Towards Energy-Aware Autonomic Provisioning for Virtualized Environments

Ivan Rodero, Juan Jaramillo,
Andres Quiroz, Manish Parashar
NSF Center for Autonomic Computing
Rutgers University, Piscataway, NJ, USA
{irodero, jjaram, aquirozh,
parashar}@cac.rutgers.edu

Francesc Guim
Intel Corporation and
Technical University of Catalonia
Barcelona, Spain
fguim@ac.upc.edu

ABSTRACT

As energy efficiency and associated costs become key concerns, consolidated and virtualized data centers and clouds are attractive computing platforms for data- and compute-intensive applications. Recently, these platforms are also being considered for more traditional high-performance computing (HPC) applications. However, maximizing energy efficiency, cost-effectiveness, and utilization for these applications while ensuring performance and other Quality of Service (QoS) guarantees, requires leveraging important and extremely challenging tradeoffs. These include, for example, the tradeoff between the need to efficiently create and provision Virtual Machines (VMs) on data center resources and the need to accommodate the heterogeneous resource demands and runtimes of the applications that run on them. In this paper we propose an energy-aware online provisioning approach for HPC applications on consolidated and virtualized computing platforms. Energy efficiency is achieved using a workload-aware, just-right dynamic provisioning mechanism and the ability to power down subsystems of a host system that are not required by the VMs mapped to it. Our preliminary evaluations show that our approach can improve energy efficiency with an acceptable QoS penalty.

Categories and Subject Descriptors

D.4.7 [**Operating Systems**]: Organization and Design—*distributed systems*; C.2.4 [**Computer - Communication Networks**]: Distributed Systems; C.4 [**Computer Systems Organization**]: Performance of Systems

General Terms

Design, Performance

Keywords

Autonomic Computing, Cloud Computing, Energy Efficiency, Virtualization, Data Center, Resource Provisioning

HPDC'10, June 20–25, 2010, Chicago, Illinois, USA.
Copyright 2010 ACM 978-1-60558-942-8/10/06 ...$10.00.

1. INTRODUCTION

The growing scale of enterprise computing environments and consolidated virtualized data centers has made issues related to power consumption, air conditioning, and cooling infrastructures critical concerns in terms of operating costs. Furthermore, power and cooling rates are increasing eight-fold every year, and are becoming a dominant part of IT budgets. Addressing these issues is thus an important and immediate task for enterprise data centers. Virtualized data centers and clouds provide the abstraction of nearly-unlimited computing resources through the elastic use of consolidated resource pools, and provide opportunities for higher utilization and energy savings. These platforms are also being increasingly considered for traditional high-performance computing (HPC) applications that have typically targeted Grids and conventional HPC platforms. However, maximizing energy efficiency, cost-effectiveness, and utilization for these applications while ensuring performance and other Quality of Service (QoS) guarantees, requires leveraging important and extremely challenging tradeoffs. These include, for example, the tradeoff between the need to efficiently create and provision Virtual Machines (VMs) on data center resources and the need to accommodate the heterogeneous resource demands and runtimes of the applications that run on them.

In our previous work [11], we investigated decentralized online clustering (DOC) and autonomic mechanisms for VM provisioning to improve resource utilization [12]. This work was focused on reducing over-provisioning by efficiently characterizing dynamic classes of resource requirements (rather than relying on generic predefined resource classes) and using these for proactive VM provisioning. In this paper we extend this concept an propose an energy-aware online provisioning approach for HPC applications running on consolidate and virtualized computing platforms. Specifically, we explore workload-aware, just-right dynamic and proactive provisioning from an energy perspective. We use decentralized online clustering to dynamically characterize and cluster the incoming job requests across the platform in terms of their system requirements and runtimes. Our approach specifically addresses virtualized cloud infrastructures with multiple geographically distributed entry points to which different users submit applications with heterogeneous resource requirements and runtimes. This results in a demand for different types of VMs with heterogeneous resource configurations. Clustering allows us to identify application tasks that require similar VM configurations. We then use these clusters for just-right VM provisioning and resource configuration so that clustered jobs with similar requirements are mapped to the same host system, allowing it to be configured (and unused subsystems and components to be powered down) to maximize energy efficiency. This is based on the observation that current servers are increasingly providing specific hardware configurations that can save energy, such as turning off or downgrading specific subsystems that are not necessary to run certain applications. We also investigate how a realistic, model-driven proactive provisioning can positively impact the utilization, energy efficiency, and performance of cloud data centers. Furthermore, we not only provision available resources proactively according to expected resource requests, but we also actively and dynamically reconfigure the physical servers if there are no resources available with the required configurations. The overall objective of our autonomic approach is to reduce the energy consumption in the data center, while ensuring QoS guarantees. This improvement in efficiency can help service providers increase their profitability by reducing operational costs and environmental impact, without significant reduction in the service level delivered.

2. BACKGROUND AND MOTIVATION

Existing research has identified many aspects of energy efficiency in different approaches including, for example, energy efficiency in cloud data centers [14, 2], efficient and on-demand resource provisioning in response to dynamic workload changes [8], and platform heterogeneity aware mapping of workloads [9]. Research efforts have also studied power and performance tradeoffs in virtualized [10] and non-virtualized environments [7] considering techniques such as Dynamic Voltage Scaling (DVS) [13, 4]. Moreover, the thermal implications of power management have also been investigated [5].

However, these existing approaches deal with VM provisioning and the configuration of resources as separate concerns, which can result in inefficient resource configurations, and resource under-utilization, which in turn results in energy inefficiencies. Different mechanisms are considered to save energy both at the resource level (DVS in the servers, dynamic reconfiguration of the servers, etc.) and at the virtualization level (VM scheduling, migration, etc.). Other related work addresses VM provisioning, but focuses on VM scheduling and dynamic configuration to adjust resource sharing and consequently reduce over-provisioning and improve utilization. In contrast to these existing solutions, our approach combines proactive VM provisioning with dynamic classes of resources, and workload characterization to provision and configure resources, while bridging the gap between optimizing energy efficiency and ensuring performance and QoS guarantees.

3. ENERGY-AWARE PROVISIONING

In a cloud environment, executing application requests on the underlying resources consists of two key activities: (1) creating VM instances to host each application request by matching the specific characteristics and requirements of the request (VM provisioning); and (2) mapping and scheduling these VMs onto distributed physical resources (resource provisioning). Our provision accomplishes these activities using a sequence of steps as show in Figure 1. The input of our approach is a stream of job requests with their resource requirements. As with most predictive approaches, the flow of arriving jobs is divided into time periods that we call *analysis windows*. Since we focus on a realistic approach, our policy is characterized by having analysis windows of a fixed duration with variable job arrival times, rather than by having a fixed number of jobs in each window. We consider an analysis window with duration of one minute in order to provide a more realistic distribution for job queuing times. Our approach attempts to optimize energy efficiency in the following ways:

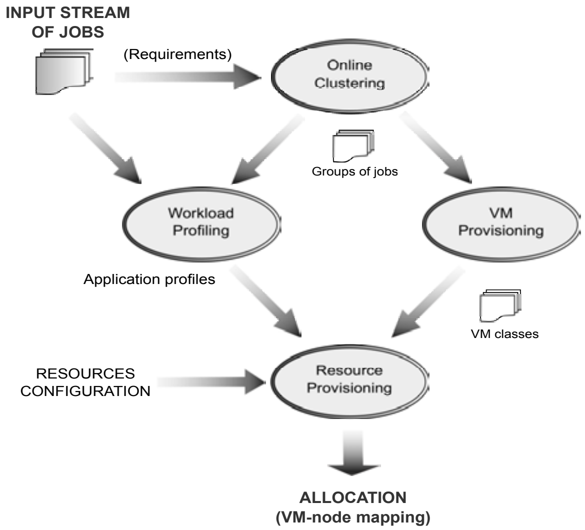

Figure 1: Overall provisioning schema.

- Reducing the energy consumption of the physical resources by powering down subsystems when they are not needed.
- Reducing over-provisioning cost (waste of resources) through efficient, just-right VM provisioning.
- Reducing re-provisioning cost (overhead of VM instantiation) through efficient proactive provisioning and VM grouping.

In the following subsections we describe the different steps of our strategy. We describe the different components of our strategy in the following subsections.

3.1 Online Clustering

For each analysis window, we cluster the job requests in the input stream based on their resource requirements. The clustering analysis returns a set of clusters (groups of job requests with similar requirements) and a set of outliers (job requests that do not match any of the clusters found). Specifically, we use the clustering algorithm presented in [12] to group the incoming requests. While clustering algorithm is capable of analyzing the input data along as many dimensions as desired, as the number of dimensions grows, so does its complexity. The dimensions that we take into account in this analysis include the requested execution time

(T), requested number of processors (N), requested amount of memory and storage, and network demand. To reduce the search space, we perform the analysis in two steps. In the first step, we run the clustering algorithm with only two dimensions: the required memory versus a derived value of execution time (considering also the requested number of processors), which represents CPU demand ($C = \frac{N \times T}{100}$). The value of 100 in the definition of C is a normalization factor that represents the duration in seconds of a reference job. In the second step, we run the clustering on requested storage and network resources over the job requests of each cluster obtained in the first step. We then use the resulting clusters to map jobs to specific VM types and determine the VM provisioning that is required.

Figure 2 shows two plots of the clustering results obtained from two different analysis windows. Each rectangle represents a cluster and contains a set of points inside it, which represent job requests that may be grouped together. The stars represent outliers. Both plots have six clusters but there are important differences between them. Consequently, the VM classes associated to the different clusters will be different for the two different analysis windows.

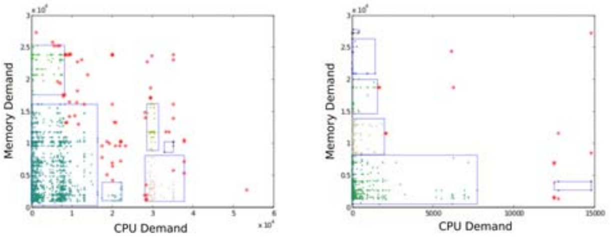

Figure 2: Clusters of two different analysis windows.

3.2 VM Provisioning

Using the clustering of jobs obtained from the online clustering step, we map each job request with a specific VM configuration (e.g. amount of memory) and use this information to determine the different classes of VM that need to be provisioned as well as the number of VMs of each class. This is different from approaches such as that used by Amazon's EC2, which only have fixed predefined VM classes. The different classes of VM identified are matched with currently provisioned VMs and if necessary, appropriate VM classes are proactively provisioned. Job requests in each cluster are then mapped to a VM in the appropriate class. Requests considered as outliers are handled independently. The main objective of this approach is balance the costs due over-provisioning (and under-utilization) and re-provisioning (the delay of configuring and loading a new VM instance).

VM provisioning is done based on the most recent clustering results from the previous analysis window, and thus it is possible to overlap the clustering computation with the creation of VM batches. For the first analysis window, the algorithm reactively creates VMs classes for all incoming job requests. Job requests within the same cluster are provisioned with the same VM class. For subsequent analysis windows, if the job request matches a cluster, the algorithm maps it to an instance of an existing VM class configured for that cluster. Otherwise, the algorithm tries to correlate the job request with other existing VM classes to find the closest match. This reduces re-provisioning costs. because the job request can usually be provisioned with an existing VM instance, therefore saving the VM instantiation delay. To correlate a given job request with a previously defined VM classes, we use the corners of the clusters' bounding boxes (i.e. the area or space occupied by the clusters) in the two-dimensional space. If the requirements of a job are completely covered by the top right corner of an existing cluster, then it can be provisioned with the corresponding VM class, because the resource configuration is sufficient to meet the job's requirements.

Figure 3 illustrates a simple scenario with three cluster from two different analysis windows (win_{i-1}, which is the previous analysis window and win_i, which is the current one). Requests of $cluster_1$ of win_i can be mapped to the VM class that satisfied the requriements of cluster of win_{i-1}. However, since some job requests within $cluster_2$ of win_i are not within the area of the cluster of win_{i-1} (shadowed area in Figure 3), not all job requests of $cluster_2$ of win_i can be satisfied by the VM configuration of the cluster of win_{i-1}.

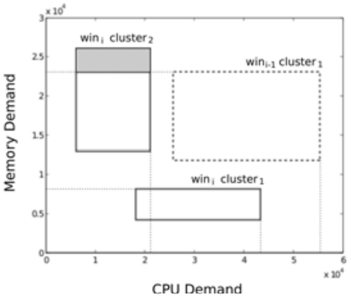

Figure 3: Example of correlation between clusters.

3.3 Resource Provisioning

Once a job request has been matched with a specific VM class, it is mapped to a specific VM instance belonging to that class. Since we proactively create VM batches, we try to reuse existing VM instances from the previous analysis window rather than creating new ones. These VMs are then allocated to physical servers. While performing this allocation, we try allocate VMs that are as closely matched as possible in their resource requirements to the same physical server. This allows us to downgrade the subsystems of the server that are not required to run the requested jobs in order to save energy. To do this, we consider specific configurations of the physical servers' subsystems to reduce their energy demand. Specifically, we use an energy model that leverages previous research on energy-efficient hardware configurations (e.g. low power modes) in four different dimensions:

- CPU speed using Dynamic Voltage Scaling (DVS). We are able to reduce the energy consumed by those applications that are, for example, memory-bound [6].
- Memory usage. For those applications that do not require high memory bandwidth we consider the possibility of slightly reducing the memory frequency or

possibly shutting down some banks of memory in order to save power [1].

- High performance storage. It may be possible to power down unneeded disks (e.g. using flash memory devices that require less power) or spin-down disks [3].

- High performance network interfaces. It may be possible to power down some network subsystems (e.g. Myrinet interfaces) or use idle/sleep modes.

We have implemented two different resource provisioning strategies: a static approach in which physical servers maintain their initial subsystem configuration, and a dynamic approach that allows the physical servers to be reconfigured dynamically. In our dynamic approach, when required physical resources are unavailable, we reconfigure an available physical server to provide the appropriate characteristics and then provision it. In order to perform an efficient mapping of the incoming job requests to physical servers, we consider their requirements in terms of subsystems utilization. Specifically, we consider CPU utilization, requested memory and storage, and communication network usage. To obtain this, we use workload profiling in terms of the resource (i.e. different subsystems) requirements.

4. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed an energy-aware online provisioning approach for HPC applications on consolidated and virtualized computing platforms. Specifically, our provisioning approach is based on the use of clustering techniques to bridge the gap between VM provisioning and resource provisioning. Energy efficiency is achieved using a workload-aware, just-right dynamic provisioning mechanism and the ability to power down subsystems of a host system that are not required by the VMs mapped to it. Our preliminary evaluations showed that our approach achieves significant improvements in energy efficiency (around 15% on average) with an acceptable penalty in QoS (less than 5% in workload execution time). The results also support our argument that just-right dynamic and proactive provisioning using decentralized clustering techniques can improve energy efficiency with an acceptable degradation of QoS.

Current and future research efforts include performing a detailed analysis of our approach on a realistic scenario using simulations. Our experiments target both static and dynamic approaches and the possible tradeoffs between them. We also plan to analyze the possible tradeoffs between VM migration, reconfiguration of servers, and hybrid solutions, such as shutting down servers when they are not used at all. As a long term goal, we will also look into the validation of the simulation results in physical scenarios.

5. ACKNOWLEDGMENTS

The research presented in this paper is supported in part by The Extreme Scale Systems Center at ORNL and the Department of Defense, and by an IBM Faculty Award, and was conducted as part of the NSF Center for Autonomic Computing at Rutgers University. This paper has been also supported in part by the Spanish Ministry of Science and Education under contract TIN200760625C0201.

6. REFERENCES

[1] H. Ben Fradj, C. Belleudy, and M. Auguin. Multi-bank main memory architecture with dynamic voltage frequency scaling for system energy optimization. In EUROMICRO Conf. on Digital System Design, pages 89–96, 2006.

[2] A. Bertl, E. Gelenbe, M. D. Girolamo, G. Giuliani, H. D. Meer, M. Dang, and K. Pentikousis. Energy-efficient cloud computing. The Computer Journal, 2009.

[3] T. Bisson, S. A. Brandt, and D. D. Long. A hybrid disk-aware spin-down algorithm with i/o subsystem support. In IEEE Intl. Performance, Computing, and Communications Conf., pages 236–245, 2007.

[4] R. Das, J. O. Kephart, C. Lefurgy, G. Tesauro, D. W. Levine, and H. Chan. Autonomic multi-agent management of power and performance in data centers. In Intl. joint Conf. on Autonomous agents and multiagent systems, pages 107–114, 2008.

[5] T. Heath, A. P. Centeno, P. George, L. Ramos, Y. Jaluria, and R. Bianchini. Mercury and freon: temperature emulation and management for server systems. In Intl. Conf. on Architectural Support for Programming Languages and Operating Systems, pages 106–116, 2006.

[6] C. Isci, G. Contreras, and M. Martonosi. Live, runtime phase monitoring and prediction on real systems with application to dynamic power management. In IEEE/ACM Intl. Symp. on Microarchitecture, pages 359–370, 2006.

[7] J. O. Kephart, H. Chan, R. Das, D. W. Levine, G. Tesauro, F. Rawson, and C. Lefurgy. Coordinating multiple autonomic managers to achieve specified power-performance tradeoffs. In Intl. Conf. on Autonomic Computing, page 24, 2007.

[8] D. A. Menasce and M. N. Bennani. Autonomic virtualized environments. In Intl. Conf. on Autonomic and Autonomous Systems, page 28, 2006.

[9] R. Nathuji, C. Isci, and E. Gorbatov. Exploiting platform heterogeneity for power efficient data centers. In Intl. Conf. on Autonomic Computing, page 5, 2007.

[10] R. Nathuji and K. Schwan. Virtualpower: coordinated power management in virtualized enterprise systems. In ACM SIGOPS Symp. on Operating Systems Principles, pages 265–278, 2007.

[11] A. Quiroz, N. Gnanasambandam, M. Parashar, and N. Sharma. Robust clustering analysis for the management of self-monitoring distributed systems. Cluster Computing, 12(1):73–85, 2009.

[12] A. Quiroz, H. Kim, M. Parashar, N. Gnanasambandam, and N. Sharma. Towards autonomic workload provisioning for enterprise grids and clouds. In IEEE/ACM Intl. Conf. on Grid Computing, pages 50–57, 2009.

[13] P. Ranganathan, P. Leech, D. Irwin, and J. Chase. Ensemble-level power management for dense blade servers. SIGARCH Comput. Archit. News, 34(2):66–77, 2006.

[14] S. Srikantaiah, A. Kansal, and F. Zhao. Energy aware consolidation for cloud computing. In USENIX Workshop on Power Aware Computing and Systems, 2008.