I/O characterization of large-scale HPC centers

Benchmarking Working Group
Open Scalable File Systems Inc.

Open Scalable File Systems, Inc. (OpenSFS) is a strong and growing nonprofit organization. OpenSFS was founded in 2010 to advance Lustre development, ensuring it remains vendor-neutral, open, and free. Since its inception, OpenSFS has been responsible for advancing the Lustre file system and delivering new releases on behalf of the open source community. Through working groups, events, and ongoing funding initiatives, OpenSFS harnesses the power of collaborative development to fuel innovation and growth of the Lustre file system worldwide.

The OpenSFS Benchmarking Working Group (BWG) was created with the intent of defining an I/O benchmark suite to satisfy the requirements of the scalable parallel file system users and facilities. The first step toward this end was identified as characterization of I/O workloads, from small- to very large-scale parallel file systems, deployed at various high-performance and parallel computing (HPC) facilities and institutions. The characterization will then drive the design of the I/O benchmarks that emulate these workloads.

As part of the characterization, the BWG released a survey at the Supercomputing Conference in 2012, to solicit participation and collect data on file systems and workloads in HPC centers. This paper summarizes the data collected and our analysis.

The survey

The survey has been drafted with the intent of collecting data on production parallel file systems, the compute partitions served by these, HPC centers or institutions running these systems, and the applications using these file systems. Most of the questions focused on file systems, including storage and interconnect, and on the workloads, including representative applications and science domains. Relevant questions of the survey are listed in Table 1.

Table 1: selected survey questions.

Site	Name	**Storage**	Technology and Interfaces
	Affiliation		Interconnect type and topology
	Activity	**Usage**	Size total/distribution (e.g. by file)
	Number of users overall		Purpose (e.g. checkpointing)
	Number of systems	**Applications**	Name
File System	Type and version		Science domain
	Format		IO patterns (e.g. random, read/write, small/large blocks, checkpoint)
	Number of client nodes/cores		
	Environment (e.g. virtualized)		Metadata operations
	Connectivity Diagram		Libraries (e.g.HDF5)

Sources of data and file systems

Several HPC centers and institutes responded and provided data about one or more file systems. The remainder of this section presents a brief profile of the centers that responded to our survey and information about their file systems, as described in response to the survey. (Note: The information presented in this document is based on the data collected during the survey period. Current details about sites and file system may differ.)

HPC Centers and Institutes Participated in OpenSFS BWG Survey

- Arctic Region Supercomputing Center (ARSC): ARSC runs two HPC systems servicing about 345 users. The center is affiliated with the University of Alaska Fairbanks and Lockheed Martin. Its predominant activities are focused on academics and research & development.
- National Institute for Computational Sciences (NICS): NICS runs six HPC systems supporting about 2400 users. The center is affiliated with the University of Tennessee and primarily supports academic research. NICS provides computing resources for the National Science Foundation and is a partner in the XSEDE project.
- Oak Ridge Leadership Computing Facility (OLCF): OLCF runs several HPC systems including Titan (the #2 system on the June 2013 Top500 list). Additionally, OLCF helps operate the Gaea supercomputer and associated file systems for NOAA's National Climate-Computing Research Center (NCRC). OLCF provides resources for government agencies, academic research, and industrial partners.
- RIKEN: RIKEN is Japan's largest research organization and encompasses several centers focusing on a variety of disciplines such as Biology, Medicine, and Computational Sciences. RIKEN's Advanced Institute for Computational Science operates the K computer.
- San Diego Supercomputing Center (SDSC): SDSC runs five HPC systems servicing over 100 users. The center is affiliated with the University of California, San Diego and provides computational resources for the National Science Foundation as an XSEDE partner. SDSC predominantly supports academic research.

Figure 1 shows a layered view of the centers and some of the file system hosted, as reported in the survey.

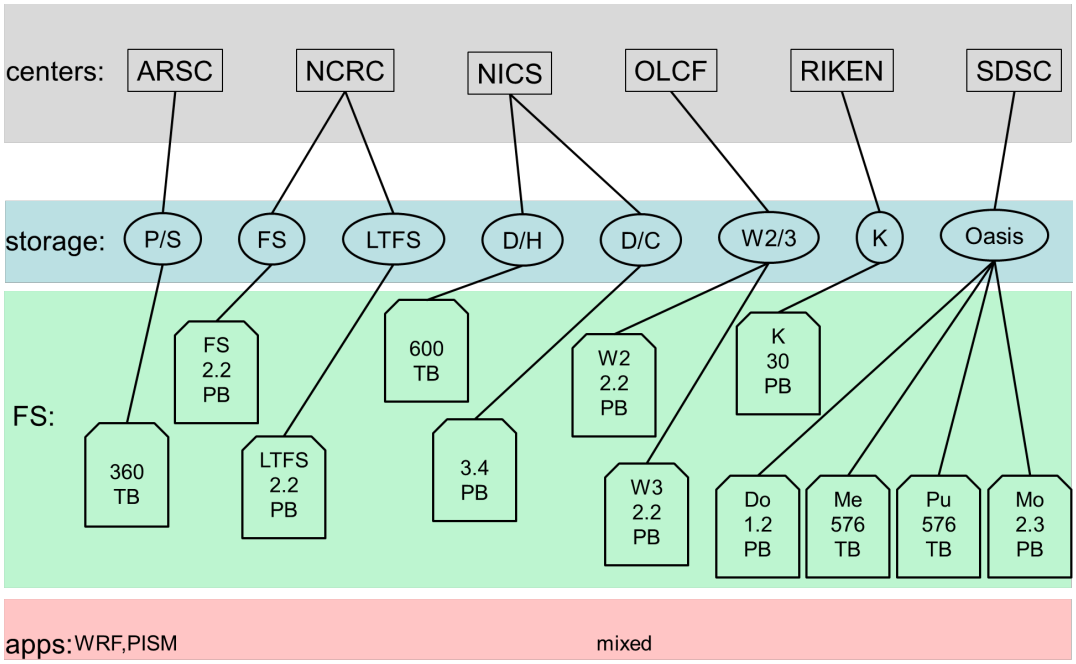

Figure 1: HPC Centers and File Systems

File Systems

- NICS (Kraken): Kraken's file system runs Lustre 1.8.4 and provides 3.36 PB of raw space. The hardware consists of six DDN S2A9900 couplets connected to Cray XT5 service nodes using DDR Infinband. These service nodes act as the Lustre servers and provide file system access to the other XT5 nodes via an internal SeaStar fabric. There are 9440 clients (112,960 cores) that use this file system as scratch space for batch jobs.
- NICS (Medusa): The Medusa file system runs Lustre 1.8.6 and provides about 1.8 PB of raw space. The storage hardware uses three DDN SFA10K couplets connected to the Lustre servers via QDR Inifinband. A QDR Infiniband fabric is also used to connect all Medusa clients (or LNET routers for some systems) to the file system servers. This file system is mounted on nearly all NICS computational resources. The roughly 10,000 clients span multiple architectures (Cray XT, SGI Altix UV, HP servers, etc.). This file system is used for scratch space for the computational resources.
- SDSC (Dolphin): The Dolphin file system runs Lustre 1.8.7 and provides 1.2 PB of raw space. The 16 OSS servers use Aeon Computing's EclipseSL product, running Lustre on the embedded system modules. Each OSS server has two 10 Gb Ethernet connections providing a link to each of two Arista 7508 Ethernet switches. Dolphin is used by the 284 Triton compute nodes, each of which uses 10 Gb Ethernet to access the file system.
- OLCF (widow2, widow3): The widow2 and widow3 file systems run Lustre 1.8.8, each providing 2.2 PB of raw space. Each file system has 64 OSS servers connected to 16 DDN S2A9900 couplets via DDR Infiniband. Each OSS server also connects to a core Infiniband fabric which is used by the roughly 19,000 clients to access the file systems. While the clients are spread among several compute resources, the vast majority of these clients are the Titan compute nodes. Titan uses LNET routers to bridge Lustre traffic between the internal Gemini network and the core IB fabric.
- ARSC: ARSC's 360 TB Lustre file system is shared by multiple compute resources. The servers run Lustre 2.1.2 while the clients run Lustre 1.8.6. Two MDS servers use a LSI 7991 array to store

metadata while the six OSS servers use DDN 6620 arrays for OST storage. LNET routers are used to forward traffic from other networks (Ethernet, Infiniband, Gemini) to the Infiniband fat-tree fabric connected to the Lustre servers. This file system is used by approximately 500 clients.

- RIKEN (FEFS): The K computer uses the Fujitsu Exabyte File System (FEFS). This file system is based on Lustre 1.8 with some special enhancements developed by Fujitsu. This file system supports over 80,000 clients and 640,000 cores. LNET routers are used to forward IO traffic from the K computer's internal 6d torus/mesh Tofu network to the fat-tree Infiniband fabric attached to the FEFS servers.

Basic characteristics of the file systems described are summarized in Table 2Table 1.

Table 2: Summary of file systems.

	OLCF (widow2)	OLCF (widow3)	NCRC FS	NCRC LTFS	NICS (Kraken)	NICS (Medusa)	SDSC	ARSC	RIKEN
# users	N/A	N/A	N/A	N/A	1650	1650	100+	345	N/A
server Version	1.8.8	1.8.8	1.8.8	1.8.8	1.8.4	1.8.6	1.8.7	2.1.2	N/A
Client version	N/A	N/A	N/A	N/A	1.8.4	1.8.6, 1.8.8	1.8.7	>=1.8.6	N/A
# clients	19042	19042	3908	40	9440	400	1638	500	88000
Interconnects (server-client)	DDR IB, Cray Gemini	DDR IB, Cray Gemini	QDR IB, Cray Gemini	QDR IB, Cray Gemini	Cray SeaStar	QDR IB	10 GigE, IB, Myrinet	IB, Ethernet	IB, Tofu
size (raw)	2.2 PB	2.2 PB	900 TB	3.1 PB	3.36 PB	600 TB	4608 TB	360 TB	10 PB (?)
# files	107M	117M	65M	38M	256M	18.3M	141M	7.1M	N/A

Data and Analysis

In this section, we analyze data received from the centers, with respect to one or two of their file systems. The systems are[1]: Kraken and Medusa, at NICS; widow3, at OLCF; Dolphin, at SDSC; and one file system at ARSC.

Distribution of data by file

The first data set represents the distribution of data into files and directories. The distribution can be observed by file number, binning files by their size and counting the number of files in the bins, or by capacity, binning files by their size and accounting for the total amount of data in each bin. Figure 2 and Figure 3 show both distributions and the corresponding cumulative distribution.

From both perspectives, the majority of the data appears to be spread across a larger number of small files. In all cases, 90% or more of the file system usage is accounted for by files 4MB or less. This distribution clearly motivates improvement in supporting metadata and small files operations.

[1] Not all centers that responded provided file system statistics, and statistics were not provided for the file systems in a single center.

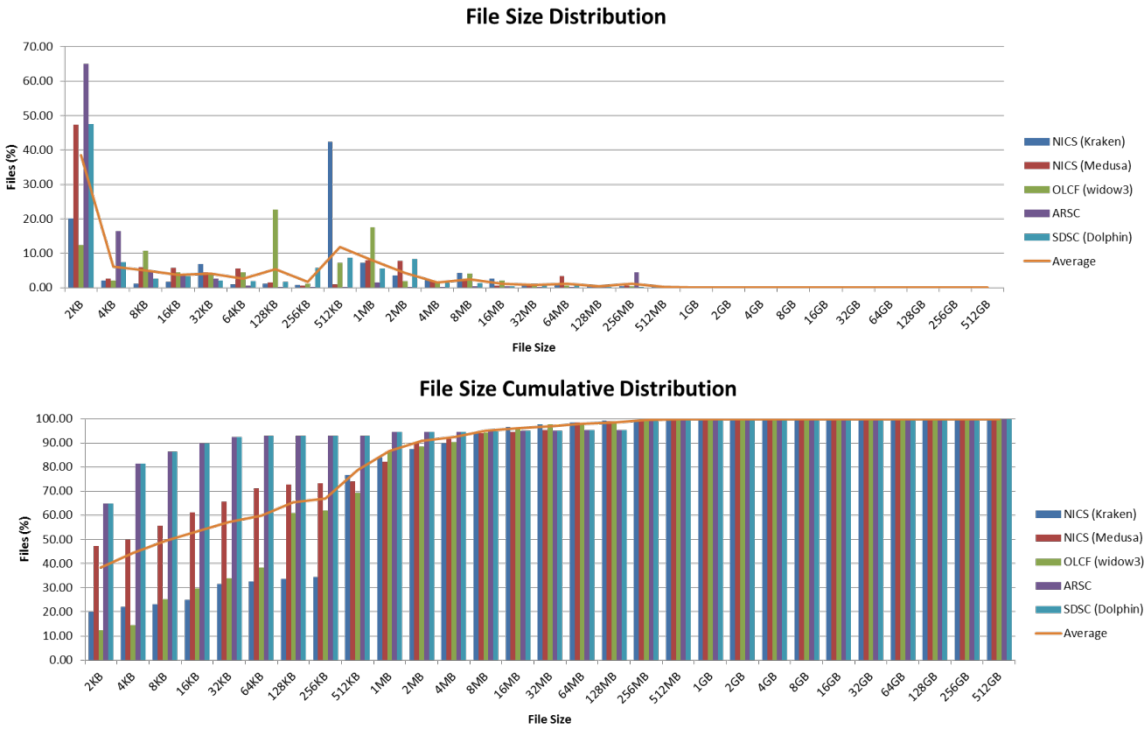

Figure 2: Size distribution

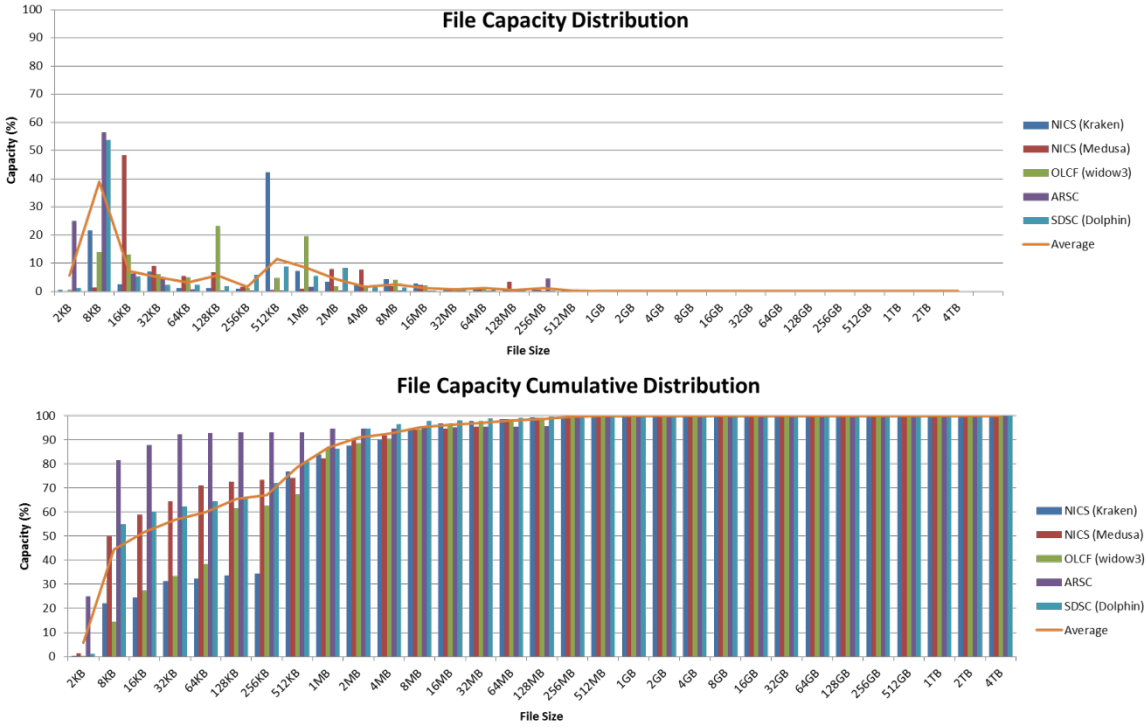

Figure 3: Capacity distribution

Distribution of data by directory

The second measure provided is the distribution of data in directories. As for files, the distribution can be by directory or by capacity. Surprisingly, more than half of the directories contain only 1 entry; further investigation revealed that this is in fact a consequence of the purge policy which is enforced on files but not on the directory tree. Less than 10% of the directories contain more than 16 entries, with the exception of the ARSC's file system; in that specific case almost half of the directories contain between 32 and 64 entries, and almost half of the directories contain between 64 and 128 entries. The distribution suggests that the amount of data contained (exclusively) in each directory is relatively low and there is a large number of directories compared to the number of files. In fact, the distribution by capacity indicates that 95% of the capacity is in directories containing 8Kb or less of data. In addition to hinting at high static meta-data load due to the high number of directories, this distribution suggests that searching individual directories involves few entries; locality may also be reflected by the directories structure, and in that case, the distribution observed could indicate either poor locality or complex locality patterns.

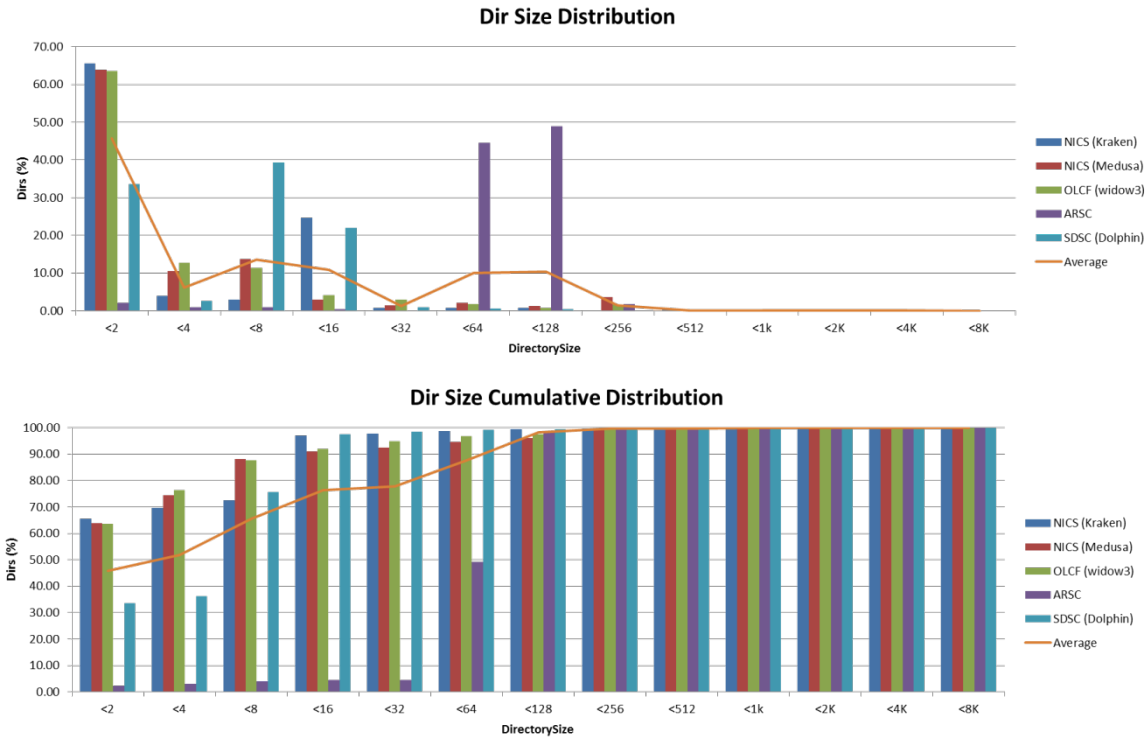

Figure 4: Directory entries distribution

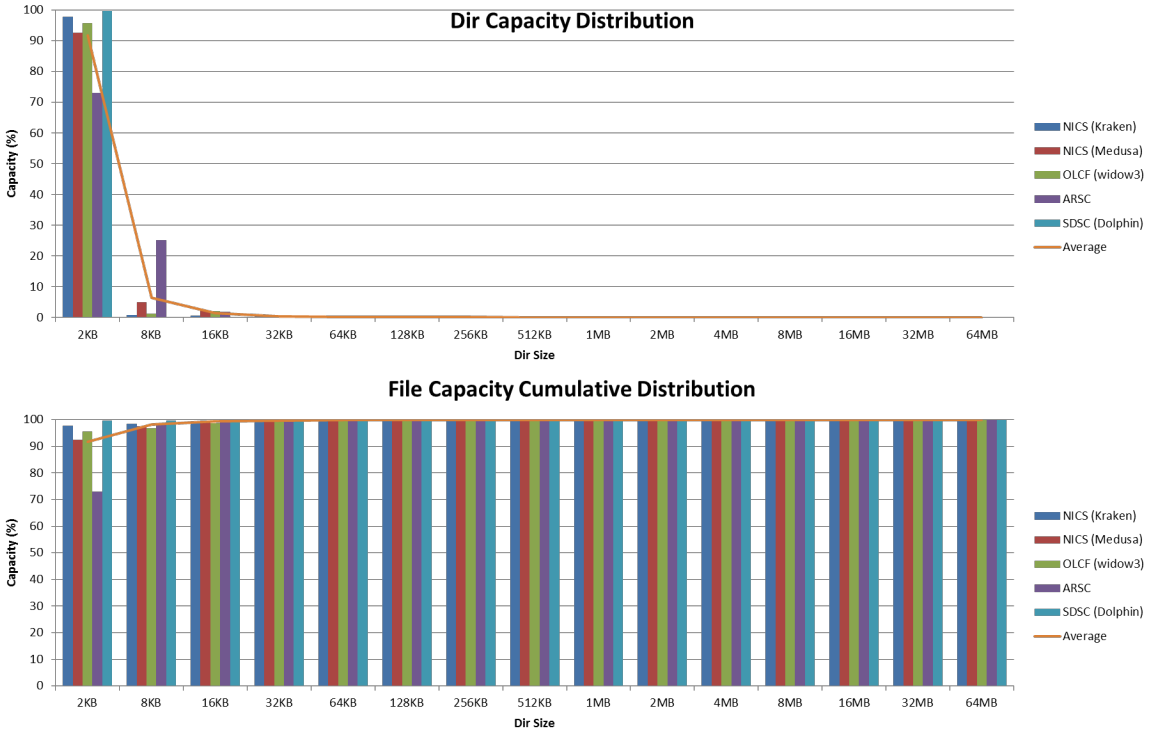

Figure 5: directory capacity distribution

Aging of data

The last part of the data analysis focuses on file system timestamps as a way to represent life-cycle of data in a file system. First we observe temporal locality in access time, as shown in Figure 6. Intuitively, data becomes of interest for a period of time and is repetitively accessed within days, and old (i.e. archived data) is accessed more infrequently; the Widow3 file system seems to be the only exception with some files accessed after hundreds of days. This observation motivates hierarchical design in which data can be moved far in the hierarchy with LRU policies. While some of the observed behavior depends on usage patterns, part is also dictated by established policies, such as purge policies, that prevent files to exist for too long (e.g. NICS has a 30 day purge policy).

Figure 7 shows the days passed since the last modification. Most of the modifications take place between 16 to 256 days (approximately, two weeks to 8.5 months); only a small fraction of files are created or modified often (i.e. within two weeks) and a significant fraction is not modified frequently (after 9 months) and probably is never modified after creation. Again, this form of locality translates to an opportunity for caching data, as most of the files and data are not modified frequently. We also caution that there is significant difference between centers. For example, SDSC center has most of its files modified only after 256 days but before 512 days. So, generic observations cannot replace the need and importance of center-specific design decisions.

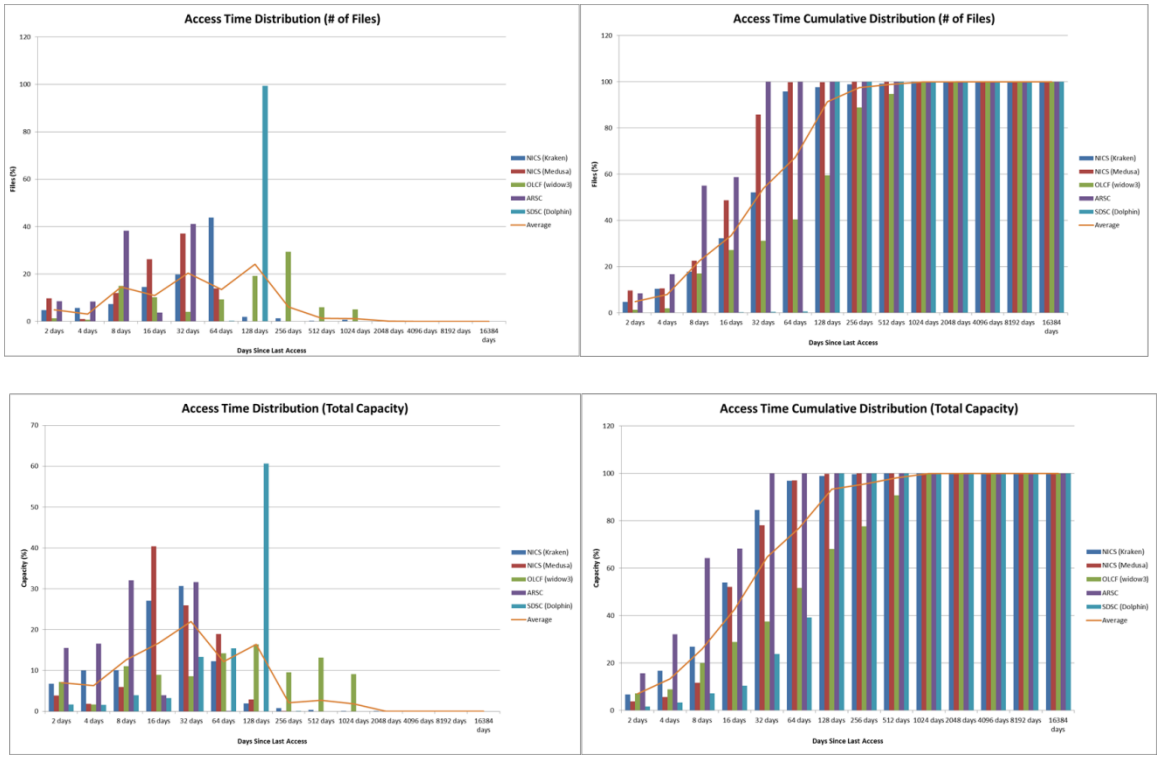

Figure 6: Access time

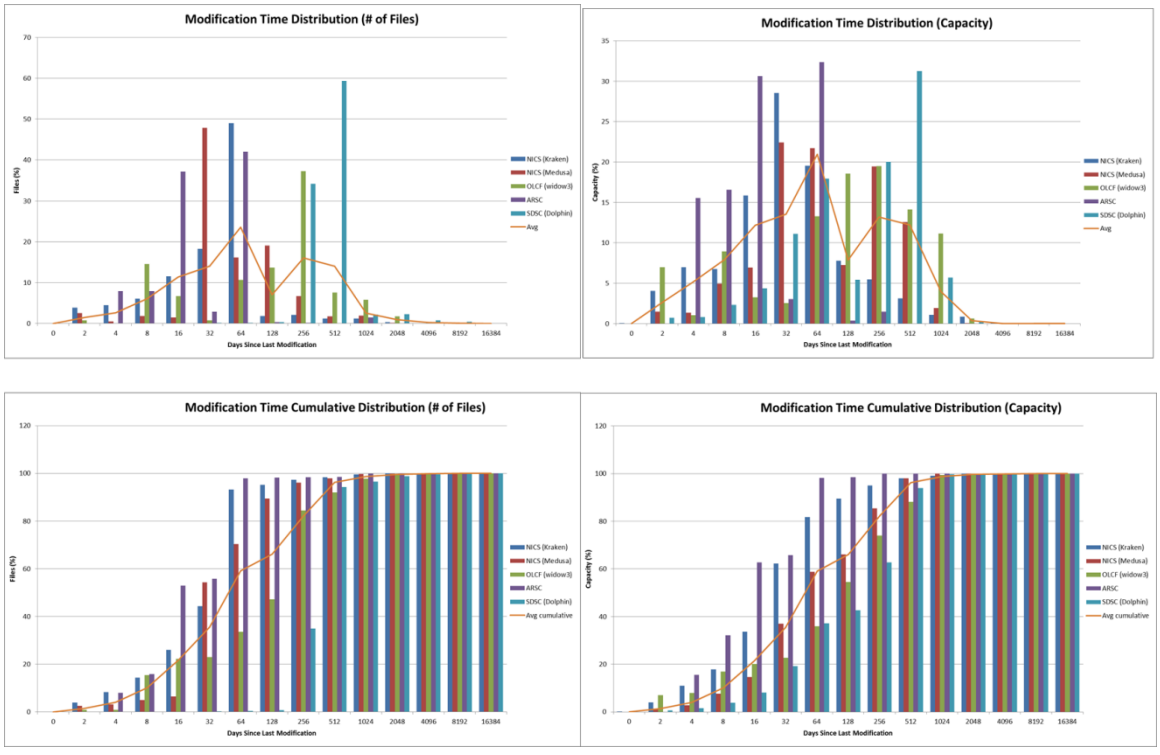

Figure 7: Modification time.

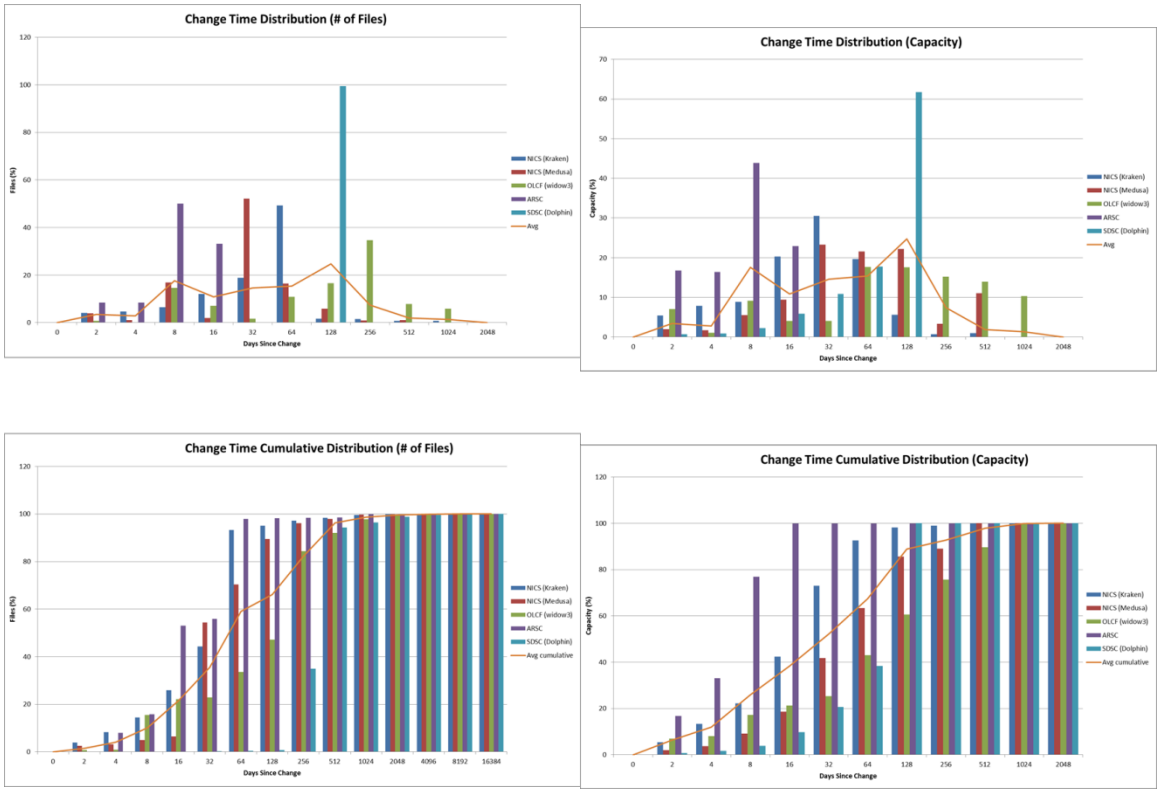

Figure 8: Change time.

Finally, change time is shown in Figure 8. Change time is updated when either the file content is modified or the file attributes are changed. The trend is very similar to the trend observed in modification time, and we suspect that it is because file content modifications dominate (compared to the file attribute changes).

Comparing access time to modification time, we observe that access is more frequent than modification, consistent with a pattern of data created once and accessed multiple times; such a reuse pattern allows for multiple cache replicas without frequent synchronizations.

Conclusion

The data presented was collected in response to a survey that the BWG released at the Supercomputing Conference in 2012. This data is informative in many ways and shows some characteristics of the file systems, some common to most of the file systems and some unique, but also exhibit some inherent limitations of the approach.

The data collected shows that most of the file systems are populated in large part by small files and directories, both in terms of number and capacity. This distribution indicates that strategies such as servicing small file operations with meta-data servers can be beneficial for reducing traffic and performance but also require the servers to be provisioned accordingly, both in terms of rate of

operation and capacity. Another similarity found in most systems is that they all exhibit some degree of temporal locality, which can be taken advantage of in hierarchical and multi-levels storage systems (HSM, burst buffer-like systems).

In both cases, some of the parameters of the distribution are constrained by policies, which also contribute to differentiate usage between centers. For example, purge policies that remove files but leave the directory structure intact tend to inflate the number of almost empty directories (containing only other directories); probably a better policy would be to have a second purge phase to remove the directory structures empty and untouched since the files were purged. Another side effect of purge policies is that it enforces a boundary on timestamps since the access time is bounded by the purge time. Because of these and other center specific patterns and events (file system build date), the generic indications emerged from this data do not always apply and should be carefully reevaluated considering additional center-specific meta-data.

Finally, we consider the lack of timing data the major limitation of this survey. While the statistics provided help create a profile of the file systems, no conclusion can be drawn on how the systems state changes and whether the average usage pattern that can be inferred by the information is actually representative of the daily usage pattern. Data including change logs would certainly be an improvement in creating a characterization of the usage patterns.

March 2014