Pflichenheft für Testsuiten.

Dr. Patrick Fuhrmann,
Owen Synge,
Elena Tews,
Dr. Christoph Wissing

DESY, Hamburg
Notkestrasse 85
22607 Hamburg
Germany

Editor : Patrick Fuhrmann, DESY
Last Modified : 02/18/07, 05:02:33 PM

Purpose of this document

This document describes the results of Task 1.4 of Subproject “The Scalable Storage Element” within work package I of the D-Grid, HEPCG project.

Coordination of the deliverable 1.4.

The goal if the deliverable has been to

1. Create a framework to run tests plug-ins which can be performed against a dCache system in order to check its proper functionality, BUG FIX regression, stability and performance. The prototype has been designed by the University of Freiburg.
2. Setup a reasonably large dCache instance. All official steps announced by dCache.ORG should be applied to this dCache system, e.g.: major updates, bug fixes and special configuration setups. It would be desirable that this instance is in use by 'real' applications. This has been done by the University of Dortmund (E5).
3. As a result of step 2 and with the experience from the international user community a set of plug-ins should be compiled which need to be run against the dCache in order to certify a new dCache version or release.
4. The test suite should be made available to the user community as a separate package in order to allow larger user communities to perform the certification testing by themselves and to possible feedback their own testing plug-ins to dCache.org.

Further reading

Further information like talks, papers, manuals as well as *dCache, the Book* are available on the dCache.ORG documentation web pages.

Introduction

This document summarizes important issues for planning, installation, operation and

developing the dCache Storage Element (SE) in the D-Grid context. A detailed description of the dCache system is given in "dCache, the Book", which is continuously updated. At various points this documents references the book for further reading.
In order to understand the following sections without reading the entire book, a short introduction to the dCache will be given here.
The dCache technology implements an advanced concept for managing large disk pools including a possible interface to HSM tape back-ends. The tape back-end is an option for larger centers and not a requirement.
The general concept for file handling is driven by the concept of "write-once, read many", i.e. a file has to created again if it should be modified. The very valuable for High Energy Physics, where the dCache is originally developed, because large data sets from experiments are produced once and then read several times afterwards for analysis. In order to efficiently access the data the dCache system can create additional copies of a file instance on other disk. Since this use pattern is also common for other disciplines dCache is used in other areas.
For the user is as transparent as possible and it is not required that the user needs to know about the exact location of the file. The files are presented in a directory tree, which looks like a mounted file system. The dCache keeps track of all replicas in the PNFS database, which one of the key components.
For the actual data access several protocols are available. dCache has its own protocol, DCAP (dCache Access Protocol), which mainly used for local data transfer, e.g. copying a file from the dCache to a CPU close by. For running the dCache in the Grid the most important data access protocol is Grid-FTP. In context of the D-Grid project another method, that is commonly used in HEP, is the xrootd technique. The components that actually run these access services are the so-called "doors". In order to have load balancing there might be several doors of a kind.
One more important component to run a dCache in the Grid is the"SRM-Interface". This interface is the entry point for the clients. It negotiates a proper transfer protocol and connection between the client and one door. Afterwards the actual transfer is performed independent of the SRM.
In order to make a Storage Element "visible" in the Grid, the SE needs to have a component named Information Provider, that communities the availability of the SE, static information like supported VOs and dynamic information like space resources. In terms of the dCache language also the disk pools are components. A pool can be a partition of a disk but also a volume of a large RAID-array.

Configuration Examples

The dCache SE is an extremely scalable technique that allows installations starting from a (good) fraction of a Tera-Byte hosted within a single machine up to the Peta-Byte scale distributed over hundreds of servers. In the following a few examples are described.

"All in one box"
For the smallest installation all components get deployed on a single server: PNFS database, SRM interface, Access doors, Information Provider and at least one Pool. Since the CPU(s) has to server all components this setup is only recommended for setups with very low access rates and small data transfers.

"Pools on Worker Node disks"
A rather popular deployment example uses the disk space in the compute nodes (Worker Nodes) of a farm. Now a days disks are rather cheap and large and only a fraction of the capacity is used by the WN itself and it can be therefore used to build a storage element by

deploying a dCache pool on each (or some) WN. The remaining components are then installed on a so called Head-Node. It is recommended to install the Doors on separate Nodes if there is more frequent access to the data. There are installation that use pools on several hundreds of Wns. Since the data are kept on less reliable disk this scenario is good example where the resilience option of dCache makes sense. In the resilience mode it made sure that at least a selectable number of copies (typically 2) is available and the failure of one disk would not lead to data loss.

"Setup with disk servers"
To build large storage elements data are hosted on large file servers. Usually dCache pools and Doors are installed on the file servers while SRM and PNFS database service remain on a Head Node. The capacity of such an installation can be increased easily by just adding new file servers. There are many midrange installations of this type running in production at various sites.

"Enterprise Installations"
The dCache storage elements scales up to the Peta-byte scale. For such an installation every (sub)component (not all were introduced in this document) are deployed on single host. The probably largest dCache SE is running at FNAL for the US-CMS Tier-1 center. The full instance is spread over several hundred servers of various kinds.

Installation

The description of the actual installation process is not topic of this document. This is described elsewhere. At the time of the beginning of the D-Grid project the installation of a dCache SE was basically a manual procedure. Along a few notes the administrator had to adapt several templates and configuration files. Also in context of the D-Grid project this procedure has been simplified a
lot. Meanwhile it is sufficient to adjust only a very few variables in a global configuration file, for the standard gLite installation tool YAIM that is also used to install other gLite components. Despite this simple procedure it is possible to deploy all scenarios mentioned above with the exception of the very complex ones. Matching the large parameter space and the requirements of particular site going for a petabyte installation is a complex that will always intervention from an experienced administrator. For those purposes the manual installation and many details about the
components are described in the Book.

Upgrade

Once installed it remains an important task for a site to install updates to the system. From the perspective of the developers it is crucial to define proper conditions which are valid to apply a dedicated update. The developers usually provide updates that fix bugs for the most recent and second most recent version. If an update needs an adjustment of configuration files the YAIM component is kept in synchronization. One critical update during the D-Grid project phase should mentioned explicitly. It was critical because it involved a change in the technology of the data base back-end used for PNFS. Failures of this upgrade would result in the worst case in a complete data lost of the SE instance. The preparation and

documentation of this upgrade has been carefully put forward by the developers. The procedure is described in detail in the Book and was verified by several sites, including two involved in the D-Grid project.

Testing

Testing is important on two sides, one is the site that runs the SE instance and the other is the development team. For the site testing ensures that the installation provides certain functionalities, e.g. access to the data via different protocols.

For the developers a testing procedure is needed to validated new releases of the software. To fulfill both requirements a Test Suite for dCache has been developed. The use of plugins allows an easy extension of the Test Suite since an increased need of tests is expected over the time. The Test Suite has become a component of a new build frame for the dCache system, that allows an automatic, packaging and testing of the product. With the flexible Test Suite it has become to test every new release against known bugs that should have been fixed.

Status of the Test-Suite as of Jan 2007

Based on the experience of the test suite prototype created by the University of Freiburg, dCache.ORG designed large Test Suite framework which is part of the generic distribution of dCache.org. The further work on this sub-package is shared between D-Grid and the US Open Science Grid. Large LCG Sites or communities are making use of the tests suite in order to re-certify a new dCache version. For dCache.ORG the test-suite is part of the automated production chain which

1. Checks-out the Software from the Code Repository
2. Compiles all components
3. Packages the system
4. Distributes the system on a set of virtual machines
5. Auto-configures the system by YAIM
6. Runs the Test Suite against this instance.
7. If 1 -6 doesn't error any error conditions, the binary package is published on the dCache.ORG web pages and in the DESY apt repository.
8. For major resp. security updates the package moves into the CERN apt repository for being tested in the pre-production systems.
9. After a OK from the pre-production sits the package finally moves into the CERN 'stable' APT repository.

As for now the following plug-ins are in us at dCache.ORG in order to certify a new dCache release:

1. **Basic gloubus_url_copy test.** The system runs FTP put and get's . This implicitly checks authentication and authorization. The plug-in tries to write data into resticted file system areas, tries to read files which are protected and tries to list system directories which are not part of the dCache name space and shouldn't be readable.
2. **Multiple Stream globus_url_copy**. FTP is operated in passive and active mode, with 1 to 20 streams.
3. **gsiDcapProtocol** : dCap is the native 'posix like' dCache access protocol. The dCap copy program 'dccp' as well as applications linked againts the dCap library are

tested. More over the dCap interposition library (PRELOAD) is tested.

4. **Pnfs File-System functional tests**. The pnfs filesystem is tested against typical filesystem operations.
5. **dCache component tests.** dCache components are tested individually. This includes, PnfsManager, PoolManager, gPlazma (Authorization) Cell, Pools, Billing module e.t.c)
6. **dCache admin interface** Is the admin interface up and functional (ssh, gui)
7. Is **dCache LDAP** system up and returning reasonable replies. This is essentially the LCG information system, providing the GLUE schema information.
8. Is **dCache web service** activate and reports correct results.
9. **xRoot testing.** I/O tests and security tests are performed on the xRoot protocol.
10. **SRM testing.** The srm is tested for all available functions. Those test only cover the PRE SRM 2.2 functionality. For SRM 2.2 there are some testing frames availabe from various sites which we hope to be able to integrate into our test frame.
11. Check of **correctness of checksum calculations** for protocols which support checksumming.

Importance of the Test Suite

The test suite is integral part of the dCache release process. No dCache version is released if it hasn't passed the test suites.
We are currently in the process of upgrading the test suite hardware system by 200 machines to integrate serious performance test plug-ins.
The test suite is a separate package in order to allow other sites to run tests against new dCache releases.