Green Computing: Optimization of Data Centers for Resource Allocation by Server Virtualization

Anil Kumar K[1] Prof. Mr. Manoj Kumar H[2]
[1] M.Tech. Student [2] Assistant Professor
[1] CNE Department
[1,2]RNS Institute of Technology, Bangalore, Karnataka, India.

Abstract— Cloud computing allows scale up and down server resource usage based on the application needs. Cloud resources come from resource multiplexing through virtualization technology. In this paper, a system is introduced that uses virtualization technology to allocate data center resources dynamically based on application demands and support green computing. A virtual machine is created by virtualizing the physical machine and it is used as a server with its own set of resources. The skewness algorithm is used to generate a alarm to indicate that the virtual machine reached its higher threshold level. When an alarm is generated a required resources can be added to the virtual machine without any downtime.

Keywords: Server Virtualization, Hypervisor, Cloud Computing, Data Center, Resource Multiplication.

I. Introduction

Virtualization and cloud computing technologies are transforming IT. The cloud computing leverages virtualization to enable a more scalable and elastic model for delivering IT services. As a result, enterprises gain a more agile and efficient IT environment that is better able to respond to business needs.

The principal driver behind the rapid adoption of virtualization has been cost reduction through server and other infrastructure consolidation. With virtualization, enterprises are no longer restricted to the traditional ratio of 1:1:1 for servers, operating systems and applications. Applications abstracted from the infrastructure enable IT to turn underutilized infrastructure into an elastic, resilient and secure pool of compute resources available to users on demand. Global IT organizations have been quick to adopt virtualization to achieve cost benefits. By replacing physical IT assets with virtual resources, organizations are achieving up to 60 percent savings in capital expenses in their data centers. Server virtualization has become a standard feature in data center environments over the last few years, with virtual machine deployment outnumbering physical server shipments in 2009.

Virtual machine monitors (VMMs) like vSphere ESXi provide a mechanism for mapping virtual machines (VMs) to physical resources. This mapping is largely hidden from the cloud users. Users with the Amazon EC2 service, for example, do not know where their VM instances run. It is up to the cloud provider to make sure the underlying physical machines (PMs) have sufficient resources to meet their needs. VM live migration technology makes it possible to change the mapping between VMs and PMs While applications are running [1], [2]. However, a policy issue remains as how to decide the mapping adaptively so that the resource demands of VMs are met while the number of PMs used is minimized. This is challenging when the resource needs of VMs are heterogeneous due to the diverse set of applications they run and vary with time as the workloads grow and shrink. The capacity of PMs can also be heterogeneous because multiple generations of hardware coexist in a datacenter.

We aim to achieve two goals in our algorithm:

- Overload avoidance. The capacity of a PM should be sufficient to satisfy the resource needs of all VMs running on it. Otherwise, the PM is overloaded and can lead to degraded performance of its VMs.
- Green computing. The number of PMs used should be minimized as long as they can still satisfy the needs of all VMs. Idle PMs can be turned off to save energy.

There is an inherent trade off between the two goals in the face of changing resource needs of VMs. For overload avoidance, we should keep the utilization of PMs low to reduce the possibility of overload in case the resource needs of VMs increase later. For green computing, we should keep the utilization of PMs reasonably high to make efficient use of their energy.

II. Related work

A. Green Computing

Many efforts have been made to curtail energy consumption in data centers. Hardware-based approaches include novel thermal design for lower cooling power, or adopting power proportional and low-power hardware. Work [3] uses dynamic voltage and frequency scaling (DVFS) to adjust CPU power according to its load. We do not use DVFS for green computing. PowerNap resorts to new hardware technologies such as solid state disk (SSD) and Self-Refresh DRAMto implement rapid transition(less than 1ms) between full operation and low power state, so that it can “take a nap” in short idle intervals. When a server goes to sleep, Somniloquy notifies an embedded system residing on a special designed NIC to delegate the main operating system. It gives the illusion that the server is always active. Our work belongs to the category of pure-software low cost solutions [4], [5], [6]. Similar to Somniloquy, SleepServer [26] initiates virtual machines on a dedicated server as delegate, instead of depending on a special NIC. LiteGreen [4] does not use a delegate. Instead it migrates the desktop OS away so that the desktop can sleep. It requires that the desktop is virtualized with shared storage.

B. Resource Allocation at the Application Level

Automatic scaling of Web applications was previously studied in and for data center environments. In MUSE, each server has replicas of all web applications running in the system. The dispatch algorithm in a frontend L7-switch makes sure requests are reasonably served while minimizing the number of underutilized servers. Work uses network flow algorithms to allocate the load of an application among its running instances. For connection oriented Internet services like Windows Live Messenger, work presents an integrated approach for load dispatching and server provisioning. All works above do not use virtual machines and require the applications be structured in a multitier architecture with load balancing provided through an front-end dispatcher. In contrast, our work targets Amazon EC2-style environment where it places no restriction on what and how applications are constructed inside the VMs.

MapReduce is another type of popular Cloud service where data locality is the key to its performance. Quincy adopts min-cost flow model in task scheduling to maximize data locality while keeping fairness among different jobs. The “Delay Scheduling” algorithm trades execution time for data locality. Work assigns dynamic priorities to jobs and users to facilitate resource allocation.

III. Proposed work

In this paper, we present the design and implementation of an automated resource management system that achieves a good balance between the two goals. We make the following contributions:

– We develop a resource allocation system that can avoid overload in the system effectively while minimizing the number of servers used.

– We introduce the concept of “skewness” to measure the uneven utilization of a server. By minimizing skewness, we can improve the overall utilization of servers in the face of multidimensional resource constraints.

Virtualization approaches use either hosted or hypervisor architecture. A hosted architecture installs and runs the virtualization layer as an application on top of an operating system and supports the broadest range of hardware configurations. In contrast, hypervisor (bare-metal) architecture installs the virtualization layer directly on a clean x86-based system. Since it has direct access to the hardware resources rather than going through an operating system, a hypervisor is more efficient than a hosted architecture and delivers greater scalability, robustness and performance.

Figure (1) shows the physical machines as a enterprise servers. A bare metal virtual machine monitor that is vSphere ESXi hypervisor is installed directly on the physical server system. The virtual machines are created with a particular operating system to use as servers. All the virtual machines are added to a centralized management system called vCenter server. The vCenter server is a centralized management tool for the vSphere suite. vCenter server allows for the management of multiple ESXi servers and virtual machines from a different ESXi servers through a single console application.

Virtual center provides statistical information about the resource use of each virtual machine and provisions the ability to scale and adjust the compute, memory, storage and other resource management functions from a central application. It manages the performance of each virtual machine against specified benchmarks, and optimizes resources wherever required to provide consistent efficiency throughout the networked virtual architecture. Besides routine management, virtual center also ensures security by defining and monitoring access control to and from the virtual machines, migration of live machines, and interoperability and integration among other Web services and virtual environments.

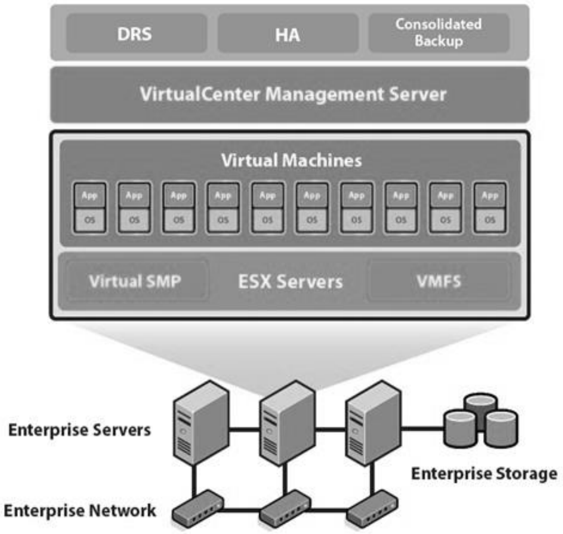

Fig. 1: System Architecture

The skewness algorithm is used to set the threshold limits to virtual machines so that when virtual machine reaches threshold level an alarm will be generated in the vCenter server interface to indicate the running out of resources in virtual machine. The vCenter server manager can able to allocate the necessary resources to the virtual machine.

IV. Implementation

A datacenter is created using the VMware virtualization technology. The global standard Virtual Machine Monitor (VMM) or Hypervisor is used to divide the physical server into multiple virtual machines. All the virtual machines should be managed through a single sign-on server called vCenter server. The virtual machines are connected through a virtual network. Multiple virtual machines shares same storage media but the one virtual machine instance will not be aware of another instance in the same storage.

A website designed to take advantages of server virtualization. It is deployed in the virtual machine environment to check the CPU, memory utilization and performance of the virtual machine. It provides an interface to storing the files and documents and also downloading files from the Data Center.

A. Skewness algorithm

```
// run the virtual machine
If (memory level of VM exceeds threshold level )
```

```
{       An alarm will be generated to notify the memory over flow
}
Else If (VM is not accessed longer time)
{       The VM will be in idle mode and save power
}Else {
        VM is healthy and accessed by clients effectively
}
```

B. Data uploading

```
// Start the XAMPP server
// Start the Apache and Mysql
If (number of files is 1 and file size is 2MB ) {
        Identifies the file type
uploads the file to the directory specified.
}
else {  Display the error message
}
```

C. Data Downloading

```
// Click on the required file
If(file selected is not an image){
Download the selected file to the client
Machine
}
Else {  Display image in gallery
}
```

The downloading and uploading files into the server is designed to check the performance statistics of the virtual machine created using server virtualization technology. Data uploading limit is fixed to 512 Kbps this is because of multiple users are going to be access the Data Center resource at the same time. Hence it satisfies the required conditions to check the virtual machines performance, CPU and memory utilizations.

V. Conclusion

The project presents, a design, implementation, and evaluation of a resource management for cloud computing services. The server virtualization multiplexes virtual to physical resources adaptively based on the changing demand. The Server virtualization achieves both overload avoidance and green computing for systems with multi resource constraints. The system maps the virtual resources to physical resources dynamically and saves energy thus supports the Green Computing. The system provides good performance, availability, efficiency and cost effective design to build better data centers.

Acknowledgement

I take this Opportunity to express my profound gratitude and deep regards to my guide Prof, Mr. Manoj Kumar H Assistant Professor, RNS Institute of technology, Bangalore, for his exempla nary guidance, and constant encouragement throughout.

References

[1] C. Clark, K. Fraser, S. Hand, J.G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, *Live Migration of Virtual Machines*, Proc. Symp. Networked Systems Design and Implementation (NSDI ’05), May 2005.

[2] M. Nelson, B.-H. Lim, and G. Hutchins, *Fast Transparent Migration for Virtual Machines*, Proc. USENIX Ann. Technical Conf., 2005.

[3] R. Nathuji and K. Schwan, *Virtualpower: Coordinated Power Management in Virtualized Enterprise Systems*, Proc. ACM SIGOPS Symp. Operating Systems Principles (SOSP ’07), 2007.

[4] T. Das, P. Padala, V.N. Padmanabhan, R. Ramjee, and K.G. Shin, *Litegreen: Saving Energy in Networked Desktops Using Virtualization*, Proc. USENIX Ann. Technical Conf., 2010.

[5] Y. Agarwal, S. Savage, and R. Gupta, *Sleepserver: A Software- Only Approach for Reducing the Energy Consumption of PCS within Enterprise Environments*, Proc. USENIX Ann. Technical Conf., 2010.

[6] N. Bila, E.d. Lara, K. Joshi, H.A. Lagar-Cavilla, M. Hiltunen, and M. Satyanarayanan, *Jettison: Efficient Idle Desktop Consolidation with Partial VM Migration*, Proc. ACM European Conf. Computer Systems (EuroSys ’12), 2012.