Data Transfer Study for HPSS Archiving

James R. Wynne III
Oak Ridge National Laboratory
OLCF
1 Bethel Valley Road
Oak Ridge, TN 37830
wynnejr@ornl.gov

Suzanne T. Parete-Koon
Oak Ridge National Laboratory
OLCF
1 Bethel Valley Road
Oak Ridge, TN 37830
paretekoonst@ornl.gov

Quinn D. Mitchell
Oak Ridge National Laboratory
OLCF
1 Bethel Valley Road
Oak Ridge, TN 37830
mitchellqd@ornl.gov

Stanley White
Oak Ridge National Laboratory
OLCF
1 Bethel Valley Road
Oak Ridge, TN 37830
whiters@ornl.gov

Tom Barron
Oak Ridge National Laboratory
OLCF
1 Bethel Valley Road
Oak Ridge, TN 37830
tbarron@ornl.gov

***Abstract*—The movement of the large amounts of data produced by codes run in a High Performance Computing (HPC) environment can be a bottleneck for project workflows. To balance filesystem capacity and performance requirements, HPC centers enforce data management policies to purge old files to make room for new computation and analysis results. Users at Oak Ridge Leadership Computing Facility (OLCF) and many other HPC user facilities must archive data to avoid data loss during purges, therefore the time associated with data movement for archiving is something that all users must consider. This study observed the difference in transfer speed from the originating location on the Lustre filesystem to the more permanent High Performance Storage System (HPSS). The tests were done with a number of different transfer methods for files that spanned a variety of sizes and compositions that reflect OLCF user data. This data will be used to help users of Titan and other Cray supercomputers plan their workflow and data transfers so that they are most efficient for their project. We will also discuss best practice for maintaining data at shared user facilities.**

***Keywords*-data transfer; HPSS; Data archiving, workflow**

I. Introduction

Archiving data to the High Perfomance Storage System (HPSS) is a crucial step for any project in an HPC setting. Most centers employ a filesystem purge that is used to keep the filesystem from filling up and becoming unusable. Having a workflow that includes archiving important data to the HPSS helps to bypass this possibly catastrophic purge. In the coming year at least one OLCF project will begin running an application that will generate data at the rate of 1 PB per day. There are currently several projects, for example see, [1], whose overall output is greater than a petabyte. With data of this size, it is important to know how fast large files can be archived because the data must leave the filesystem expediently to keep the filesystem operational.

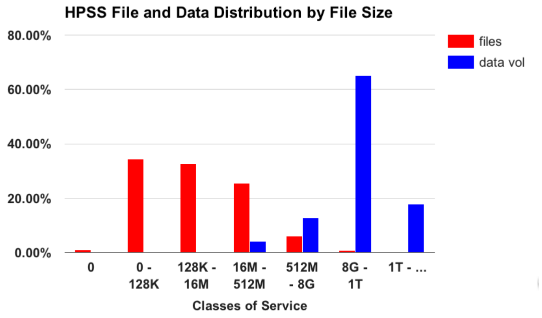

Figure 1. The current file size distribution on OLCF's HPSS

OLCF recently added a new Class of Service for the HPSS to help users efficiently store larger files. This Class of Service makes use of a Redundant Array of Independent Tapes (RAIT), so that the data is distributed over several storage tapes with enough redundancy that it can be recovered in the event of tape failure from surviving tapes. The old method of data redundancy was to make dual copies of the data. RAIT will allow more efficient use of the tape storage without sacrificing data integrity. The current size distribution of files in HPSS is shown in Figure 1. The red bars indicate the number of files in each size bucket and the blue bars indicate the aggregate size of all the files in each bucket. The buckets are sized to correspond with the Classes of Service discussed in Section IV. From the graph, we can see that while most of the files are relatively small (under 512M), the larger files (over 512M) account for most of the data. In the future, as machine memory and simulation sizes grow, we expect that average file size will increase as will the amount of data stored in larger files relative to that stored in smaller files. We wanted to gather information help users decide what would be the fastest and most user-friendly way for them to archive data on the HPSS. What is fastest and easiest is relative to each user's needs. For example, some users may archive overnight, so speed would be less of a factor for them than the reliability of the method to be scripted and left to run without manual intervention. Thus, our sample set of representative user data consisted of files ranging in size from 100KB to 1.1TB. This choice was also motivated so that the tests would trigger each of the different Classes Of Service on the HPSS, which determine how the data is ingested. In this set, both directories of files and equivalently sized single files were also tested. Tens of tests were run for each case on each of the transfer platforms to factor in the variability caused by the shared user system. In this study we also tested the ingest rate, from the OLCF file system to the HPSS disk system for the new RAIT Class of Storage. We choose this step, rather than the transfer from HPSS disk to HPSS tape, because it is the step that the users see, where as the subsequent tape transfer time is hidden from the user. There are several factors that determine the speed of transfer to the HPSS at OLCF. Three chief factors are the speed of reading the data on the Lustre filesystem, the speed of moving the data over the network and the speed of writing that same data on the disk systems of the HPSS. Typically user facilities have different teams of systems administrators to maintain the center filesystems, network and data archive. While these teams tune their systems to work well together, the work is usually done as benchmarking under ideal conditions, with no other users on the systems. Regular testing from the user perspective under normal loads can help find additional optimization for the collective function of the systems as our tests did in this study. At OLCF the HSI and HTAR transfer tools are used to interact with the HPSS. OLCF offers users three types of transfer platforms specialized for interactive and scripted transfers with the HPSS using these tools. The first type is the HSI Transfer Agent, triggered from the Titan external login nodes. It automatically utilizes multiple data transfer nodes to move a single file or set of files more efficiently. The second type of data transfer platform is the interactive data transfer nodes, which allow several users to utilize a single data transfer node at once. To offer an option that eliminates user contention, a set of batch scheduled data transfer nodes that allow a single user to utilize the entire node is provided as a third option. Our tests were run on each of these systems under normal loads. We will share our performance results and the pros and cons from a user perspective of each of these transfer platforms. We will also discuss best practice for maintaining data at shared user facilities.

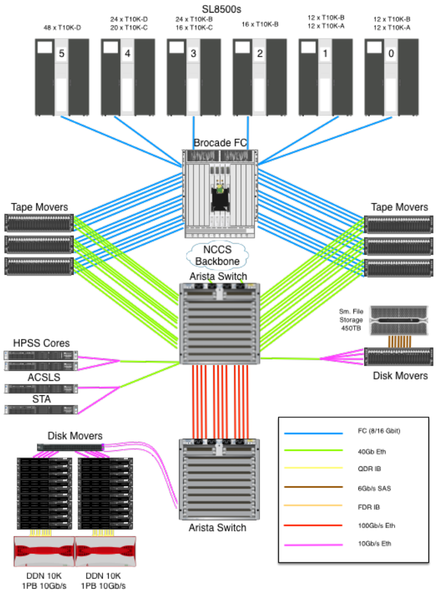

Figure 2. HPSS Layout

II. Transfer Tools

A. HSI

The Hierarchical Storage Interface, or HSI, was developed by Gleicher Enterprises for use with HPSS. HSI provides a FTP like environment to transfer local files from Lustre or other filesystems to the HPSS namespace. The HSI tool is most commonly used for a single file per transfer. [2].

B. HTAR

Gleicher Enterprises' HTAR is used when the number of files to be transferred is too large to use HSI. When HTAR is invoked on the commandline, it will attempt to bundle the specified directory of files into a single tar file. The usage of this tool emulates Unix's tar command. HTAR will also create an index file that is transferred with the tar archive to HPSS. This index file holds data that is used to know the offset of each of the tarred files into the tar archive. The HTAR tool was originally designed to be used with several tens of thousands of files that are less than three MB in size each [3]. However HTAR use is limited to individual files that are less than 64 GB each. This limitation in size is because HTAR is POSIX compliant so it has an upper limit of 64 GB per file.

III. OLCF Infrastructure Serving the HPSS

A. The HPSS Layout

HPSS is an Hierarchical Storage Management system, or HSM. The OLCF HPSS infrastructure consists of many systems, and has the ability to store data across many different layers of technology and speed. As data is used less frequently it migrates to deeper and slower levels of storage and as data is requested by the user, it is brought back to faster tiers, configurable via system policies.

The filesystem metadata is stored and distributed across three NetApp MDT 2600 disk arrays which are Fibre connected to the HPSS core server.

The core server is the brain of the entire HPSS, directing and orchestrating the other parts of the system and keeping track using a DB2 database to maintain all of the location and configuration metadata to manage the HPSS systems and the stored data. Stored data is moved to or from disk with a disk mover, and tape with a tape mover. When the core server requests a move due to a user or request, the core will direct the mover to carry out the operation. As the HSI gateway server, Disk movers, and/or Tape movers move data, the Core server is updated with the status and location of data in the system. The Core Server then enters the metadata into DB2. Each file ingested by the HPSS has an associated Class of Service (COS). The HPSS's Core Server determines the appropriate COS based upon the size of the file, see Table I. The COS that the file triggers will control how the file will be transferred and stored in HPSS. HPSS also has a storage hierarchy consisting of multiple levels of storage, each representing a different storage class (SC). Thus the COS are used to sort data into the levels in the storage hierarchy that are best suited for its size.

Disk movers are connected to a NetApp 5500 disk array for small files and extra small files. The NetApp arrays store the small COS files for the excellent performance with small file transfers. All medium COS and larger files are stored on Data Direct Network (DDN) 10K arrays, for the large file streaming capabilities. There is currently about 2.5 PB of disk across the NetApp and DDN arrays for ingest and tape staging.

Tape movers are each Fibre Channel (FC) attached to a subset of Oracle Enterprise class tape drives, T10K(A,B,C,D), in one of the six Oracle SL8500 tape library complex. There are currently 60,000 tape slots across the SL8500 complex and a number of T10K technology drives: 72 T10KD, 36 T10KC, 64 T10KB and 24 T10KA. Each library is interconnected with its nearest neighbor and all are configured full frame. Each drive is connected to the FC Brocade Director switch, and zoned to their respective Tape movers. Because of the number of devices and physical separation of the HPSS system two Arista switches are utilized, interconnected at 1.2 Tb/s. Tape movers and Disk movers are interconnected over 10Gb or 40Gb Ethernet through the Arista HPSS backbone. All Data Transfer Nodes, Titan external transfer nodes, and HPSS HSI Gateway systems sit outside the HPSS core infrastructure in the National Center for Computational Sciences (NCCS) network space which is connected back to the Arista HPSS backbone at 80 Gb/s. Other auxiliary systems reside within the HPSS infrastructure to facilitate HPSS or tape functions, such as Oracle management and analytics systems.

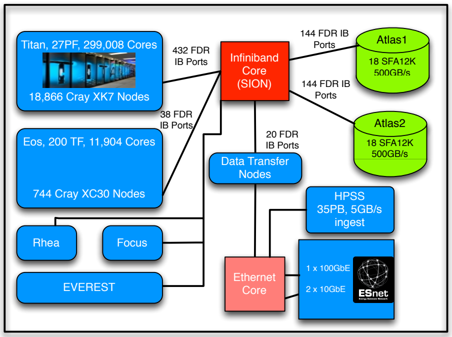

Figure 3. Compute, storage, and networking systems at OLCF in 2014. [4]

B. The Transfer Agent

The HSI Transfer Agent is a part of the Gleicher Enterprises suite of tools. Our configuration leverages a group of seven Data Transfer Nodes (DTNs) that exist in the NCCS network. These DTNs are triggered by HSI transfers originating only from the external Titan login nodes where the Class of Service (COS) associated with the transfer is Large, X-Large, or XX-Large. Any HTAR transfer, or any HSI transfer that does not trigger the Transfer Agent from the external Titan login nodes will be transferred over the 1Gb/s or 10Gb/s network link on the node. The Transfer Agent will split the data between the Transfer Agent's DTNs and send it in parallel. If the data being transferred triggers the Large COS, only two threads are launched on the Transfer Agent to parallelize the transfer, else if the COS is X-Large or above, HSI will employ eight threads. By using this parallel transfer technique the time taken to transfer the data to the HPSS will theoretically decrease drastically.

C. The Data Transfer Nodes

The DTNs are divided by function. Two nodes are dedicated for interactive use, ten are set up for batch-scheduled transfers, and seven are reserved for the HSI Transfer Agent. The two interactive nodes are the most heavily used and the Ethernet interfaces are often fully utilized. The scheduled data transfer nodes are intended to give users a platform that is free from on-node contention for long running transfers to the HPSS and remote sites. The scheduled nodes can easily be integrated with workflows on Titan and the analysis clusters since they share the same batch scheduling system. With this division of transfer nodes by function, OLCF users experience less contention for these nodes and the hardware is configurable to the needed capacity of each function.

As shown in figure 3, one side of the data transfer nodes faces the OLCF's Infiniband fabric, named SION (Scalable I/O Network), and the other side connects to the OLCF Ethernet backbone. SION's main purpose is to provide the highest possible I/O bandwidth between Titan and the Lustre

COS Name	Lower size boundary	Upper size boundary
X-Small	0 KB	128 KB
Small	128 KB	16 MB
Medium	16 MB	512 MB
Large	512 MB	8 GB
X-Large	8 GB	1 TB
XX-Large	1 TB	256 TB

Table I
THE FILE SIZE RANGES FOR EACH COS ON THE HPSS

COS	HSI Size	HTAR SIZE
X-Small	100 KB	10 10 KB files
Small	1 MB	10 100 KB files
Medium	500 MB	10 50 MB files
Large	1 GB	10 100 MB files
X-Large	64 GB	64 1 GB files
XX-Large	1.1 TB	18 61 GB files

Table II
FILE SIZES USED

filesystems with 1TB/s of aggregate capacity between the two. The HPSS is linked directly to the OLCF Ethernet backbone with a theoretical maximum ingest rate of five Gb/s. The DTNs are currently equipped with a 10Gb/s Ethernet interface and OLCF is in the process of upgrading the DTNs with 40Gb network interfaces.

IV. STUDY DESIGN

As described in section III-A, each file that is ingested by HPSS has an associated COS. The HPSS's gateway determines the appropriate COS based upon the size of the file as well as the usage characteristics. Table 1 shows the breakdown for the six primary COS at OLCF. All data sizes are given in powers of 1024. The six COS allow HPSS to match the file with the data storage target that will be the most efficient for that file. For example, storing the larger files on the DDN disks instead of the NetApp disks because the DDNs perform better for larger files.

To ensure that all six COS were tested, the sizes of the test cases were 100KB, 1MB, 500MB, 1GB, 64GB, and 1.1TB as shown in Table II. Files in both test sets were timed from each of the available transfer platforms by using HSI and HTAR. Since HTAR specializes in transferring directories of smaller files, the test case for the transfers using HTAR were directories of smaller files whose sum is equal to the target size for the COS. Therefore, the test cases for the HSI based transfers were single files of the appropriate size for each COS. Note: because of the 1Gb/s link on the Titan login node versus a 10Gb/s link on the DTNs, HTAR was not tested on the Titan login nodes.

During this study, an XX-Large COS was implemented to help store files larger than 1 TB that are expected to be used in the future. The introduction of this new COS prompted the design of a new round of testing that would compare the speeds of the new XX-Large COS with the previous largest X-Large COS. To do this, the same 1.1TB file from the XX-Large tests was used, but the target COS was changed by a flag in HSI and HTAR. This allowed the test to override the default COS (XX-Large) and used the X-Large COS instead.

COS	File Size	Interactive DTN	Scheduled DTN
X-Small	128 KB	0.01±0.002s	0.01±0.003s
Small	1 MB	0.02±0.01s	0.01±0.003s
Medium	500 MB	2.70±0.01s	0.05±0.01s

Table III
HSI AVERAGE TIME STANDARD DEVIATION FOR THE THREE SMALLEST TEST FILES.

COS	Dir. Size	Interactive DTN	Scheduled DTN
X-Small	128 KB	0.05±0.01s	0.02±0.006s
Small	1 MB	0.03±0.01s	0.01±0.002s
Medium	500 MB	2.84±0.63s	3.86±3.90s

Table IV
HTAR AVERAGE TIMES WITH STANDARD DEVIATION FOR THE THREE SMALLEST TEST DIRECTORIES

Users are interested in both the time to archive a file and the rate of transfer, so both are shown. Rates were calculated based on the transfer time and the size of the files in binary Megabytes. Because we are testing on shared resources on a production system there is variance in our sample. We wanted to capture the variance but also give numbers that represent the most commonly occurring times and rates for our transfer tests. For our study we display the average time with the sample standard deviation. The sample standard deviation here is meant to be a gauge of the variability of the transfer performance.

Times that differed from the average by more than 3 times the standard deviation were excluded. The general form of our trials had the majority of the data points within a sample with similar values. However, most tests contained data points that, while less than 3 standard deviations different than the average, were still very different from the mode. Thus for the rates associated with the timing data, we choose to show the median rather than the average for each test, as it is a better representation of the most frequently occurring values.

V. STUDY RESULTS

A. Small Files

The Transfer Agent is not triggered for files in the size ranges of X-Small, Small, and Medium COSes, so we discuss the times for only the interactive and scheduled DTN. As shown by the standard deviations given with the average times in tables III and IV for the X-Small, Small, and Medium files there is large variation in the rates for the trials regardless of where the transfer occurred or which transfer tool was used. These transfers occur over such short time intervals that they are susceptible to transient contentions for resources on all of the systems involved

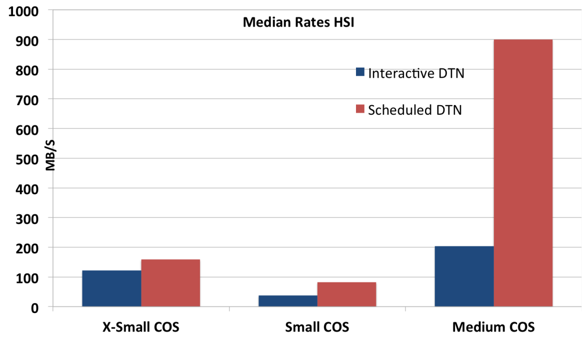

Figure 4. Median rates for HSI transfers form the interactive DTN and the Scheduled DTN.

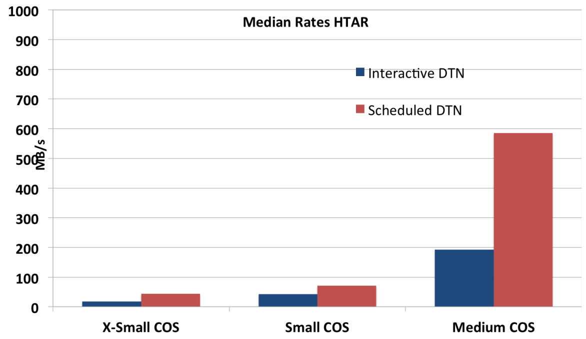

Figure 5. Median rates for HTAR transfers form the interactive DTN and the Scheduled DTN.

in the transfer. The Medium COS tests on the scheduled DTN experienced the largest variation for the small files. The most extreme case was the Medium COS HTAR sample on the scheduled DTN, where the mean transfer time was 3.6s, but the median time was only 1.12s. The values for individual data points in this sample ranged from 0.5s to 9.4s and, while there were more low values than high, the data was well spread over the range. No data points met the three standard deviations criterion for exclusion. Running the same test a few days later yielded an average time of 0.8s and a median time of 0.9s. There was little spread in the values of the data. This case illustrates how transient contention for the resources the systems can impact performance. It also illustrates how difficult it is to give users a meaningful estimate of the performance of a shared system.

Figures 4 and 5 show the rates associated with the times given in tables III and IV. The scheduled DTN is shown to be a faster platform for launching transfers than the interactive DTN across all files tested in this sample. This is not surprising because the scheduler eliminates the on-node contention that is common on the interactive DTNs due to simultaneous use from multiple users. It is unclear why the scheduled DTN Medium COS tests show so much of an over-performance relative to the tests on the interactive DTN. It is healthy to keep in mind that all of these samples have a great deal of variance and a much larger sample should be gathered over time to get the best estimate of expected performance. Even with a good estimate, the variance is such that a particular transfer might be significantly faster or slower than the estimate would indicate. Even a good estimate is not a guarantee.

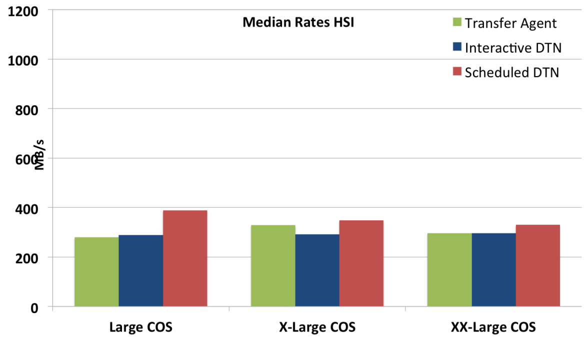

Figure 6. Median rates for HSI transfers before tuning the Transfer Agent.

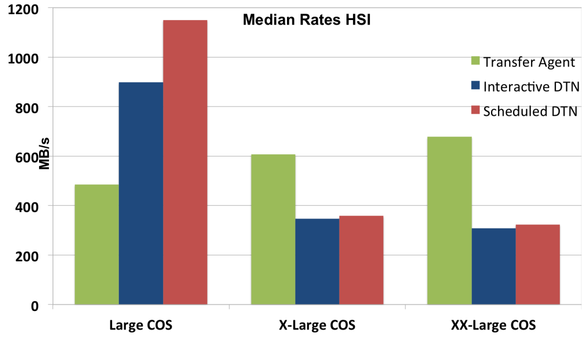

Figure 7. Median rates for HSI transfers after tuning the Transfer Agent.

B. *Large Files and the Transfer Agent*

The Transfer Agent is triggered for HSI transfer using the largest three COSs. Tables V and VI show the transfer times for the three largest COS tests before and after the tuning that was done on the Transfer Agent during this study. Figures 6, and 7 show the median rates associated with the before and after timing data. Even though the tuning only impacted the Transfer Agent, tests using the scheduled and nteractive DTNs were re-run for the after tests to give a more consistent reference for the Lustre and HPSS condition on the day of the after tests. The ”before” tests show that Transfer Agent did not give a performance advantage over the DTNS as is clear in figure 6. This test prompted an investigation of the HSI Transfer Agent’s configuration.

We tested to see if the issue was in the Lustre read, the HPSS disk write, or in the combination of the two. To test the Lustre response, a 10 GB file was copied to /dev/null, a device file that discards all data. To test the combination, this 10 GB file was transferred to HPSS using HSI. To test the HPSS separate from Lustre, the 10 GB file was pulled

COS	File Size	Transfer Agent	Interactive DTN	Scheduled DTN
Large	1GB	4.20±0.002s	3.62±0.93s	4.51±4.02s
X-Large	64GB	199.36±0.36s	267.40±65s	187.12±4.9s
XX-Large	1.1TB	3545.33±16s	3520.45±0.151s	3177.73±17s

Table V
HSI AVERAGE TRANSFER TIMES WITH STANDARD DEVIATION BEFORE TRANSFER AGENT TUNING

COS	File Size	Transfer Agent	Interactive DTN	Scheduled DTN
Large	1 GB	2.11 ±0.36s	1.32±0.65s	1.99±3.0s
X-Large	64 GB	101.89±25s	190.00±7.6s	183.67±11s
XX-Large	1.1 TB	1605.27±171s	3399.88±118s	3230.26±160s

Table VI
HSI AVERAGE TRANSFER TIMES WITH STANDARD DEVIATION AFTER TRANSFER AGENT TUNING

	Lustre	HPSS	Combination
Transfer Agent	17s	5s	13s
IDTN	12s	15s	9s

Table VII
TRANSFER COMPONENT TEST

off the HPSS into /dev/null. While this test is a read rather than a write, it gives us a measure of the HPSS response. We ran these three tests on the Transfer Agent and on the interactive DTN for reference. Ten sets of these tests were run over the course of a day. Table VII shows the average timings for each of these tests.

For interactive DTN, there was little difference between the speeds of the two components and the HSI using both of them. In the case of the Transfer Agent, the speed without Lustre was three times faster than the same test on the interactive DTN. The speed of the combined system for HSI with the Transfer Agent was closer to the speed of Lustre. This indicated that there could be an issue with the tuning of the Transfer Agent for Lustre. Tuning tests showed that the cause of the poor performance was due to the I/O block size used in the HSI Transfer Agent. Lustre's block size is set at 1MB; tuning the HSI Transfer agent to better ingest a 1 MB block size lead to an boost in transfer speed. Once this configuration adjustment was made, the full set of tests were re-run.

Table VI shows that after the tuning the times for HSI transfers decreased by at least half relative to the before test. This table also shows that the reference tests on the scheduled and interactive DTNs run on the same day, were still faster than the transfer agent. In fact, all transfer rates for the large COS and the X-Large COS were better for the test shown in 7 compared the tests shown in 6. The improvement on the scheduled and interactive DTNs is not related to the tuning of the transfer agent, as they are separate systems. We used the most utilized Lustre File system, Atlas1, for all the tests except the XX-Large COS tests, which were run on the less utilized Lustre filesystems, Atlas 2. We chose this because we did not want to further burden Atlas1 with the 1.1TB files used in the XX-Large tests. Between the before and after tests users did a voluntary removal of files from their Lustre projects spaces on Atals1. Thus, when the after tests ran, Atlas 1 Lustre was less full and may have been performing better. Atlas 2 had little change in utilization between the tests, so it is not surprising that our before and after reference tests for the XX-Large COS are more consistent.

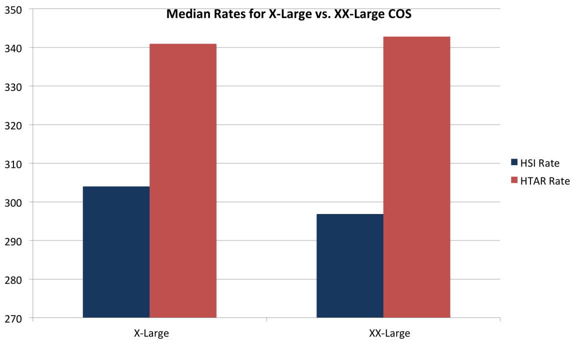

Figure 8. Median rates for transferring the same file in both the X-Large and XX-Large COS

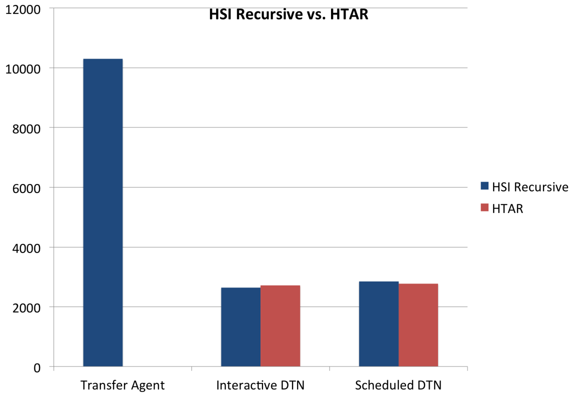

Figure 9. Median rates for recursive HSI of 1.1 TB directory and HTAR of the same directory

In tables V and VI the standard deviation for the large

COS	File Size	Interactive DTN	Scheduled DTN
Large	1 GB	1.23±0.69s	1.17±0.72s
X-Large	64 GB	180.32±5.68s	158.97±8.27s
XX-Large	1.1 TB	3108.32±44.04s	3062.14±74.45s

Table VIII
HTAR AVERAGE TRANSFER TIMES WITH STANDARD DEVIATION

1	Transfer Agent	DTN	Scheduled DTN
Time (s)	814.80±4.25s	3177.54±50.4s	2944.58±50.4s

Table IX
RECURSIVE HSI OF 1.1TB DIRECTORY

COS timing data from the scheduled DTN is comparable or larger than the mean. This indicates that the spread in the data values between individual points was very large for these trials. This could be due to transient contention during the trials. The scheduled DTNs are free from on-node contention from other users on the DTN itself, but this is only one source of many possible sources of contention. A larger sample of tests will be necessary to develop a better understanding of the expected the range of rates for the large COS tests.

The tuning helped the rate of transfer for the Transfer Agent for X-Large and XX-Large as is evidenced how much the Transfer Agent over performs the DTNs in the reference tests run on the same day. The large COS only uses two threads on the Transfer Agent, rather than the eight that are triggered by the X-large and XX-Large COSes. Further tuning and tests of the Transfer Agent may be warranted for the large COS.

Our next tests were motivated by the new guidance we give users to bundle files into archives of 1 TB or greater to take advantage of the RAIT redundancy that comes automatically with the XX-Large COS. The test set was the same as the HTAR tests for the XX-Large COS: a directory of 18 64 GB files. We wanted to know if users would be taking a loss in transfer speed by doing the bundling. Since RAIT will likely be available for the three largest COSes we compare all three. For the HTAR transfer tests there is little difference in the performance of the transfer on the scheduled and interactive DTNs, as can be seen in figure 9. There is also little difference in the time required for the 1.1 TB directory transfer with HTAR and transfer with a recursive HSI unless the Transfer Agent is used. For the 1.1 TB file the recursive HSI with the Transfer Agent is nearly 2X faster than the fastest HTAR transfer. HTAR does not currently trigger the Transfer Agent, so there is no parallel transfer option for large directories that will give the advantage of RAIT, unless all the files in the directory are large enough to trigger the XX-Large COS.

After the XX-Large COS was implemented, tests were run to see if there were a performance difference between the XX-Large COS and the X-Large COS. As you can see in figure 8, the times were roughly the same. From the ingestion side of HPSS, the only difference in the COSes is that the smaller COSes use NetApps for the disk cache whereas the larger COSes use the DDNs. Any variance in time can be explained by contention within Lustre or the DTNs.

VI. CONCLUSIONS AND BEST PRACTICES

A. Observations

For this study, the variation of the data was large, especially for the transfers that could complete in less than 10 seconds. This illustrates how difficult it is to accurately inform users about the particular expected rate of performance of a shared system. The key takeaway should not be a single set of expected rates, but rather a range of reasonable rates. Based on our tests, for large files across all platforms a transfer rate between 200 MB/s and 600 MB/s is reasonable. The fastest way to put on the HPSS was using the HSI transfer using the Transfer Agent, especially for COSes X-Large and XX-Large. For the largest file tests this can be up to two times faster than the same transfer on the scheduled DTNs. The scheduled DTNs showed a speed advantage over the interactive DTNs in most trials. They are also on the same batch scheduler as our primary resources so users can automatically trigger the file archive from the batch script that creates that data.

B. Recommendations for Transfers

If speed is the primary concern for larger files and directories, the Transfer Agent is the best tool and a recursive HSI of a directory is the fastest method on the Transfer Agent. However unless all the files in the directory are grater than 1 TB, a recursive HSI will not trigger the XX-Large COS and will not yield the automatic redundancy of RAIT. Users should bundle their data into files larger than 1 TB to get the data security of RAIT whenever possible. If all the component files are less that 64 GB, HTAR, from one of the scheduled DTN will save users the time needed to tar the files in a separate step. The scheduled DTNs should be used when automatic archiving at the end of a data generation or analysis job is needed.

C. Recommendations for Testing

As was shown in this study, testing from the user perspective may reveal additional optimization for tunable systems. Therefore we recommend that tests are run under the same conditions experienced by users at least once a month. This allows better statistics to build the expected performance range and ensure that user perspective troubleshooting is occurring at a regular interval. Future work may include implementing this and reporting on the results of a yearlong study. Coupling these tests with tests of Lustre will lead to better understanding of the performance and variance.

VII. Acknowledgment

We would like to acknowledge Mitch Griffith, Jason Hill, and Jason Anderson for helpful discussion about the HPSS and HSI. This research used resources of the Oak Ridge Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725.

References

[1] K. Heitmann, N. Frontiere, C. Sewell, S. Habib, A. Pope, H. Finkel, S. Rizzi, J. Insley, S. Bhattacharya ”The Q Continuum Simulation: Harnessing the Power of GPU Accelerated Supercomputers”, arXiv:1411.3396

[2] ”Gleicher Enterprises.” HSI. Web. 15 Apr. 2015. http://www.mgleicher.us/index.html/hsi/.

[3] ”Gleicher Enterprises.” HTAR. Web. 15 Apr. 2015. http://www.mgleicher.us/index.html/htar/.

[4] S. Parete-Koon, B. Caldwell, S. Canon, E. Dart, J. Hick, J. Hill, C. Layton, D. Pelfrey, G. Shipman, D. Skinner, H.A. Nam, J. Wells, J. Zurawski ”HPC’s Pivot to Data” Proceedings of CUG 2014, Lugano, Switzerland – (2014)