ISTITUTO NAZIONALE DI FISICA NUCLEARE

Sezione di Perugia

INFN/TC-10/01
February 9, 2010

INSTALLATION AND CONFIGURATION OF DCACHE STORAGE SYSTEM WITH INFNGRID PROFILE FOR GLITE MIDDLEWARE

Hassen Riahi[1], Flavio Vella[2]
[1)] *INFN, Sezione di Perugia, Dip. Matematica ed Informatica, Univ. di Perugia, Italy*
[2)] *INFN, Sezione di Perugia, Italy*

Abstract

The CMS experiment is expected to produce a few Peta Bytes of data per year and distribute them globally. Within the CMS computing infrastructure, some user tasks can be performed at level of CMS Tier-3 sites without any supports or sites reliability guaranteed by CMS collaboration. In this note, we present the integration of dCache storage system and Glite middleware to allow users to perform physics analysis.

We describe briefly, in this paper, Perugia data center and its distinctive features with an overview of dCache and Glite middleware. We show in detail the followed steps and the solutions to possible problems. Finally, we describe perspectives in integrating dCache storage system with Glite middleware.

PACS: 89.30.Gg

Published by **SIS-Pubblicazioni**
Laboratori Nazionali di Frascati

1 Introduction

The computational farm of INFN-Perugia was setup in 2004 from the unification of local computational resources from different experiments with the aim to share and optimize them. Many of these experiments use the local Grid and also the World-wide LHC Computing Grid (WLCG)[1]/EGEE[2] for their researchs. In particular for the Compact Muon Solenoid (CMS) experiment[3] users, the Grid is needed to transfer remote data for the following local analysis. It was therefore needed a storage system with Grid integration support and able to deliver optimal performances: the final choice was dCache[4] which allows, better than other storage systems, to satisfy the needs of CMS users in Perugia.

Actually, the data center offers 230 cores and about 30 TeraByte of storage capacity in private network. It is used locally by 30 users. Many services as computing element, user interfaces, worker nodes and network services run on virtual machines based on Xen technology[5]. We have chosen to use virtual machines for the following reasons:

1. Optimization of hardware resources.
2. More efficient services management
3. Intrinsic aspects of high reliability and availability

Considering the benefits of such solution to the local Perugia site configuration, it was chosen to virtualize the dCache master node.

2 DCache and Glite overview

dCache is a sophisticated system developed jointly by DESY and Fermilab. It allows transparent access to files on disk or stored on magnetic tape drives in hierarchical storage managers (hsms). dCache has proven to be capable of storing and exchanging hundreds of Terabytes of data, transparently distributed among dozens of disk storage nodes. Access to the data is provided by various ftp dialects, including gridftp, as well as by a native protocol (dccp), offering regular file system operations.[6]

The GLite distribution is an integrated set of components designed to enable resource sharing. In other words, this is the middleware for building a Grid. The distribution model is to build different services from these components and then provide easy installation and configuration for different platforms. The GLite middleware is developed by the EGEE project and pulls together contributions from many other projects, such as LCG and VDT[7]. It is currently deployed on hundreds of sites as part of the EGEE project and enables global science in a number of disciplines particularly for LCG project.[8]

3 Installation procedure

dCache was not originally developed for the Grid and its setup, as a Glite service, is not yet optimized. This is the reason why dCache gLite profile is not used for installing and configuring it, forcing many sites to customize their own solution. We have decided to debug the INFN Grid[8] release of dCache gLite profile in order to simplify and automate as much as possible the installation.

1. Repository

   To install dCache admin node, the following repositories will be used:

   ```
   ca.repo cernextra.repo cern.repo cernupdate.repo
   dag.repo glitebdii.repo glitegeneric.repo
   glitese_dcache_admin_postgres.repo glitese_dcache_info.repo
   ig.repo jpackage.repo
   ```

2. Java[9]

   Dcache depends on Java Virtual Machines

   ```
   yum clean all
   yum install java1.5.0suncompat
   ```

3. Postgres[10] installation

   dCache uses Postgres to manage mainly PNFS databases. To install PostgresServer, the following repositories must be disabled:

   ```
   /etc/yum.repos.d/cern.repo
   /etc/yum.repos.d/cern-update.repo

   yum install postgresql-server
   ```

4. dCache, Certification authority (CA), host certificate and resource-BDII (Berkley Database Information Index)[11] packages installation

   The previous repositories must be enabled again before running the following installation commands:

```
yum install ig_SE_dcache_admin_postgres
yum install lcg-CA
yum install ig_bdii

chmod 644 /etc/grid-security/hostcert.pem
chmod 400 /etc/grid-security/hostkey.pem
```

5. dCache configuration

   To configure dCache node with glite middleware, three files must be edited:

   (a) dCache section in ig-site-info.def configuration file

   (b) Configurations files of dCache itself:

   ```
   /opt/d-cache/etc/node_config
   /opt/dcache/dCacheSetup
   ```

6. Network file system (NFS)[12]

   The NFS server must run on the machine where the virtual file system PNFS[13] will be mounted:

   ```
   start portmap
   start nfs
   ```

7. ig-site-info.def file

   This is the file where site related parameters are defined and used for Grid nodes configurations. To define a dCache admin node, the following variables must be set in dCache section:

   ```
   DCACHE_ADMIN="adminNode.DnsDomainName"
   DCACHE_POOLS="Pool1.domain: Size :/path/to/pool1"
   DCACHE_POOLS="Pool2.domain: Size :/path/to/pool2"
   ```

   Ports that will be used in the admin node and doors to run on:

   ```
   DCACHE_PORT_RANGE="20000,25000"
   DCACHE_DOOR_SRM="door_node1[:port]"
   ```

```
DCACHE_DOOR_GSIFTP="door_node1[:port] door_node2[:port]"
DCACHE_DOOR_GSIDCAP="door_node1[:port] door_node2[:port]"
DCACHE_DOOR_DCAP="door_node1[:port] door_node2[:port]"
DCACHE_DOOR_ROOT=" door_node1[:port] door_node2[:port] "
```

In order to use PNFS, it is needed to set the following value in /opt/d-cache/etc/node-config file:

```
NAMESPACE=pnfs
```

8. Node configuration

   Now, it is possible to configure the node using ig-yaim command:

```
/opt/glite/yaim/bin/ig_yaim -r -s ig-site-info.def \
-n ig_SE_dcache_admin_postgres -f config_edgusers
/opt/glite/yaim/bin/ig_yaim -c -s ig-site-info.def \
-n ig_SE_dcache_admin_postgres
```

   But the Chimera[14] namespace will be installed rather than PNFS namespace. It is than needed to add the following variables in ig-site-info.def file:

```
DCACHE_NAME_SERVER="adminNode.domain"
DCACHE_PNFS_SERVER="adminNode.domain"
```

   Configuration with ig-yaim command (second iteration):

```
/opt/glite/yaim/bin/ig_yaim -c -s ig-site-info.def
-n ig_SE_dcache_admin_postgres
```

   Also in this case, PNFS namespace is mounted as Chimera namespace. To correctly install it, is then needed to run:

```
cd /pnfs
ln -s /pnfs/fs ftpBase
ln -s fs/usr DnsDomainName
```

   Now, dCache admin node can be configured successfully using ig-yaim command:

```
/opt/glite/yaim/bin/ig_yaim -c -s ig-site-info.def \
-n ig_SE_dcache_admin_postgres
```

9. Logs checking

```
tail -f /var/log/*.log
```

10. Setup checking

    Check the correctness of the information collected by the site-BDII from various resource-BDII:

```
ldapsearch -x -H ldap://site-BDII.DnsDomainName:2170 -b \
mds-vo-name=SiteName,o=grid
```

    Query the storage-BDII:

```
ldapsearch -x -H ldap://dCacheAdminNode.DnsDomainName:2170 -b \
mds-vo-name=SiteName,o=grid
```

    Check the installation using the transfer commands:

```
srmcp \
srm://dCacheSRMDoorHost:8443/pnfs/DnsDomainName\
/data/yourVO/user/yourDir/my-test-file
file:////tmp/test4.tmp
srmls srm://dCacheSRMDoorHost:8443/pnfs/pg.infn.it/data

globus-url-copy -dbg \
gsiftp://dCacheGsiftpDoorHost/pnfs/DnsDomainName\
/data/yourVO/user/yourDir/my-test-file
file:////tmp/test4.tmp

dccp -d 23 /pnfs/yourVO/user/yourDir/my-test-file.
```

11. Site's specific configuration

    Due to its multiple network interfaces, the Storage Resource Manager (SRM) interface returns a wrong TURL, which creates a certificate conflict . This problem is reported when asking to SRM to use GSIFTP protocol.

    The solution is to add in the gridftpdoor.batch an explicit configuration that can tell to SRM which is your first network interface:

```
create dmg.cells.services.login.LoginManager GFTP-${thisHostname} \
"${gsiFtpPortNumber} \
-export \
-listen=${Your first interface}\
diskCacheV111.doors.GsiFtpDoorV1 \
-prot=raw \
-clientDataPortRange=${clientDataPortRange} \
```

12. Upgrade

    When upgrading dCache, the number of experiments in PNFS space must be kept (or increased) to avoid issues with PNFS Database. In the last case, the consistency of databases in PNFS must be checked.

4 Conclusion

In this paper, it has been described a procedure to install dCache with glite middleware proposing several solutions to problems that can be met and giving suggestions based on Perugia Grid site experience.

Using this procedure, dCache can be easily maintained and fully integrated within the gLite middleware.

This work can be generalized to different tiers and storage systems. The procedure can be extended to the one where the types of file system and storage can be added directly to the gLite middleware configuration.

References

[1] LHC Computing Grid (LCG), Web Page, http://lcg.web.cern.ch/LCG/ and LCG Computing Grid - Technical Design Report, LCG-TDR-001 CERN/LHCC 2005-024, (2005)

[2] EGEE: http://www.eu-egee.org/

[3] The CMS Collaboration, JINST 3 (2008) S08004.

[4] dCache: http://www.dcache.org/

[5] XEN: http://www.xen.org/

[6] Managed Data Storage and Data Access Services for Data Grids, M. Ernst, P. Fuhrmann, T. Mkrtchyan, DESY, Hamburg, Germany J. Bakken, I. Fisk, T. Perelmutov, D. Petravick, Chep04.
Link: http://www.dcache.org/manuals/chep04.michael.paper.pdf

[7] Building and testing a production quality grid software distribution for the Open Science Grid, by Alain Roy et. al., 2009 Journal of Physics: Conference Series. 180 012052 (6pp).

[8] http://www.italiangrid.org/middleware

[9] James Gosling, W Joy and G Steele, The Java language specification, Published by Addison Wesley ,ISBN: 0-201-63451-1, 2000

[10] Postgres: http://www.postgresql.org/

[11] BDII: https://twiki.cern.ch/twiki/bin/view/EGEE/BDIIv4

[12] Brent Callagham, NFS illustrated, Published by Addison Wesley, ISBN:0-201-32570-5, 2000

[13] PNFS: http://www-pnfs.desy.de/

[14] Chimera - a new, fast, extensible and Grid enabled namespace service, Mr. Mkrtchyan Tigran, Dr. Fuhrmann Patrick, Mr. Gasthuber Martin DESY, Hamburg, Germany, Chep06