The dCache Storage Element and more

Patrick Fuhrmann[1] for the dCache team

Deutsches Elektronen Synchrotron
Notkestrasse 85, 22607 Hamburg

Abstract. In 2007 the *Large Hadron Collider(LHC)* at CERN will go online and will produce a sustained stream of data in the order of 300MB/sec which has to be distributed and persistently stored at several dozens of sites around the world. dCache/SRM is a storage system which has proven to cope with those requirements and is being able to store data in the petabyte range distributed among hundreds of disk storage nodes. dCache strictly separates file name space and file location and with that is able to transparently manage dCache internal or external file location changes due to configuration or load balancing constraints. dCache allows access to the data by various transfer protocols like http, gridftp, xrootd and a posix like protocol and supports the high level *Storage Resource Manager (SRM)* protocol for storage and transfer control. Furthermore dCache provides mechanisms to optimize interactions with tertiary storage systems.

1 Contributors and Dissemination

dCache/SRM is a joined effort between the Deutsches Elektronen-Synchrotron[1] in Hamburg and the Fermi National Accelerator Laboratory[2] near Chicago with significant distributions and support from the University of California, San Diego, INFN, Bari as well as from the GridPP people at Rutherford Appleton Laboratory, UK[4] and CERN[3].

At the time of this publication, dCache is in production at more than 35 locations in Europe and the US. The size of installations span from tapeless, single hosted systems to setups exceeding 200 TBytes of disk storage attached to tertiary systems. Typically LHC Tier I sites, like SARA (Amsterdam), IN2P3 (Lyon), gridKa (Karlsruhe), Brookhaven (US) and FermiLab (US) are running dCache installations connected to a variety of tape storage systems, while Tier II centers make heavy use of the high availability resilient dCache mechanism. Peak throughputs, we learned about, have been in the order of 200 TBytes/day to more than 1000 clients simulataneously.

2 Technical Specification

2.1 The general picture

The idea behind the dCache storage middleware is basically three-folded. dCache is primarily designed to combine petabyte disk storage, located on thousends of storage nodes to a single rooted virtual file system. Various access methods are provided, including posix like access as well as http, ftp and xrootd. dCache may handle multiple copies of a single file within or outside its sphere of responsibility, to cope with user configuration and/or performance bottlenecks. Based on this, dCache operated stand alone, or may serve as a disk cache in front one or more tertiary storage system. A set of mechanisms is build-in to optimize access to such a backend. Moreover, a toolkit will be provided, allowing system administrators to customize dCache - Tape system interactions. Finally, dCache implements all interfaces required by the LCG[6] Storage Element definition. This includes the Storage Resource Manager (SRM[10]) Control Protocol, the GridFtp[14] transport protocol as well as mechanism to publish information into the LCG information provider system. Furthermore, dCache is prepared to be compliant with the Storage Element definition of the Open Science Grid (OSG[21]) initiative.

2.2 The namespace

dCache is based on the idea of having the file name space strictly separated from the actual location of the file within or outside of its refuge. As described below, this easily allows having necessary file replication operations done without interfering with user activities. Although the current implementation of the file name-space still covers the needs, a migration to higher scalability is already in preparation. As a first step we will allow to partition the name space service into independent sections which may run on different server machines. This give us the necessary time to introduce the next generation name space service, Chimera[23]. Chimera is based on the experience we and our customer made over the last years of heavy production. Chimera is fully database based and intrinsically scalable.

2.3 Data access protocols

The dCache framework allows to plug-in transfer protocols to access data within the system. At the time of this publication, the following protocols are supported. dCap and xRootd, both of which allow posix like random access to the data, by which xRootd currently is in the process of being tested. In the category of wide area streaming protocols, dCache implements http and various FTP dialects including GsiFtp.

2.4 Dataset replication mechanisms

With the increasing size of a storage system installation, administrators tend to assign certain properties with storage nodes. This might be simply the distinction between write and read storage nodes or, more sophisticated, the assignment of storage nodes to user groups or Virtual Organizations. Internal file movement might become necessary to fulfill those administrator defined rules, which are based on attributes of the file and the type of the transfer. Currently the IP number of the requesting client host, the location of the file within the file system tree, the transfer direction and the type of the protocol are inspected to select appropriate storage location candidates. Within this set of possible locations, dCache selects the most appropriate storage node by keeping track of system properties like the number of active data transfer movers, available space and the time files have been accessed last. This allows to react on so called 'hot spots', storage nodes, more heavily used than others. Moreover, the system may be driven in resilient mode in which it makes sure that a minimum of copies of each single file is available on different storage nodes allowing node failures or intended shutdowns of individual nodes without interrupting overall system availability.

2.5 Tertiary storage system connectivity

Being used as a cache system, dCache moves data from/to tertiary storage systems transparently on request or following certain rules. Files accessed by clients, but not yet in cache, are automatically fetched from backend storage. Files written into the cache are collected and moved towards tertiary storage after a certain amount of data has been collected or a timeout has expired. Those steering parameters can be attached to particular data types and need not be be global. During the most recent LHC data challenges, it turned out that most disk arrays only provide limited performance if data is written to and read out of those systems simultaneously with high demanding throughput. This effect degrades dataflow performance when data is coming in from external sources and is supposed to go to tape as fast as possible. With the next release of dCache we therefor allow the system to chose pools to only receive data or only flush data. Switching is done automatically.

There is a manifold of issues related to dCache tape system interactions we have to face for the medium term future, mainly because of a missing standard of tertiary storage systems concerning the access protocols and behaviours. Therefor we are still in heavy discussion with our customers to provide a generic solution.

2.6 dCache partitioning

dCache can be run in various modes, ranging from a simple cache in front of a tertiary storage system up to an autonomous resilient dCache. As dCache systems grow, system administrators tend to combine those modes within a single instance. In the past this caused suboptimal behaviour of the load balancing mechanisms mainly because all steering parameters have been treated globally. With the next release, dCache instances may be partitioned, treating dCache subsections differently concerning hot spot detection and load balancing.

2.7 LHC Grid compatibility

In order to let the dCache interact with LCG grid middleware, it has to implement certain data control, data transport and information provider protocols. Concerning data movement, GsiFtp and a local, posix like method has to be provided, which is the case for dCache as described above. Transfer protocol negotiation, name space operations and space management is done by means of the Storage Resource Manager Protocol (SRM)[10]. The way global information on Storage Elements is distributed, varies among different types of grid middleware. dCache is compliant to the LCG methods and by implementing Clarens[22], is prepared to talk to OSG middleware as well.

References

1. DESY : http://www.desy.de
2. FERMI : http://www.fnal.gov
3. CERN : http://www.cern.ch
4. Rutherford Appleton Laboratory : http://www.cclrc.ac.uk/
5. Large Hadron Collider : http://lhc.web.cern.ch/lhc/
6. LHC Computing Grid : http://lcg.web.cern.ch/LCG/
7. Fermi Enstore http://www.fnal.gov/docs/products/enstore/
8. High Performance Storage System : http://www.hpss-collaboration.org/hpss/
9. Tivoli Storage Manager : http://www-306.ibm.com/software/tivoli/products/storage-mgr/
10. SRM : http://sdm.lbl.gov/srm-wg
11. CASTOR Storage Manager : http://castor.web.cern.ch/castor/
12. dCache Documentation : http://www.dcache.org
13. dCache, the Book : http://www.dcache.org/manuals/Book
14. GsiFtp http://www.globus.org/ datagrid/deliverables/gsiftp-tools.html
15. Secure Ftp : http://www.ietf.org/rfc/rfc2228.txt
16. NFS2 : http://www.ietf.org/rfc/rfc1094.txt
17. GridKA : http://www.gridka.de/
18. Cern CMS Experiment : http://cmsinfo.cern.ch
19. Grid GFAL http://lcg.web.cern.ch/LCG/peb/GTA/GTA-ES/Grid-File-AccessDesign-v1.0.doc
20. D-Grid, The GErman e-science program : http://www.d-grid.de
21. The Open Science Grid : http://www.opensciencegrid.org
22. JClarens : http://clarens.sourceforge.net/jclarens/
23. The Chimera Filesystem service : http://www.dcache.org/manuals/chimera-paper-chep06.pdf