High Performance Computing and Virtualization

Mit Desai
Department of Computer Science
University of Southern California
Los Angeles, CA.
mitdesai@usc.edu

Research Paper: CSCi-555 Advanced Operating Systems (Fall 2012)

Abstract:
These days, Virtualization is being widely used in the enterprise. But still, it does not have any major contribution in High Performance Computing\(^1\)(HPC). Not just limiting to Cloud Computing, virtualization has some other greater benefits in the High Performance Computing like Fault Tolerance, Job Migration, Scalable Hypervisors and many more. In this paper, I have discussed these topics and also the pros and cons in this new emerging area of computing.

1. Introduction:

The working atmosphere in the HPC has a great difference from what it is in the traditional enterprise. HPC uses computer clusters for solving computational problems. In this approach, large number of processors are used in close proximity to each other addressing a single problem.[17][2]

However, using computer clusters to solve a problem, would be a costly option. Virtualization has many advantages like having a customized OS, system security and performance isolation. So having a single computer with running many Virtual Machines\(^2\) (VM) over it can be a possible alternative to it.

This paper discusses the virtualization technology in HPC and its pros and cons.

Further in this paper, Section 2 consists of the overview of the Virtualization in HPC. Section3 discusses how heterogeneity adds a benefit to it. Section4 describes the scalable

\(^1\)High Performance Computing Referenced as HPC in the rest of the paper.
\(^2\)Virtual Machines referenced as VM in rest of the paper
hypervisors and multi-core virtualization. Section 5 describes the Reliability and fault tolerance followed by a description of VMM Bypass I/O in Section 6. At the end, the paper discusses the security and the issues that are to be faced while Virtualization is used in HPC.

2. Virtualized HPC:

In computing, virtualization is the creation of a virtual version of a resource that is not visible to the operating system, end user or the application that is interacting with it.[3] There are two types of virtualizations that have to be kept in mind while considering HPC. Hardware Virtualization and Desktop Virtualization.

Hardware Virtualization refers to the creation of a virtual machine that acts like a real computer with an operating system. Software executed on these virtual machines is separated from the underlying hardware resources[3][4]. Hardware Virtualization is in-turn divided into three sets. 1) Full Virtualization[6] 2) Partial Virtualization[6] 3) Para Virtualization[5]

Desktop Virtualization sometimes also referred as Client Virtualization is a technique where the desktop environment of a machine is separated from the physical machine. Virtual Desktop Interface is the server computing model enabling desktop virtualization including the hardware and software systems required to support virtualized environment[7]

For creating and running the virtual machines, an extra piece of software is needed. This is known as a Hypervisor or a Virtual Machine Monitor³ (VMM). VMM allows the host machine (the machine on which it is running) to run multiple virtual machines (known as the Guest Machines). Each of this guest machine can have its own customized environment. The XEN Virtual machine Monitor is discussed below.

XEN[23], the VMM is at lowest level directly accessing the hardware. The VMM runs on the most privileged processor level. VMs containing the guest OS run on the top of the VMM. A special domain, Dom0 is created at the boot time. It has the access to control the interface provided by hypervisor and can control the creation, termination and migration of the Guest OS. Domains communicate with each other through channels and shared pages. Interrupts are generated when send operations from one side of an event are to be received at the other side. A domain interacting with another domain, gives access to its local shared memory pages to that particular domain and then sends a notification event to that domain. The shared pages are then used for data transfer. [8]

The figure if XEN Hypervisor is shown below which would make it more clear to understand the details.

³Virtual Machines Monitors referenced as VMM in rest of the paper
3. *Heterogeneous Virtual Systems:*

HPC uses computer clusters for solving computational problems. In this approach, large number of processors are used in close proximity to each other addressing a single problem.[17][2] These cluster of nodes are of the same kind. That means, all the nodes have a mutual agreement on what operating system to be used.

Such an agreement is always beneficial as far as administrative perspective is taken into consideration. Also identical applications can run on any of the node in the cluster. But such a system suffers greatly in flexibility. For an instance we can consider a homogeneous system that provides services to users with different applications. These applications in turn run on different platforms (Windows, UNIX, Solaris etc). Resource sharing can be a headache in such a scenario. A user will be unable to use the shared resources due to platform dependencies. Reconfiguring and rebooting a node can be an option But if it is considered, the aim for high performance cannot be achieved.[9]

This issue can be best approached by Virtualization. A virtualized system can be created where a single physical machine can host one or more virtual machines and use it as a customized operating system. So a user can now use the resources as they would be on the same workstation and he can even freely make choice of the operating system. The greater benefit of the customized operating system would be to make a light weight operating system
for any specific applications. The VM image would include the minimum of the OS and runtime variables as possible. The booting and transportation of such a small VM image would be very fast and hence can give a fair amount of performance gain.

4. **Multi-Core Virtualization:**

There are two main issues for which the hardware should take an active role in management of resources. 1) There is an uncertainty in efficiently mapping the computation to the hardware. And to overcome these certainties, requires a detailed knowledge of the architecture of the core. Which cannot be easily acquired. 2) The capabilities and hardware change rapidly. But the cost of implementing a scheduling decision into the OS has increased as compared to the cost of computation.[10]

Considering these issues, the hardware needs to be playing an active role in managing the on-chip cores. The details on the on-chip cores can be abstracted using a multi-core virtualization layer homogeneous Virtual Processors (VCPUs) can be exposed to the OS through the hardware and software interfaces (ISA). The multi-core hardware adapt to the dynamic heterogeneity of the actual physical cores. So, the lowest level of the system software remains completely unmodified.[10]

Figure 2 illustrates multi-core virtualization and its two basic abilities. First, it supports the ability to move a software-visible VCPU from one core to another, and the ability to temporarily suspend execution of a VCPU when there are no appropriate cores on which it can run. Second, is preserves the illusion that all VCPUs are simultaneously executing, even if a subset of them are suspended[10].

---

**Figure 2: Multi-core Virtualization[10]**

(*Virtual Processors referenced as VCPUs in rest of the paper*)
In the figure shown below, Four VCPUs are exposed the software, only three cores are actually present. VCPUs V0, V1, and V3 have been transparently migrated, while VCPU V2 has been transparently suspended.

**Scalable Hypervisors with multiple Cores**

The scenario in the enterprise environment where virtualization is implemented is completely different that what it is in the HPC. As the resource requirement is less in the enterprise environment, many VMs are hosted on a single physical machine. Compared to it, in the HPC environment, the number of VMs is not so high. It would not exceed the number of cores available for computation. Less VMs can have access to larger amount of resources and thus can lead to a higher performance.

As for the current implementation of the hypervisors is considered, they are monolithic. Due to such implementation, all the cores that are in the system will execute for the same VMM operation. As a result, there will be much overhead as the number of context switches increases. This might lead to cache pollution, resulting into a performance hit, and TLB trashing. An alternative to this approach is the approach of sidecore[11]. Here, each core is responsible for the execution of certain set of instructions rather than the complete set of VMM functions. The guest OS sends a request to a sidecore.

But for putting this approach into implementation, there are certain things that are to be kept in mind. The guest OS needs minor modifications for adapting to this approach. Also this approach wastes numerous clock cycles for checking the incoming requests from the guest OS. The sidecores have their dedicated functions. So they cannot be used for any other purpose other than for what they are designed.

Another approach for high performance computing with virtualized systems is by the use of high-end virtualization[25]. In place of accumulating and assigning multiple cores to a single machine, multiple machines can be combined together making them act like a single high-end machine. This is known as High-End Virtualization. The single virtual system thus aggregated is much more powerful. This technique known as aggregation is highly preferred in HPC.

5. **Reliability and Fault Tolerance:**

Since last many years, it was believed that the hardware is a reliable resource. The algorithms and the software infrastructure so far were dependent upon this belief and the fact has almost remained true. But, maintainability of the hardware reliability is becoming a new challenge.
And this is due the technology scale making the devices more and more likely to have hardware faults caused by lots of factors.[10]

Now, if there is any failure in any core, the effect is limited only up to that VM. This is because virtualization provides complete isolation to the machines. Virtualization provides reactive as well as proactive fault tolerance.

Reactive fault tolerance is a technique which takes an appropriate action once a failure is detected. It does this by restoring the running instance of OS to the last checkpoint. Checkpointing is done for every running VM periodically. It is even expensive to do this. In virtualization the entire state of the VM is captured and its snapshot is saved. The process of saving this data to disk should be quick for avoiding the hardware failures. This technique is helpful only till the rate of the failure is less. Once the failure rate goes high, checkpointing becomes a difficult job and the fault tolerance mechanism will not function as it should.

Proactive fault tolerance on the other hand deals with the faults before they actually occur. It does so by the management agents who predicts the application failures and migrates them to another stable node to avoid the failure. The *Live Migration* mechanism provided by XEN is explained below.

**Live Migration:**

Xen's live migration transfers an application on a guest OS to any other healthy node before it fails. It also provides the pre-migration mechanism where all the states are transferred before the application starts executing on the new healthy node. The sequence in which the live migration occurs is explained below:[12]

a) Pre Migration: Whenever the migration is initiated, host inquires for available resources and if found, it will reserve them for the process.

b) Pre Copy: All the pages are sent to the new node while the guest OS still continues the execution of the application. The changes made during this time are sent as dirty bits so that they can be identified on the new node.

c) Still in the Pre Copy phase, the host now sends all the dirty bits.

d) Now, the host stops and sends all the modified pages to the new node where the VM is restarted.
6. **VMM Bypass I/O:**

In VMs, the I/O operations need the involvement of the VMM which is a much more overhead and reducing it can lead to a very high performance gain. The VMM Bypass I/O[8] extends the idea of OS Bypass[25]. It removes the bottleneck of the I/O access through VMM.

For this new mechanism, a new device driver is needed for the guest OS known as the guest module. It is responsible for handling the privileged access to the device. To bypass the VMM for the I/O operations, the guest OS creates virtual access points and maps them to the user processor. Another driver known as the backend software module is needed for accessing the device driver. Again, if the devices are accessed by the VMM, this backend module can also be implemented into the VMM. The VMM bypass I/O design is described in the figure shown below.

![Figure 3: VMM Bypass I/O](image)

7. **Security:**

All the applications that are running, run on a single physical machine. But for every application, it executes in its own VM. VMM provides isolation among all the running VMs. It is a key factor in providing the security to a virtualization system. The data that is used by one VM is not visible to any other VM on that VMM.
In a traditional HPC system, all the applications or the users have to be trusted with privileges for loading any modules. With the use of VM, a user is allowed to execute such privileged instructions on its specific VM. As the VMs are isolated, the instructions executed by any one VM will never affect any other VM. Moreover, the VMM controls the integrity of a physical machine. So if any instructions on any guest OS runs some malicious code and the VM crashes, it can be easily recovered from the snapshot that was recently captured.

8. **Problems with the use Virtualization in HPC:**

Although virtualization has many advantages and it has a great potential, there are several things that needs to be studies and ways to handle them have to implemented.

The problems described here needs to be studies well in order to implement VM in a HPC System.

1. In case of a traditional servers, there are a few VM that are deployed at a time. managing them is not a problem. But in HPC, there might be cases where large number of VMs need to be deployed. Ways for managing the resource allocation to such dynamic deployment has to found out

2. The traditional servers has a single VMM that abstracts the underlying hardware. In case of the heterogeneous HPC, a separate VMM is needed per node. So there must be some way for the coordination and communication between all these VMMs

3. HPC Systems are to be built to handle the dynamic load. Presented a set of applications, the distributed resource manager will make a static decision based on the currently available cluster of resources. Once allocated, application would either run to complete execution or fail.

9. **Conclusion:**

With the introduction of Virtualization into the High Performance Computing, there are significant gains in what HPC is today. Virtualization in HPC would take it to the new heights with the features that it will introduce into the HPC like customization, scalability, reliability, fault tolerance and security.

But still there are certain areas that are unexplored and needs to be studied. A solution to the problems currently faced in the implementation of Virtualization into HPC is to be found out. Once these issues are resolved, HPC will set new heights for virtualization.
10. **References:**


2. *Analysis and performance results of computing betweenness centrality on IBM Cyclops64* by Guangming Tan, Vugranam C. Sreedhar and Guang R. Gao The Journal of Supercomputing Volume 56, Number 1, 1–24 September 2011


4. "Virtualization in education". IBM. October 2007. Retrieved 6 July 2010. "A virtual computer is a logical representation of a computer in software. By decoupling the physical hardware from the operating system, virtualization provides more operational flexibility and increases the utilization rate of the underlying physical hardware."


6. VMware (11 Sep 2007). "Understanding Full Virtualization, Paravirtualization, and Hardware Assist"


16. Lamia Youseff, Rich Wolski, Brent Gorda and Chandra Krintz, *Paravirtualization for HPC Systems*, Workshop on XEN in HPC Cluster and Grid Computing Environments (XHPC), held in conjunction with The International Symposium on Parallel and Distributed Processing and Application (ISPA 2006), December 2006


25. *Opportunities for Leveraging OS Virtualization in High-End Supercomputing*, by Kevin T. Pedretti and Patrick G. Bridges