Broadening the Battle-zone

Some Thoughts on the Extension of Grid/Cluster Computing into the Desktop and Pool PC Domain

Michael Janczyk, Dirk von Suchodoletz

Professorship for Communication Systems, Institute of Computer Science, University of Freiburg

Abstract. Virtualization is established really well on the desktop of software engineers and testers or people who would like to run another operating system within their favorite desktop environment and it sells rather well for computer center machinery consolidation. This paper would like to explore the suitability of virtualization concepts for the idea of distributed computing. The idea itself is not really new, but the introduction of virtualization here might solve a set of open problems in this field like data security issues, separation of functionality and management of each domain: The desktop and the cluster node environments. Optimally both could be kept away from each other allowing completely different entities to manage their domain without interfering with the other. Linux operating systems especially in stateless deployment allow a high grade of flexibility and ease of use in large scale setups, which are of interest to cycle stealing endeavors. Thus the research outlined is focused on this host and cluster environment but open to other options too if applicable.

1 Introduction and Basic Idea

The number of networked machinery is still rising. Nearly every workplace in scientific institutions will be computer equipped. Additionally you will find a number of public workplaces and lecture pools in educational institutions. In the last few years the cluster computing gained some significant attention to meet the rising demand for number crunching. All these machines are to be installed, managed and operated. Cluster and grid computers tend to be separate infrastructure duplicating much of the already offered resources and services: They implement their own network infrastructure, need administration and system management, filling up 19" racks and requiring a substancial amount of cooling.

Bringing together unused compute power of workstations and pool machines might utilize already installed resources better leads to a set of questions: How these resources could be used more effectively and efforts to be bundled? Is there any option of cooperation of completely different faculties to share their machinery? Which requirements and conditions are to be met? The answer to these questions will influence the consulting services of the Computer Center, the resource planning and decisions on purchasing of new equipment.

Virtualization for secure cycle stealing There exist two relevant modes of operation in this field: The traditional mode of changing the operation environment in off hours into cluster computing. Or: Using virtualization technologies to utilize idle CPU cycles without interfering and mixing with the prior ranking desktop operation. There are different strategies of virtualization to be evaluated in their potential.

This paper will outline the basic requirements for idle cycle stealing, the several virtualization scenarios, the setup options and the research questions to prove. In this paper a Linux operating environment because of its predominance in the field of number crunching is presumed. The host environment of the research discussed here is Linux too, offering a broader range of virtualization options. Some of the tools discussed later on are transferable other host environments too.

Advantages of network booting The time of autonomous workstations is over: The majority of machines in larger organisations is centrally managed. The machines are provided with IP configuration via DHCP service. Their setup is done automatically with either software distribution frameworks, several kinds of netinstall or completely stateless operation. Nowadays networks offer huge bandwidths and thus allow fast network setup or network dependent operation. Most of the every day tasks are network dependent so there is no loss in productivity if the operating systems are provided over the network too.

Most interesting option under the viewpoint of installation and maintenance is stateless operation mode for the main and cycle stealing environment on netbooted nodes. This would offer a fast roll-out and easy update without the need to install anything relevant on the nodes itself.

The Matrix The constraints of using standard workplace environment differ from the usual grid or cluster setups:

- In the best case the average desktop user shouldn't detect that someone else is using his/her machine too.
- Neither the desktop user should be able to interfere with the cluster operation nor the other way round.
- Both domains should be kept as far as possible from each other: Users logged on into one or the other environment shouldn't be able to read and alter data of the other ones.
- The overall overhead of different virtualization methods is to be estimated. The less any deployed environment consumes the more is left for computing.

For a later deployment of the results the complexity to master is of interest: The different virtualization tools and environments might require special kernels and/or modules or additional tools to be installed on the host machine. Options to separate the setup and operation of the different environments: Optimally the cluster system could be maintained rather independently from the host system and vice versa.

2 Different Virtualization Concepts and Cluster Computing

Virtualization will be used to separate easily the cluster from the desktop environment. There are different concepts of virtualization around deploying different concepts. For the futher analysis the following virtualization techniques will be evaluated:

Full virtualization This refers to a technique, which allows to simultaneously run an unmodified guest operating system on the same hardware as the host system. Traditional X86 architecture is not virtualizable. The first implemented solution to this problem is **binary translation**. The instructions are analyzed and if required emulated on the fly.

Representatives of this type are **VMware Server and Virtual Box**. The first one was chosen because of the maturity of the product – VMware beeing among the pioneers of X86 virtualization and the installation is quite simple and it could be easily operated on a stateless client easing large scale roll-outs and tests. It does not require any further modification of the system kernel. VMware uses kernel modules which provide the necessary modifications to trap non-virtualizable instructions of the guest operating system.

Further testing involves creating of guests and exporting them via NFS or NBD to the workstations. First the exports are setup read-write, but later on it should be preferable to export read-only. VMware offers with the option '`independent-nonpersistent`' an elegant way to local guest file-system changes without modifying the common source of just on cluster node installation for all clients involved. VMware itself does not offer any means to control the share of compute power consumed, so it has to be done in the host system.

To overcome the shortcomings of the X86 architecture Intel and AMD introduced virtualization extensions to their CPUs. It is called **hardware-assisted virtualization**. These extensions implement besides the existing kernel and user mode, a root and non-root mode.

Kernel-based Virtual Machine (KVM) and **Xen 3** are two applications which use the virtualization extensions. KVM is part of the Linux kernel since version 2.6.20. It implements only an interface for the virtualization extensions, which can be used by other programs. The user-space part for this type of virtualization is a modified **QEMU**. Guest systems can be created with the command-line interface or with external programs. The complexity of getting KVM-QEMU to work in a stateless environment is about the same to the one of VMware Server which makes it a candidate for further analysis. But QEMU lacks CPU limitation and at the moment sound usability and stability.

Paravirtualization Another approach to the X86 architecture without virtualization support is paravirtualization. This technique requires a modification of the kernel of the host operating system and the virtualized kernel. The hypervisor is executed in kernel mode.The drawback of this technique is, that only open-source kernels can be para-virtualized.

Xen is the best-known paravirtualization for Linux at the moment and thus object in the evaluation field. Guest systems can be created via external programs or '*debootstrap*'. For the special environment of a stateless client a few changes have to be made, including early bridge configuration in case of using a network bridge. To finally boot Xen PXELinux with multiboot capabilities is needed.

Xen with bridge configuration seems to be the most complex solution. In stateless mode the bridge has to be configured very early, so that the network connection does not get interrupted later on which would freeze the client. The fact that the system kernel is loaded in ring 1 makes this solution even more complex. The load distribution options between the desktop hypervisor system and the cluster node guest are to be explored in depth.

Operating system-level virtualization This techniqe is also called jails. Which describes the character of this virtualization. For each operating system a container is created. Each container corresponds to a system partition and contains the user-space part. The kernel is modified to operate well with all user-space instances.

Applications using os-level virtualization are **Linux-VServer**, **Virtuozzo** or its open-source part **OpenVZ**. For the first tests Linux-VServer is chosen. Later other Products could be tested for advanced capabilities in limiting CPU and Memory. For the installation first a vanilla kernel is patched witch VServer extensions. These extensions and functionalities can be controlled via '*util-vserver*' tools.

3 Conclusion and Perspectives

This paper discussed how the several virtualization concepts could be brought to an average (stateless Linux) desktop workstation. This delivers the prerequisites for the evaluation of virtualization in cycle stealing scenarios which will follow in ongoing work. There are a number of cluster related issues to be discussed:

- The stability of the tools involved might be less proven than of a standard setup without virtualization. Nevertheless the user might restart his/her machine any moment.
- The period of uninterrupted operation might be shorter than on a cluster worker node. Calculate the economic impact: What is cheaper – deploying more dedicated cluster nodes (not only the direct costs of the machines, but the whole infrastructure of power supply and air conditioning, network sockets and rack space) should be taken into account.
- Compare virtualization mode to direct operation of the same machines: Are better results feasible if the machine is rebooted into cluster mode in off-hours.
- Could checkpoint/resume strategies improve the overall results: Recover from interupts faster or move whole virtual compute environments around from one machine to another.