Table of Contents

PandaDataServiceTest

Introduction

This page is meant to provide information on installing the test versions of DQ2 components to support DQ2 v0.2.12 for USATLAS sites. The instructions provide details on installing DQ2 0.2.12 site services and a Local Replica Catalog (LRC).

DQ2 Site Services

System Requirements For Testing

A machine for running DQ2 site-services and the LRC

Testing so far has been on 32-bit SLC3/RHEL3 machines and we are beginning to work with SLC4. At present DQ2 site services is not working with SLC4. The current recommendation is that the LRC be on a 32bit SLC3 machine. The DQ2 site services will occupy around 500MB of disk space while the LRC will occupy ~1.5GB

A gridftp server

This can be an existing gridftp server or you can install a copy of the VDT on your test machine.

Storage

You should have adequate storage resources available to the gridftp server to cache files while testing the DQ2 installation. This should be on the order of 100GB.

Pre-installation Steps

Grid Certificates

Neither DQ2-site services or the LRC require the use of a host certificate for the machine, but your gridftp server will need one. DQ2 requires the use of a user certificate. This certificate must be registered in the CERN VOMS for proper interaction with the BNL FTS server.

entry

Your test site should be listed in the TiersOfATLASCache.py file used by DQ2. You should identify the following information:

- DQ2 Site NAME (e.g. UTA_TEST, BU_TEST)
- LRC Contact information (Host, port, LRC Read account/password)
- Base URL for storage (host,path e.g. gsiftp://gk02.swt2.uta.edu/ifs1/dq2_test/Storage)

Once you identify this information, please send the data to someone who can modify the TOACache file. Wensheng and Patrick (others may have access as well) can update the file.

FTS Channel

A channel will need to be created within the BNL FTS server to support transfers to your test site. If the gridftp server is different from your production system you should contact Hiro or Wensheng to get the channel created.

Decide on which components to use

Beginning with this release, you have the option of using preinstalled versions of python and MySQL to support DQ2 site services and the LRC (still needs work). The python should be version 2.3.4 or later and MySQL should be 4.0.24 or later.

Expected commands

The installation process expects to find certain commands in $PATH to succeed. The commands are:

- mysql_config
- curl-config

Installing

You will need to install pacman. Pacman versions 3.18.5 and 3.19 were used for most test installations. Older versions may be compatible but have not been tested. The latest version is available at:
http://physics.bu.edu/pacman/sample_cache/tarballs/pacman-latest.tar.gz

DQ2 site services

If you wish to use an installed version of python you should verify that it is the first python found in $PATH If you wish to use an installed version of MySQL, you should make sure that the MySQL binaries are in $PATH and $LD_LIBRARY_PATH contains the path to the MySQL libraries or is in the default library search path.

The pacman command to get things rolling is:

```
 pacman -get http://www.atlas-swt2.org/pacman/DQ2:DQ2-Test
```

You will be prompted several times during the installation. The first two questions concerning trusting the pacman cache at UTA and for the VDT components. The next question will ask for your DQ2 Site name. The last two question will ask if you want to install private versions of python and MySQL. If you answer "Y" to these questions, the installation process will download and configure these components. If you wish to use system components you should answer with a "N"

The final messages should be:

```
Installation has completed
see post_install/README for more information
```

Note:
To setup the installation, you should source:
config/SITE_NAME/environment.sh
This has changed from previous installations.

Post installation Configuration

If you are using a pre-installed version of MySQL, the next section will show you how to make the MySQL server ready for use with DQ2. If pacman installed a version of MySQL this is not necessary. Production servers may need to modify the TOA timeout but this is not expected to be necessary for test installations. An LRC needs to be installed and configured. The post install steps of the LRC will provide information on configuring DQ2 to use the LRC.

MySQL

If you installed a private version of MySQL for DQ2 there are no configuration actions necessary. However, if you used a pre-installed version of MySQL, there are some extra steps needed to make the DQ2 installation functional:

Create an account in for DQ2 operations

The appropriate mysql syntax should look something like:

```
GRANT ALL PRIVILEGES ON queued_transfers_XXXXX.* TO dq2user@localhost IDENTIFIED BY 'dqpwd';
```

*Notes: *
You should replace XXXXX with your DQ2 site name (e.g. UTA_TEST)
The example above assumes that the DB service is installed on the same machine as the DQ2 services. If this is not the case, replace localhost with name of the machine hosting DQ2.
You may want to specify that the dq2user can access the system from the host returned by `hostname`. This can be done with an additional GRANT statement
The account/password is not required to be dq2user/dqpwd. You should change these to more appropriate values if your DB service can be accessed by anyone other than the installer.

Create the database needed by DQ2

The mysql command to use should look like:

```
CREATE DATABASE IF NOT EXISTS queued_transfers_$DQ2_SITE;
```

Modify the DQ2 configuration

The file $DQ2_LOCATION/config/$DQ2_SITE/executor_conf.py contains a variable that controls how DQ2 contacts the DB service. The installation will contain the following lines (the second is commented out)

```
queued_transfers_cat =
'mysql://dq2user:dqpwd@/queued_transfers_XXXX?unix_socket=/data/mysql.sock'
#queued_transfers_cat = 'mysql://dq2user:dqpwd@hostname:3306/queued_transfers_XXXX'
```

Only one of these forms should be enabled. The first is appropriate if you want to connect to a db service that is local to DQ2 site host AND you are attempting to connect through the file socket. If you use this form, you should verify the username/password AND path to the file socket is correct.

The second form is appropriate when DQ2 must connect to a remote DB service or a local DB service will be used and access is through a TCP port. You should verify that the username/password/hostname are correct for your situation. You should protect this file as read-only to the installation account:

```
chmod 600 $DQ2_LOCATION/config/$DQ2_SITE/executor_conf.py
```

Populate the database with tables

This is the first chance to run any code that is DQ2 related. Do the following:

- Verify that $PATH $LD_LIBRARY_PATH are correct for the versions of python and MySQL
- Verify that the MySQL server is running
- Source $DQ2_LOCATION/config/SITE_NAME/environment.sh (This has changed from previous pacman installations)
- python $DQ2_LOCATION/DQ2/siteservices/executor/classic/RecreateDatabase.py

Create the lrcs_conf.py file

This is a DQ2 configuration file that provides contact details for the LRC. The file should look something like:

```
lrcs = {'XXXX' : 'mysql://lrc_writer:writePWD@your_host_name:3306/localreplicas',}
```

Where XXXX is your site name, lrc_writer is the DB account that can write to the LRC, writePWD is the DB account's password, your_host_name is the host name of the machine hosting the LRC DB, and 3306 is the port used by the LRC DB.

Logging information

DQ2 logs information while operating to $DQ2_LOCATION/config/SITE/subscriptions.log and can be somewhat customized through the settings in the executor_conf.py file (same directory as the log). The items from the configuration file that have an affect are:

```
site_log = '$DQ2_LOCATION/config/SITE/subscriptions.log'
log_level = logging.DEBUG
MB = 10 ** 6
log_max_bytes = 600 * MB
```

The first entry determines what file is used for logging. Unless otherwise specified, it is highly recommended to use the entry as generated during the pacman installation. The next entry determines the level of information that is contained in the log files. Your choices here are logging.DEBUG, logging.INFO, logging.ERROR. The current recommendation is to keep the log_level at the DEBUG setting. The next two entries determine how DQ2 manages the log file to provide a limit on the size of the log. The example settings show that the log file is allowed to grow to 600 MB before the file is rotated. The files are rotated by appending a numeral (1, 2) to the existing log file. The instrinsic behavior of DQ2 is to create up to two old logs, before information is thrown out. This implies that DQ2 will use up 1.8GB to maintain logging information

The pacman installation provides an alternative to the log rotation mechanism built into DQ2. The installation will use the system command logrotate to manage the size of the log file. There are two parts to this management. An entry in the cron file will call logrotate, nightly, at midnight to create daily logs of DQ2 operation. A configuation file ($DQ2_LOCATION/util/log.conf) is passed to logrotate to control how old logs are managed. The file distributed with the pacman installation looks like:

```
rotate 7
daily
missingok
compress
copytruncate

$DQ2_INSTALLATION/config/MYSITE/subscriptions.log {}
```

The result is that old log files are compressed with gzip and kept for seven days. If you want to keep more information, adjust the value for the rotate item to a larger number. As a guide, the compression appears to provide about a 20:1 savings in space.

Starting and Stopping services

Starting DQ2 Services

There is a file created in $DQ2_LOCATION/util/agents.cron that are appropriate for starting/DQ2 services. This file may need to be merged with a crontab generated by the LRC installation. If you are using a pre-installed version of MySQL, you should delete the lines:

```
# DQ2_MySQL
```

```
*/20 * * * * (source $DQ2_SETUP; $DQ2_MYSQL/mysql.sh start > /dev/null 2>&1)
```

The Fetcher process is configured to run every five minutes but the exact time is randomized to ease the burden on the Central Catalog server.

If you want to run the DQ2 services outside of the crontab file, you should run the following (assuming that your environment is correct):

```
python $DQ2_LOCATION/DQ2/siteservices/executor/classic/LoggingServer.py
```

```
python $DQ2_LOCATION/DQ2/siteservices/executor/classic/SpawnAgents.py
```

```
python $DQ2_LOCATION/DQ2/siteservices/executor/classic/AssignAgents.py
```

You can then run the Fetcher manually with:

```
 python $DQ2_LOCATION/DQ2/siteservices/executor/classic/Fetcher.py YOUR_SITE_NAME
```

Stopping Services

Stopping the DQ2 site services is more complicated in v0.2.12. You need to kill a process by hand, run a python script, and then kill two remaining processes.

1.) Remove the crontab if enabled.

2.)First find the process responsible for executing SpawnAgents.py

```
ps -auxw | grep SpawnAgents
```

and kill the appropriate process Execute the python script.

3.)Next, kill off the various agents

```
python $DQ2_LOCATION/DQ2/siteservices/executor/classic/StopAgents.py
```

This will kill off the various agents within DQ2. It may take some time, but you should be left with two running processes, The Logger and AssignAgents.

4.) Lastly, You should kill the AssignAgents process, then the Logger process using the same mechanism as SpawnAgents.

Notes:

Upgrading an existing DQ2 0.2.11 site

Pre-instalation steps

Backup your crontab (just in case anything goes awry)

Determine how your current DQ2 site services uses MySQL. It is likely that your DQ2 site is using a version of MySQL that was installed when DQ2 was first configured. You should be able to determine which MySQL instance is supporting site services by finding which MySQL versions are running on your system, finding which MySQL instance contains a database named queued_transfers_SITE, finding if your crontab file is starting an MySQL instance.

If you are using a MySQL instance that was installed for DQ2, you can perform the installation steps above while your existing service runs. If you are using a system supplied MySQL instance, and you are going to reuse the same MySQL instance for the new site services, you should stop your dq2 services before upgrading.

Installing DQ2 0.2.12

Choose a directory to hold the new installation, e.g. /opt/dq2_0.2.12. This MUST be separate from your existing DQ2 directory!

Perform the installation above. It is recommended to let pacman install MySQL unless you have a pressing need to use a different version, or your LRC is using a different version. If you are reusing a system MySQL to support the DQ2 installation, then you should reuse the value for queued_transfers_cat in $DQ2_LOCATION/config/SITE/executor_conf.py.

When the pacman installation is complete you should look through the README file concerning local modifications to the DQ2 installation.

There is a step necessary to configure the new DQ2 site-services to use the LRC database. You should copy the lrc_confs.py file from the prevuous DQ2 installation into the new installation. Th file is found in $DQ2_LOCATION/config/SITE/. If you don't have an lrcs_conf.py, you can continue, but you should learn why you should have this file.

Modifications for the LRC

The web service component of the LRC uses the DQ2 installation to read information from the TiersOfATLASCache.py file to determine information about your site. It would be good form to modify your LRC configuration to read information from the new DQ2 installation area. This should be considered a requirement if your site rotates storage locations by locally modifying the TiersOfATLASCache.py.

The necessary change will depend on how your LRC is configured to start the httpd server. In some installations, there is a specific file for setting up the environment for the httpd server. This should be located in $DQ2_LOCATION/setup_httpd.sh (where DQ2_LOCATION is your old DQ2 path). The file should contain a line that looks like:

```
export PYTHONPATH=/opt/dq2/DQ2/:$PYTHONPATH
```

where "/opt/dq2" is the old DQ2_LOCATION. This line should be updated to use the new $DQ2_LOCATION:

```
export PYTHONPATH=/opt/dq2_0.2.12/DQ2/:$PYTHONPATH
```

If you don't have a setup_httpd.sh file, you will need to determine which file is used to setup the environment for the httpd daemon. Your existing crontab file should contain a line that periodically restarts httpd. You should identify and modify the appropriate line that adds your previous $DQ2_LOCATION/DQ2 path to $PYTHONPATH.

Once you have modified the file configuraiton file, you should restart the web-service to pickup the new value. From a clean environment:

```
. OLD_DQ2_LOCATION/setup_httpd.sh
```

```
OLD_DQ2_LOCATION/httpd/bin/apachectl -k stop
OLD_DQ2_LOCATION/httpd/bin/apachectl -k start
```

Modifications to the crontab

The new DQ2 installation creates a crontab file that can be used to start the DQ2 site-services. This file will need to be modified so that it can also start the LRC services.

Make a backup of your new crontab file.

Find the entries from your old DQ2 crontab file that are responsible for starting the MySQL service (for the LRC) and the web server. They should look like:

```
DQ2_LOCATION=/opt/dq2
DQ2_SETUP=/opt/dq2/setup.sh
*/5 * * * * (source $DQ2_SETUP; source $DQ2_LOCATION/externals/mysql/4.0.24/slc3_ia32_gcc323/mysq
*/20 * * * * (source $DQ2_LOCATION/setup_httpd.sh >/dev/null 2>&1; cd $DQ2_LOCATION;  httpd/bin/a
```

You should add these lines to your new crontab files, but modify the names `DQ2_LOCATION` and `DQ2_SETUP` to `OLD_DQ2_LOCATION` and `OLD_DQ2_SETUP`, respectively.

Install the new crontab.

Major updates:
-- Main.mcguigan - 12 Jan 2007

Responsible: PatrickMcGuigan

Never reviewed

This topic: PanDA > PandaDataServiceTest
Topic revision: r8 - 2007-02-27 - PatrickMcGuigan