Enabling Consolidation and Scaling Down to Provide Power Management for Cloud Computing

Frank Yong-Kyung Oh   Hyeong S. Kim   Hyeonsang Eom   Heon Y. Yeom
School of Computer Science and Engineering
Seoul National University
Seoul, Korea, 151-744
{*ykoh,hskim*}*@dcslab.snu.ac.kr*, {*hseom, yeom*}*@snu.ac.kr*

Abstract

Consolidation and scaling down can be used as enabling techniques to effectively manage power for distributed systems. To support power management for cloud computing with consolidation and scaling down, it is necessary to study the effects of mixed workloads and the consolidation on the performance of application in the virtual machines.

Accordingly, we present two studies investigating the effects of VM co-location and CPU thermal management along with the performance in this paper. We conducted experiments to show the effects of performance interference when VMs with different characteristics are consolidated in a single server and analyzed the results so that we can use the results to design a workload-aware VM live scheduler with distributed power management for cloud computing.

1 Introduction

There have been lots of research focused on improving energy efficiency. Scaling down the cluster in distributed systems is an effective method to improve energy efficiency [2, 5, 8]. Server consolidation is another way to improve utilization of physical machines (PM), which consequently reduces the overall power consumption. It is beneficial to integrate scale-down and server consolidation through virtualization since consolidation eases scaling down the cluster through live migration. By dynamically adjusting the number of servers, we can improve the power efficiency of the whole clusters. Several virtual machine (VM) management frameworks have addressed the problem of placing VMs among multiple PMs to minimize the number of PMs while accommodating current loads [1, 3, 6]. Most of them use heuristic methods since VM placement is an instance of bin packing problem [7, 12].

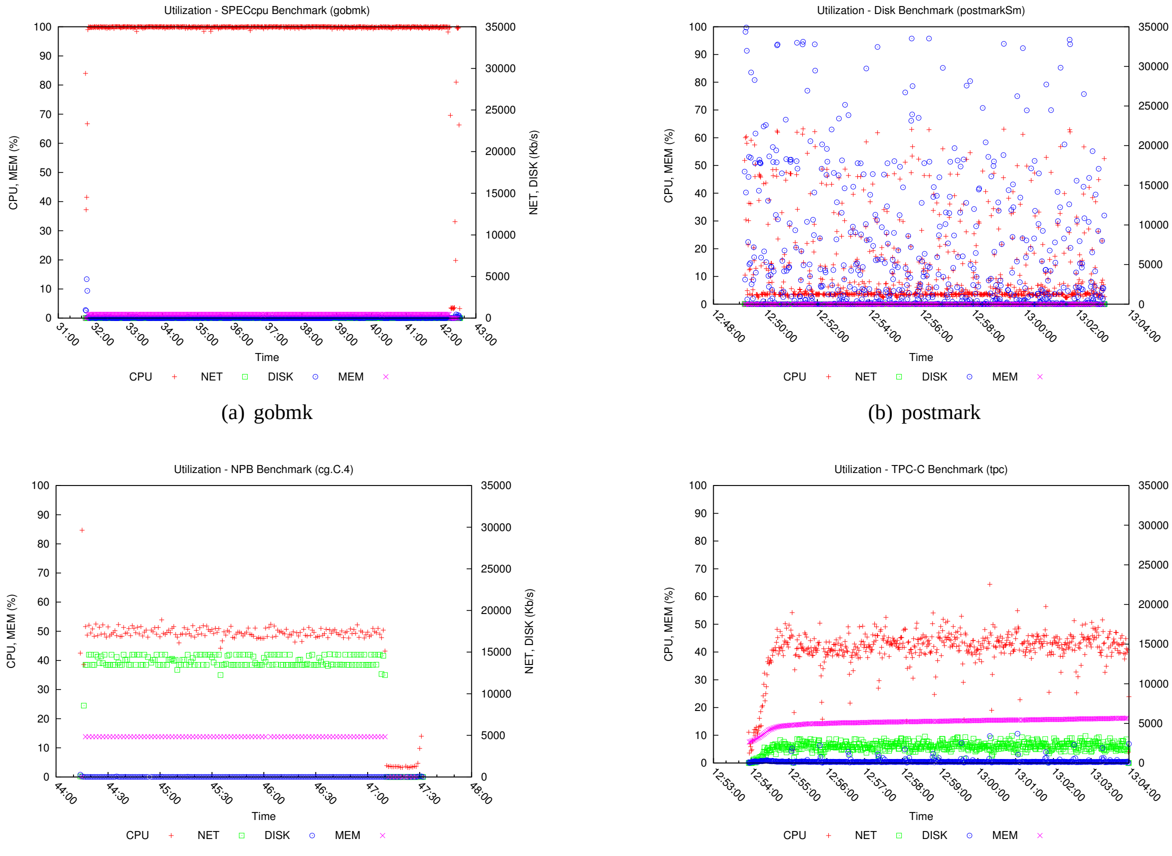
Although previous literature provides several power management frameworks, additional problems still exist that need to be addressed to manage power consumption effectively. Thus, in this paper, we try to answer two fundamental questions: How do we map VMs into PMs so that we can balance performance and power consumption and what is the side effects of consolidating virtual machines in a single server? If VMs run different workloads and they compete for a shared resource such as memory and I/O devices, the performance of VMs can suffer from resource contention. Therefore, it is necessary to consider VM workloads and minimize the performance interference by running different types of VM workloads together.

To design a VM scheduler with aforementioned issues we have to study the effects of VM co-location and other system constraints on the performance. It is clear that mixing different VMs affect the performance of VMs, but other system constraints may also affect the performance. For example, we found that when the system is highly utilized, CPU automatically throttles its frequency so that the system is not damaged due to the high CPU temperature (this is called the CPU thermal throttling). The VM consolidation can lead to such system constraints. If CPU activates thermal throttling, all the cores dynamically adjust its frequency and consequently, the performance of applications is degraded. Therefore, excessive consolidation can cause violation to the Quality of Service in the cloud computing environments. We investigate more on this issue in the following sections.

This paper is a study on the performance variance of applications in virtual machines with various system constraints. Our paper is the first attempt to answer the fundamental questions which arise when we use consolidation and scaling down to improve power efficiency of distributed systems.

2 The Effects of VM Consolidation

In this section, we briefly describe the implementation of our monitoring framework and study the effects of VM



(a) gobmk

(b) postmark

(c) cg.C.4

(d) TPC-C (running a client on a remote machine)

Figure 1: Resource utilization of four applications out of 32 applications.

consolidation.

2.1 Monitoring Framework

In our monitoring framework, we assume that a VM runs either an application in its entirety since the behavior of running application characterizes the behavior of VM itself. There are many different Xen monitoring tools available such as xenoprof [13] and xentrace [4]. Both tools give fine-grain level of gathering architectural data events like cycles, instructions retired, interrupt, etc. However, since we are only interested in per-VM resource utilization, we have used *xentop* tool in our monitoring module as the base gathering command.

The monitoring module periodically records the CPU utilization and other I/O activities per domain. The recorded information is then parsed to extract the resource usage details per VM. During the benchmark runs, we also included process-level monitoring implementation to gather the memory usage of a specific process since xentop only provides current fixed allocated memory instead of providing actual memory utilization.

2.2 A Study on VM Co-location Effects

To study the effects of co-locating VMs in a single machine, we used an Intel Quad-core 2.83Ghz server with Xen 3.3 hypervisor installed. The benchmark suites used in this paper are shown in Table 1. We used SPEC CPU2006 [10], NAS Parallel Benchmarks (NPB) [9], TPC-C [11], disk benchmarks and

Table 1: Benchmark suite used in our experiments. Bold text indicates selected programs that represent different classified groups.

Benchmark	Programs	Remarks
SPECcpu2006	CINT(**gobmk, mcf**), CFP	CPU,MEM-intensive applications
NPB	**cg**, **mg**, ep, lu, sp, bt	CPU+network (NPROCS=4)
TPC-C	**tpc**	CPU + I/O
Disk	**postmark**, copy, bonnie, iozone	CPU + I/O
netperf	**netperf** client	network

netperf. SPEC CPU2006 is an industry-standardized CPU-intensive benchmark suite, comprised of compute-intensive floating point and integer real-world programs. NPB benchmark suite is widely used to evaluate parallel machines and TPC-C is a database benchmark for OLTP systems.

As our first step to identify the characteristics of each application, we repeated the following sequences for all the applications at least three times and used the average value.

1. Set up a virtual machine and deploy the corresponding application into the VM.

2. Run the benchmark application and measure the resource usage (CPU, memory, I/O activity) of inside and outside the VM.

The resource utilization of different applications is shown in Fig. 1. Most SPEC CPU benchmark applications like the Fig. 1(a) fully utilize CPU resource during the most of benchmark run and perform no I/O other than reading the input and writing the results. The Fig. 1(b) shows the utilization of Postmark, the one of I/O intensive benchmarks. As we suspected, it clearly shows lots of I/O activities along with same CPU utilization pattern due to handling the interrupts. Since NPB cg is a MPI-based program which runs in parallel with other machines, it uses a great deal of network resources to exchange data among them. And upon receiving the data, each machine performs moderate computation, showing the 50% CPU utilization. Fig. 1(d) shows the trace of database (DB) benchmark. We see that more network resource is used because a client accesses the database remotely. Since most applications show constant resource usage (behavior), we took the average of each resource utilization and plotted the resource usage in a graph like Fig. 2.

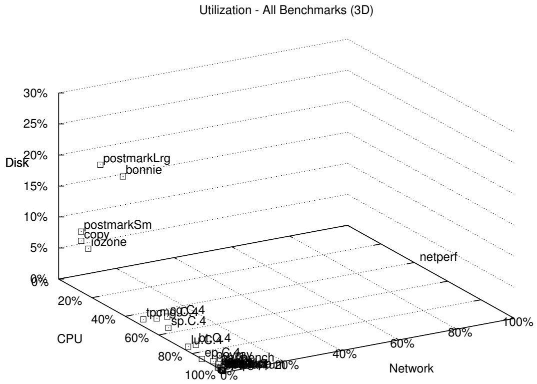


Figure 2: 3D Visualization of resource utilization plot.

According to the resource usage, we classified into seven groups and selected a representative application for each group. Since we want to consolidate VMs efficiently in terms of performance and power, we create a list of combinations (a mix of three VMs) that would utilize different parts of resources in the system.

For each VM, we configured it with 2 GB of memory and 20 GB of VM image. The mix results are shown in Fig. 3.

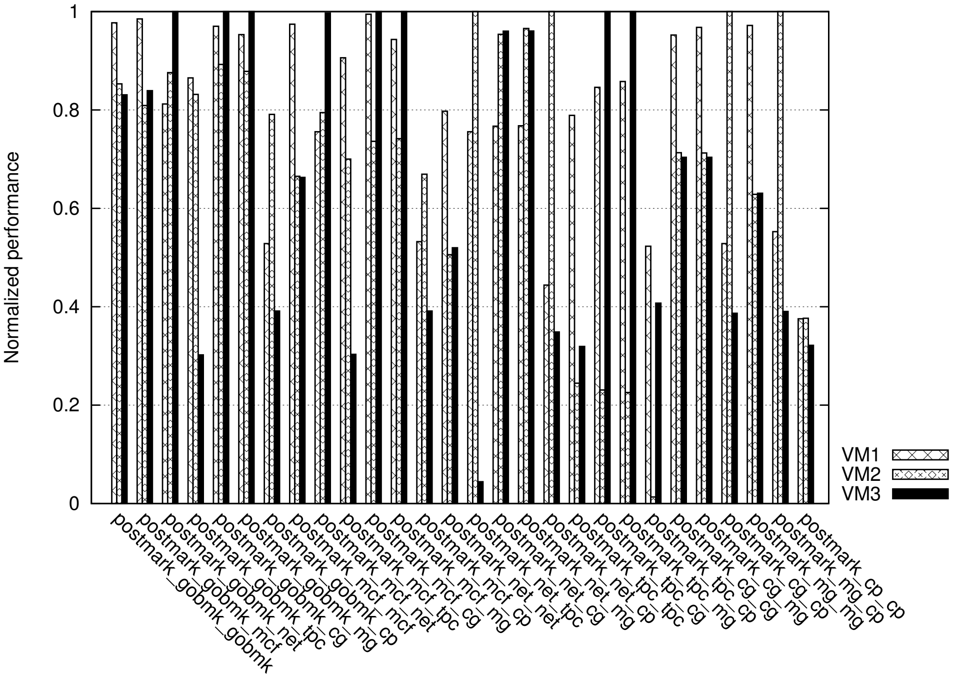


Figure 3: Impact of co-located virtual machines. We measured the normalized performance of each application.

We first explain the performance of gobmk, which is a CPU intensive application. The gobmk shows about 15-20% of performance degradation when it is mixed with postmark, and mcf and gobmk itself. However, the performance degradation becomes less as gobmk is mixed with tpc/net/cp. This is because gobmk fully utilizes CPU and the other applications use less than 50% of CPU. The memory intensive application, mcf, shows poor performance in all the cases. Among those with combined memory intensive application, mcf itself and cp show the worst performance and net shows the least interference. On the other hand, cg and mg have no performance penalty except when it is mixed with net. There is a minor performance penalty due to the network contention between mg/cg and net. When both postmark and cp run together, their performance shows consistently poor performance even if other applications are mixed. For network intensive application, it shows decent performance when is mixed with all other applications except running two network intensive applications together. The tpc shows a large range of different performance outputs. The tpc uses both of CPU and disk moderately. This prevents tpc from boosting in the Xen VM scheduler. Boosting allows I/O VM which went out of the running queue to jump to the head of the queue so that it can be scheduled when the interrupt arrives. However, since tpc moderately uses CPU, tpc is still in the run queue and does not get boosted.

2.3 A Study on Thermal Effects

To study the effects of thermal effects, we stress the target system and inspect how the system reacts. We use the same server in our previous study to identify the effect of thermal change on performance. Since the target system has four cores, we execute four CPU intensive processes simultaneously and measure the power consumption of the system. We found that even if the utilization remains the same at 100%, the power consumption continuously rises for a moment. This is because the temperature affects the power consumption of the CPU. We also found that at a certain point, the power suddenly drops because the CPU has reached a critical temperature. The thermal effect is shown in Fig. 4. In this figure, we show the histogram of frequencies of the four cores while the CPU temperature stays at its critical value. We used the *ondemand* DVFS governor where it periodically checks the current CPU load and maximizes its frequency if the current load is higher than the system threshold. Since the governor is set to ondemand, the frequency of the cores should stay at their maximum frequency while at the peak load. However, as shown in the figure, the frequency of the cores does not constantly remain at 2.83Ghz. Instead, the frequency of all cores is dropped to 2.00 GHz during the one-fourth of the total execution time

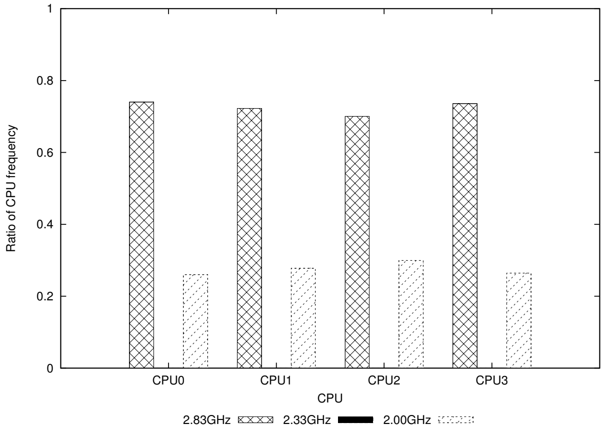


Figure 4: The distribution of time during which each core CPU[0-4] was in one of three available frequencies, 2.00 GHz, 2.33 GHz and 2.83 GHz.

During the experiment, we also measured performance of applications when the system is in the thermal throttle mode. The performance is normalized to that of the maximum frequency. As shown in Table 2, the application under the throttle mode experiences 8% of performance degradation compared to the performance under the maximum CPU frequency. For the performance under the minimum frequency, it is reduced by 29%. Since CPU thermal throttling affects all the cores within the chip, all the applications running on that chip will experience such performance degradation.

In short, the thermal effect on CPU power consumption cannot be neglected since high temperature can result in extra power consumption and the CPU thermal throttling. Therefore, consolidating multiple VMs into a single machine can lead to unexpected performance degradation, which have to be handled carefully.

Table 2: Normalized performance under CPU throttling.

frequency	maximum	minimum	throttle
real value	1	1.40	1.09
derived value	1	1.41	1.09

3 Workload-based Scheduling Design

In this section, we introduce the prototype design of the virtualized management and explain overall design of the architecture. As shown in Fig. 5, there are several different modules in the resource manager. The first two modules, monitoring + prediction and power profiler both use

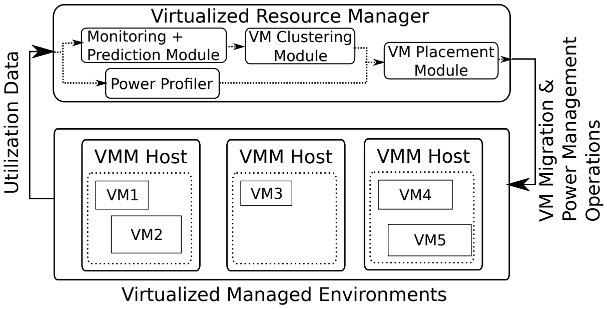

Virtualized Resource Manager
Monitoring + Prediction Module
VM Clustering Module
VM Placement Module
Power Profiler
Utilization Data
VM Migration & Power Management Operations
VMM Host
VMM Host
VMM Host
VM1
VM2
VM3
VM4
VM5
Virtualized Managed Environments

Figure 5: The architecture of virtualized management

collected resource usage from the repository. The prediction module predicts the forecast resource demand using the AutoRegressive Moving Average (ARMA) and power profiler estimates power consumption. Based on the predicted demand of VMs, k-means clustering is performed to classify VM workloads.

Once the classification of VMs is done in the clustering module, next we need to consider how we are going to place VMs into PMs while meeting the power and performance requirements. This solution implemented in the VM placement module. In the module, the mapping of VMs into PMs is determined and it performs the VM migration and power management operations by shutting down unused physical machines. For packing VMs into PMs, we use First Fit Decreasing (FFD) heuristic bin packing algorithm. In addition, we use multi-dimensional bin packing algorithm since we have to consider CPU, memory, and I/O resources.

As discussed in Section 2, we are going to consider different issues such as the CPU frequency throtting and heterogeneous workloads when VMs are paired up during the packing process. In order to quantify preference which VM should be migrated each other, we come up with Pairwise Preference matrix (PP-matrix). The preference value is between zero and one, and close to one indicates more suitable preference to pair up each other. Each value in the PP-matrix is defined according to the effects of VM co-location.

4 Conclusion

In this paper, we mainly have presented studies on the performance effects of VM consolidation with heterogeneous workloads and the thermal effect on performance. And also we have introduced the architecture design as a prototype. Our current implementation are only coded as a simple simulator, but we plan to finalize our implementation of the virtualized management by integrating with OpenNebula. Furthermore we aim to evaluate our approach with real workloads such as web server benchmarks.

Acknowledgment

This research was supported by Future-based Technology Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (20100020731).

References

[1] BELOGLAZOV, A., AND BUYYA, R. Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud data centers. In *Proceedings of the 8th International Workshop on Middleware for Grids, Clouds and e-Science* (2010), pp. 4:1–4:6.

[2] CHUN, B., IANNACCONE, G., KATZ, R., LEE, G., AND NICCOLINI, L. An Energy Case for Hybrid Datacenters. In *Workshop on Power Aware Computing and Systems (HotPower)* (Big Sky, Montana, Oct. 2009), USENIX.

[3] DHIMAN, G., MARCHETTI, G., AND ROSING, T. vGreen: a system for energy efficient computing in virtualized environments. In *Proceedings of the 14th ACM/IEEE international symposium on Low power electronics and design* (2009), pp. 243–248.

[4] GUPTA, D., GARDNER, R., AND CHERKASOVA, L. Xenmon: Qos monitoring and performance profiling tool. Tech. rep., HP Lab, Tech., 2005.

[5] HARNIK, D., NAOR, D., AND SEGALL, I. Low power mode in cloud storage systems. In *Proceedings of the 2009 IEEE International Symposium on Parallel & Distributed Processing* (Rome, Italy, 2009), IEEE Computer Society.

[6] HERMENIER, F., LORCA, X., MENAUD, J.-M., MULLER, G., AND LAWALL, J. Entropy: a consolidation manager for clusters. In *Proceedings of the ACM international conference on Virtual execution environments* (2009), pp. 41–50.

[7] HYSER, C., MCKEE, B., GARDNER, R., AND WATSON, B. J. Autonomic Virtual Machine Placement in the Data Center. Tech. Rep. HPL-2007-189, HP Laboratories, 2008.

[8] KRIOUKOV, A., MOHAN, P., ALSPAUGH, S., KEYS, L., CULLER, D., AND KATZ, R. NapSAC: design and implementation of a power-proportional web cluster. In *Proceedings of the first ACM SIGCOMM workshop on Green networking* (New Delhi,India, 2010), ACM, pp. 15–22.

[9] NASA Parallel Benchmark. `http://www.nas.nasa.gov/Resources/Software/npb.html`.

[10] SPEC CPU 2006. `http://www.spec.org/benchmarks.html`.

[11] TPC-C. `http://www.tpc.org/tpcc/`.

[12] WILCOX, D., MCNABB, A., SEPPI, K., AND FLANAGAN, K. Probabilistic virtual machine assignment. In *Proceedings of the First International Conference on Cloud Computing, GRIDs, and Virtualization* (Nov. 2010), pp. 54–60.

[13] xenoprof. `https://xenoprof.sourceforge.net/xenoprof_2.0.txt`.