Using virtual Lustre clients on the WAN for analysis of data from high energy physics experiments

D Bourilkov[1], P Avery[1], M Cheng[1], Y Fu[1], B Kim[1], J Palencia[2], R Budden[2], K Benninger[2], D Shrum[3] and J Wilgenbusch[3]

[1] University of Florida, Gainesville, Fl, USA
[2] Pittsburgh Supercomputing Center, Pittsburgh, PA, USA
[3] Florida State University, Tallahassee, Fl, USA

E-mail: bourilkov,avery,cheng,yfu,bockjoo@phys.ufl.edu
josephin,rbudden,benninge@psc.edu
dcshrum,wilgenbusch@fsu.edu

Abstract. We describe the work on creating system images of Lustre virtual clients in the ExTENCI project (Extending Science Through Enhanced National CyberInfrastructure), using several virtual technologies (Xen, VMware, VirtualBox, KVM). These virtual machines can be built at several levels, from a basic Linux installation (we use Scientific Linux 5 as an example), adding a Lustre client with Kerberos authentication, and up to complete clients including local or distributed (based on CernVM-FS) installations of the full CERN and project specific software stack for typical LHC experiments. The level, and size, of the images are determined by the users on demand. Various sites and individual users can just download and use them out of the box on Linux/UNIX, Windows and Mac OS X based hosts. We compare the performance of virtual clients with that of real physical systems for typical high energy physics applications like Monte Carlo simulations or analysis of data stored in ROOT trees.

1. Introduction

Today virtualization is moving in the mainstream of software deployment. Starting from virtualized servers and virtualized infrastructure, attention is growing as well on the client side. The provision of virtual images as appliances enables clients to use different operating systems and complete application suites on top of the same underlying host hardware. The ease and convenience of building and distributing virtual images allow users to concentrate their energy on the tasks at hand, without spending substantial time on installing, maintaining and updating software.

The Lustre parallel, distributed filesystem [1] is used in some of the world's largest and most complex computing environments. It provides high performance, scaling to tens of thousands of nodes and petabytes of storage with excellent I/O and metadata throughput. In ExTENCI we use Lustre 2 series [2] augmented with Kerberos authentication [3] to provide security and single sign-on over the wide area network (WAN). Combined with virtual clients, this is a powerful tandem for data analysis, well suited for the current trend towards big data in many fields.

2. Virtual Clients Test Bed

The ExTENCI project explores the use of virtual technologies to provide pre-built images which can be used by clients in high energy physics and many other applications. These clients can be “light-weight” or “rich”, combining a Linux OS with CERN and/or project specific software, and adding on demand access to large amounts of data through a distributed Lustre file system over the WAN.

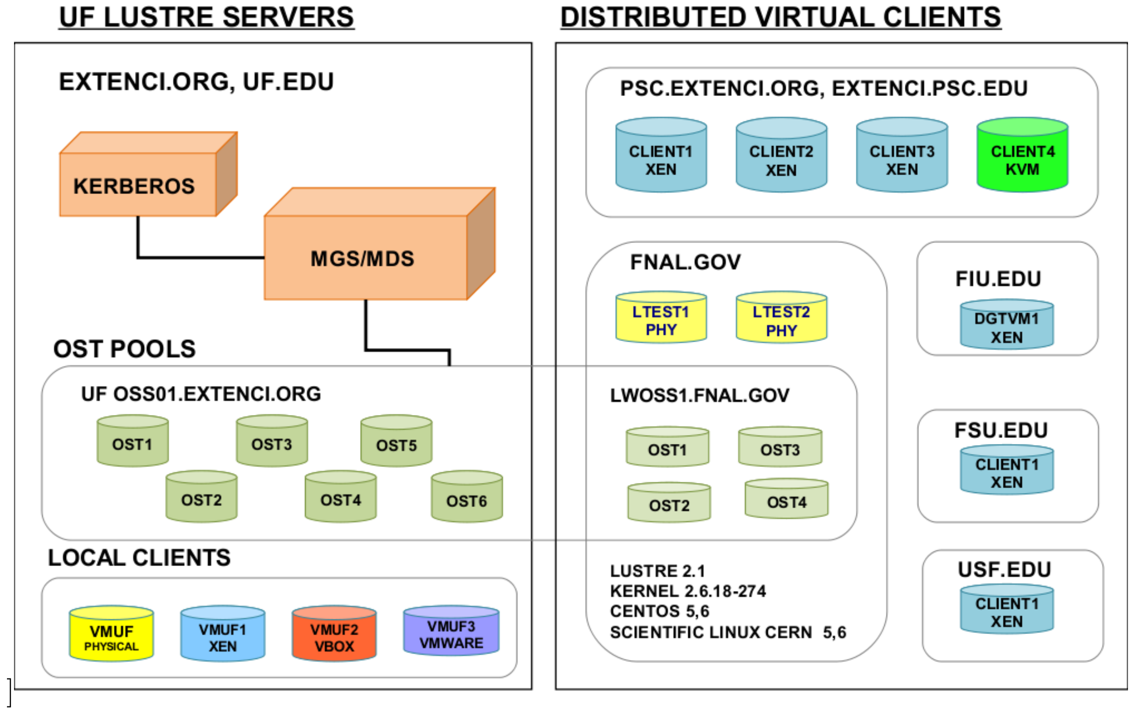

Figure 1. ExTENCI virtual clients test bed.

Figure 1 shows a detailed overview of the ExTENCI test infrastructure [4], including virtual clients which can access data on the Lustre WAN filesystem.

The Kerberos realm `EXTENCI.ORG` was established to create the secure Lustre network that only authorized systems and users can access. The University of Florida (UF) manages the Kerberos distribution center (KDC), the Lustre metadata server (MDS) and object storage server (OSS), and one pool of object storage targets (OST), while Fermilab handles another pool of OST. In addition to UF and Fermilab servers, virtual clients running various virtualization software - Xen [5], VMware [7], VirtualBox [8] and KVM [9]), are booted from pre-configured images by participating sites that securely mount the `/extenci` filesystem after being authorized and granted unique keytabs by UF. The Kerberos keytab file is an encrypted, local, on-disk copy of the host’s secret key. Any machine within a Kerberos domain must have a keytab file, called `/etc/krb5.keytab`, in order to authenticate to the Key Distribution Center (KDC).

System images pre-configured with Kerberos and Lustre (quota, ACLs) as well as the application software stack (CMS [10], ATLAS [11], CernVM-FS [12], ROOT [14]) make the setup and administration of the systems easier. We support various virtualization technologies including Xen, VirtualBox, VMplayer and KVM. The key features of these virtualization technologies are summarized in table 1. “Guest SMP” in this table specifies if the virtualized guest operating system supports symmetric multiprocessing (SMP) and is able to boot multiple virtual cores. “Para guest I/O” in the same table means that guests with paravirtualization bypass the emulation for disk and network I/O, thus giving improved performance. Paravirtualization is a virtualization technique introduced by Xen. In contrast to full virtualization in which an unmodified guest operating system runs like under a real machine, a paravirtualized guest kernel needs to be modified to be Xen-friendly. With paravirtualization, the guests are aware that the underlying environment is Xen hypervisor instead of real hardware. Paravirtualization bypasses the emulation for disk and network I/O, thus giving much better performance than full virtualization [6]. “Running snapshot” in the table means creating images from running instances of virtual guest systems. The various client images can be downloaded by users, and booted up to mount the filesystem. Currently, Lustre on a Xen client is more optimized than on VirtualBox, KVM and VMplayer. The host and client operating system details of the images used are given in table 2 and table 3.

Table 1. Key features of four virtualization technologies

	Xen	VBox	KVM	VMplayer
License	GPL	Oracle	GPL	VMware
Guest SMP	Yes	Yes	Yes	Yes
amd-V/intel-VT	Support	Support	Require	Support
Host OS	Modified kernel	Load modules	Load modules	Load modules
Guest OS	Modified kernel	Native	Native	Native
Guest I/O	Para	Emulation	Emulation	Emulation
GUI	No	Yes	No	Yes
Running snapshot	Yes	Yes	Yes	No
image format	raw	vdi	qcow2	vmdk

Table 2. Configurations of physical host

CPU	Quad-Core AMD Opteron(tm) Processor 2378
Cores	8
Memory	16 GB
OS	Centos 5
Kernel	2.6.18-274.18.1.el5 for VBox, KVM, and VMplayer 2.6.18-274.17.1.el5xen for Xen
Network	10 Gb/s

Table 3. Configurations of virtual guest

CPU	Quad-Core AMD Opteron(tm) Processor 2378
Cores	1
Memory	2 GB
OS	Centos/SL 5
kernel	2.6.18-274.17.1.el5xen

CernVM-FS [12], a caching, web-based, read-only filesystem has been added to the client’s software stack. This useful CERN tool is optimized to deliver the latest application software on-demand over the network using http and fuse [13] to mount the virtual filesystem. It verifies file checksums (SHA1) against the catalog obtained over https and can also be scaled up with additional squid caches.

The ROOT [14] data analysis package, developed at CERN, is an application responsible for the reading and writing of CERN LHC data stored in ROOT trees. It is fundamental to all of the LHC experiments and is called upon directly by CMS and ATLAS for I/O from their huge frameworks. It intelligently compresses and decompresses CMS or ATLAS data, stores them in trees and leaves, and can sort and place similar leaves from many events next to each other, thus increasing the packing efficiency and the resulting file density. It has many optimizable parameters (e.g. read-ahead) and is very fast reading a few variables from each event, but can slow down when reading complete events. Consequently, I/O can also become CPU–intensive depending on the occurrence and frequency of decompression/compression.

3. Performance Tests and Results

Our test bed is built around a physical machine configured as shown in table 2. It serves both as a Lustre physical client and the host for virtualization. To fairly compare performances of virtualization technologies, all virtual guests have the same configuration shown in table 3.

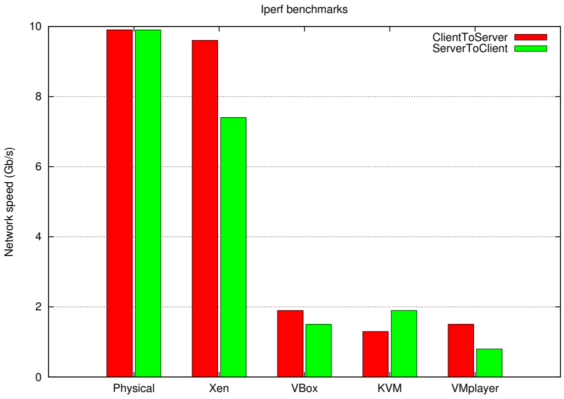

Figure 2. Iperf network tests for I/O performance of physical and virtual clients using different technologies.

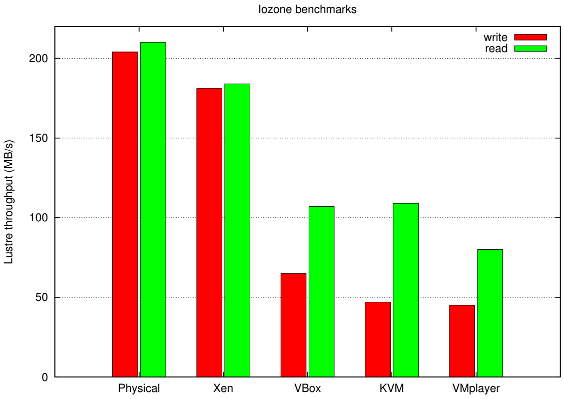

Figure 3. IOzone I/O rate tests for access to the Lustre file system of physical and virtual clients using different technologies.

The I/O performance of physical and virtual clients using different technologies is shown in figure 2 for Iperf [15] network tests, and in figure 3 for IOzone [16] tests to access the Lustre file system. Xen gives the best results in all cases, thanks to its unique paravirtualization bypassing device emulation.

To investigate Lustre scalability for multiple ROOT instances, we use ROOT 5.30 installed on the physical client to perform Lustre reads on a non-striped file (stored on one OST). The comparisons of reading ROOT trees with different branch/leave structures from local files or from the distributed Lustre file system are shown in figure 4 and 5. Customized ROOT files filled with random numbers are generated and read with a "thin" ROOT client ($\sim$ 4x10^3 lines of code). With the tuning of the ROOT file, full control of the data format is retained. Various data tree structures are tested (by varying the number of branches and leaves per branch) from the very simple to that closely resembling complex CMS data. I/O-intensive ROOT files are also designed to saturate the Lustre filesystem. Our findings indicate that in contrast to I/O performed on the local partition, Lustre exhibits very good scalability with increasing number of ROOT instances, as shown in figures 4 and 5. In the first test (ROOT tree with two branches and only two leaves per branch, and file size of 20 GB filled with random numbers), the scalability is

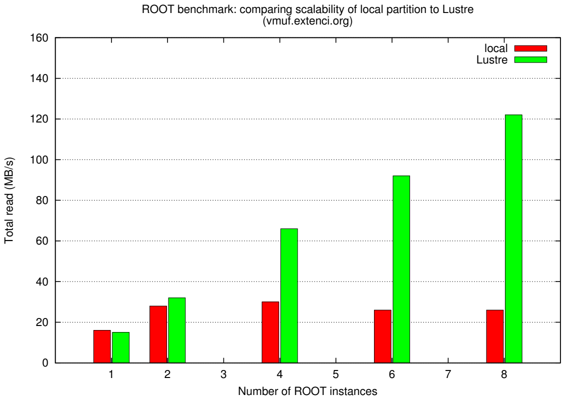

Figure 4. Perfect linear scalability with Lustre for increasing number of ROOT instances using a ROOT file with 2 branches, 2 leaves per branch and 20 GB of random numbers.

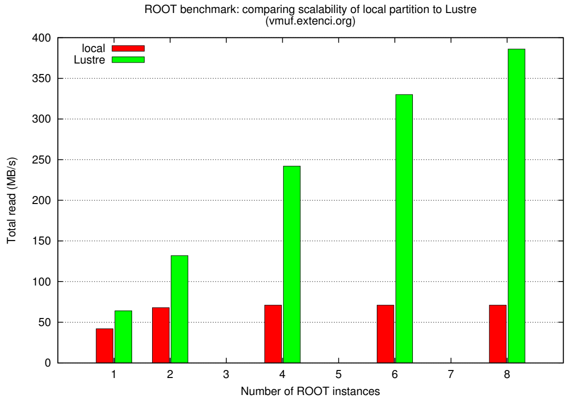

Figure 5. Lustre scalability of multiple ROOT instances using a ROOT file with 20 branches, 5000 leaves per branch and 5 GB of random numbers.

linear, but the rates are not very high (15 MB/s for one client). In the second test, the first root instance benchmarked at a decent 64 MB/s, the second at close to twice that rate and so forth. Still, in this case (the ROOT file corresponding to 20 branches and 5000 leaves per branch) we do not observe perfect linear scaling with Lustre, with the rate saturating around 400 MB/s for 8 instances.

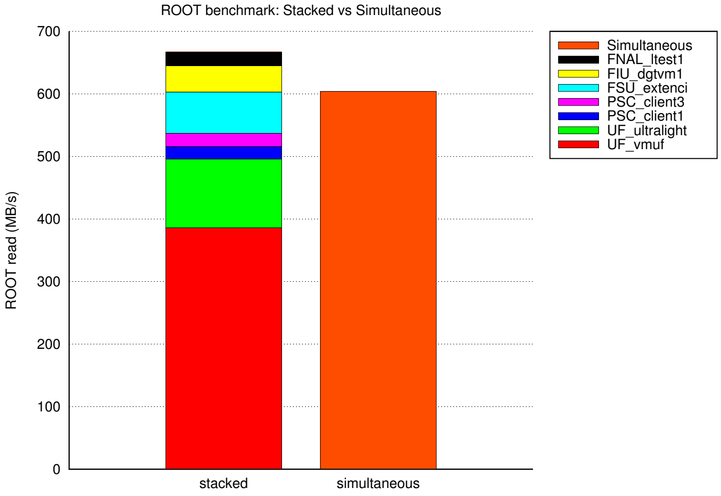

Figure 6. Runs with multiple ROOT instances at several sites in parallel to test the scalability of the distributed Lustre filesystem.

To test the global ROOT Lustre scalability of our system we have performed runs with a realistic mix of clients accessing data over the LAN and the WAN. First we ran sets of tests sequentially (stacked) in order to determine the best performance of our system for different client/server combinations. Then we ran all clients in parallel. As shown in figure 6, the simultaneous run gives I/O rates close to the stacked rates. The simultaneous I/O throughput on the OSS storage is benchmarked by Collectl [17] when all the clients are running in parallel. This proves that Lustre gives scalable total I/O throughput to multiple and increasing number of clients.

CernVM-FS is a network file system based on http and optimized to deliver experiment software. CernVM-FS provides complete CMS and ATLAS software installations, enabling the

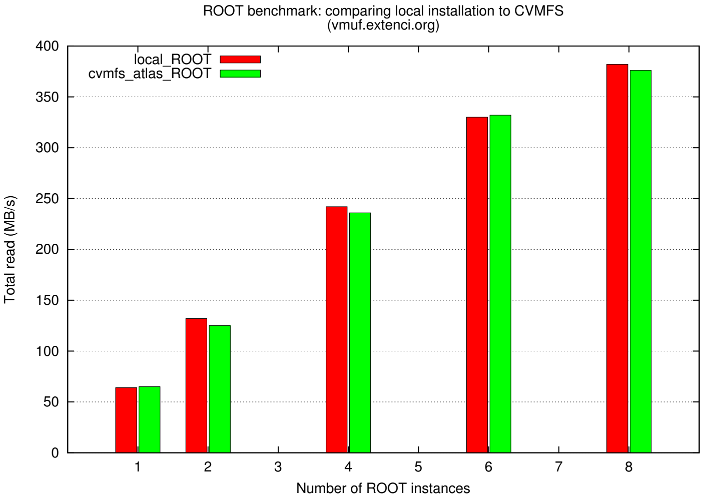

Figure 7. Runs with multiple ROOT instances to test of the scalability of the distributed Lustre filesystem. ROOT taken from CernVM-FS has similar performance as local ROOT, both achieving close to 400 MB/s for 8 ROOT instances.

building of light virtual clients. The comparison of the performance of ROOT installed locally, or taken from CernVM-FS, is shown in figure 7. We find that the performance of applications using CernVM-FS is close to that of the locally installed software after the initial cache population, with minimal overhead.

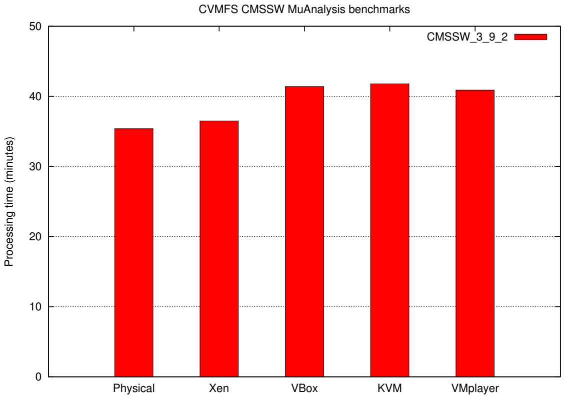

Figure 8. Run times for CMS analysis of muon data with CMSSW taken from CernVM-FS, using different virtual images. Shorter times are better.

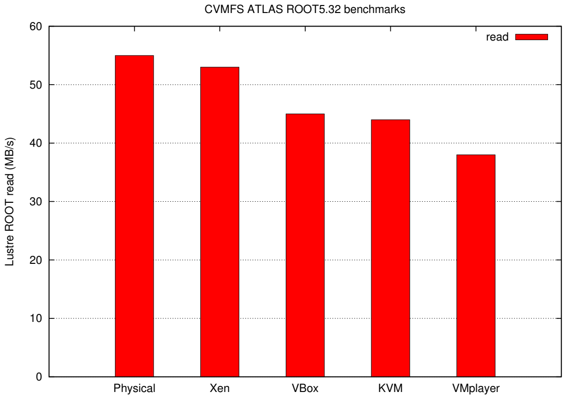

Figure 9. I/O rates for a typical ATLAS application taken from CernVM-FS, using different virtual images. Simulated data are read.

Examples of runs with virtualized CMS and ATLAS applications are shown in figures 8 and 9, and compared with runs on similarly configured physical hosts (1 core with memory 2GB). As the applications are CPU intensive, the difference in I/O rates is less noticeable. Still we observe the best virtual performance with Xen.

4. Outlook

We have successfully built and deployed virtual images which can be used “out-of-the-box” by clients in high energy physics and many other fields. We have combined virtual clients with a Lustre file system, distributed over the WAN, to provide ease of access to large volumes of data directly from the user desktop or laptop. We have seen encouraging results when performing scalability tests for the I/O rates of virtual clients. The users have to be aware that different virtual technologies can have substantial differences in I/O performance.

Acknowledgments

We acknowledge the funding from the National Science Foundation, NSF Proposal 1007115 ExTENCI with the OSG. DB and MC thank Dave Dykstra for help with installing the CernVM-FS.

References

[1] The Lustre filesystem. Available at `http://www.lustre.org` .
[2] The Lustre 2 filesystem. Available at `http://wiki.whamcloud.com/display/PUB/documentation` .
[3] MIT Kerberos. Available at `http://web.mit.edu/kerberos` .
[4] Palencia J *et al.* “Using Kerberized Lustre over the WAN for High Energy Physics Data,” *Proceedings of the Lustre Users Group 2012 Meeting*, Austin, TX, April 2012. Available at `http://www.opensfs.org/wp-content/uploads/2011/11/lug2012-v20.pdf` .
[5] Xen virtualization. Available at `http://xen.org/` .
[6] Xen paravirtualization. Available at `http://wiki.xen.org/wiki/Xen_Overview` .
[7] VMware player virtualization. Available at `http://www.vmware.com/products/player` .
[8] VirtualBox virtualization. Available at `https://www.virtualbox.org/` .
[9] KVM virtualization. Available at `http://www.linux-kvm.org/page/Main_Page` .
[10] The CMS experiment. Available at `http://cms.web.cern.ch/content/cms-physics` .
[11] The ATLAS experiment. Available at `http://www.atlas.ch` .
[12] The CernVM-FS filesystem. Available at `http://cernvm.cern.ch/portal/techinfo` .
[13] Filesystem in Userspace (Fuse). Available at `http://fuse.sourceforge.net/` .
[14] The ROOT analysis framework. Available at `http://root.cern.ch` .
[15] Iperf network benchmark tool. Available at `http://openmaniak.com/iperf.php` .
[16] IOzone filesystem benchmark tool. Available at `http://www.iozone.org/` .
[17] Collectl: a benchmarking and monitoring tool. Available at `http://collectl.sourceforge.net/` .