Chargeback Model for Resource Pools in the Cloud

Daniel Gmach, Jerry Rolia*, Ludmila Cherkasova
HP Labs
Palo Alto, CA, USA; *Bristol, UK
e-mail: {firstname.lastname}@hp.com

***Abstract*—This paper presents three methods for apportioning server costs among workloads in shared resource environments such as computing clouds. We consider a fine sharing of resources, the impact of time varying resource usage, large ratios for peak to mean workload demands, and the influence of random choices for the co-placement of workloads on shared servers. These features can affect the quantity of servers needed to support workloads as well as the robustness of the cost values assigned to each workload. We compare the three methods for apportioning costs and recommend the method that assigns costs in the most repeatable manner. A detailed study involving 312 workloads from an HP customer environment demonstrates the result.**

Keywords-component; Cloud Computing, Cost models, Shared Resources, Virtualization

I. INTRODUCTION

Virtualization technologies are maturing and the computing capacity of individual servers is increasing rapidly. As a result more and more computing can be conducted in shared resource pools such as private and public clouds. A new hot topic in cloud computing and the virtualized world is how to account for the shared infrastructure usage and how to chargeback services running on top of the underlying physical infrastructure. In the recent past, before the virtualization era, the accounting model was relatively simple and straightforward: the server hardware, its power usage, and software costs were directly associated with the deployed application using these resources, while the storage and networking costs were typically apportioned on a usage basis. When multiple virtual machines are deployed to a shared resource pool, the virtual machines may migrate between hosts periodically or frequently. The virtual machines may have different resource requirements yet the hosts have bounded capacity so the number of virtual machines that can be assigned to a host while satisfying service level requirements depends on the time varying workloads of the applications in the virtual machines. The question is "who is responsible for the incurred costs" for cost recovery? The focus of this paper is on the notion of cost recovery or charge back for shared resource pools, as opposed to pricing or what customers are willing to pay for resources.

A common sense approach for costing is to extend the usage-based model, i.e., from virtualization layer monitoring information one can derive average resource usage per application for a costing interval, e.g., three weeks, and then the physical server cost can be split up respectively. Currently, many service providers employ such simplified usage–based accounting models [1─4]. However, the relationship between application workloads and costs is actually more complicated.

Some workloads may have a large peak to mean ratio for demands upon server resources. We refer to such workloads as *bursty*. For example, a workload may have a peak CPU demand of 5 CPU cores but a mean demand of 0.5 of a CPU core. Such ratios may have an impact on shared resource pools. A pool that aims to consistently satisfy the demands of bursty workloads will have to limit the number of workloads assigned to each server. This affects the number of servers needed for a resource pool. Thus, burstiness affects costs. Some service providers limit the effect of burstiness by capping virtual machine sizes, thereby forcing application owners to acquire more virtual machines for busy periods.

Further, server resources are rarely fully utilized. Even though many services can be assigned to a server some portion of the resources will remain unused over time. The amount of unused resources may depend on, from the perspective of workloads, random workload placement/consolidation choices. The costs of unallocated resources must also be apportioned across workloads.

In this paper, we discuss these issues, present three methods for apportioning server costs among workloads that share servers in such environments, and demonstrate the implications on reported costs for the different methods in a detailed case study with 312 workloads from an HP customer environment. Two of the methods are new. Our main contribution is a method that takes into account burstiness and is tolerant of alternative workload placement decisions. The method provides for repeatable cost estimates for workloads within a shared resource pool.

This paper is organized as follows. Section II formally introduces the notion of costs and three models for apportioning costs. Section III presents a workload characterization for a server consolidation exercise considered in the paper. Section IV presents a performance case that compares the apportioning methods. Section V offers summary and concluding remarks along with a description of our next steps.

II. COSTS AND APPORTIOINING COSTS

The total costs of a resource pool include the acquisition costs for facilities, physical IT equipment and software, power costs for operating the physical machines and facilities, and administration costs. Acquisition costs are often considered with respect to a three year time horizon and reclaimed according to an assumed rate for each costing interval. We assume that workload placement remains the same within each

costing interval. Without loss of generality, this paper focuses on server and virtualization software licensing costs only.

Below, we define three categories of resource usage that can be tracked separately for each server resource, e.g., CPU, memory, for each costing interval. To simplify the notation, the equations we present only consider one server resource at a time, e.g., CPU or memory for one costing interval. Then the corresponding costs over all resources are summed up to give the total cost for all server resources for each costing interval. The final cost for an application workload is the sum of its costs over all costing intervals. The three categories of resource usage are:

- **Direct resource consumption by a workload**: $d_{s,w,}$ the average physical server utilization of a server s by a workload w. $d_{s,w}$ in *[0,100]*. $d_{s,w}$ is *0* if a workload w does not use a server s.
- **Burstiness for a workload and for a server**: $b_{s,w}$, the difference between peak utilization of a server s by workload w and $d_{s,w}$. $b_{s,w}$ in *[0,100]*. Additionally, b_s, the difference between the peak utilization of a server s and its average utilization. b_s in *[0,100]*.
- **Unallocated resource for a server**: the difference between 100 and the peak utilization for a server s, a_s in *[0,100]*. *a* refers to unallocated resource.

Next, we present 3 different methods for apportioning cost. We refer to these as server-usage, server-burst, and pool-burst.

First we consider a *server-usage* approach that considers only the direct resource consumption by W workloads. Suppose a server s has a cost C_S that represents the costs associated with the server. We define a workload's cost share based on its server-usage as $\prod^{\text{server-usage}}_{s,w}$:

$$\prod_{s,w}^{server-usage} = C_S \frac{d_{s,w}}{\sum_{w'=1}^{W} d_{s,w'}} \quad (1)$$

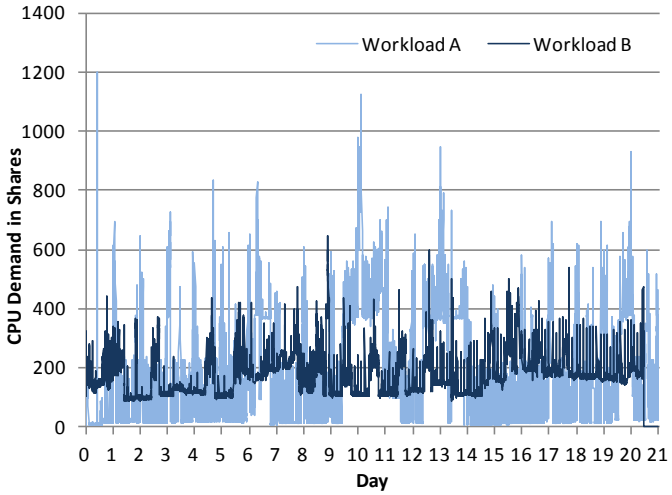

Figure 1: CPU demand of workloads

The *server-usage* approach does not take into account the impact of workload burstiness on costs. Figure 1 motivates our concern regarding burstiness. It shows the CPU demands for two workloads for three weeks. In the figure, 100 shares correspond to one 1 GHz CPU. Both workloads exhibit a similar average CPU demand: 161.8 CPU shares for Workload A versus 170.3 for Workload B. Using Eq. (1), for a certain consolidation scenario c that considers many workloads for a three week consolidation interval, the CPU costs for hosting Workload A are $36.27 whereas Workload B has costs of $39.12. This clearly does not reflect the real hosting costs for these two workloads. Workload A has much higher variability and much higher peaks than Workload B, 1200 CPU shares compared to 645 CPU shares. The burstiness of Workload A actually causes less dense workload placements and thus lower average server utilization and the need for more servers.

To take into account burstiness and unallocated resources we partition C_s based on utilization to get C^d_s, C^b_s, C^a_s, respectively, where C^d_s corresponds to costs associated with the average utilization of the server resource; C^b_s corresponds to the burstiness b_s of the server resource; and C^a_s corresponds to the unallocated server resource, i.e., the difference between 100% and the peak utilization.

For the *server-burst* approach, we divide the burst portion of costs for a server in a manner that is weighted by the burstiness of each workload on the server. In a second step, the server's unallocated resources are apportioned based on the bursty costs. Server-burst $\prod^{server-burst}_{s,w}$ is defined as:

$$\prod_{s,w}^{server-burst-temp} = C_s^d \frac{d_{s,w}}{\sum_{w'=1}^{W} d_{s,w'}} + C_s^b \frac{\varepsilon + b_{s,w}}{\sum_{w'=1}^{W} (\varepsilon + b_{sw'})} \quad (2)$$

$$\prod_{s,w}^{server-burst} = \prod_{s,w}^{server-burst-temp} + C_s^a \frac{\prod_{s,w}^{server-burst-temp}}{\sum_{w'=1}^{W} \prod_{s,w'}^{server-burst-temp}}$$

The ε value, a small value, in the numerator and denominator of the 2nd term of the first equation ensures that the denominator does not evaluate to zero for cases where there is no difference between peak and mean resource usage. Using Eq. (2) with consolidation scenario c, the total CPU costs are $54.7 for Workload A and $21.7 for Workload B. The difference stems from the fact that Workload A is much burstier than Workload B.

We note that dividing costs in this way can still lead to a lack of robustness for workload costs. The costs are sensitive to the placement of workloads on servers. To provide for a more robust cost estimate, the following *pool-burst* approach for apportioning costs partitions burstiness and unallocated resources using measures for the S servers in a resource pool instead of for individual servers.

$$\prod_{s,w}^{pool-burst-temp} = C_s^d \frac{d_{s,w}}{\sum_{w'=1}^{W} d_{s,w'}} + \left(\sum_{s'=1}^{S} C_{s'}^b\right) \frac{\varepsilon + b_{s,w}}{\sum_{s'=1,w'=1}^{S,W} (\varepsilon + b_{s',w'})} \quad (3)$$

$$\prod_{s,w}^{pool-burst} = \prod_{s,w}^{pool-burst-temp} + \left(\sum_{s'=1}^{S} C_{s'}^a\right) \frac{\prod_{s,w}^{pool-burst-temp}}{\sum_{s'=1,w'=1}^{S,W} \prod_{s',w'}^{pool-burst-temp}}$$

Using Eq. (3) with consolidation scenario c, the total CPU costs are $62.2 for Workload A and $23.0 for Workload B. Section IV presents a case study that compares cost

apportioning results for Eq. (1), (2) and (3). The formulas are applied separately to various resources such as CPU and memory. The sum of the resulting costs represents the total costs for a workload.

III. Workload Characterization

To evaluate the effectiveness of different apportioning functions, we obtained three months of workload trace data for 312 workloads from one HP customer data center. Each workload was hosted on its own server so we use resource demand measurements for the server to characterize its workload's demand trace. Each trace describes resource usage, e.g., processor and memory demands, as measured every 5 minutes.

We define CPU capacity and CPU demand in units of CPU shares. A CPU share denotes one percentage of utilization of a processor with a clock rate of 1 GHz. A scale factor adjusts for the capacity between nodes with different processor speeds or architectures. For example, the nodes with 2.93 GHz CPUs in our cases study were assigned 293 shares. We note that the scaling factors are only approximate; the calculation of more precise scale factors is beyond the scope of this paper. The memory usage is measured in GB.

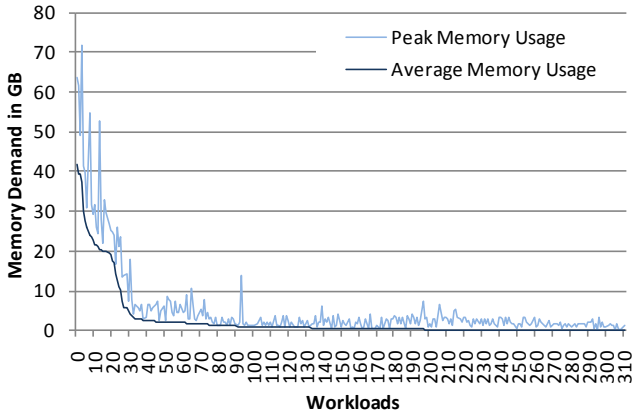

Figure 2: Workload Memory Usage

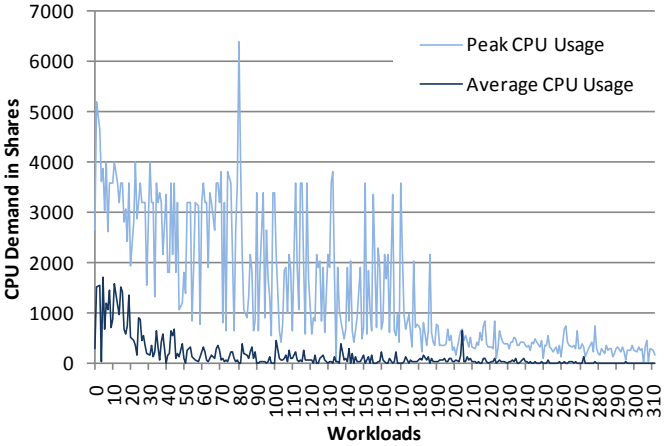

Figure 3: Workload CPU Usage

Figure 2 and 3 summarize the memory and CPU usage for the workloads under study. Figure 2 shows the average and maximum memory usage for each workload. Note that we order workloads by their average memory usage for presentation purposes. Figure 3 shows the average and maximum CPU usage of corresponding workloads. There are a few interesting observations:

For 80% of the workloads, the memory usage is less than 2 GB. While the maximum and average memory usage are small and very close in absolute terms the peak to mean ratios are still high. For 10% of the workloads the memory usage is much higher, 10─70 GB; the maximum memory usage can be very large in absolute terms but the peak to mean ratios are less than 3. There are strong correlations; workloads with a high memory usage have higher peak and average CPU usage. Figure 3 shows that the first 50 workloads have high memory usage and higher average CPU usage than the remaining workloads. Most workloads have very bursty CPU demands: while most of the time these workloads have low CPU usage—85% of the workloads use on average less than 293 CPU shares—their maximum CPU demand is rather high—42% of the workloads have a peak usage of more than 1000 CPU shares. The average peak to mean ratio for CPU usage was 52.6, with some workloads having a peak to mean ratio above 1000.

IV. Case Study

We conducted a comprehensive case study using the workload data for the 312 workloads and resource pool configuration described in Section III to evaluate our proposed cost apportioning approaches. We consider the following shared resource pool configuration: each server consists of 24 x 2.93 GHz processor cores, 128 GB of memory, and two dual 10 Gb/s Ethernet network interface cards for network traffic and virtualization management traffic, respectively. The total acquisition cost for each of these servers was estimated as $58,000, including licensing costs. The costs were approximately $31,000 for CPU and $27,000 for memory. Using a linear depreciation and assuming a lifetime of three years the cost for three weeks is $1,112 per server.

For consolidation, we employ a consolidation engine that minimizes the number of servers needed to host the workloads while satisfying their time varying resource demand requirements [5]. The engine is able to offer many solutions that are near-optimal. To evaluate the robustness, i.e., repeatability, of costs assignments for our approaches, we consider 100 consolidation solutions for a three week costing interval. For the 100 solutions, the consolidation engine reported solutions that assigned the 312 workloads to between 18 and 20 physical servers in the resource pool causing the fine sharing of resources.

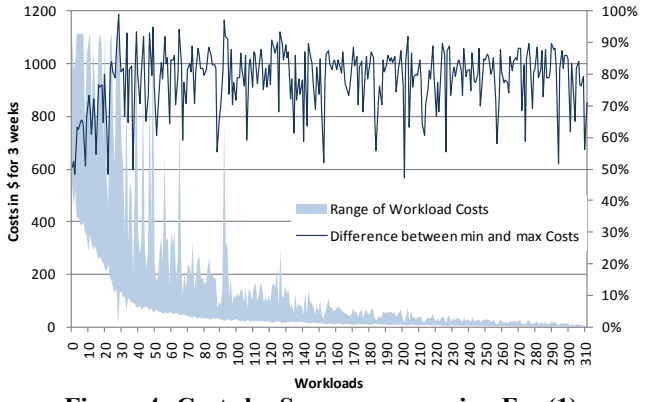

Figure 4: Costs by Server-usage using Eq. (1)

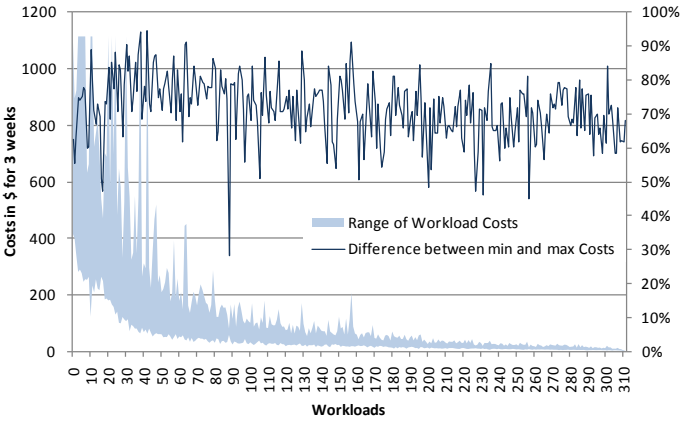

Figure 5: Costs by Server-burst using Eq. (2)

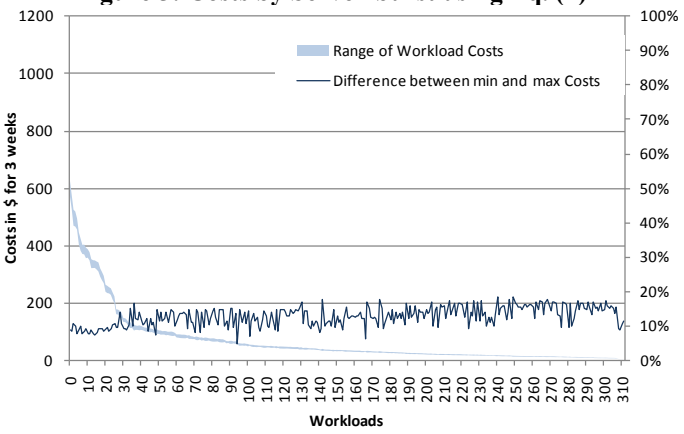

Figure 6: Costs by Pool-burst using Eq. (3)

Figure 4, 5, and 6 show the costs for the workloads as calculated using the *server-usage*, Eq. (1), *server-burst*, Eq. (2), and *pool-burst*, Eq. (3), approaches, respectively. Visual inspection shows that the server-usage and server-burst approaches have very wide ranges for the assigned costs. The average differences between min and max costs assigned to the workloads are 79% and 72%, respectively. Taking into account burstiness decreases the variability in assigned costs, but does not yet yield robust cost assignments. For example, for Workload 5 of Figure 5 cost could range from $250 to $1100 for the same work. Such big differences would make it very hard to plan and charge for customers workloads. Figure 6 shows the results for the pool-burst approach. Its cost assignments are much less sensitive to workload placement decisions and thus have a much tighter range. The average difference between min and max costs is reduced to 13%, which reflects the difference that arises from consolidating to between 18 and 20 servers.

Figure 7 gives a breakdown of the average sum of CPU and memory costs over the 100 consolidation scenarios as apportioned by direct usage (according to $d_{s,w}$), bursty usage (according to $b_{s,w}$), and unallocated usage a_s with respect to Eq. (3). The workloads are sorted by total cost. The figure shows that for most workloads, the largest components of the costs are due to direct resource usage and usage burstiness. As defined by Eq. (3), the relative costs for unallocated resources are similar for all workloads. In this study, the unallocated costs were almost entirely due to memory costs. Finally, the figure shows that the ratio between costs for direct usage and for burstiness differs significantly between the workloads. This is expected as the burstiness of the workloads differs significantly in the set under study.

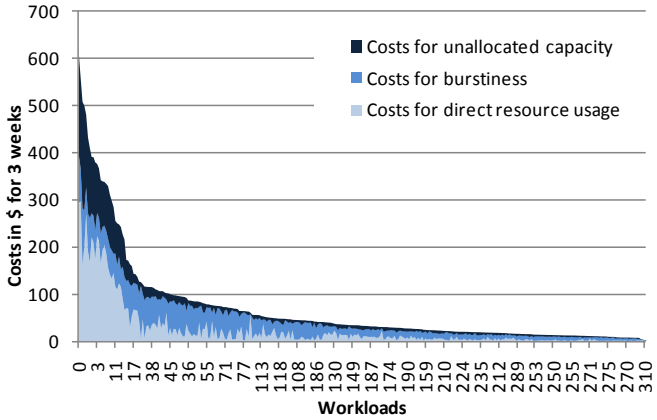

Figure 7: CPU + Memory Costs per Workload

V. Conclusions and Future Work

This paper describes our preliminary work on apportioning costs in cloud computing environments where many application workloads share a resource pool. We described workload performance features that impact resource pool costs and show that these must be taken into account if the true impact of workloads on resource pool costs is to be considered. We have shown that different apportioning approaches have an impact on the robustness of cost assignments and present an approach that offers robust, i.e., predictable, cost assignments that take into account burstiness and is tolerant of different workload placements. The costing information supports charge back and can also be useful in support of more elaborate pricing models.

Our future work includes: applying and extending the proposed method to other aspects of cost including infrastructure, power, and human operation costs; planning for resources that are not used all the time and the relationship with pricing models; and applying the methods to more example workloads. Finally, we will also explore the impact of using other high percentiles for resource usage rather than the peak resource usage, i.e., the 100 percentile, in our apportioning formulas.

Acknowledgment

We thank Wade Satterfield and Paul Miedona of Hewlett Packard for their support of this work and help with data.

References

[1] Amazon web services. http://aws.amazon.com/

[2] IBM Tivoli Usage and Accounting Manager Virtualization Edition. http://www-01.ibm.com/software/tivoli/products/usage-accounting/index.html

[3] HP Insight Dynamics: HP Virtual Server Environment. http://h18004.www1.hp.com/products/solutions/insightdynamics/vse-overview.html

[4] VMware: Virtualize Your Business Infrastructure. http://www.vmware.com/virtualization/

[5] J. Rolia, L. Cherkasova, M. Arlitt, and A. Andrzejak: "A Capacity Management Service for Resource Pools". In Proc. of the 5th Intl. Workshop on Software and Performance (WOSP). Palma, Illes Balears, Spain, pages 229–237.