SC3 experiences

Ron Trompert

SARA
Why dCache?

- **dCache provides an srm I/F**
  - We use DMF in our HSM environment for which there is no SRM implementation

- **dCache provides flexibility with respect to HSM backends**
  - If we need to switch to another HSM setup for some reason

- **dCache provides functionality promised by the SRM standard but not supported by DMF**
  - File pinning
SC3 Infrastructure
SC3 infrastructure

Pool nodes
- 4x dual Opteron’s, 4GB memory, 2x 1GE
- 2TB disk cache, 12x 250GB SATA, 3ware RAID controller, disk I/O
  200MB/s RAID0 (used during SC3) and 100MB/s RAID5, XFS

Admin node
- dual Xeon, 4GB memory, 2x 73GB internal disk, 2x 1GE

MSS gateway nodes (disk servers)
- 2x dual Xeon, 4GB memory, 2x 73GB internal disk, 2x 1GE, dual HBA
  FC, 1.6 TB CXFS filesystem (SAN shared filesystem)
- runs CXFS client, read/write data directly to/from CXFS filesystem
- and rfio daemon to put/get data to/from pool nodes

MSS server (CXFS/DMF)
- 4 cpu R16K MIPS, 4GB memory, 12x FC, 4x GE, 2x 36GB internal
  disk, 1.6 TB CXFS filesystem (SAN shared filesystem), 3x STK 9940B
  tape drives
- CXFS MDS server, regulates access to CXFS filesystem
- DMF (Data Migration Facility = HSM system), migrates data from disk
to tape and back

Network
- dedicated 10GE network between CERN – Amsterdam
- GE internal network between pool nodes and MSS gateway nodes
dCache configuration

**dCache 1.2.2-7-3**

**Admin node**
- ia32
- SL304 with 2.4.21-32.0.1 kernel
- Runs pnfs server, srm and gridftp door

**Pool nodes**
- amd64
- Debian (sarge) 2.6.8-10 kernel
- Pool node s/w is in java
- j2sdk 1.5.0
- Got source rpm of CASTOR client to build rfio
- Three minor issues encountered installing the dCache pool S/W.
- Pools with XFS filesystem
- Run gridftp door
dCache configuration

The default number of I/O movers was 100
- Leads to very high loads
  We have set it to approximately 5

The default heartbeat was 120
- Leads to poor load balance over the pools.

A single transfer request with multiple files would dump all transfers on a single pool.

A pool may get a considerable number of transfers before other pools are taken into account

We have set it to 10
**dCache configuration**

**Number of streams per transfer**

- Using the Globus gridftp server on a dedicated 10 Gb link with a small number of streams (1-2) is optimal (1 GB file, 50 MB/s, 1 stream)

- A 1 stream transfer with a dCache gridftp server leads to 1.6 MB/s for a 1 GB file. This performance scales almost linearly with the number of streams (1-10 streams -> 1.6 MB/s-16 MB/s)

  => probably an implementation issue and not networking.

- Bad for transparency which is desirable in a Grid environment.

- Kept it at 10 which is the default.
Tuning of kernel parameters on pool nodes

- `vm.lower_zone_protection = 200`
- `vm.dirty_expire_centisecs = 250`
- `vm.dirty_writeback_centisecs = 250`
- `vm.dirty_ratio = 10`
SC3 Results

- **Disk2disk: 100-110 MB/s**
  - Problems with stability of the nodes

- **Disk2tape: 50 MB/s**
  - Not enough bandwidth, SAN not dedicated
Sc3 observations

- SrmPut requests lead to gridftp doors receiving files and passing them on to pool nodes.
  - Puts unnecessary load on nodes
  - Uses bandwidth which can be used for useful transfers.
  - FTS (srmPut) => 100-110 MB/s
  - Srmcp (srmCopy) => 180 MB/s
SC3 observations

- Left srmPut, right srmCopy
Timeouts in returning turls.

- `getRequestStatus` timeouts
- Restarting dCache did not help by itself. Cleaning out the postgres db and restarting dCache did.
- Happens when transfers are going on for a couple of days.
SC3 observations

- With a full disk pool everything kept running happily
SC3 observations

Dips in network traffic

dCache if fine. There is no relationship with events in the gridftp logs and srm logs with the dips. Also no relation with dips and failed transfers.
SC3 observations

- With a constant number of files no constant throughput
SC3 observations
SC3 observations

- **Uuid is not always unique**
  - Sometimes failed transfers because of attempt to overwrite an existing file
  - Adding a timestamp to the file name solved this
Post SC3 tests

- dCache 1.6.5-1
- No gridftp door on admin node
Crash tests

- dCache 1.6.5-1
- 5 I/O movers per pool
- Normal shutdown of one pool node
- Kill -9 -1 as root on pool node
- Genuine crash of a pool node due to overload. Thanks to the CERN people.

- Max. number of I/O movers is 5.
- Max. number of files is 6 for the shutdown and 20 for the overload crash.
Crash tests

- shutdown
- Kill -9 -1
- real crash

Overall hourly averaged throughput during the last 24 hours
Stress tests

- Gradually increase the number of files from 4-40.

- For each pool:
  - Max. number of I/O movers = 2
  - Max. number of store movers = 3
Stress tests
Performance tests

Disk2disk

- Test1
  - 4 I/O movers
  - No copies to CXFS disk servers
  - 30 files

- Test2
  - 2 I/O movers, 2 store movers
  - Copies to CXFS disk servers
  - 30 files
### Pool Request Queues

<table>
<thead>
<tr>
<th>CellName</th>
<th>DomainName</th>
<th>Movers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Active</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>rembrandt3_1</td>
<td>rembrandt3Domain</td>
<td>4</td>
</tr>
<tr>
<td>rembrandt3_2</td>
<td>rembrandt3Domain</td>
<td>0</td>
</tr>
<tr>
<td>rembrandt4_1</td>
<td>rembrandt4Domain</td>
<td>4</td>
</tr>
<tr>
<td>rembrandt9_1</td>
<td>rembrandt9Domain</td>
<td>4</td>
</tr>
</tbody>
</table>

**Total**

<table>
<thead>
<tr>
<th>CellName</th>
<th>DomainName</th>
<th>Active</th>
<th>Max</th>
<th>Queued</th>
</tr>
</thead>
</table>
Performance tests: test1

Overall hourly averaged throughput during the last 24 hours

- rembrandt3
- rembrandt4
- rembrandt9
## Pool Request Queues

<table>
<thead>
<tr>
<th>CellName</th>
<th>DomainName</th>
<th>Movers</th>
<th>Restores</th>
<th>Stores</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Active</td>
<td>Max</td>
<td>Queued</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>6</td>
<td>26</td>
<td>20</td>
</tr>
<tr>
<td>rembrandt3_1</td>
<td>rembrandt3Domain</td>
<td>2</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>rembrandt3_2</td>
<td>rembrandt3Domain</td>
<td>0</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>rembrandt4_1</td>
<td>rembrandt4Domain</td>
<td>2</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>rembrandt9_1</td>
<td>rembrandt9Domain</td>
<td>2</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>6</td>
<td>26</td>
<td>20</td>
</tr>
</tbody>
</table>
Performance tests: test2

Overall hourly averaged throughput during the last 24 hours

- rembrandt3
- rembrandt4
- rembrandt9

throughput (MB/s)

time