LARGE SCALE, GRID-ENABLED, DISTRIBUTED DISK STORAGE SYSTEMS AT THE BROOKHAVEN NATIONAL LAB RHIC/ATLAS COMPUTING FACILITY

B. Gibbard, Z. Liu, R. Popescu, O. Rind, J. Smith, Y. Wu, D. Yu, X. Zhao,
Physics Department, Brookhaven National Laboratory, Upton, NY 11973, USA

Abstract

The BNL RHIC/ATLAS computing facility provides large scale, production quality, grid-enabled, distributed disk storage systems for the RHIC/ATLAS experiments based on the dCache project developed by DESY/FNAL. This paper presents the deployment and usage of the two instances of dCache. System performance and tuning experiences during LHC service challenges are also described.

INTRODUCTION

The Brookhaven RHIC/ATLAS Computing Facility serves as both the tier-0 computing center for RHIC and the tier-1 computing center for ATLAS in the United States. The increasing challenge of providing local and grid-based access to very large datasets in a reliable, cost-efficient and high-performance manner is being addressed by a large-scale deployment of dCache[1], the distributed disk caching system developed by DESY/FNAL.

dCache provides a system for users to store and retrieve huge amounts of data, distributed among a large number of server nodes or stored in a Hierarchical Storage Manager (HSM), under a single virtual file system tree with a variety of standard access methods. In addition, it significantly improves the efficiency of connected tape storage systems through caching, i.e. gather & flush, and scheduled staging techniques. The system maintains load balance and fault tolerance through the use of cost metrics and inter-pool transfers, dynamic replication of files upon detection of hot spots, and multiple distributed administrative servers for each access method. The system is highly scalable due to the use of distributed data movers and access points (doors), highly distributed storage pools, and direct client–disk (pool) and disk (pool)–HSM (HPSS) connections. The various access protocols supported include a local access protocol - DCAP (POSIX-like), GsiFTP data transfer protocol (secure wide area data transfer), and the Storage Resource Manager Protocol (SRM)[1].

Currently at BNL, there are two large dCache deployments, one for the ATLAS experiment and one for the PHENIX experiment.

BNL dCache systems employ the worker nodes utilized by the RHIC and ATLAS analysis clusters, making use of the large amount of low-cost, locally mounted disk space available on the computing farm. On read pool servers, which are the majority, the resources are shared with the computing worker nodes. Within this hybrid storage/computing model, the worker nodes function simultaneously as file servers and compute elements, providing a cost-effective, high throughput data storage system.

In order to ensure reliability and high performance, both systems also deploy a small number of dedicated servers, which are used for critical dCache components, e.g. PNFS node, door nodes, and write pool nodes.

The BNL dCache systems also serve as caching front-ends to the HPSS Mass Storage System. By integrating with a backend tape prestaging batch system, such as the Oak Ridge Batch System, access to the data on tape is greatly optimized.

In the next sections, the deployment and usage of the USATLAS dCache[2] and PHENIX dCache will be presented. System performance and tuning experiences during LHC service challenge activities [3] will also be discussed in the USATLAS section.

USATLAS DCACHE SYSTEM

System deployment

Figure 1 shows the architecture of the USATLAS dCache system. Figure 2 shows the status of the current system setup.

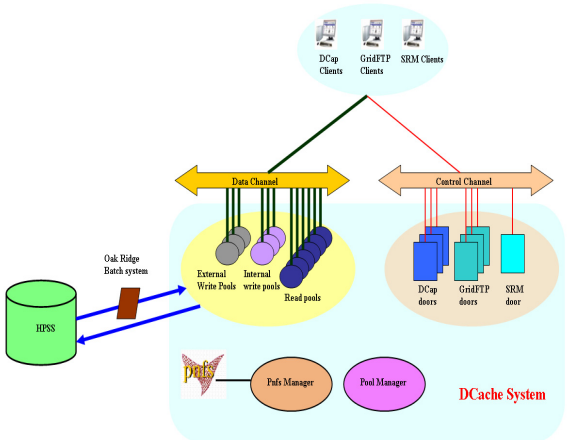

Figure 1: USATLAS dCache system

Server Type	Numbers of servers	Disk cache space
PNFS Core server node	1 (dedicated)	N/A
SRM server node	1 (dedicated)	N/A
GridFTP and DCAP Core server nodes	4 (dedicated)	N/A
Internal/External Read pool nodes	322 (shared)	145 TB
Internal/External write pool nodes	8 (dedicated)	2 TB
Total	**336**	**147 TB**

Figure 2: Servers in USATLAS dCache. "Shared" means that servers share resource with worker nodes.

Usage of the system

USATLAS dCache has been in production service for ATLAS users since November 2004. An on-site user with a local BNL account has read permission through local or grid protocols, and write permission after a dCache work area is assigned to the UID. A remote user in the ATLAS/USATLAS VO has read permission through grid protocols (like GridFTP or SRM), and write permission after their grid DN is mapped into a local account which has been assigned write permission to a work area.

As of Jan 31st 2006, 152 TB of ATLAS production data have been written into the dCache name space.

Besides production usage, USATLAS dCache has also been working as a BNL Storage Element during a series of LHC Service Challenge (SC) activities.

The system has exhibited quality performance through a series of Service Challenges and US ATLAS production runs.

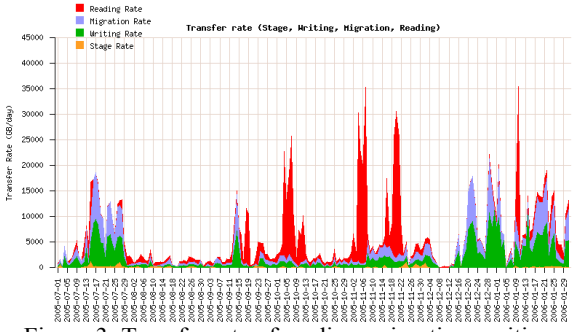

Figure 3: Transfer rate of reading, migration, writing, stage

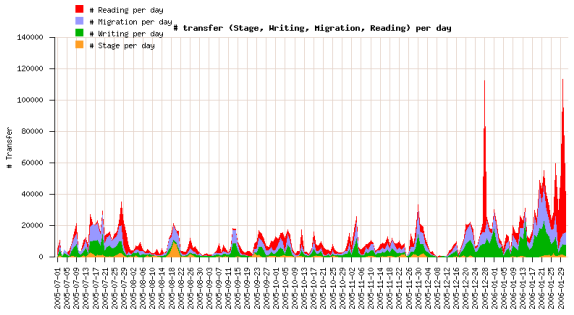

Figure 4: The number of transfers per day for reading, migration, writing, stage

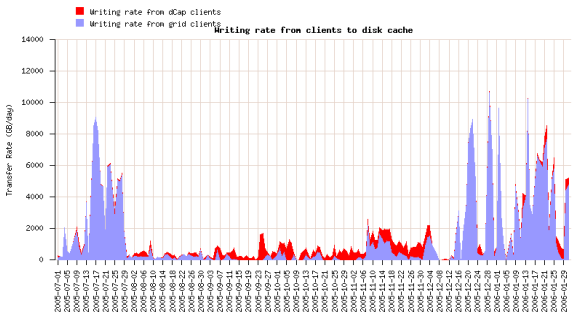

Figure 5: The transfer rate of writing from dCap clients and grid clients

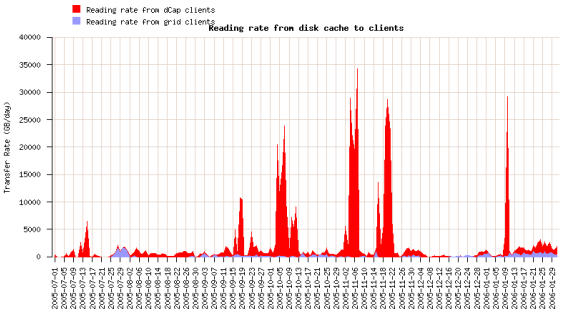

Figure 6: The transfer rate of reading from dCap clients and grid clients

Figures 3~6 show the statistics of data transfer in USATLAS dCache from July 2005 to January 2006. Note, reading represents transfer from dCache disks to clients, writing represents transfer from clients to dCache disks, stage represents transfer from HPSS to dCache disks, and migration represents transfer from dCache disks to HPSS.

System performance during LHC Service Challenge

USATLAS dCache has also been working as a BNL Grid-enabled Storage Element during a series of LHC Service Challenge activities, i.e., SC3 throughput and Service phase, SC3 re-run[5].

The transfer rate goals for BNL during past SC activities have been achieved.

During the SC3 throughput phase in July 2005, the disk-to-disk transfer rate from CERN to BNL was sustained at about 120 MB/sec, and the disk-to-tape rate from CERN to BNL was sustained at about 80 MB/sec.

During the SC3 re-run in January 2006, the disk-to-disk transfer rate from CERN to BNL was sustained at about 90 MB/sec and the disk-to-tape rate from CERN to BNL was sustained at about 50~60 MB/sec.

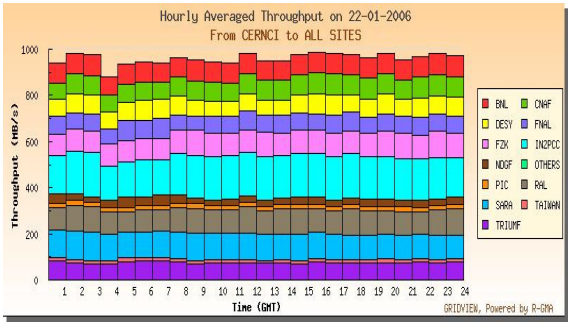

Figure 7: Data transfer from CERN to 12 major computing centers during SC3-rerun

System tuning experiences during LHC SC

By participating in LHC Service Challenges, we have gained some system tuning experience as follows.

The RHEL4 OS shows better performance over RHEL3 on write servers. A higher disk I/O rate is achieved with lower load.

XFS shows much higher disk I/O rate and lower load compared to EXT3 for write servers.

Increasing the kernel stack size on write servers makes them more stable. It avoids system crashes caused by stack overflow problems with high disk I/O on XFS systems. The kernel stack size has been increased from 4K to 8K on the write servers.

Increasing free memory can avoid dCache application Java page allocation errors.

TCP network tuning does not seem helpful at this point since the network is currently the bottleneck from CERN to the BNL USATLAS dCache system. In the future, when the network is upgraded, this method is expected to be useful for performance tuning.

Bonnie disk I/O testing has been done to get the optimized number for maximum parallel write threads on the write servers. According to the testing, the optimized number for concurrent write threads in our system is six. When the number of write threads is increased above six, the disk I/O performance drops dramatically.

Hyperthreading is disabled on door servers and write servers for better system CPU utilization. Each server has two physical CPUs, originally set up to use virtual processors, making four virtual CPUs per server. However, since there are only one or two dCache Java processes running on each write server or GridFTP door server, and no other CPU-intensive user applications, disabling hyperthreading shows better CPU usage and performance.

On write pool servers, allowing I/O in only one direction at a time shows better performance, i.e. only allowing either inbound traffic from clients, or outbound traffic when flushing/copying data into HPSS/read pools. Concurrent inbound and outbound traffic downgrades the disk I/O performance. Current dCache software does not provide an intrinsic feature to do this. However, the next version of dCache will implement smart tape system flushing and enable the tuning[4].

Future Directions

The ATLAS experiment will generate data volumes each year on the Petabyte scale starting in 2007. As the USATLAS tier-1 center, the long-term goal for storage services at BNL is to build a Petabyte-scale, dCache-based, grid-enabled Storage Element. The future system will utilize Petabyte scale disk space on thousands of farm nodes to hold the most recently generated/used data on disk. HPSS, as the tape backend, will hold all historical ATLAS production data.

PHENIX DCACHE SYSTEM

The PHENIX experiment is one of four operating at BNL's Relativistic Heavy Ion Collider (RHIC) and has been taking data since 2000. During most of this period, the collaboration has relied upon a centralized SAN, served via NFS to their farm of Linux servers, to meet their dynamic storage needs. However, over the last few years, cost and scalability problems with this model have increased with the size of the data store and the growing multiplicity of compute nodes. This has heightened the interest in tapping the vast amount of low-cost distributed local storage on the compute farm through a software-managed solution. Increasingly, dCache has been providing this solution.

The current PHENIX dCache deployment comprises over 25 TB of locally-mounted storage on 212 pools spanning 128 servers. Most of the pool servers reside on the compute farm, behind a firewall, with local and outbound-only access available. These servers contain either 1 or 2 pools each, providing a total of either 111 or 262 GB of storage. Newer servers will supply more than 400 GB per host. A small subset of dedicated servers are configured as write pools with available inbound access from outside BNL. These servers contain four pools with about 420 GB of total storage apiece.

As mentioned in the introduction, the PHENIX dCache, like the USATLAS dCache, has been deployed according to a "hybrid model" in which the pool servers also double as computing elements on the Linux farm. The primary mode of user access to the farm is through a Condor batch system, which limits the number of running jobs per compute node. Thus, under this model, the number of data servers tends to increase in proportion to the number of clients, providing a naturally scalable and cost-effective system.

The drawback is instability resulting from unpredictable user jobs contributing to server downtime. The dCache system handles this situation smoothly, although the demands on the tape storage system can increase as requested files on downed systems are retrieved from backup. This has further implications on an administrative policy that relies on server redundancy and does not guarantee 24/7 uptime for individual hosts.

The performance of the PHENIX dCache is shown in figures 8 and 9. Peak transfer rates of more than 50 TB/day and 60K files/day have been observed. The high transfer rate for dCap reflects a buffer incompatibility issue when using the PHENIX ROOT implementation that sometimes causes more data to be transferred than was actually required by the application.

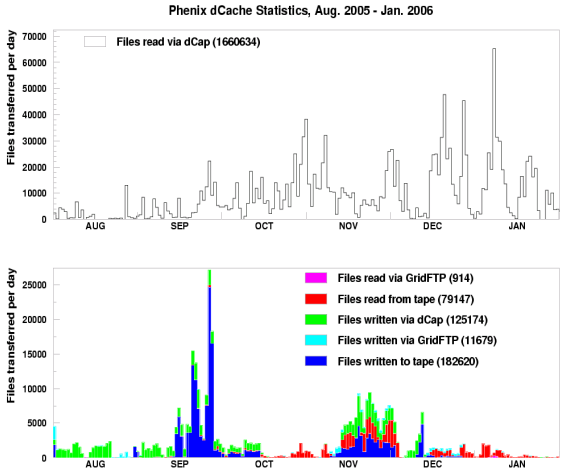

Figure 8: Number of PHENIX dCache file transfers per day in a recent six month period. Transfers via native and grid protocols as well as to/from mass storage are indicated with cumulative totals.

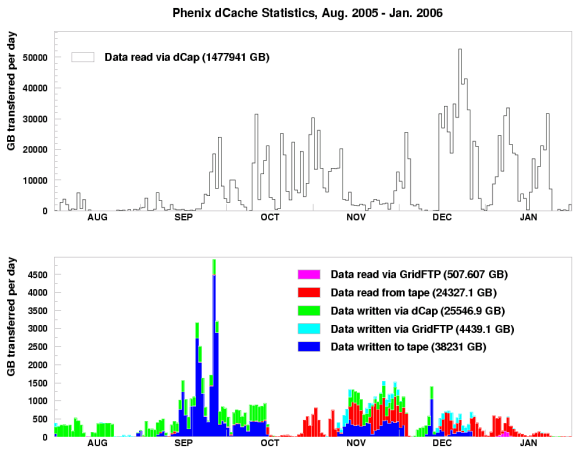

Figure 9: PHENIX dCache data volume transferred per day in a recent six month period. Transfers via native and grid protocols as well as to/from mass storage are indicated with cumulative totals.

Backend interface to HPSS

The dCache backend interface to HPSS was developed locally using a drop-in script and hooks provided by the dCache developers. The primary data transfer protocol is Parallel FTP (PFTP) with a Hierarchical Storage Interface (HSI)[6] in use to manage the data tree. All files written directly into dCache are backed up in HPSS under a single user account (the special account "dcphenix"). This does not affect ownership within the dCache file system itself. The PNFS file tree structure is replicated underneath the top-level dcphenix account in the HPSS filesystem.

dCache does not currently have a built-in method for global throttling of mass storage access. PHENIX had already deployed a software layer to manage file restore requests before dCache was implemented, so this was a natural choice for the backend interface. This software, known as the "data carousel,"[7] consists of a set of PERL scripts that pass requests to a tape access optimization layer. A MySQL database is used at an intermediate level to provide policy-driven controls and to monitor the status of tape requests. Another tracking script, activated on the selected pool when a restore request is made, interfaces to this database to track the progress of the request and returns notification to the pool when the file has transferred. In this way, fine-grained control of dCache restore requests is attained, in particular during periods of conflicting priority. Flushing of precious files to tape uses PFTP directly and is controlled by limiting the number of concurrent processes at the individual pool level.

Interfacing to the PHENIX file catalog

In moving to a dCache file access model, PHENIX was looking for a unified storage solution; thus, they had to confront both the fact that they already had a large (many hundreds of terabytes) data store and a production file catalog with an established data stream from control room to mass storage. Creating secondary data streams and/or directly loading previously written data into dCache was not an efficient or even feasible short-term solution. Instead, scripts were developed to create file metadata entries in dCache by writing the catalog information directly into the PNFS tags. In this way, more than 240 TB of reconstructed data has been registered into dCache so far, without requiring file stages from HPSS. In addition, an update process periodically checks for new entries in the PHENIX catalog and continues to replicate this metadata in the dCache.

Future Directions

The 2006 RHIC polarized proton run begins in March. During this run, an expansion of the existing dCache deployment is envisioned with up to 110 TB of additional storage space. In addition, dCache will serve a greater role in transferring data to and from offsite clients at collaborating institutions.

ACKNOWLEDGEMENTS

We would like to thank the dCache developers from DESY and FNAL for their contributions to the dCache software and for providing us with essential technical support. We also would like to thank other dCache sites for sharing their experiences with us.

We also would like to thank the CERN tier-0 coordinators for their support during the LHC Service Challenges.

Thanks to the BNL RCF/ACF HPSS team for their great efforts in backend maintenance and support.

Thanks to Scott O'Hare for his work in developing the HPSS backend interface and to Irina Sourikova for her support in further integrating the PHENIX data carousel.

REFERENCES

[1] dCache project website: http://www.dcache.org/
[2] BNL USATLAS dCache website: http://www.atlasgrid.bnl.gov/dcache/manuals/
[3] LHC Service Challenge website: https://uimon.cern.ch/twiki/bin/view/LCG/LCGServiceChallenges
[4] P. Fuhrmann, "dCache, the Upgrade", CHEP06, Mumbai, India, Feb. 2006.
[5] BNL Service Challenge website: http://www.usatlas.bnl.gov/twiki/bin/view/Projects/ServiceChallenges
[6] HSI website: http://www.sdsc.edu/Storage/hsi/
[7] BNL PHENIX Data Carousel website: http://www.phenix.bnl.gov/software/tutorials/datacarousel.html