Setting up Castor 2

Demo-scale setup of all Castor components

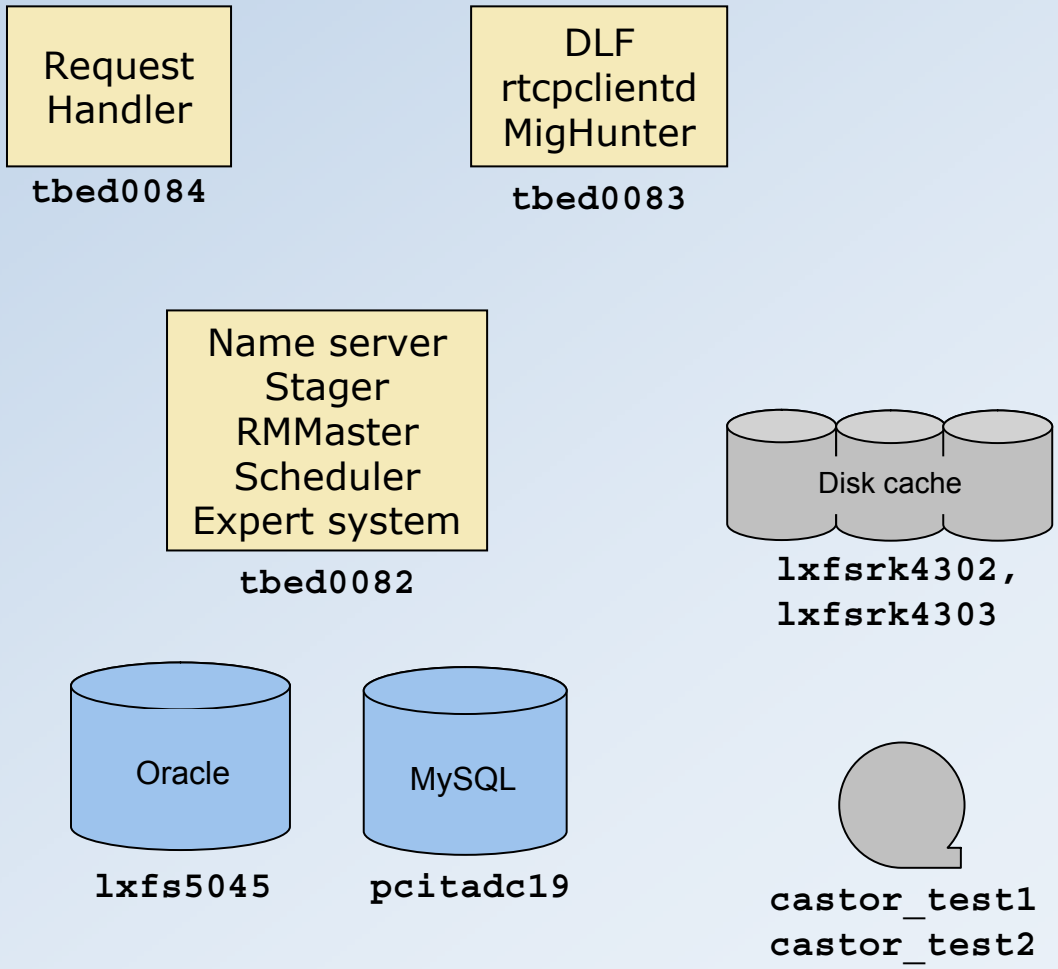
Giuseppe Lo Presti
CERN/IT/FIO & INFN/CNAF

giuseppe.lopresti@cern.ch

Castor External Operation Workshop, CERN, June 13-14, 2005

Castor components overview

- Databases setup
  - Name server
  - Distributed Logging Facility (DLF)
  - Catalog DB
  - GC
- Main daemons setup
  - Request Handler
  - Stager
  - Resource Manager Master
  - Scheduler (LSF or MAUI)
  - Expert system
  - MigHunter
  - DiskServer
  - TapeServer



Name server setup

- 1. Install software from RPM
  - castor-ns-server
- 2. Edit configuration script `/etc/castor/NSCONFIG`
  - castor/xxxx@castor_nsdb (user/passwd@db)
- 3. Create DB schema
  - `ns/Cns_oracle_tbl.sql` or `ns/Cns_mysql_tbl.sql`
- 4. Start NS service
  - `$ service castor-ns-server start`

DLF setup

- 1. Install software from RPM
  - castor-dlf-server
- 2. Edit configuration script `/etc/castor/DLFCONFIG`
  - dlf/xxxx@castor_dlfdb (user/passwd@db)
- 3. create the DLF DB schema
  - manual procedure:
    - run `dlf_oracle_drop.sql` to drop the schema
    - run `dlf_oracle_tbl.sql` to create the schema
    - enter the facilities using the `dlfenterfacilities` command
  - **`dlforasetup.sh`** / **`dlfmysetup.sh`** scripts do it at once
- 4. Start DLF service
  - **`$ service castor-dlf-server start`**

Catalog DB setup

- 1. Create the catalog DB schema
  - Run the **`oracle.sql`** script (drops and recreates all)
- 2. Insert basic metadata
  - SvcClass: “default” provided
    - **`$ enterSvcClass`**, **`modifySvcClass`** and **`deleteSvcClass`** for managing custom SvcClasses
  - FileClass
    - **`$ enterFileClass`**, **`modifyFileClass`** and **`deleteFileClass`**
    - import from old stager with --GetFromCns option:

```
$ foreach cname (`nslistclass | grep NAME | awk '{print $2}'`)
        enterFileClass --Name $cname --GetFromCns
  end
```

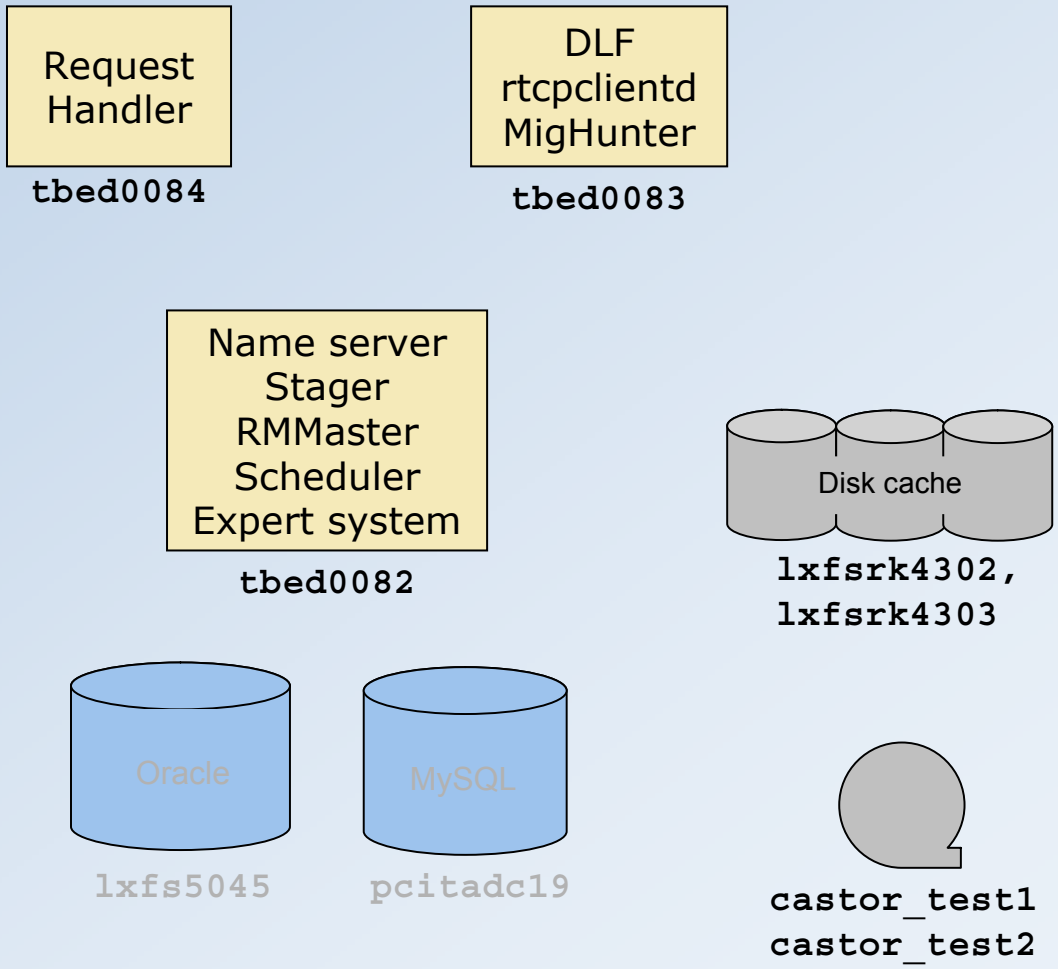
  - DiskPool: “default” provided
    - **`$ modifySvcClass --AddDiskPools dpool or --RemoveDiskPools`**
  - TapePool
    - **`$ modifySvcClass --AddTapePools tpool or --RemoveTapePools`**

Garbage Collection

- GC workers run on each diskserver
  - they ask the gc_service of the stager about what to delete
  - the stager takes info from the database
- “GCHunter” running inside the db as a stored procedure to mark files to be deleted
  - according to different policies
- Setup
  - Define policies in the catalog db
  - Modify svcClasses to use them

Castor components overview

- Databases setup
  - Name server
  - Distributed Logging Facility (DLF)
  - Catalog DB
  - GC
- Main daemons setup
  - Request Handler
  - Stager
  - Resource Manager Master
  - Scheduler (LSF or MAUI)
  - Expert system
  - MigHunter
  - DiskServer
  - TapeServer



Request Handler setup

- 1. Install software from RPM
  - castor-lib-oracle / castor-lib-mysql
  - castor-rh-server
- 2. Edit configuration script `/etc/castor/castor.conf`

```
OraCnvSvc        user    main
OraCnvSvc        passwd  xxxxxxxx
OraCnvSvc        dbName  castor_maindb
OraStagerSvc     user    main
OraStagerSvc     passwd  xxxxxxxx
OraStagerSvc     dbName  castor_maindb

DLF              HOST    tbed0083
STAGER           HOST    tbed0082
RHLog            LOGALL  x-dlf://tbed0083.cern.ch/
```

- 3. Start RH service
  - `$ service castor-rh-server start`

Stager setup

- 1. Install software from RPM
  - castor-lib-oracle / castor-lib-mysql
  - castor-stager-server
- 2. Edit configuration scripts
  - **/etc/castor/castor.conf:**

```
OraCnvSvc        user    main
...
OraStagerSvc     dbName  castor_maindb
EXPERT           HOST    tbed0082
DLF              HOST    tbed0083
stager           LOGALL  x-dlf://tbed0083.cern.ch/
```

  - **/etc/castor/RMNODECONFIG:**

```
RMMASTER_HOST    tbed0082
```

  - **/etc/castor/EXPCONFIG:**
    More later in Expert system
- 3. Start stager service
  - **$ service castor-stager-server start**

RMMaster setup

- 1. Install software from RPM
  - castor-rmmaster-server
- 2. Edit configuration scripts
  - /etc/castor/castor.conf:

```
rmmaster   LOGALL   x-dlf://tbed0083.cern.ch/
RM         CONNECTTIMEOUT   120
RM         TIMEOUT          120
RM         MAXMAUIGETJOBS   5000

RFIO       CONNTIMEOUT      10
RFIO       CONRETRY         10
RFIO       CONRETRYINT      1
RFIO       NFS_ROOT         /shift
RFIO       CONNECT_TCP_NODELAY      YES
```

  - Initialize cluster

```
$ rmadminnode --node host --feature movers:svcClasses --partition partName
```

    In our demo:

```
$ rmadminnode --node lxfsrk4302 --feature rfio:default --partition default
$ rmadminnode --node lxfsrk4303 --feature rfio:default --partition default
```

- 3. Start service
  - `service castor-rmmaster-server start`

Scheduler setup

- Support provided for MAUI and LSF
  - For this demo we use MAUI (open source)
- 1. Install software from RPM
  - *not yet packaged*
- 2. Edit configuration script
  - Default values provided
  - **`/usr/local/moab/moab.cfg`**

```
NODECFG[DEFAULT]      MAXLOAD=5.0
NODECFG[lxfsrk4302]   MAXJOB=1000
NODECFG[lxfsrk4303]   MAXJOB=1000
```

- 3. Start service
  - **`/usr/local/moab/sbin/moab`**

Expert system setup

- 1. Install software from RPM
  - castor-expert-server
  - clips >= 6.21
- 2. Edit configuration scripts
  - `/etc/castor/EXPCONFIG:` policies to be used (more later)

```
EXP_RQ_FILESYSTEM    /etc/expert/          /usr/bin/clips -f bestfs.clp
EXP_RQ_MIGRATOR      /etc/castor/expert/   /usr/bin/perl migrator.pl
EXP_RQ_RECALLER      /etc/castor/expert/   /usr/bin/perl recaller.pl
```

  - `/etc/castor/expert/`
    Path for expert system policies
- 2. Start service
  - `service castor-expert-server start`

Migration setup

- 1. Install software from RPM
  - castor-lib-oracle / castor-lib-mysql
  - castor-rtcopy-clientserver
- 2. Edit configuration scripts
  - `/etc/castor/castor.conf:`

```
OraCnvSvc        user    main
...
OraStagerSvc     dbName  castor_maindb
rtcpcld            LOGALL  x-dlf://tbed0083.cern.ch/
migrator           LOGALL  x-dlf://tbed0083.cern.ch/
recaller           LOGALL  x-dlf://tbed0083.cern.ch/
MigHunter          LOGALL  x-dlf://tbed0083.cern.ch/
TapeErrorHandler   LOGALL  x-dlf://tbed0083.cern.ch/
```

- 3. Start service
  - `$ service castor-rtcopy-clientserver start`
  - `$ modifySvcClass --Name svcClass --AddTapePool tpName [--NbDrives count]`
    `$ /usr/bin/MigHunter -t pollingTime -d svcClassName`
  - In our demo:

```
$ modifySvcClass --Name default --AddTapePool alicemdc6_3592 --NbDrives 2
$ /usr/bin/MigHunter -t 20 -d default
```

Diskserver setup

- 1. Install software from RPM
  - castor-gc-server
  - castor-rfio-server
  - castor-rmnode-server
  - castor-job
- 2. Edit configuration scripts
  - `/etc/castor/castor.conf:`

```
DLF         HOST        tbed0083
STAGER      HOST        tbed0083
GC          INTERVAL    60
job         LOGALL      x-dlf://tbed0083.cern.ch/

RFIO    DAEMONV3_RDSIZE 2097152
RFIO    DAEMONV3_WRSIZE 2097152
RFIO    NFS_ROOT        /shift

RFIOD   WTRUST  tbed0082 tbed0082.cern.ch
RFIOD   RTRUST  tbed0082 tbed0082.cern.ch
RFIOD   XTRUST  tbed0082 tbed0082.cern.ch
RFIOD   FTRUST  tbed0082 tbed0082.cern.ch
```

  - `/etc/castor/RMNODECONFIG:`

```
RMMASTER_HOST    tbed0082
/shift/lxfsrk4303/data01
/shift/lxfsrk4303/data02
...
```

- 3. Start diskserver job
  -
```
service castor-gc-server start
service castor-rfio-server start
service castor-rmnode-server start
```

Tapeserver setup

- 1. Install software from RPM
  - castor-tape-server
  - castor-rfio-server
  - castor-rtcopy-client
  - castor-vdqm-client
  - castor-vmgr-client
- 2. Configure as in the old castor
- 3. Start service

Demo session

2nd part: Administrator functionalities

Giuseppe Lo Presti
CERN/IT/FIO & INFN/CNAF

giuseppe.lopresti@cern.ch

Castor External Operation Workshop, CERN, June 13-14, 2005

Summary

- Migration policies
- GC policies
- MySQL support status
- Wrap up

Migration policies

- Policies for tape pools can be enforced
  - different criteria: e.g. file size
- Example
  - a tape pool is used for big files (better transfer troughput)
  - a different tape pool is used for small files (better access time)
- We'll setup this policy in the expert system and see the migration processes according to it
  - Commands
    - `$ modifySvcClass --Name default --MigratorPolicy sizeMigrPolicy.pl`
    - `$ printSvcClass --Name default`
    - `$ vmgrlisttape -P tapePool` (castor_test1 or castor_test2)
  - Query
    - SELECT TapePool.name, CastorFile.fileid, Stream.id
      FROM CastorFile, TapeCopy, Stream2TapeCopy, Stream, TapePool
      WHERE TapeCopy.castorFile = CastorFile.id
        AND Stream2TapeCopy.child = TapeCopy.id
        AND Stream.id = Stream2TapeCopy.parent
        AND TapePool.id = Stream.tapePool
        AND CastorFile.fileid in (<set of fileIds>);

GC policies

- Policies for GC on disk pools can be enforced
  - criteria: e.g. main task currently running
- Example
  - A disk pool is used for data taking: delete files as soon as they are migrated to tapes (oldest-first)
  - Another disk pool is used for analysis: use some clever policy to decide which file has to be deleted from disk
    - for this demo we use the file size > biggest-first
- We'll setup the two policies and see the GC in action
  - Create the policies as stored functions in the DB
  - Create the svc classes and associate with disk pools
    - `$ enterSvcClass --Name biggestFirst --GcPolicy biggestFirstGCPolicy`
    - `$ modifySvcClass --Name biggestFirst --AddDiskPool biggestFirstDiskPool`
    - `$ printSvcClass --Name ...`
  - For demo purposes, limit the Min/Max Free Space on the FileSystems
  - Create a bunch of files and stage them, then look at the DB

MySQL development

- Goal
  - Develop a persistency layer to run the stager database over MySQL
  - Implement missing SQL-level functionalities in C++
- Status
  - Started on mid March 2005
    - using MySQL version 5.0.2-alpha
  - First persistency layer working on mid May 2005
    - MySQL 5.0.4-beta
    - Autogenerated classes for stager data access, using the same framework already in place for Oracle
    - Most stored procedures ported
  - Expected fully functional prototype by August 2005
    - Hopefully running over MySQL 5.0 production version
    - Full stager services

MySQL demo

- RH server storing requests to a MySQL instance
  - Data stored in MySQL
  - Traceable in DLF
- New VDQM server (pre-release under development)

- Demo
  - change the config file to load the mysql **`.so`** module
    - ```
      #SvcMapping     DBCNV     4 #SVC_ORACNV
      SvcMapping      DBCNV    11 #SVC_MYCNV
      DynamicLib      ORACNV   libcastorCommonOra.so
      DynamicLib      MYCNV    libcastorCommonMysql.so
      ```
  - start the **`rhserver`** (on **`pcitadc19`**)
  - run stager client commands
  - start the **`vdqmserver`**
  - Look in DLF
      ```

Conclusions and wrap up

- Technical overview of the new stager architecture
  - Key points: DB centric, stateless daemons
- Configuration overview of a full castor setup
  - For demonstration purposes, yet a good exercise to make all components (almost) working as expected
- Some new functionalities have been shown
  - DLF, PrepareTo
  - Policies for administrators
- Next steps
  - Deployment @ CERN going on
  - Deployment @ external sites

Castor components

- Castor daemons
  - LSF (scheduler)
  - RTCopyClient
  - Stager
  - Request Handler (RH)
  - Distributed Log Facility (DLF)
  - Expert system & Resource monitor
  - SRM?
- Oracle servers
  - Stager db
  - DLF db
  - Dataguard for stager db
- Disk servers

From the old castor:

- rfiod

- tpdaemon (tpconf) -> Ben

- rtcpd