PROCEEDINGS OF SCIENCE

# Investigation of Storage Options for Scientific Computing on Grid and Cloud Facilities

**Gabriele Garzoglio**[1]
*Fermi National Accelerator Laboratory, P.O. Box 500, Batavia, IL, 60510, USA*
*E-mail:* garzoglio@fnal.gov

**Keith Chadwick**[2]
*E-mail:* chadwick@fnal.gov

**Ted Hesselroth**
*E-mail:* tdh@fnal.gov

**Andrew Norman**
*E-mail:* anorman@fnal.gov

**Denis Perevalov**
*E-mail:* denis@fnal.gov

**Doug Strain**
*E-mail:* dstrain@fnal.gov

**Steve Timm**
*E-mail:* timm@fnal.gov

In recent years, several new storage technologies, such as Lustre, Hadoop, and BlueArc, have emerged. While several groups have run benchmarks to characterize them under a variety of configurations, more work is needed to evaluate these technologies for the use cases of scientific computing on Grid clusters and Cloud facilities. This paper discusses our evaluation of the technologies as deployed on a test bed at FermiCloud, one of the Fermilab infrastructure-as-a-service Cloud facilities. The test bed consists of 4 server-class nodes with 40 TB of disk space and up to 50 virtual machine clients, some running on the storage server nodes themselves. With this configuration, the evaluation compares the performance of some of these technologies when deployed on virtual machines and on "bare metal". In addition to running standard benchmarks such as IOZone to check the sanity of our installation, we have run I/O intensive tests using experiment specific applications. This paper presents how the storage solutions perform in a variety of realistic use cases of scientific computing.

*The International Symposium on Grids and Clouds and the Open Grid Forum*
*Academia Sinica, Taipei, Taiwan*
*March 19 - 25, 2011*

[1] All authors from the same address
[2] Speaker

 

## 1.Introduction

In recent years, several new storage solutions have become of interest to the Grid and Cloud community. In order to familiarize ourselves with the features available with these new technologies, the Grid and Cloud Computing department of the Fermilab Computing Division has started an evaluation of four storage technologies for the user cases of data intensive science on Grid and Cloud resources. The evaluation so far has focused on Lustre [1] with a future comparison to Hadoop [2], BlueArc [3], and OrangeFS [4]. This evaluation is managed as a collaborative project across several departments of the Computing Division., including the Data Movement and Storage department [5] One of the focuses of our investigation is to compare storage performance when the server environment is installed on a "bare metal" machine or is virtualized.

The evaluation method includes an initial study of typical High Energy Physics storage access patterns [6], in order to set the expected required scale for the technologies under evaluation. The core evaluation includes measurements of performance and fault tolerance, as well as observations on operational qualities of each storage system. In particular, the project performs measurements of performance following three different methodologies:

1. *Standard benchmarks*: we use standard storage benchmarks, mainly IOZone [7], to measure aggregate bandwidth to storage and metadata access performance;
2. *Application-based benchmarks*: we use the offline framework of the NOνA experiment to measure the storage performance with a root-based application.
3. *HEPiX Storage Group benchmarks*: in collaboration with the HEPiX Storage Working Group [8], we have integrated the NOvA computing framework with the HEPiX storage test bed. The HEPiX group will use this application to evaluate storage technologies not covered under our investigation (e.g. AFS/VILU, NFS4, GPFS, etc.). From preliminary results, the solution that allows clients to read the maximum number of events within a time window of 14 minutes is AFS/VILU. AFS caches client requests using Lustre on the back end, a configuration not evaluated by this project.

The project has currently completed the evaluation of Lustre and has preliminary results for Hadoop and BlueArc.

In sec. 2 we describe our test bed. We discuss the results for Lustre from the standard benchmarks in sec. 3 and from the application-based benchmarks in sec. 4. We present a few considerations on operations for Lustre in sec. 5. We present preliminary results from Hadoop and BlueArc in sec. 6. We conclude with a summary of our findings for Lustre in sec. 7 and with acknowledgments in sec. 8.

## 2.Test Bed Specifications

To evaluate a storage technology we first install and commission it on four machines of the FermiCloud cluster. These are server class machines with dual quad core CPU Xeon E5640 2.67GHz with 12 MB of cache and 24 GB of RAM. Each machine has 6 SATA disks, 2 TB

each, arranged as a RAID 5 and 2 additional system disks. Basic disk benchmarks (hdparm) on the raided disks show a maximum access bandwidth of 377 MB/sec. Each machine is provided with a 1 GBit Ethernet and an InfiniBand interface, the latter currently not used by the project.

For clients, we use 7 machines of the FermiGrid Integration Test Bed cluster (FG ITB) with dual quad core CPU Xeon X5355 2.66GHz with 4 MB cache and 16 GB RAM. Each machine has a single 500 GB disk (hdparm speed 76 MB/s) and a 1 GBit Ethernet interface. Each of them hosts 3 XEN virtual machines, each using 2 cores and 2 GB of RAM, for a total of 21 client virtual machines.

We tested Lustre 1.8.3 with 1 Metadata Target (MDT) and 3 Object Storage Targets (OST) with different striping settings. Fig. 1 shows a diagram of the two configurations tested: "bare metal" and "virtual" Lustre server.

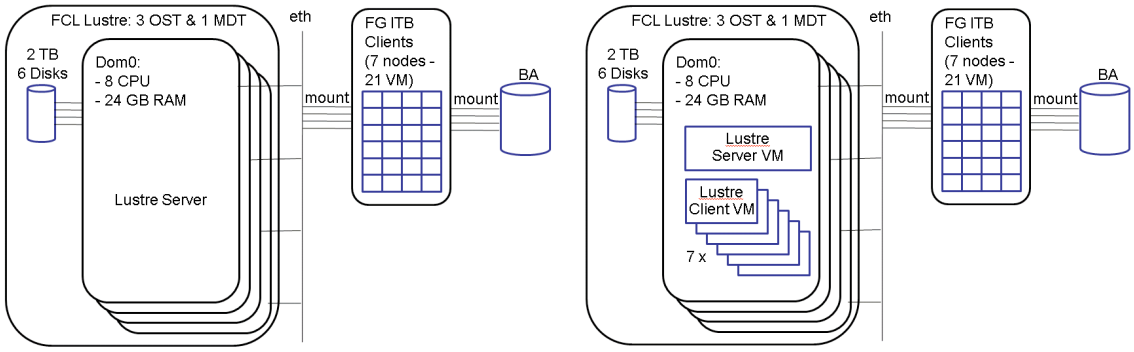


*Fig. 1: Two diagrams of the Lustre test bed.* ***On the left****, the Lustre servers are installed on "bare metal".* ***On the right****, the Lustre servers are installed on virtual machines; each "bare metal" machine hosts also 7 client virtual machines. For both configurations, 21 client virtual machines are instantiated at the FG ITB cluster, each mounting a BluArc (BA) disk with the software, home areas, etc.*

The "virtual" Lustre server configuration is of interest for small deployments, where it is expected that data can be served with a fraction of the 8 cores available to each servers. In these cases, the additional cores can be used by instantiating "client" virtual machines. It should be noted that this configuration can be achieved only by running the Lustre server also on a virtual machine. In fact, running the Lustre server on "bare metal" does not allow this configuration because the Lustre kernel, necessary to run the server, does not support virtualization.

## 3. Results from Standard Benchmarks

We use standard benchmarks to measure the performance of Lustre v1.8.3 on metadata operations and different patterns of read / write access. We focus on the latter in this paper.

### 3.1 Data Access Measurements

We have used the IOZone test suite to measure the read/write performance of Lustre in the configurations described in sec. 2. We configured IOZone to write and read 2 GB files from 3 to 48 clients on 3 virtual machines.

#### 3.1.1 ITB clients vs. "bare metal" Lustre

For these tests, the Lustre server was installed on "bare metal" at the FermiCloud resources (FCL), while the clients were run from FG ITB virtual machines, as shown in fig.1

left plot. The data was not striped i.e. whole files resided on any one of the three OSTs. Results are shown in fig. 3.

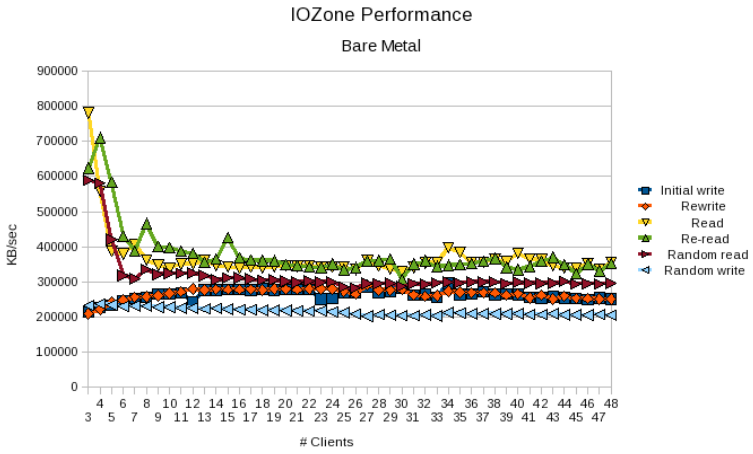


*Fig. 3: IOZone plots of ITB clients vs. Lustre on "bare metal" for different data access patterns.*

The results of approximately 350 MB/s for read and of 250 MB/s for write form a baseline for the comparison with all other configurations.

### 3.1.2 ITB clients vs. "virtual" Lustre

These tests compare the performance of Lustre servers deployed on "bare metal" with respect to Lustre servers deployed on virtual machines. For these we used KVM virtual machines with 1 CPU core and 3 GB of RAM and different IO drivers to optimize performance, as shown in fig. 4.

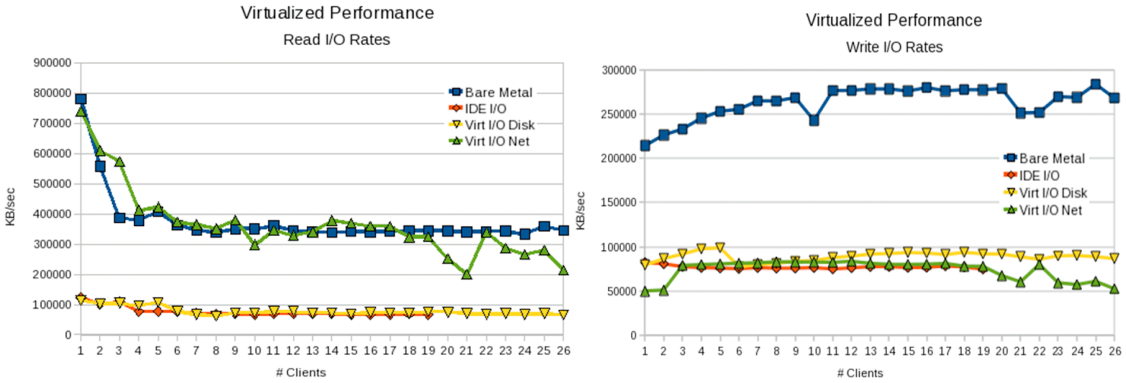


*Fig. 4: IOZone tests of Lustre servers on KVM virtual machines with different disk and network IO drivers. The measurements are performed with 1 to 26 clients on 3 FG ITB virtual machines.*

The results show that the use of network VirtIO drivers is crucial to optimize performance. The read speed of 350 MB/s is comparable to the speed of Lustre servers deployed on bare metal. Despite the use of VirtIO drivers, however, the optimal write speed of 70 MB/s is 4 times slower than for Lustre on bare metal. The results also show that the use of the default IDE drivers or of disk VirtIO drivers has little impact on the performance optimization.

In order to improve the write speed, we have attempted to allocate more cores and memory to the virtual machines running the Lustre servers. These tests did not show any improvement in performance, since the server process is not CPU bound.

### 3.1.3 FCL clients vs. "virtual" Lustre

These tests compare the performance of different client configurations accessing a Lustre server deployed on virtual machines. In particular, we compare the performance of clients ("on-board") running on the same host machine as the Luster server (FCL cluster) with respect to clients ("external") running on a different cluster (FG ITB). Fig 6 shows the measurements.

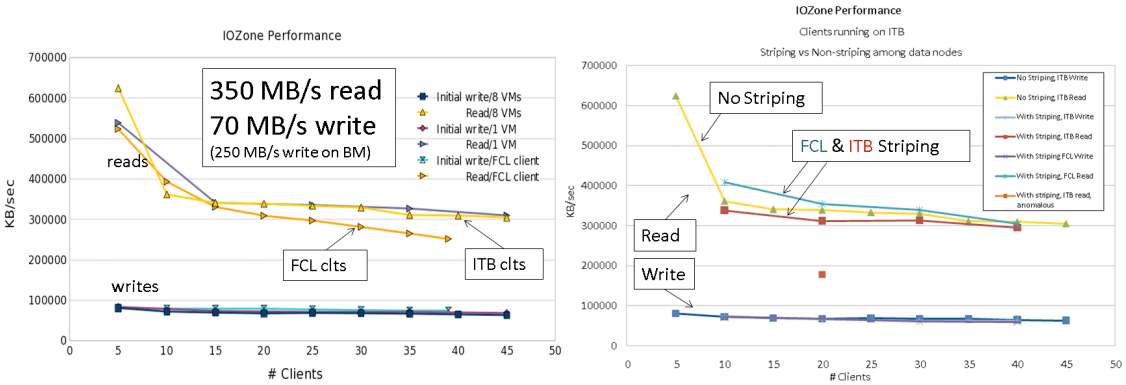


*Fig. 6: IOZone plots of 1 to 45 clients on 3 virtual machines for Lustre deployed on virtual machines.* ***Left Plot****: this plot compares the performance of "on-board" (FCL) vs. "external" (ITB) clients.* ***Right Plot****: this plot compares the performance of clients on FCL and ITB clusters with and without file striping (4MB stripes).*

The left plot shows that read rates are about 15% lower for "on-board" (FCL) clients with respect to "external" (ITB) clients. As discussed in sec. 4, this effect is reversed when using application-based benchmarks. Write performance is not affected by the client configuration. The number of virtual machines running on the same host that runs the Lustre server does not affect the measurement for ITB clients. The measurements were taken with only the Lustre server virtual machine ("1 VM" in the plot) or with the Lustre server and seven client virtual machines ("8 VM").

The right plot shows the effects of file striping on read and write performance for FCL and ITB clients. For a small number of clients, FCL clients have slightly higher read rates with file striping as compared to ITB clients (with or without file striping). This difference, however, tends to disappear for a larger number of clients. Write rates are not affected by the striping configuration.

## 4. Application-based Benchmarks

We tested the performance of Lustre using real High-Energy Physics analysis applications. We developed two test benchmarks, one based on "Ana", the framework of the NOvA experiment at the time, another based on "Loon", the framework of the MINOS experiment. Both frameworks are based on root. The applications read a fraction of each event as fast as possible (for NOvA about 50%), effectively simulating an event "skimming" application. We access 10 files of 1.5 GB in size in random order and, before starting each test, we clear the client buffer cache. We run one client per machine and measure utilized bandwidth by looking at the amount of data transferred from the statistics of the client network interface.

### 4.1 Write Bandwidth Measurements

While using two different frameworks was good to introduce diversity in our tests, we soon realized the Loon-based benchmark was always CPU intensive. MINOS data, in fact, are kept compressed on disk to optimize storage space and the resulting typical read / write bandwidth for 21 concurrent clients was 3 MB/s. The benchmark based on Loon, therefore, was not stressing IO access to storage and was not a very good gauge of Lustre performance.

Similarly, our Ana-based benchmarks for write bandwidth were always CPU bound, mainly due to the allocation of root data structures. The measured read/write bandwidth to storage was also about 3 MB/s. Note that this effect is not present when using the benchmark to only read data, simulating the extreme case of a skimming application that does not select any events for its output.

### 4.2 Lustre on "Bare Metal" vs. Lustre on Virtual Machines.

Focusing on the read bandwidth for the Ana-based benchmark, we measured 15.6 ± 0.2 MB/s for a single client for Lustre on "bare metal", and 15.3 ± 0.1 MB/s for Lustre servers deployed on virtual machines. The two results are clearly compatible.

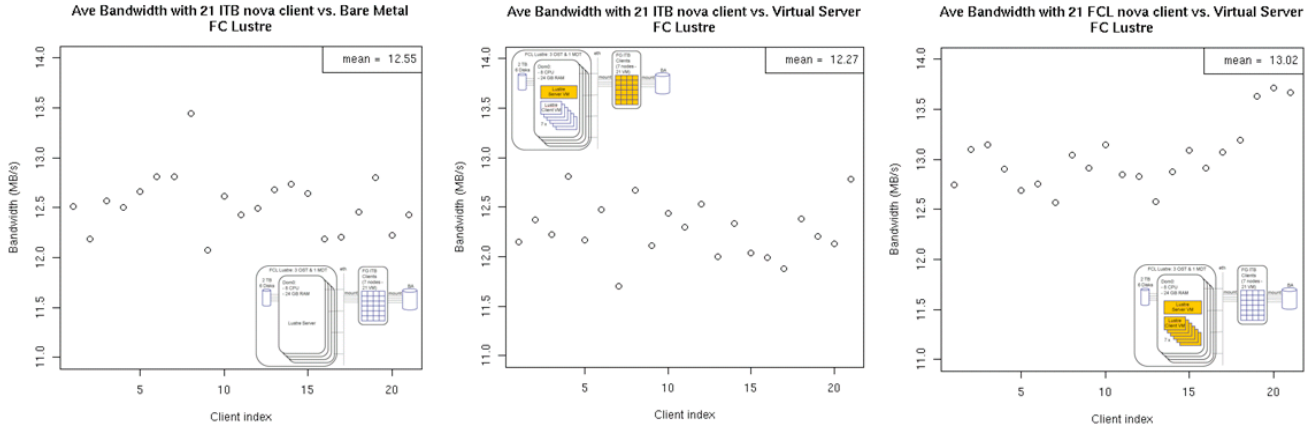


*Fig. 7: Read bandwidth for 21concurrent NOvA clients, for non-striped Lustre in different configurations. The **first plot** to the left shows the performance for Lustre on bare metal; the **second plot,** for Lustre on virtual machines with clients from the FG ITB cluster; the **third plot,** for Lustre on virtual servers and "on-board" clients (FCL NOvA clients).*

Using 21 clients, the average read bandwidth per client becomes smaller, as shows in figure 7. Note that the aggregate bandwidth is about 270 MB/s, slightly below the saturation of the three 1 GBit network interfaces to the OST servers. For Lustre on "bare metal", we measure 12.55 ± 0.06 MB/s, very similar to 12.27 ± 0.08 MB/s for Lustre on virtual machines and clients on the FG ITB cluster. For "on-board" clients, we measure 13.02 ± 0.05 MB/s, a performance only slightly better. Comparing these last two configurations, it should be noted that the IOZone tests did better for clients on the FG ITB, a result opposite to this one.

For our root-based benchmark, we conclude that Lustre on "bare metal" has an equivalent read performance as Lustre on virtual machines, for an aggregate client bandwidth slightly below the network saturation of the 3 OSTs. For "on-board" clients, the performance is slightly better.

### 4.3 More Consistent Bandwidth with File Striping

We have studied the effects of file striping on read bandwidth. Figure 8 shows our measurements with file striping (4 MB per stripe) for 21 clients accessing Lustre on Virtual machines. These plots should be compared with the two rightmost plots of figure 7. There, the configuration is exactly the same, except that striping is turned off.

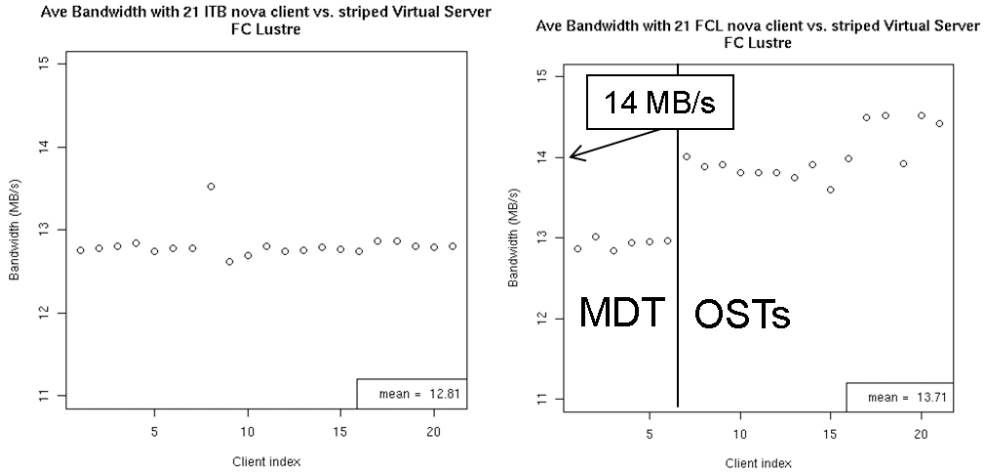


*Fig. 8: measurements of bandwidth with file striping (4MB per stripe) for Lustre on virtual machines.* ***Left plot****: 21 clients on the FG ITB clusters.* ***Right plot****: 21 clients "on-board". Note the difference in bandwidth when the clients are on the same node that runs the MDT or the OSTs.*

The overall results are very similar to the ones of figure 7. When clients are on the FG ITB, we measure a read bandwidth of 12.81 ± 0.01 MB/s with file striping vs. 12.27 ± 0.08 MB/s when file striping is turned off. Similarly, when clients are "on-board", we measure in average 13.71 ± 0.03 MB/s with file striping vs. 13.02 ± 0.05 MB/s when file striping is turned off. There are 2 features worth noting. First, with file striping the bandwidth measurements are very similar among all clients i.e. each client has a more consistent access to bandwidth. Second, for "on-board" clients running on the same nodes as the Lustre OSTs (where the data is), we measure an increase in access bandwidth of about 1 MB/s as compare to clients running on the node hosting the MDT or on the FG ITB cluster. In this case, there is a positive effect on bandwidth due to the fact that all files have some stripes on a disk of the same physical machine as the clients.

### 4.4 Fairness in the Distribution of Bandwidth

We have tested access to bandwidth of 49 clients for Lustre on virtual machines. In this regime, there is contention for resources. With 49 clients, the three 1 GBit network cards to the OSTs should saturate (single clients can retrieve data at 15 MB/s) and there are almost three clients per spindle (18 disk spindles overall).

In this condition, we ask whether the system is fair in distributing the available limited bandwidth to clients. In a fair system, we expect for all clients running concurrently to receive approximately the same amount of bandwidth, finishing almost at the same time.

Figure 9 shows that the last client takes almost twice as long (1.77 times) to finish as the fastest client, with almost a flat distribution of times between the two. The bandwidth

distribution is also almost flat between 8 and 11 MB/s, about a 15% spread (up and down) around the average.

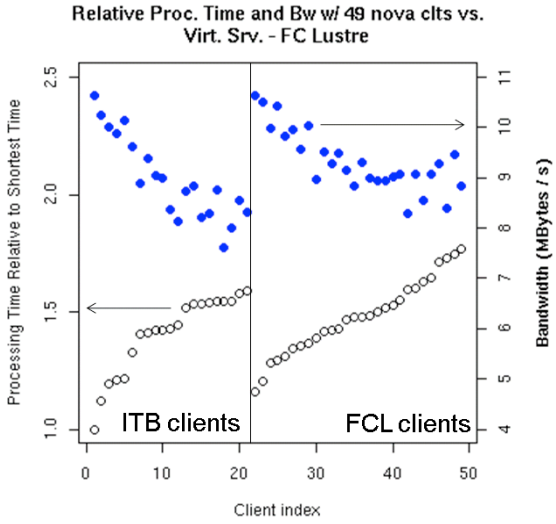


*Fig. 9: Measurements of bandwidth (right axis) and completion time relative to first client to finish (left axis) with 49 clients. The bandwidth has a distribution of about 15% around the average.*

The system does not make a distinction between "on-board" and "external" clients. The average measurements of time spread and bandwidth between the two types of clients are compatible. For ITB clients, the average completion time factor with respect to the fastest client is 1.41 ± 0.04 and the average bandwidth 9.0 ± 0.2 MB/s; for on board clients the values are respectively 1.48 ± 0.03 and 9.3 ± 0.1 MB/s.

We conclude that the system is fair in the distribution of the available bandwidth within 15% with no distinction between "on-board" vs. external clients.

## 5.Operational Considerations

Using a test bed to evaluate a technology for a short time is not the ideal environment to uncover the operational issues that arise when running in production. In any case, our limited observations on the operational properties of Lustre are the following.

All configuration parameters are accessible through luster commands and are typically encoded in the luster partition. Some are available as parameters in the */proc* file system directory, even if the manual recommends to access them through the command line interface. This indirect access to reading and changing configuration parameters (as opposed to e.g. editing a file) makes it more difficult for the non-expert to diagnose and debug problems.

We found that some errors in the log are easy to understand and point to the source of the problem. Other errors are incomprehensible to the non-expert, with long error codes and references to the source code.

In our tests we also observed the effects of the read-ahead algorithm of Lustre. The root-based benchmark from NOvA reads about 50% of each event. In our tests of Hadoop and BlueArc, we see that only about 50% of each file is transferred through the network interface, as expected. That number is 85% for Lustre. The default configuration of Lustre increases the read ahead size from 1 MB to 40 MB with linear increments when two or more sequential client

reads fail to be satisfied by the Linux buffer. In this case, the read ahead algorithm transfers portions of each event that, in the end, are not used. In a production environment, optimizations of these parameters should be attempted considering the layout of the event on disk (e.g. how root branches are clustered).

## 6. Status and Future Work

The storage investigation project has currently finished the evaluation of Lustre in "bare metal" and "virtual" configurations. It is now focusing on making a similar evaluation for Hadoop and BlueArc.

The preliminary results of Hadoop and BlueArc show very good performance using IOZone benchmarks. For Hadoop, we deploy one name node and three data nodes on "bare metal", similarly to Lustre, and we make data available to clients through a semi-POSIX Fuse interface. The fact that not all POSIX interfaces are implemented make the use of Hadoop not best suited for interactive computing of for general purpose jobs. We measure a bandwidth of 150 –290 MB/s for read and of 110 – 290MB/s for write, depending on the number of file replicas (bandwidth increases when the number of replicas decreases). For BlueArc, we use the laboratory production deployment and mount the file system through NFS. We measure a bandwidth of 300 MB/s for read and of 330 MB/s for write. As shown above, remember that for Lustre on "bare metal" we measured 350 MB/s and 250 MBs respectively.

Using our application-based benchmarks show similar performance for Hadoop and Lustre. For 21 concurrent client reads, we measure a bandwidth of 7.9 ± 0.1 MB/s for Hadoop and 8.15 ± 0.03 MB/s for BlueArc, compared to 12.55 ± 0.06 MB/s for Lustre. It should be noted that these benchmarks, based on root, cannot write directly to Hadoop because of a lack of required POSIX interfaces.

## 7. Conclusions

The storage investigation project at Fermilab is evaluating storage technologies for the use cases of High-Energy Physics stakeholders running on Grid and Cloud facilities. The project has finished the study of Lustre and is evaluating Hadoop and BlueArc, before studying Orange FS.

With regards to the performance of Lustre, we observe that Lustre servers on virtual machines allow reads at the same speed as Lustre on "bare metal", but writes are three times slower. The use of VirtIO drivers is necessary but does not make up the gap. The HEP applications tested do not have high demands for write bandwidth; therefore, the flexibility of the virtual server may be valuable and access bandwidth adequate.

For the "virtual" Lustre configuration, "on-board" and "external" clients have the same performance, within 15%. We also observe that data striping increases bandwidth stability to clients but has minimal (5%) impact on read and none on write bandwidth. Also, for a large number of clients, the fairness of bandwidth distribution is within 15%. It should be noted that Lustre read ahead algorithm reads 85% of data, as opposed to the necessary 50% by design, as

Hadoop and BlueArc. Data on additional storage solutions will become available to our community through our partnership with the HEPiX Storage group.

Our evaluation of fault tolerance for Lustre shows that the non-blocking fail-over policy provides graceful performance degradation as expected. The switch between fail-over and fail-out policies, however, must be done with care to avoid the risk of losing the data. In our case, we could not recover from the de-synchronization of MDT and OST statuses.

Our observations on the operational properties of Lustre are limited. We observe that some log errors are easy to understand and some very hard for non-experts. In addition, the need to access the configuration through Lustre commands makes problem diagnosis and debugging not immediate for non-experts.

Our preliminary evaluation of BlueArc and Hadoop (for a small number of file replicas) shows results comparable to Lustre from our standard benchmarks. Both storage technologies have about 66% the read bandwidth performance of Lustre from application-based benchmarks, instead.

## 8. Acknowledgments

We thank Andrei Maslennikov from the HEPiX storage group; Robert Hatcher and Art Kreymer from MINOS; Neha Sharma for her support on FermiCloud; Alex Kulyavtsev and Amitoj Singh for the consulting time on Lustre. This work is supported by the U.S. Department of Energy under contract No. DE-AC02-07CH11359

## References

[1] Lustre web site. Accessed on Apr 2011. http://www.lustre.org

[2] B. Bockelman, “Using Hadoop as a grid storage element”, 2009 J. Phys.: Conf. Ser. 180 012047 doi: 10.1088/1742-6596/180/1/012047

[3] Blue Arc web site. Accessed on Apr 2011. http://www.bluearc.com/

[4] Orange File System project web site. Accessed on Apr 2011. http://www.orangefs.org

[5] A. Kulyavtsev, M. Crawford, S. Fuess, D. Holmgren, D. Litvintsev, A. Moibenko, S. Naymola, G. Oleynik, T. Perelmutov, D. Petravick, V. Podstavkov, R. Rechenmacher, N. Seenu, J. Simone, “Lustre File System Evaluation at FNAL”. Proceedings of CHEP'09, Prague, March 23, 2009

[6] G. Garzoglio, “A study of data access patterns for the DZero and CDF experiments at Fermilab”. Links accessed on Apr 2011: DZero: http://home.fnal.gov/~garzogli/storage/dzero-sam-file-access.html CDF: http://home.fnal.gov/~garzogli/storage/cdf-sam-file-access-per-app-family.html

[7] IOzone File System benchmarks web site. Accessed on Apr 2011. http://www.iozone.org/

[8] The HEPiX Storage Wokring Group. Accessed on Apr 2011. http://w3.hepix.org/storage/