dCache Extension Points

The talk Patrick is afraid of

Gerd Behrmann

behrmann@nordu.net

Extension points

- gplazma2
  - auth plugins
  - mapping plugins
  - account plugins
  - session plugins
  - identity plugins
- pool manager
  - pool selection unit plugins
  - partition plugins
- pool
  - pool file store plugins
  - pool meta data store plugins
  - (pool repository)
- xrootd
  - authentication plugins
  - authorization plugins
  - mapping plugins
- name space
  - pnfs manager plugins
  - chimera SQL dialect
  - chimera file system provider
- Services plugins
- Property plugins

Installing plugins

- Default plugin locations

  ```
  /usr/share/dcache/plugins
  /usr/local/share/dcache/plugins
  ```

- Subdirectories are searched for
  - Custom JAR archives (.jar)
  - Custom defaults (.properties)
  - Custom services (.batch)

- `/usr/share/dcache/plugins/myplugin-1.0.0/`

  ```
  myplugin.properties
  myplugin.batch
  myplugin.jar
  ```

New service

- myplugin.jar

```
class org.example.MyCell extends CellAdapter
```

- myplugin.properties

```
myplugin.mySetting=foobar
myplugin/cell.name=mycell
```

- myplugin.batch

```
create org.example.MyCell ${cell.name} \
       “${myplugin.mySetting}”
```

- dcache.conf

```
myplugin.mySetting=blabla
```

- Layout file

```
[yourDomain/myplugin]

cell.name=yourcell
```

Class plugin

- myplugin.jar

```
class org.example.MyRepository
    implements MetaDataStore
```

- myplugin.properties

```
metaDataRepository=org.example.MyRepository
```

ServiceLoader plugin

- myplugin.jar

  -
    ```
    class org.example.MyAuthorizationProvider
     implements AuthorizationProvider {
       createFactory(pluginName, properties)
       {
          if (pluginName == “myplugin”) {
            ....properties[“myplugin.mySetting”]....
            return new MyAuthorizationFactory(...)
          }
       }
    }
    ```

  -
    ```
    META-INF/services/
     org.dcache.xrootd.plugins.AuthorizationProvider
    ```
- org.dcache.xrootd.plugins.AuthorizationProvider
  ```
  org.example.MyAuthorizationProvider
  ```
- myplugin.properties
  ```
  myplugin.mySetting=foobar
  ```
- dcache.conf
  ```
  myplugin.mySetting=blabla
  xrootdAuthzPlugin=myplugin
  ```

MyAuthorizationFactory

```
public class MyAuthorizationFactory
    implements AuthorizationFactory
{
    public String getName()
    {
        return “myplugin”;
    }

    public String getDescription()
    {
        return "My xrootd mapping plugin";
    }

    public MyAuthorizationHandler createHandler()
    {
        return new MyAuthorizationHandler();
    }
}
```

MyAuthorizationFactory

```
public class MyAuthorizationFactory
    implements AuthorizationFactory
{
    public String getName()
    {
        return “myplugin”;
    }

    public String getDescription()
    {
        return "My xrootd mapping plugin";
    }

    public MyAuthorizationHandler createHandler()
    {
        return new MyAuthorizationHandler();
    }
}
```

MyAuthorizationHandler

```
public class MyAuthorizationHandler
    implements AuthorizationHandler
{
    public String authorize(
        Subject subject,
        InetSocketAddress localAddress,
        InetSocketAddress remoteAddress,
        String path,
        Map<String, String> opaque,
        int request,
        FilePerm mode)
    {
       return path.replaceFirst(
              “^/global/cms/MYSITE(/|$)”,
              “/pnfs/mysite.dk/data/cms/foo$1”);
    }
}
```

Plugin ideas

Network aware load balancing

Partition manager

Write a network aware pool selection algorithm; a distributed dCache like the one at NDGF should ideally take the network topology, distances, and bandwidth into account when choosing a pool.

Pool group by convention

Pool selection unit

Write a plugin that relies on pool naming conventions to group pools into pool groups, assign them to VOs, detect dedicated write pools/read pools/p2p pools, etc. This would allow you to completely skip the PSU configuration. Most sites have a naming convention like this already.

Plugin ideas

Multi-partition pools

File store

Write a plugin that supports storing files in multiple partitions.

Hierarchical data directory

File store

Write a plugin that organizes files in the data directory into “bucks” to avoid large directories.

Pool meta data in PostgreSQL

Meta data store

Write a plugin that stores the meta data in a relational database (eg postgresql).

Pool repository in Casandra DB

Pool repository

Write a plugin that stores pool data and meta data in Casandra DB (a distributed database).

Plugin ideas

Replace xrootd path prefixes
xrootd mapping

Replace path prefixes with site local prefixes according to a mapping file.

Replace xrootd path prefixes
xrootd mapping

Replace path prefixes through callout to external catalog service.

Plugin ideas

Distributed name space in Casandra DB

Name space

Write a name space backend that stores the dCache name space in a distributed database to elliminate the single point of failure.

Auto-import files

Name space

Write a name space wrapper that detects files accesses and triggers some action, eg silently downloads the file from another system.

Overlay name spaces

Name space

Write a name space plugin that “overlays” another name space and allows external storage systems to be “mounted”.

Plugin ideas

scp/sftp door

Door and mover

Write a plugin providing door that supports scp and sftp.

Extend it by allowing the user to login through ssh and expose a minimal command line interface to ls, cp, mv, rm, upload and download files.

ssh plugin for gPlazma

gPlazma

Write a gPlazma authentication and mapping plugin to support ssh public keys.

Wish list

- Write Maven archetypes for the different types of plugins.
- Write “hello world” versions of the different plugin types.
- Move all plugin interfaces to use ServiceLoader.
- Versioning mechanism for plugins.
- Repository for plugins.

Warning

- We do **not** guarantee a stable interface for plugins
- We will not change the interface for fun
- but if a new feature, or a bug fix requires it then it will be changed
- we may also change it to improve consistency