Large File System Backup

NERSC Global File System Experience

A Mokhtarani, M. Andrews, J. Hick, W. Kramer -- NERSC - LBNL

Introduction

Global file systems require significant increases in capacity each year to continue meeting ever-increasing data requirements from scientific applications. NERSC provides the NERSC Global File system (NGF), a large amount of on-line storage, to users for their daily storage needs. NGF is accessible from all compute systems at the center. Due to the importance of the global file system, the center has established a policy of providing backup of users' data with the goal of complete restoration of the file system to a known state in the event of a catastrophic failure. As the size of NGF increases to keep pace with user demand, NERSC should focus on the strategy it uses to perform the backups and ensure it will meet demand for years to come.

As of March 2008 there are more than 100 projects that range from 1 - 10 TB of data each. Figure 1 shows the projects with the largest amount of data in NGF.

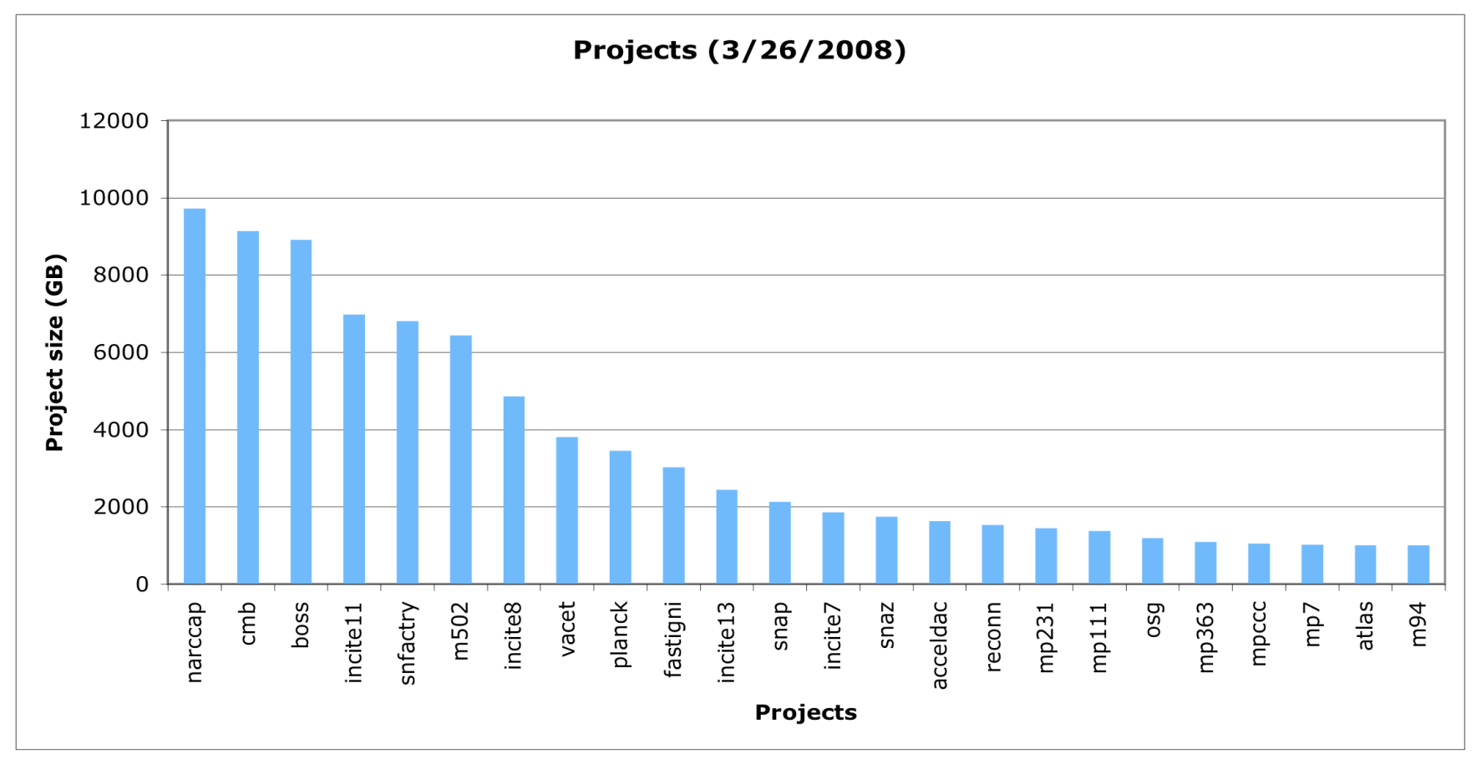

Fig. 1 Projects at NERSC

The center continually upgrades and adds new hardware to meet users' needs. Figure 2 shows the growth in storage usage for a two-year period, a seven fold increase in data; an average quarterly rate of about 29%. If this trend continues, by 2011 the amount of data in NGF will pass the *Petabyte* point.

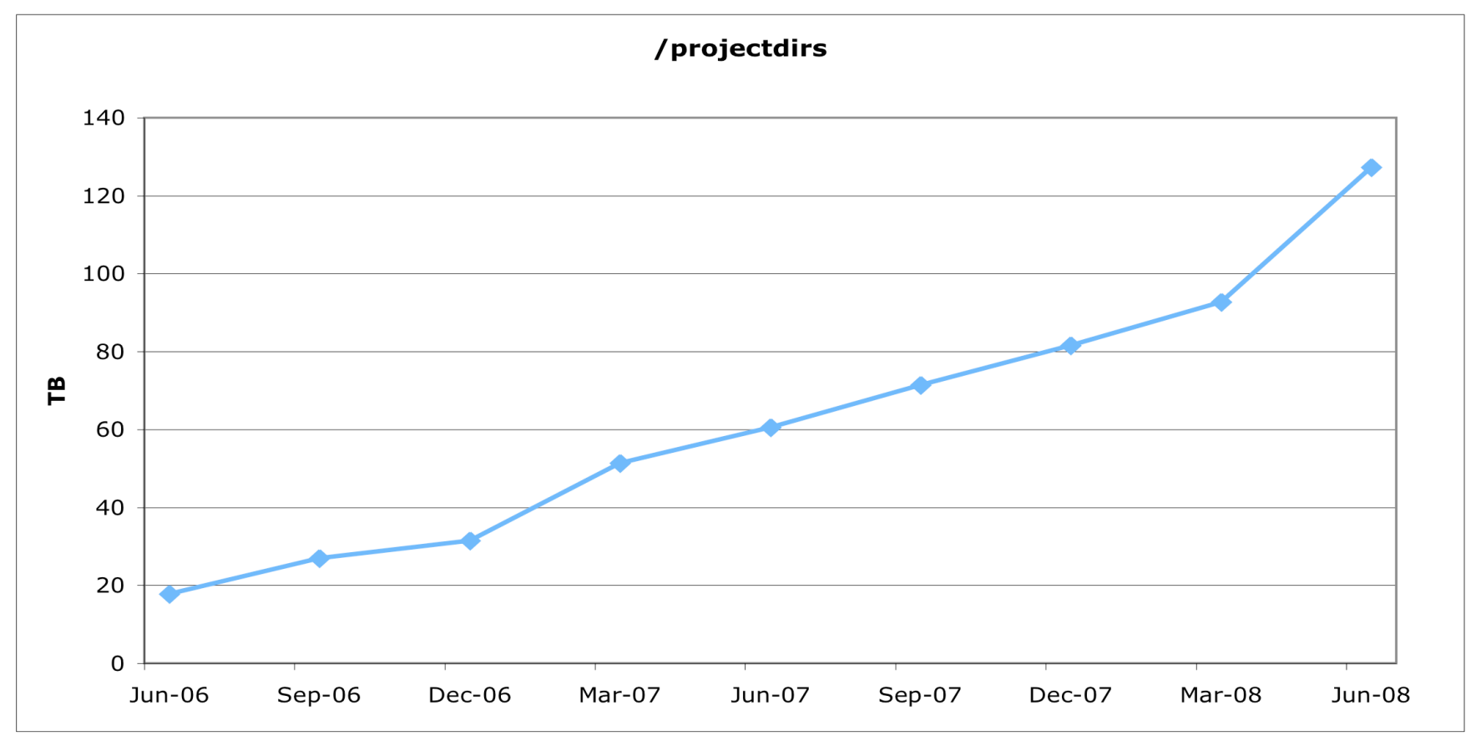

Fig. 2 Growth in users' data in NGF

NERSC Global File system (NGF)

The current NGF backup solution is dependent on the software currently used for the global file system and its configuration. The global file system software is IBM's GPFS version 3.1. Figure 3 shows the current NGF configuration and its components. TSAILUN is a cluster of servers servicing user's requests. Major characteristics of the system are:

•230 TB end user storage with 50+ million inodes accessible from all compute systems (70 TB is allocated to Deep Sky project and is not backed up).
•24 I/O server nodes, 2 service nodes
•GPFS 3.1 PTF 20
•DDN 9500 with SATA & FC, IBM SATA & FC, Sun SATA, 32 2/4 Gb/s FC
•5.5 GB/s bandwidth for streaming I/O

A traditional full backup of the file system is performed periodically using a set of custom scripts. The scripts can be configured to backup a selected set of projects. The backup files are stored in the center's High Performance Storage System (HPSS). A dedicated server in the TSAILUN cluster is used for backup purposes. Initially, a single process was responsible for the whole task but later versions of the script allow for parallel processes to participate in the task. Performing parallel I/O improved the time needed to complete backups significantly.

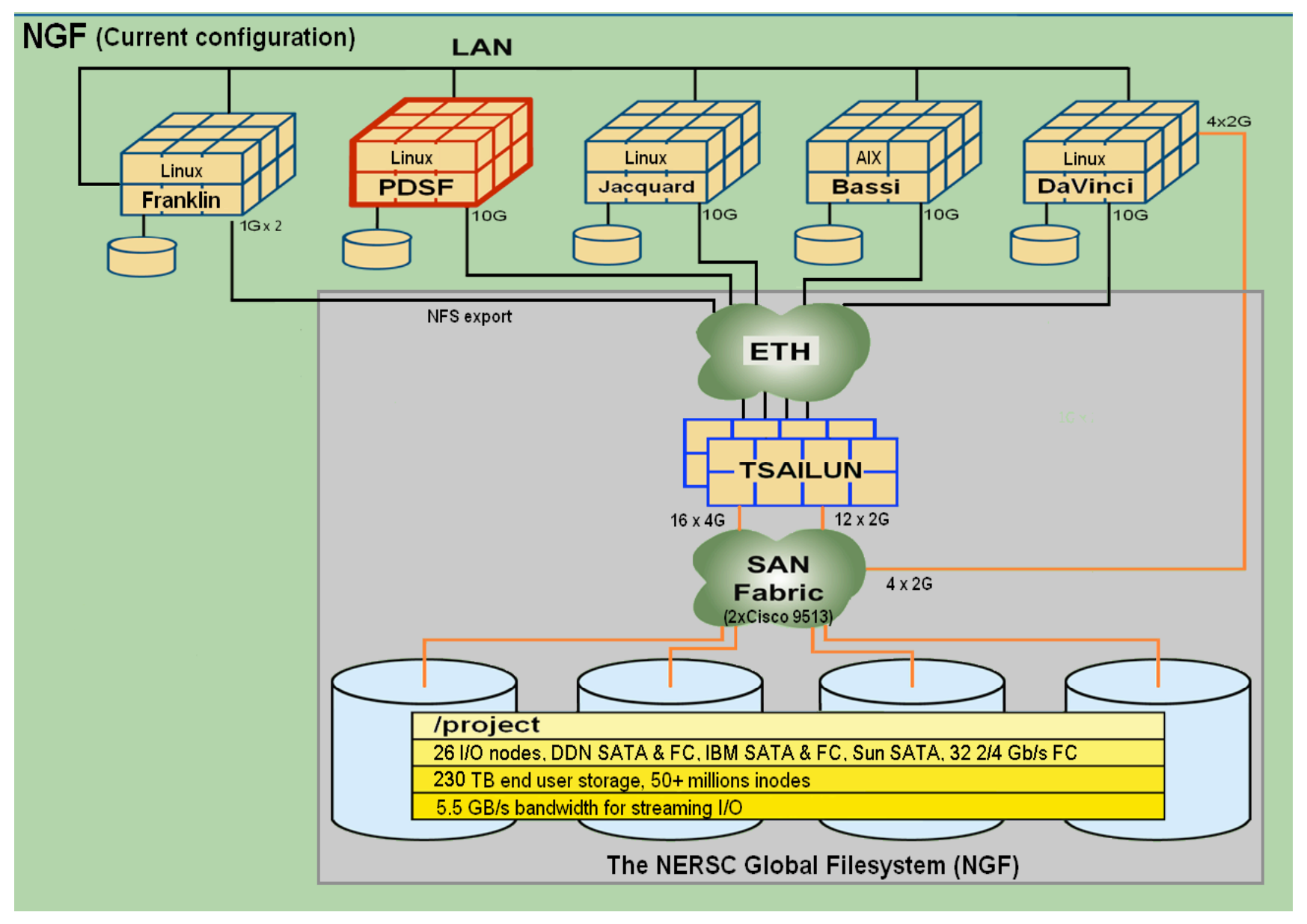
NGF (Current configuration)
LAN
Linux
Franklin
1G x 2
Linux
PDSF
10G
Linux
Jacquard
10G
AIX
Bassi
10G
Linux
DaVinci
10G
4x2G
NFS export
ETH
TSAILUN
16 x 4G
12 x 2G
SAN
Fabric
(2xCisco 9513)
4 x 2G
/project
26 I/O nodes, DDN SATA & FC, IBM SATA & FC, Sun SATA, 32 2/4 Gb/s FC
230 TB and user storage, 50+ millions inodes
5.5 GB/s bandwidth for streaming I/O
The NERSC Global Filesystem (NGF)

Fig. 3 NGF Configuration (June 2008)

Backup Process

The scripts perform the backup of the */project* area in the following steps:

1. Create a fast scan of 'inodes' using the GPFS native interface
2. Sort the list in descending order by project size
3. A scheduler assigns each project to an agent; there are usually eight agents (processes) for this on a single server. We can have more than one server and more agents.
4. Each agent creates a series of 10 GB files using 'star' into a specific reserved part of NGF, and sends them to HPSS using HPSS's pftp client.
5. Once an agent finishes its project, the scheduler assigns it a new project from the list.

The fast inode scan generates one record per directory entry containing inode number, file size, file mode, access and modification times, and full file path. This is accomplished in three steps: a) open an inode scan and read the content of each inode, b) read all directory entries starting from a given root directory, and c) merge these records to construct the one record per directory entry. The first two steps use GPFS APIs and the third is to match inode information with file path. It should be noted that the fast inode scan is only useful for incremental backup where a list of changed files can be extracted from the inode scan without costly examination of every file to construct the list.
Figure 4 illustrates the data flow for the above steps. The connection to HPSS has been upgraded to a 10Gb network resulting in slightly better transfer rates.

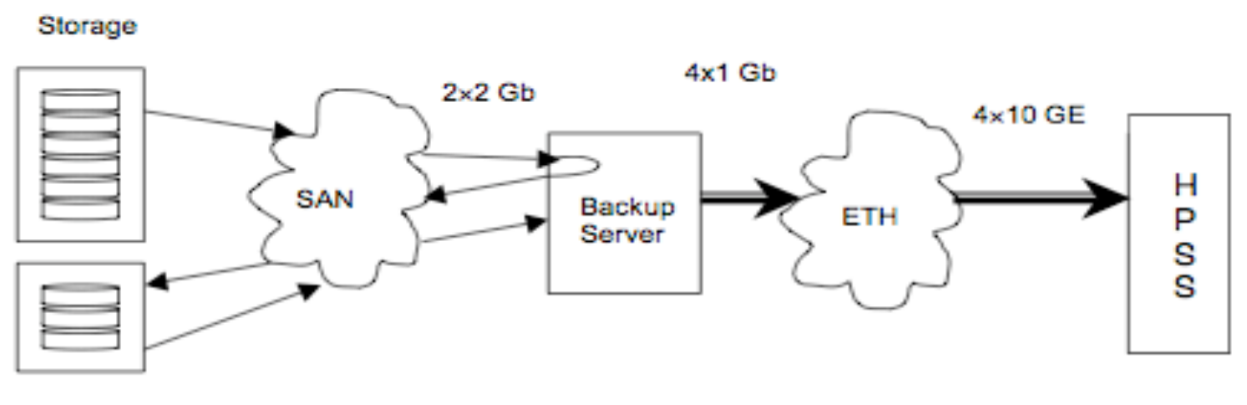

Fig. 4

Figure 5 shows the amount of data backed up and the time, in days, for each. The times include creating the full backup, and transferring that data to HPSS, but not the time it takes for HPSS to migrate the data to tape. The last full backup of ~110 TB of data took about 13 days to complete. The sharp drop in time for the September 2007 backup is due to increasing the number of processes participating in the process from 1 to 8, thus parallelizing the task. Each agent is assigned one project at a time and creates a series of 10 GB files and sends them to HPSS for tape archival using HPSS's Parallel FTP (PFTP) client. It waits until all transfers succeed before getting a new project from the scheduler. The slight improvement in the March 2008 backup is the result of upgrading four HPSS data mover 1Gb network connections to 10Gb
It is clear from Figure 5 that the existing tools are not adequate to provide efficient and reliable backup policies for NGF as it continues to grow. There are significant issues currently with the length of time it takes to complete a backup.
.

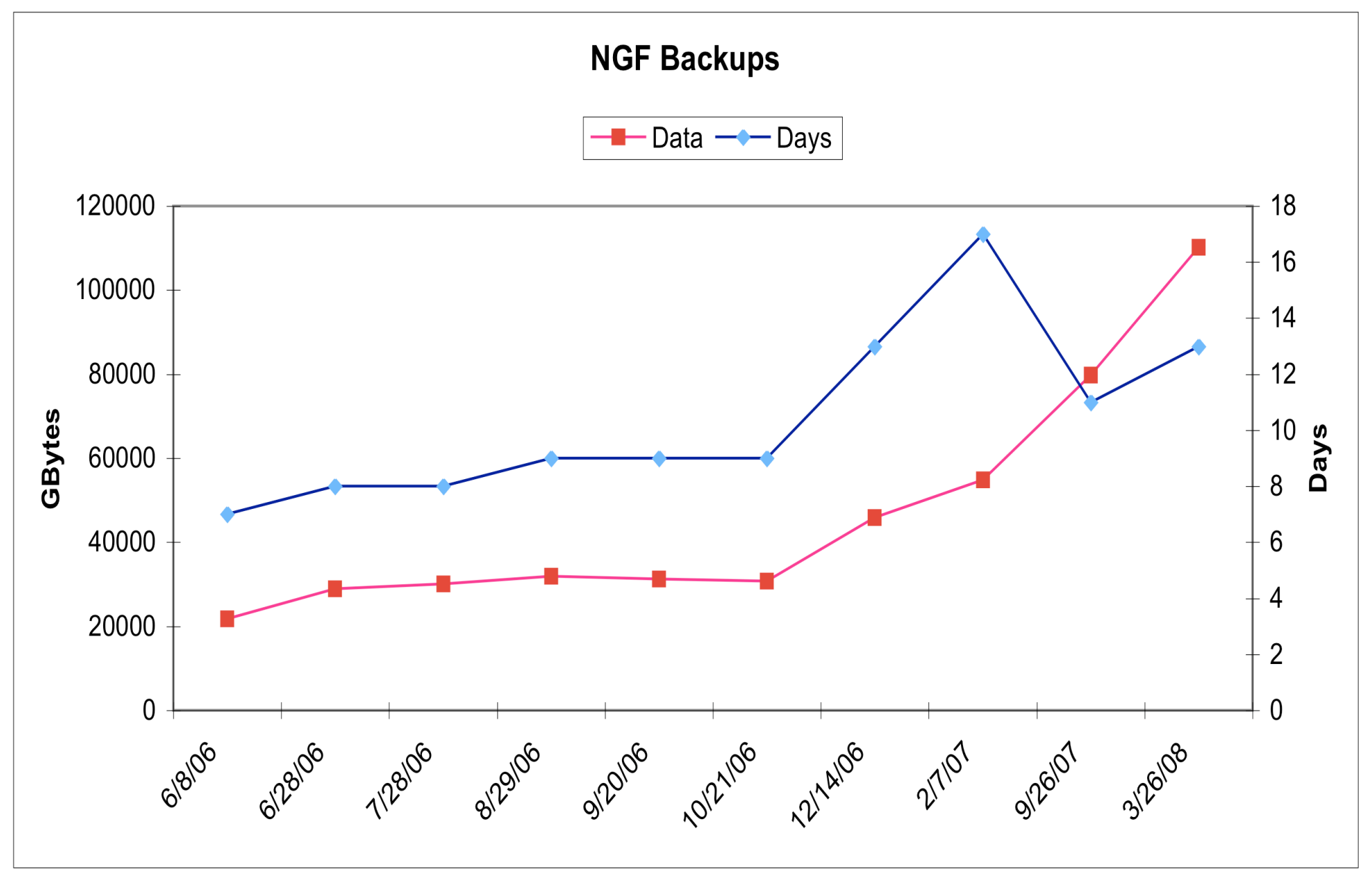

Fig. 5

Future Improvements

The single factor having the most impact on the backup process is the communication bandwidth among different (sub-)systems participating in the process. Here is a list of specific recommendations for improving the performance:

1. The existing server has two 2 Gb FC connections to the storage arrays. These connections are used for reading from the main storage arrays, writing the files to temporary storage, and reading them back from the temporary storage for transfer to HPSS. This is the main bottleneck in the present setup (we can get, at best, 200 MB/s rate). Eliminating this step by adding direct attached storage to the server should allow higher bandwidth to read files from NGF and speed up the process, assuming that the DAS would have enough bandwidth to avoid contention among processes for write and read.
2. Adding more servers would also reduce backup time greatly. We estimate that with the present setup we can run the system with 3-4 servers without saturating the rest of the system, excluding the temporary storage spools, with essentially very little impact on the file system. Since all agents share a set of spindles for temporary storage, addition of more servers and agents would saturate this component very quickly. An alternative would be to assign each agent, or group of agents, to different set of disks and minimize the contention.
3. The additional time spent in writing to and reading from the temporary storage can greatly be decreased if solid state disks (SSD) are used inside the server and thus eliminating the extra paths through the SAN fabric. This also allows higher bandwidth to read files from NGF.
4. Each agent waits until the transfer to HPSS is completed before starting on the next set of files. This constitutes approximately 30% of the total time. HPSS is capable of handling much higher transfer rates. Making the transfer asynchronous can reduce the total time for backup if the system is not always saturated. It is difficult to get an accurate estimate since there are overlaps in operations among agents. A single process backup yielded a 40 % 45% of time spend in transfers to HPSS
5. Consider direct-to-tape transfer in HPSS due to the size of NGF backups.
6. Consider streaming the transfers to HPSS, thus eliminating the step for temporary storage. This, however, requires significant software support from HPSS. The new releases of HPSS, 6.2 and 7.1, have this feature implemented.

Conclusions

We have examined the existing backup utilities at NERSC for NGF full backup. The present trend of 29% quarterly increase in users data makes the current approach impractical within the next few years. If frequent backups of users' data will be required to restore the global file system to a known state in a Petascale or larger environment, careful consideration should be paid to the requirements and design of tools. A number of steps are outlined that can improve the performance significantly. However, periodic full backup of large file systems will be impractical and will impose a large overhead on available resources. A more practical approach is either a selected list of files/directories, determined by users, are backed up, or frequent incremental backup of the file system.
Any backup policy needs to consider the impact on the file system in servicing user's request. Having too many agents participating in the process can impact file system response time, thus degrading performance for end users. Calculations based on the existing 5.5 GB/s of NGF streaming with a 5% overhead on the file system for backup processes give ~250 MB/s bandwidth, and about 110 hours of backup time for every 100 TB of data.
The backup policy should also consider the value of retaining user data from backups spaced farther apart. Roughly, about 20% of files have modification date of less than120 days and if, for instance, a quarterly backup is taken and a disaster occurs requiring restoration of the file system, some user data might render invaluable if they have changed significantly since the last full backup.