	12
D	Beverage	722	6

Table 2: Workload Types

We now describe the $four$ workloads we select for this study. For this study, we only considered non-virtualized Windows servers from the $four$ data centers. Unix-based servers often employ aggressive filesystem page caching, which may over-estimate memory usage. Similarly, hypervisors may employ ballooning and/or memory deduplication, which may lead to error in monitored memory usage. Since we were working with an external monitoring framework that we could not modify, we chose to only use physical Windows servers, where monitored memory accurately reflects real memory demand, for this study. We filter out any servers for which monitoring data or the specifications of the server is not available in the data warehouse.

Our first workload A is from the production data center of a Fortune 100 Bank. Our second workload B is from one of the data centers of one of the largest Airlines in the world. Our third workload C is the primary data center of a Fortune 500 natural resources company engaged in mining and minerals. Our last workload D is one of the largest beverage company in the world. We loosely classify all applications as either web-based workloads or computational/batch processing jobs. For a given application, all servers hosting any component of the application is accordingly labeled as web-based or batch (for example, any database server that is part of a web-based application is also labeled as web workload server). Workload A has the highest fraction of web-based workload servers, followed by D, B and C. We study these workloads over the period between June, 2012 to November, 2012. A summary of the workloads is captured in Table. 2.

4. UNDERSTANDING ENTERPRISE WORKLOADS

In this section, we take a look at important aspects that influence VM consolidation in production data centers. The key driver for VM consolidation is burstiness in enterprise workloads. Hence, we first study the variance in CPU and Memory usage for enterprise workloads.

4.1 Study on variance in CPU and Memory

One of the prime motivation for server consolidation is low average server utilization. Dynamic consolidation can provide additional infrastructure savings for bursty workloads. Hence, we first study the variability in CPU usage for enterprise workloads. We use two different metrics to capture variability.

- *Peak Average Ratio*: The first metric we use is the ratio between the peak demand of a resource and the average demand for the resource. Static consolidation needs to size workloads based on peak, whereas dynamic consolidation may potentially size a workload based on average demand over a period of time. Hence, this ratio is an indicator of the performance of dynamic consolidation over and above static consolidation.
- *Coefficient of Variability (CoV)*: Our second metric is the traditional coefficient of variability (CoV), defined as the ratio of standard deviation to the mean of a distribution. A CoV of 1 or more indicates a heavy-tailed distribution and a CoV less than 1 indicates low burstiness.

Figure 2 captures the peak to average ratio of CPU demand for the 4 workloads studied in this work. Since dynamic consolidation can be performed with different consolidation periods, we estimate the CPU demand for consolidation periods of duration 1 hour, 2 hours and $4hours$. The peak demand and average demand over a period of a month is then computed and its cumulative distribution frequent (cdf) is plotted. We observe that the $Banking$ workload is very bursty with more than 50% of the workloads exhibiting a Peak-to-Average ratio of more than 5 for consolidation intervals of 1 and 2 hours. With 4 hour consolidation intervals, the Peak-to-Average ratio is more than 4 for 50% of the workloads. Further, more than 30%, 15% and 5% workloads exhibit a Peak-to-Average ratio greater than 10 for consolidation intervals of 1, 2 and 4 hour respectively. The $Airlines$ and $Natural$ $Resources$ workload exhibit relatively modest burstiness (more than 50% workloads having a ratio greater than 2) but still have significant potential savings with dynamic consolidation. The $Beverage$ workload is very similar to the $Banking$ workload with the impact due to consolidation intervals being less significant.

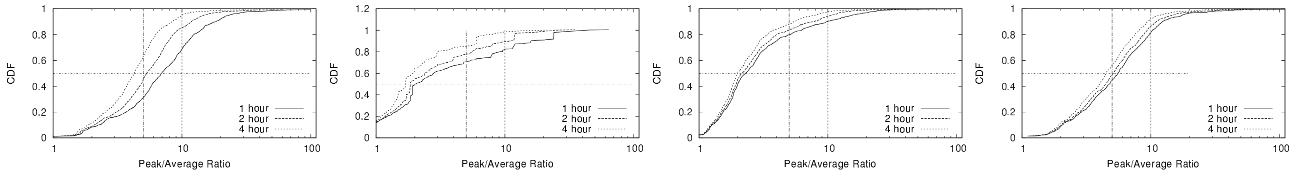
The study on Coefficient of Variability exhibits a similar trend. More than 50% of servers that constitute the $Banking$ workload are heavy-tailed ($CoV \geq 1$). Approximately 30% and 15% servers for $Airlines$ and $Natural\ Resources$ workload are heavy-tailed. The $Beverage$ workload exhibits a similar degree of burstiness to the $Banking$ workload. One may note that the $Banking$ workload has a much higher proportion of web-based workloads than other workloads. Since web workloads are known to be heavy-tailed [7], this reflects in higher degree of burstiness for the $Banking$ workload. The $Natural\ Resources$ workload has the highest fraction of custom batch applications, leading to lower burstiness.

Even though there has not been such a detailed and large-scale study on burstiness, these observations match earlier studies (e.g., [5, 11, 27]) at a high level. This study shows a higher degree of burstiness than our previous study [27] carried out five years ago (highest CoV of 4 as opposed to CoV of 10 seen in this study). We conjecture that workloads have become more bursty over the last 5 years due to a larger fraction of web-based applications (however, this can not be validated as the applications in the studies are different). We summarize our key findings below.

OBSERVATION 1. *CPU Utilization of servers vary greatly over time with Peak to Average Ratio of* 5 *and a CoV of* 1 *or more for more than* 25% *of servers studied.*

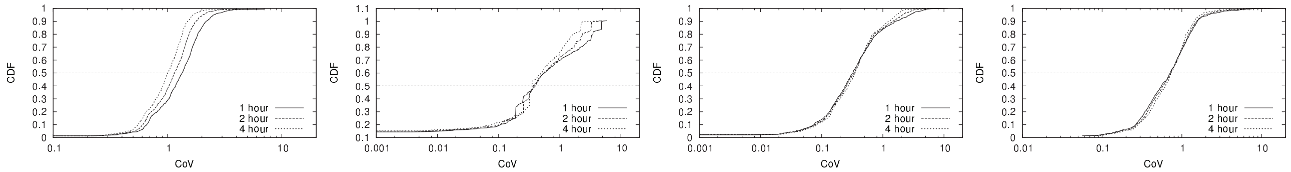
CPU has been often identified as the primary bottleneck resource for multi-tier applications [8]. Hence, consolidation approaches have primarily focused at CPU consumption. We next study the main memory demand for the selected workload. We again consider consolidation periods of 1, 2 and 4 hours. The peak and average demand is computed for each server. We plot the cumulative distribution function (CDF) of the peak-to-average ratio in Fig. 4. We also estimate the CoV of memory demand for each server and plot the CDF in Fig. 5.

The relative burstiness between the 4 workloads for memory demand exhibit similar trends to CPU demand. The $Banking$ work-



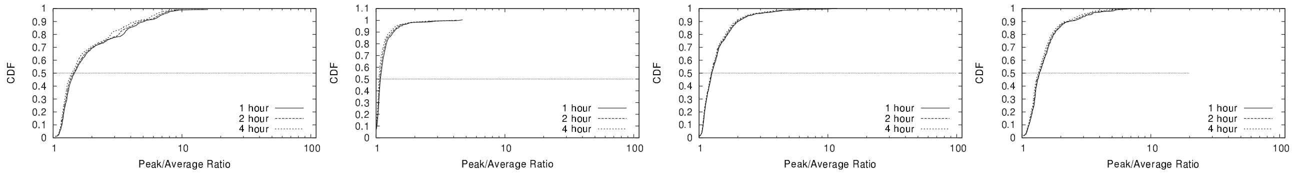
(a) (b) (c) (d)

Figure 2: CDF of Peak to Average Ratio for CPU. (a) Banking (b) Airlines (c) Natural Resources (d) Beverage



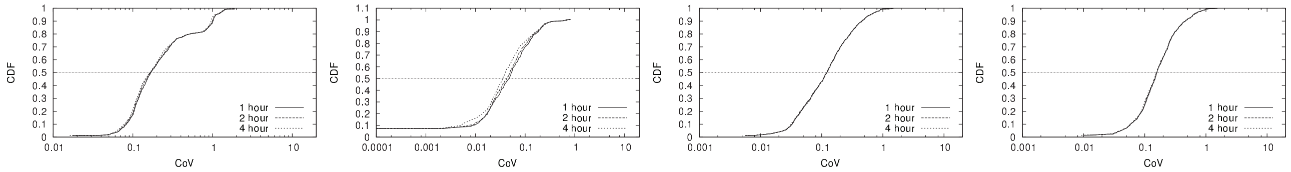
(a) (b) (c) (d)

Figure 3: CDF of Coefficient of Variability for CPU. (a) Banking (b) Airlines (c) Natural Resources (d) Beverage



(a) (b) (c) (d)

Figure 4: CDF of Peak to Average Ratio for Memory. (a) Banking (b) Airlines (c) Natural Resources (d) Beverage



(a) (b) (c) (d)

Figure 5: CDF of Coefficient of Variability (CoV) for Memory. (a) Banking (b) Airlines (c) Natural Resources (d) Beverage

load has higher Peak-to-Average ratio than other workloads. However, the absolute memory demand of the servers show some surprising trends. Firstly, the peak-to-average ratios are much smaller for memory than CPU. Even for the $Banking$ workload, there are hardly any servers with Peak-to-Average ratio greater than 10. More than half of the servers have a Peak-to-Average ratio of 1.5 or less (much lower compared to 5 for CPU demand). 90% and 60% of $Airlines$ and $Natural\ Resources$ workload have Peak-to-Average ratio less than 1.5.

The modest burstiness in memory demand is revalidated with the CoV study. There are about 20% servers in the Banking workload with a CoV greater than 1. For the $Airlines$ and the $Natural\ Resources$ workload, none of the servers show heavy-tailed memory demand as CoV is less than 1 for all servers. The $Beverage$ workload has few heavy-tailed servers but the overall proportion is less than 10%. Overall, we observe that memory demand exhibits an order of magnitude smaller burstiness than CPU demand. This observation implies that the potential of dynamic consolidation to save memory is a fairly modest 50% as compared to 500% for CPU. In hindsight, such a behavior is not surprising. We experimented with some web benchmarks and observed that marginal increase in memory with increase in throughput is much more modest than the increase in CPU. For example, we varied the throughput for Olio benchmark from 10 to 60 operations/sec on a Xeon dual core server and observed the variance in CPU and memory. We observed that for a $6X$ increase in application throughput, CPU demand increased from 0.18 core to 1.42 cores ($7.9X$ increase), whereas the memory demand only increased by $3X$.

OBSERVATION 2. *Memory demand of servers vary moderately over time with Peak to Average Ratio of* 1.5 *and a CoV of* 0.5 *or less for more than* 80% *of servers studied.*

4.2 Consolidation is constrained by CPU or Memory?

We have observed that the variability in CPU demand is significantly higher than memory demand. Most prior research has focused on CPU as a dominant resource for servers. We next investigate if the assumption holds true for workloads in a consolidated environment as well.

In this experiment, we consider all the servers that constitute a workload. For a given consolidation interval, we aggregate the CPU demand across all the servers to estimate a total CPU demand for the interval. We similarly compute the total memory demand across all the servers. The metric used for CPU demand is the IDEAS RPE2 Relative Server Performance Estimate v2 [22], one of the most popular benchmarks for server compute performance. Memory demand is reported in MBs. The ratio between the CPU and memory demand is computed as the resource ratio for the interval. Fig. 6 captures the CDF of the resource ratio across all consolidation intervals in the two week period for our four workloads.

We observe a wide disparity in the resource ratio across the workloads. The *Banking* workload exhibits the highest CPU intensity, whereas the *Airline* workload exhibits the highest memory intensity. We can rank the 4 workloads in the following order in terms of CPU intensity: (i) *Banking*, (ii) *Beverage*, (iii) *Natural Resources* and (iv) *Airlines*. In order to provide a perspective to what the actual numbers mean, we compare them against the ratio for the IBM HS23 Elite blade server with 2 processor and $128GB$ of RAM. HS23 Elite blades are especially designed for running virtual machines and have extended memory making it one of the blade servers with the highest memory/CPU ratio. We observe that *Banking* workload are memory-intensive for 30% of the time. The *Airline* and *Natural Resource* workload are bottleneck by memory for the entire duration of the experiment. The *Beverage* workload is dominated by memory for more than 90% of all consolidation intervals.

It may seem surprising that aggregate workloads are memory constrained even though individual workloads are CPU constrained. However, it is pertinent to note that CPU becomes a bottleneck resource at high load. At low and average loads, CPU demand reduces significantly, whereas memory reduction in memory demand is not that steep as we noted earlier. Aggregating demand across a large number of workloads leads to an average workload scenario, where memory demand starts to dominate CPU demand. We summarize our findings in the following observation.

OBSERVATION 3. *Data centers with server consolidation are constrained by memory more often than CPU (even after using extended memory blade servers).*

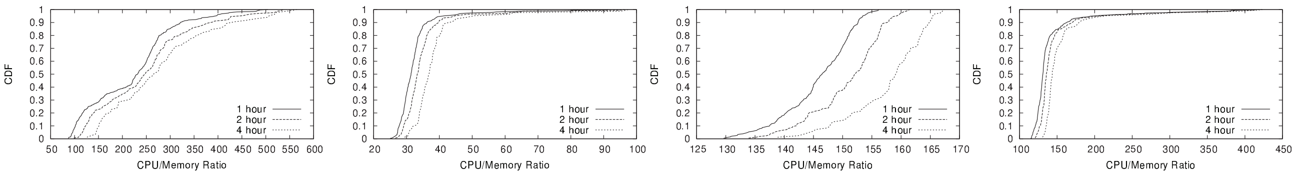
4.3 Impact of live migration on VM Consolidation

Dynamic VM consolidation leverages live VM resizing and live VM migration for fine-grained resource allocation. Live VM migration consists of a pre-copy phase, where the memory allocated to a virtual machine is transferred from the source physical server to the target physical server. All memory pages are marked as read-only and writes are flagged. All pages that were made dirty in a pre-copy round are copied again in the next round. The pre-copy completes when either a very small number of dirty pages remain or the number of dirty pages do not reduce between consecutive rounds. Even though specifics differ, all know live migration implementation [6, 18]. follow this overall design.

It is obvious that live migration requires CPU resources, memory resources and network resources. Further, since the entire active memory of a VM is copied, the resource requirement is significant. Clark *et al.* report a downtime of $210ms$, migration of 62 seconds and throughput drop of 10% for SpecWeb [6] during the pre-copy phase. Verma *et al.* show that significant resource contention can happen for high memory utilization as well as high CPU utilization [29]. The actual impact of migration depends on the precise workload, infrastructure (e.g., network bandwidth) [1], as well as any other co-located VMs [29]. Nelson *et al.* showed in 2005 that reserving 30% of a server for migration minimizes the pre-copy time [18]. VMWare's latest release (in 2013) still recommends leaving aside 30% of servers' resources for live migration [13].

We have experimented with live migration across hypervisor platforms in the past including IBM's pHyp [28], ESX 3.0 [25] and ESX 3.5 [29]. We had observed that one should not load the server beyond 75% host CPU utilization to ensure stable live migration without significant performance impact on the applications [29]. We repeated experiments with ESXi 4.1 to validate earlier studies. We observed that if the CPU utilization is below 80% and memory committed is below 85%, we can perform live migration reliably (with performance impact on the applications). In a real world with production workloads, the consequences of a prolonged live migration can be disastrous. Any application of dynamic VM consolidation thus requires bounds on the duration and performance impact of live migration. Since these bounds depend on the underlying infrastructure and the application profile, rules of thumb calculations are used to reserve resources for live migration. We use a thumb rule of reserving 20% resources for reliable live migration. This is lower than the 30% reservation that VMWare officially recommends but we believe is a pragmatic balance between performance risk and capacity utilization.

OBSERVATION 4. *In order to support dynamic consolidation,*



(a) (b) (c) (d)

Figure 6: Ratio of CPU to Memory usage. (a) Banking (b) Airlines (c) Natural Resources (d) Beverage. In comparison, the CPU to memory ratio for a high-end blade server is 160.

it is recommended to reserve at least 20% *of a physical server's resources for live migration.*

5. EXPERIMENTAL COMPARISON OF CONSOLIDATION APPROACHES

In this section, we report consolidation planning results for the workloads described in Section. 3.

5.1 Compared Algorithms

We compare the suitability of applying dynamic consolidation for the selected workloads. We run the following consolidation planning algorithms in this work.

- *Semi-Static Consolidation*: This is vanilla semi-static algorithm that uses peak expected resource demand for sizing and first-fit-decreasing for placement.
- *Stochastic Consolidation*: This is the consolidation algorithm inspired from the PCP algorithm in [27]. We use the following PCP parameters in the reported experiments : (i) Body of the distribution = 90 percentile (ii) Tail of the distribution = Max.
- *Dynamic Consolidation*: We use a state-of-the-art dynamic consolidation scheme that compares various adaptation actions possible and selects the one with least cost. The actual sizing function used in this case is the estimated peak demand in the consolidation window.

5.2 Experimental Settings

Metric	Value
Experiment Duration	14 days
Dynamic Consolidation Interval	2 hours
Number of Intervals	168
CPU reserved for VMotion	20%
Memory reserved for VMotion	20%

Table 3: Baseline Experimental Settings

It is not possible to use competing algorithms in a production environment as workloads can't be replayed. Further, we do not have the permission to test multiple consolidation options in a production environment. Hence, we use an emulator for this comparison. The emulator uses as input a set of resource usage traces for each physical server and returns consolidation statistics for the server. The input traces capture the resource demand from individual virtual machines on a server. The emulator is part of our consolidation tool suite and shares algorithms with our dynamic consolidation systems [25, 28]. The emulator captures the impact of virtualization overhead as well as memory savings due to deduplication in a configurable fashion. We have verified the accuracy of the emulator using two synthetic workloads $RuBIS$ and $daxpy$. For verification, we created a resource model for the workload (e.g., number of RuBIS clients versus CPU and memory used). We also implemented a micro-benchmark that can use either a specified amount of memory or consume a specific number of cores. Given the resource consumption in a trace, we run the workload at the appropriate intensity to consume at least one of the two resources. The other resource is then consumed using the micro benchmark. Hence, the workload and the micro benchmark together attempt to consume the same amount of CPU and memory as specified in the trace. We observed that the 99 percentile error bound of our emulator is 5% for RuBIS and 2% for $daxpy$ workload. For more details of the benchmarks and profiling methodology, please refer to [28].

We perform the consolidation planning using the emulator for a 14 day window. Semi-static consolidation variants assume that consolidation can be re-performed after the 14 days by taking a downtime and either doing VM relocation or live migration. Dynamic consolidation uses a consolidation period of 2 hours. This leads to a total of 168 consolidation intervals for dynamic consolidation. Average measures across the 168 intervals are reported for dynamic consolidation. We reserve 20% CPU and memory on all physical servers for reliable live migration. All the experimental parameters and their baseline values are reported in Table. 3

5.3 Performance Parameters

We compare the selected consolidation approaches in their ability to optimize the following parameters.

- Space and Hardware: The most important cost parameter in a data center is the cost of facilities and hardware. This cost is derived based on the number of servers and their specifications, the size of the racks and their occupancy, and the space cost of raised floor for the datacenter.
- Power Cost: Power cost is calculated based on the number of operational servers and their utilization is a given consolidation interval. For dynamic consolidation, average power across all consolidation intervals is reported.
- Server Utilization: We report the ability of a consolidation approach to improve the CPU utilization of servers in the data center.
- Resource Contention: Resource contention for a physical server captures the additional demand from virtual machines that can not be met within the server's capacity.

5.4 Baseline Results

We first report a comparison of space and power cost savings by the three consolidation approaches. For confidentiality reasons, we do not report absolute numbers and only report the cost normalized with respect to the cost of the *Vanilla* semi-static approach.

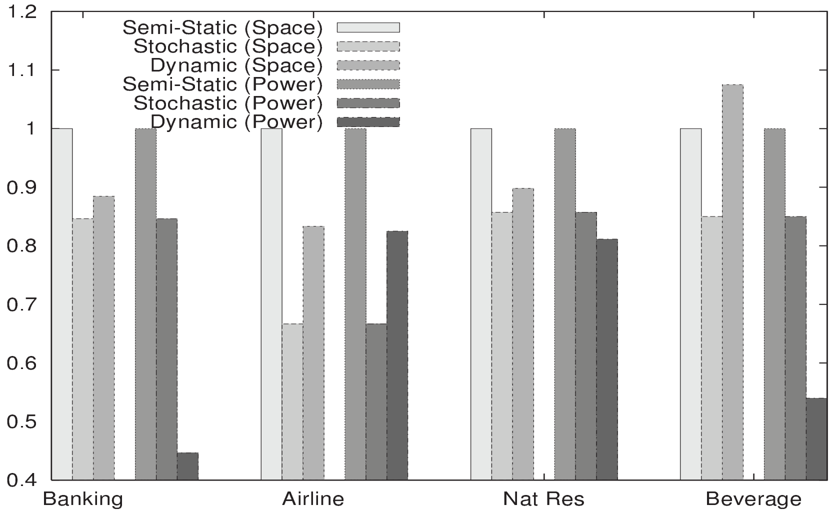


Figure 7: Infrastructure Cost Comparison

Our first significant observation is that *Stochastic* semi-static consolidation outperforms dynamic consolidation in terms of the space cost. It has been widely assumed that dynamic consolidation enables fine-grained elasticity, allowing workloads to be served from a smaller footprint by moving resource across the VMs on demand. Our experiments show that the assumption does not hold true in real world. Dynamic consolidation does outperform vanilla *semi-static approach* for 3 of the 4 workloads.

The findings, though surprising at first, are not hard to explain. The fact that dynamic consolidation should reserve resources for live migration has been largely ignored. This reservation ensures that dynamic consolidation starts with a handicap of about 20%. Further, as we have shown earlier, most workloads are memory-constrained and the burstiness in memory is an order of magnitude smaller than CPU. Finally, recent stochastic techniques improve the performance of *Semi-Static* consolidation by more than 15%. Prior dynamic consolidation work has primarily ignored stochastic consolidation. Hence, the gains due to fine-grained workload sizing and resource allocation in dynamic consolidation over and above stochastic consolidation is unable to compensate for the resources reserved for live migration.

We next look at the power cost for the various approaches. The overall numbers for power cost are more in line with past observations. For both *Banking* and *Beverage* workload, dynamic consolidation significantly outperform both vanilla *semi-static* approach as well as *stochastic* approach. For the *Banking* workload, dynamic consolidation reduces power by almost 50% over stochastic consolidation. As opposed to space cost, which is dictated by the largest number of servers provisioned across all consolidation intervals, power cost is computed independently for each consolidation interval based on the number of active servers in the interval. Hence, power can be saved with dynamic consolidation for intervals, where the aggregate resource demand is low.

Another significant observation is the muted power savings by dynamic consolidation for *Airline* and *Natural Resources* workload. This clear difference in behaviour across the 4 workloads can be explained by taking a closer look at the burstiness and resource ratio for the 4 workloads (Fig. 2, 3, 4, 5, 6). The *Banking* workload has the highest variability in CPU as well as memory. Further, it is more CPU intensive than other workloads (Fig. 6(a)), which allows consolidation to be driven by highly bursty CPU demand. The *Airline* workload is mostly constrained by memory (ratio of less than 50 compared with 160 for the HS23 server) and hence dynamic consolidation can only leverage the fairly modest burstiness in memory demand (Fig. 5(b)). The *Natural Resource* workload has high CPU burstiness and moderate memory burstiness. Further, more than 90% of consolidation intervals are memory constrained (CPU/memory ratio less than 160), leaving very few intervals for dynamic consolidation to be effective. The *Beverage* workload has more CPU-intensive consolidation intervals than all workloads other than the *Banking* workload, allowing it to save power in many more intervals. Further, its burstiness in memory is higher than *Airline* or *Natural Resource* allowing it to leverage fine-grained consolidation, even in periods with memory contention.

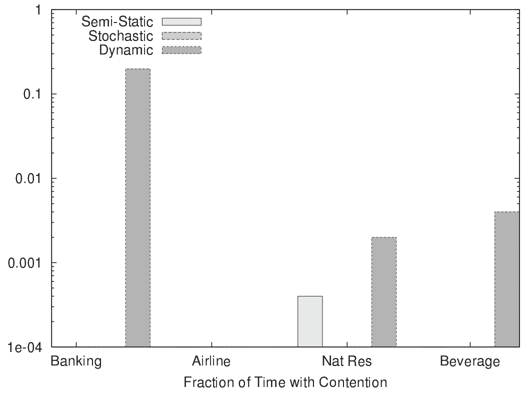
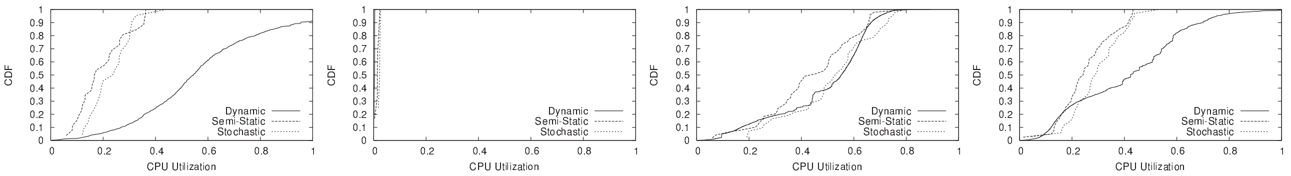


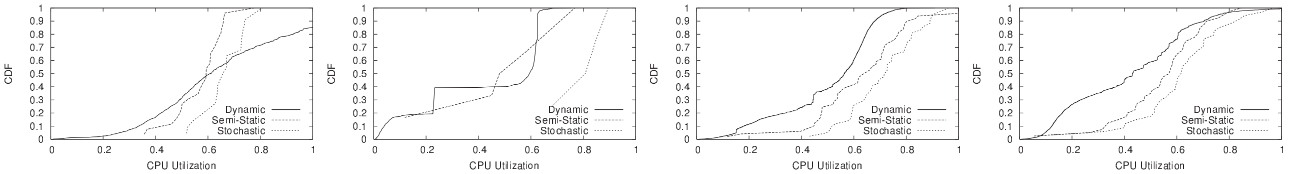
Figure 8: Fraction of time with contention. Absence of bar indicates zero contention.

We next study the resource contention due to consolidation on the virtualized servers. Fig. 8 captures the fraction of time (number of hours of the 336 hours) for which contention was experienced by a server. We observe that number of hours with resource contention is small for all workloads other than the *Banking* workload.



(a) (b) (c) (d)

Figure 10: CDF of Average Utilization. (a) Banking (b) Airlines (c) Natural Resources (d) Beverage



(a) (b) (c) (d)

Figure 11: CDF of Peak Utilization. (a) Banking (b) Airlines (c) Natural Resources (d) Beverage

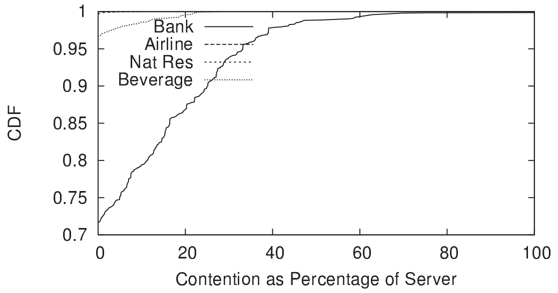


Figure 9: Distribution of CPU Contention. Absence of line for $Airline$ **indicates no contention.**

The *Banking* workload has spikes, which lead to contention with dynamic consolidation. The *Beverage* workload also has intervals, where dynamic consolidation leads to resource contention. The *Semi-static* consolidation scheme also has an isolated case of resource contention for the *Natural Resources* workload. Overall, we observe a strong correlation between the burstiness of a workload and the likelihood of resource contention with dynamic consolidation.

Fig. 9 captures the distribution of contention with the *Dynamic* consolidation scheme. We note that the highly bursty *Banking* workload can lead to very high contention. The contention is measured as the additional demand on the physical server as a fraction of the server's capacity. Even though the additional demand can not be measured, we estimate the demand based on the actual demand observed in monitoring data with standalone servers. The fact that the *Banking* workload has CPU as the dominating resource combined with the extremely high CoV leads to fairly serious CPU contention.

We next study the average CPU utilization achieved by VM consolidation. Fig. 10 captures the CDF of the average CPU utilization of each workload using the different consolidation approaches. Our first observation is the really low CPU utilization for the *Airlines* workload, which is a direct consequence of the high memory usage of the workload. Of the other three workloads, we observe that the vanilla *Semi-Static* and *Stochastic* techniques do not lead to any servers with high average utilization for *Banking* and *Beverage* workload. This is a result of the high variability of these two workloads, which prevents semi-static approaches from driving up the average server utilization high while avoiding resource contention. For the *Natural Resource* workload, all schemes lead to approximately the same distribution of the average utilization. The *Natural Resource* workload exhibits moderate variability and hence dynamic consolidation does not achieve any significant performance improvement over the other schemes. This was also reflected in the power cost for this workload, where all three techniques had similar power costs.

We also study the peak utilization achieved by the various workloads. *Peak Utilization* is correlated with resource contention. Workloads with peak utilization close to 1 are likely to exhibit resource contention. We observe this correlation in Fig. 11. We find that the workload with the highest resource contention (*Banking* with Dynamic consolidation) has the highest peak utilization (15% servers cross 100% CPU utilization). All other variants have very few servers with peak utilization close to zero.

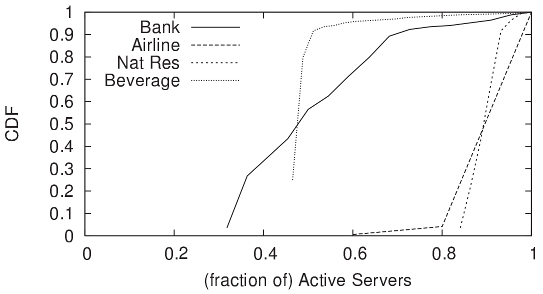


Figure 12: Distribution of Running Servers with Dynamic Consolidation

We have observed a trend that workloads with significant burstiness exhibit high dynamism. This reflects in power savings as well as the possiblity of resource contention. To validate this hypothesis, we also plot the distribution of the number of active servers with dynamic consolidation. We observe (Fig. 12) that both the *Banking* and *Beverage* workloads have a much wider distribution of servers. The *Banking* workload switches off upto 70% of deployed servers in some consolidation interval. The *Beverage* workload keeps only

50% of servers active for 90% of consolidation intervals. These observations validate our hypothesis that dynamic consolidation is relevant only for workloads with high burstiness.

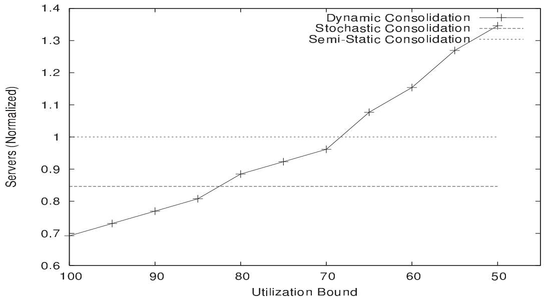


Figure 13: Performance of Banking workload

5.5 Sensitivity Analysis

Our experimental study shows that *Dynamic* consolidation is not useful in reducing space and hardware costs. Further, it only leads to power savings for highly variable workload. However, dynamic consolidation is significantly impacted by the amount of resources reserved for live migration. Since the required reservation varies with hypervisors and their implementation, it is important to vary the reservation. Hence, we conduced a sensitivity analysis of dynamic consolidation with change in the resources reserved for live migration. We next report the key findings of this sensitivitly analysis.

Fig. 13 captures the number of servers provisioned for the *Banking* workload with change in the utilization bound for the physical server. For a utilization bound of U, $1 - U$ fraction of all server resources are reserved for live migration. For example, our baseline setting had a utilization bound of 0.8 with 0.2 fraction of a server's CPU and memory reserved for live migration.

We observe that *Dynamic Consolidation* is very sensitive to the utilization bound. In fact, with a utilization bound of 0.85 (or 15% resource reservation for live migration), dynamic consolidation starts to outperform *Stochastic* consolidation. If no resources were reserved for live migration, *Dynamic Consolidation* can reduce the number of servers by 18%. On the other hand, if we need to reserve more resources for live migration, *Dynamic Consolidation* performs even worse than Vanilla *Semi-Static* consolidation.

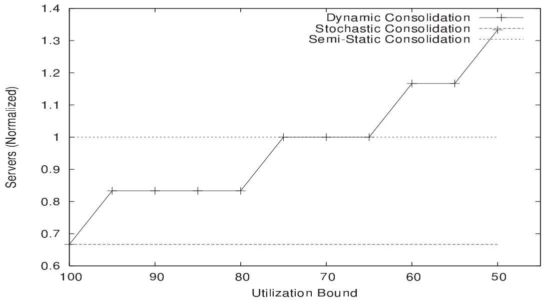


Figure 14: Performance of Airline workload

We observe a similar trend for the other three workloads as well. For the *Airline* workload, Dynamic consolidation achieves the same performance as *Stochastic* with a utilization bound of 100%. The *Natural Resource* workload is the best performing at utilization bound of 90% and achieves 17% improved performance compared to *Stochastic*, if 100% resources were available.

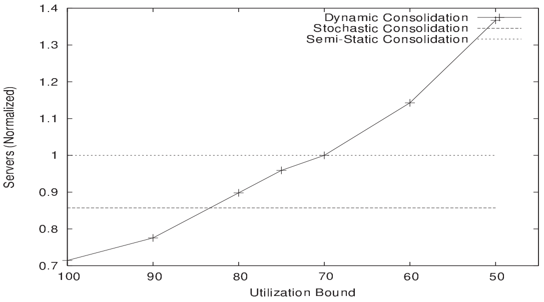


Figure 15: Performance of Natural Resource workload

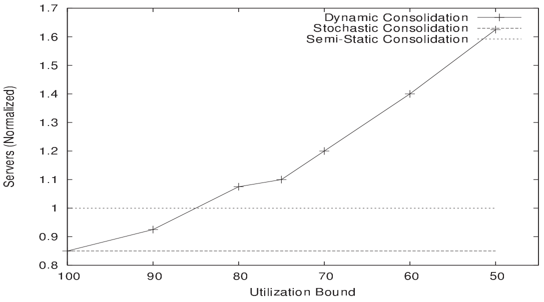


Figure 16: Performance of Beverage workload

Our experimental study can be summarized by the following key observations.

OBSERVATION 5. *Dynamic consolidation does not lead to space and hardware savings over intelligent semi-static consolidation for many workloads. We believe that one of the primary reason that semi-static consolidation performs well is because correlation between workloads is stable over time [27].*

OBSERVATION 6. *Dynamic consolidation leads to power savings for workloads that exhibit high burstiness. However, these savings may be associated with resource contention.*

OBSERVATION 7. *If the resources reserved for live migration can be reduced without impacting the reliability of migration, then dynamic consolidation can achieve space and hardware savings as well.*

6. RELATED WORK

The related work in this area can broadly be classified into three categories.

6.1 VM Consolidation

Server consolidation has been a popular area of research for the past ten years. Server consolidation has been driven by the goals of power minimization, space minimization and managing SLA violations. Most of the research in this area deals with dynamic VM sizing and placement. In [20], the authors propose a way to assign the right amount of CPU to a set of virtual machines running the different components of an application. They extend their work to deal with multiple resources in [19]. The placement problem has been more widely investigated where a system provisions the appropriate amount of resource to virtual machines that host the application components and place them on appropriate hosts [4, 25, 26, 17, 10, 15, 28, 30]. Variations of the core consolidation scheme includes factoring adaptation cost [15], power budget [28], and hardware cache usage [26]. Recently, VM placement also been extended to include network costs [16] in a static setting but the core technique can be easily extended to dynamic VM placement. [3] propose a technique to explicitly handle time-varying network loads.

In this work, we do not propose any new VM placement algorithm. On the other hand, we take a careful look at the problem of VM consolidation in real world and identify the constraints under which VM consolidation operates.

6.2 Workload Characterization

Our work fits more in the workload modeling and characterization space. Classical workload modeling has focused on aggregate workload characterization. Aggregate workload characterization of a web server by Iyenger *et al.* [14] and workload models of a large scale server farm by Bent *et al* [2] fall in this category. Recently, hybrid provisioning using both static and dynamic provisioning has also been proposed in [9] to combine predictive and reactive approaches. Individual server utilization, which is closer to our study, has been studied in [5, 11, 4, 27]. In [5], Bohrer *et al* use peak-trough analysis of commercial web servers to establish that the average CPU utilization for typical web servers is fairly low. Similar observations on the peak-trough nature of enterprise workloads have been made in [11]. In [4], Bobroff *et al* perform trace analysis on commercial web servers and outline a method to identify the servers that are good candidates for dynamic placement. Verma *et al.* present a study on the CoV of CPU demand and correlation between workloads [27].

However, all these workload studies focus only on CPU whereas we show that memory is a more constrained resource in virtualized data centers. Further, we make the choice of consolidation at a more coarse level (e.g., data center or cluster) instead of individual server, which Bobroff *et al.* do. Finally, we are not aware of any study that compares all variants of consolidation or performs any evaluation at the data center scale that we do.

6.3 Performance Models for Virtualized Servers

Our work relies heavily on the prior work, which models the performance impact of virtualization and live migration on enterprise workload. In [24], the authors present a profile-driven modeling for multi-component services and use it to estimate the application performance. Singh *et al.* [23] propose a mix-aware dynamic provisioning scheme for non-stationary workloads using k-means clustering. One of the first studies on the performance impact of live migration are [18, 6], which report downtimes of $100ms$, migration duration in the order of a minute and throughput impact upto 10%. Jung *et al.* [15] report impact on the response time for live migration. The impact of live migration has also been well studied by [1, 29]. Verma *et al.* show that more than 25% of all VMs may need to be live migrated in each consolidation interval. Our work leverages these studies to estimate the amount of resources that need to be reserved for the live migration process in dynamic consolidation.

7. DISCUSSION

In this work, we presented a detailed study of VM consolidation, while factoring in real-life constraints experienced in data centers. This work captures 4 representative data centers but the overall findings are consistent with our experience in more than 30 consolidation engagements. We observe that memory demand is far less bursty than CPU demand and virtualization has led to memory being the more constrained resource on physical servers. As a consequence, we found that the benefit of fine-grained resource allocation in dynamic consolidation is far more modest than previously believed. We found no significant savings in space and hardware due to hardware consolidation and power savings upto 50% for workloads with high burstiness and less memory contention. However, these savings were also associated with a higher risk of SLA violations due to a combination of high burstiness and aggressive fine-grained consolidation. Dynamic consolidation, in summary, has mixed benefits in today's data center. Intelligent semi-static consolidation was proposed as an alternative to dynamic consolidation for data centers that can not support live migration due to technical reasons (e.g., direct attached storage, expensive license costs) [27]. However, the observations proposed to motivate stochastic consolidation (stability of aggregate statistics and pair-wise correlation [27]) combined with the high overhead of dynamic consolidation make stochastic consolidation an equally, if not more, appealing proposition for data centers.

We did observe that the performance of dynamic consolidation is very sensitive to the amount of memory reserved for live migration. We believe further research is needed to improve the performance of dynamic consolidation and practically realize its benefits.

Enabling Shorter Consolidation Intervals In this work, we have consider dynamic consolidation interval of 2 hours. This is a practical number based on the time taken by live migration today as well as the network speeds in data centers built over the past few years. Improvements in network bandwidth as well as advances in live migration implementation can allow shorter dynamic consolidation intervals to become practical. This will enable more fine-grained consolidation, reducing the overall hardware footprint as well as providing more opportunities for saving power.

Improving live migration efficiency One of the key insights of our study is the significant amount of resources needed for live migration. Further, these resources are needed on a source server, which is likely overloaded and migrating VMs out of it. In order to ensure the reliability of migration, we need to reserve resources for live migration, effectively preventing server resources to be effectively used. We believe further research is needed on the efficiency of live migration. In particular, one may note that most activities required for live migration are performed on the source host. Further, significant memory copying is performed from the source server to the target. Offloading some of this work to the target server ((e.g., the copying process) can improve the efficiency of live migration. Further, if some work can be offloaded outside the operating system, it may reduce the burden on the server (e.g., use of RDMA [21] for the memory copy). There are obvious challenges in these approaches but have the potential to improve the effectiveness of dynamic consolidation as well as other system management activities that leverage live migration.

8. CONCLUSION

Our work attempts to compare the relative performance of various variants of VM consolidation. We conclude that all variants of VM consolidation are useful for specific workloads. Highly bursty and predictable workloads with high CPU contention can benefit from dynamic consolidation. However, there are many workloads with high memory contention and we recommend semi-static consolidation for such workloads. Semi-static consolidation avoids live migration and associated performance issues making it suitable for critical applications. Our work also establishes the need of a comprehensive consolidation planning analysis prior to VM consolidation in the wild.

9. REFERENCES

[1] S. Akoush, R. Sohan, A. Rice, A. Moore, and A. Hopper. Predicting the performance of virtual machine migration. In *IEEE MASCOTS*, 2010.

[2] L. Bent, M. Rabinovich, G.Voelker, and Z. Xiao. Characterization of a large web site population with implications for content delivery. In *WWW*, 2004.

[3] O. Biran, A. Corradi, M. Fanelli, L. Foschini, A. Nus, D. Raz, and E. Silvera. A stable network-aware vm placement for cloud systems. In *CCGRID*, 2012.

[4] N. Bobroff, A. Kochut, and K. Beaty. Dynamic placement of virtual machines for managing sla violations. In *IM*, 2007.

[5] P. Bohrer, E. Elnozahy, T. Keller, M. Kistler, C. Lefurgy, C. McDowell, and R. Rajamony. The case for power management in web servers. In *Power aware computing*, 2002.

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

[7] M.E. Crovella, M.S. Taqqu, and A. Bestavros. Heavy-tailed probability distributions in the world wide web. In *A Practical Guide to Heavy Tails*, pages 3–26. Chapman & Hall, 1998.

[8] C. Amza *et al.* Bottleneck characterization of dynamic web site benchmarks. In *Rice University Computer Science Technical Report TR02-388*.

[9] A. Gandhi, Y. Chen, D. Gmach, M. Arlitt, and M. Marwah. Minimizing data center sla violations and power consumption via hybrid resource provisioning. In *IGCC*, 2011.

[10] D. Gmach, J. Rolia, L. Cherkasova, G. Belrose, T. Turicchi, and A. Kemper. An integrated approach to resource pool management: Policies, efficiency and quality metrics. In *Proc. DSN*, 2008.

[11] D. Gmach, J. Rolia, L. Cherkasova, and A. Kemper. Workload analysis and demand prediction of enterprise data center applications. In *IISWC*, 2007.

[12] VMWare Inc. VmwareÍ distributed power management concepts and use. In *VMWare White Paper. http://www.vmware.com/files/pdf/Distributed-Power-Management-vSphere.pdf*.

[13] VMWare Inc. Vmware vsphereÍ vmotionÍ architecture, performance and best practices in vmware vsphereÍ 5. In *VMWare Technical White Paper. http://www.vmware.com/files/pdf/vmotion-perf-vsphere5.pdf*, 2013.

[14] A. Iyengar, M. Squillante, and L. Zhang. Analysis and characterization of large-scale web server access patterns and performance. In *WWW*, 1999.

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

[16] X. Meng, V. Pappas, and L. Zhang. Improving the scalability of data center networks with trafÞc-aware virtual machine placement. In *Infocom*, 2010.

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

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

[19] 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.

[20] P. Padala, K. Shin, X. Zhu, M. Usyal, Z. Wang, S. Singhal, A. Merchant, and K. Salem. Adaptive control of virtualized resources in utility computing environments. In *Eurosys*, 2007.

[21] Remote direct memory access. http://en.wikipedia.org/wiki/Remote_direct_memory_access.

[22] About IDEAS Relative Performance Estimate 2 (RPE2). http://www.ideasinternational.com/performance/.

[23] R. Singh, U. Sharma, E. Cecchet, and P. Shenoy. Autonomic mix-aware provisioning for non-stationary data center workloads. In *ICAC*, 2010.

[24] C. Stewart and K. Shen. Performance modeling and system management for multi-component online services. In *Usenix NSDI*, 2005.

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

[26] A. Verma, P. Ahuja, and A. Neogi. Power-aware dynamic placement of hpc applications. In *Proc. of ACM ICS*, 2008.

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

[28] 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.

[29] A. Verma, G. Kumar, R. Koller, and A. Sen. Cosmig: Modeling the impact of reconfiguration in a cloud. In *IEEE MASCOTS*, 2011.

[30] M. Wang, X. Meng, and L. Zhang. Consolidating virtual machines with dynamic bandwidth demand in data centers. In *Infocom*, 2011.