Highly Available dCache

2016-10-19, HEPiX Fall 2016, Berkeley

Mattias Wadenstein
Overview

- News in dCache 3.0
- HA support in dCache
- HA dCache as deployed by NDGF-T1
- Future outlook
- Next dCache workshop
Thanks to

- Actual work by Gerd Behrmann and the rest of the dCache.org team
- Much slide content from Paul Millar, who presented this in the September pre-GDB meeting
- The dCache team likes to credit:
The upcoming 2.17 release will be dCache 3.0

dCache server releases
... along with the series support durations.

<table>
<thead>
<tr>
<th>Series</th>
<th>Release Type</th>
<th>Years</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.17 series</td>
<td>(anticipated release)</td>
<td>2016-2017</td>
</tr>
<tr>
<td>2.16 series</td>
<td>(golden release)</td>
<td>2015-2017</td>
</tr>
<tr>
<td>2.15 series</td>
<td></td>
<td>2014-2016</td>
</tr>
<tr>
<td>2.14 series</td>
<td></td>
<td>2013-2015</td>
</tr>
<tr>
<td>2.13 series</td>
<td>(golden release)</td>
<td>2012-2014</td>
</tr>
<tr>
<td>2.12 series</td>
<td></td>
<td>2011-2013</td>
</tr>
<tr>
<td>2.11 series</td>
<td></td>
<td>2010-2012</td>
</tr>
<tr>
<td>2.10 series</td>
<td>(golden release)</td>
<td>2009-2011</td>
</tr>
</tbody>
</table>
New things in dCache 3.0: CEPH

- dCache now has built-in **CEPH integration**: Sites can deploy a dCache pool that provides access to a CEPH pool.

- dCache files are written as **RBD images**: These can also be accessed independent of dCache, if you know the PNFS-ID of the file.

- All **protocols** and **high-level features** are available: Sites with tape integration may need to tweak their scripts.
New things in dCache 3.0: srmfs

- **srmfs** is an interactive client that provides fast access to an srm storage
- Similar idea to lftp or similar
  - Use cd, ls, get, put, stat commands in interactive shell
  - If you ever were under the impression that “srm is slow”, try it out
New things in dCache 3.0: HA dCache

• **What** is HA dCache?
  – Multiple instances of core components can run concurrently,
  – Doors updated to support load-balancers (e.g., HAProxy).

• **Why** HA dCache?
  – Symmetric deployment (making life easy),
  – Horizontal scaling (no CPU bottlenecks),
  – Fault tolerance (no single-point-of-failure),
  – Rolling bug-fix updates (no downtimes).

• Using **zookeeper** for location and some state
  – PoolManager state persistent in zookeeper, not in poolmanager.conf
HA dCache: SRM

• **Split** the GSI “front-end” from “SRM engine”

• Allow **multiple front-ends**:
  
  horizontal scaling for encryption overhead

• Allow **multiple back-end “SRM engines”**:
  
  each scheduled request is processed by the same SRM engine, load-balancing and fault-survival.

• Support for **HAProxy protocol**
  
  using TCP mode, rather than HTTP mode.
Pencil sketch of possible deployment

Clients ----> SRM front-end ----> SRM back-end ----> Rest of dCache
Pencil sketch of possible deployment

Clients -> HAPerxy

SRM front-end -> SRM back-end

Rest of dCache

SRM front-end
Pencil sketch of possible deployment

Clients

HAProxy (live)

HAProxy (hot spare)

CARP: “shared” IP address

SRM front-end

SRM front-end

SRM front-end

SRM back-end

Rest of dCache

NB: works fine with just two node
Pencil sketch of possible deployment

- Clients
  - HAProxy (live)
    - HAProxy (hot spare)
  - SRM front-end
  - SRM back-end
  - SRM front-end
  - SRM back-end
  - SRM front-end

- Rest of dCache

CARP: “shared” IP address

NB: works fine with just two node
HA dCache: general protocol remarks

- Should work fine for **TLS-based** protocols (SRM, gsiftp, webdav, gsidcap)
  
  Needs **load-balancer hostname** as a Subject Alternate Name (SAN) in the X.509 certificate

- Can have SRM redirects clients to individual doors, rather than using HA proxy:
  
  SRM already provides load-balancing.

- **HAProxy** protocol used to discover **client IP address**: de facto industry standard.
Pencil sketch of possible deployment

Clients

HAProxy (live)

HAProxy (hot spare)

CARP: “shared” IP address

Gsiftp Door

Rest of dCache

Gsiftp Door

Gsiftp Door

NB: works fine with just two node
HA dCache: FTP

- Updated to understand **HAProxy protocol**.
- **IPv4 and IPv6** supported.
- **Data channels** connect directly to pool or door, bypassing HAProxy.
HA dCache: other protocols

- **WebDAV**: nothing major needed
- **xrootd**: updated to understand HAProxy protocol. As usual so-called “GSI” xrootd sucks:
  - special care needed over x.509 certificate
  - kXR_locate returns IP address; makes host name verification hard.
- **dcap**: updated to understand HAProxy protocol; No other major changes needed.
- **NFS**: not updated to support HA.
HA dCache in practice

- NDGF only has two physical machines: zanak and clom for central services
  - Running postgresql for dCache on HW
  - And a bunch of virtual machines in Ganeti on the same HW
- This is running in production as of last week
  - Some parts have been in production longer, like rempgr management of database failover
Backend technology for HA dCache

- **Handling postgresql failover with repmgr**
  - Somewhat manual for now, hard to make a majority decision with less than three nodes
  - But repmgr makes failover and promotion of the old failed node to an up to date secondary automatically when it comes up, etc
  - `dcache.db.host=clom,zanak`

- **ZooKeeper for directory and data services**
  - We run 3 dedicated VMs for ZooKeeper
  - `dcache.zookeeper.connection = zoo1.ndgf.org:2181,zoo2.ndgf.org:2181,zoo3.ndgf.org:2181`
Our production HA dCache

- Two fat virtual machines with all the central services
  - Named kermit and piggy
  - 2x SRM, gsiftp, webdav, PnfsManager, PoolManager, etc
  - Almost symmetrical (except for billing log files on piggy)
  - We try to lay these out so that we get a zookeeper quorum and one of these machines on the primary postgresql server
    - Live migration with Ganeti makes this easy to change
    - This way we can lose the other one without much interruption
    - Loss of primary node will take some manual work before we’re up: this is a strong case for a third machine with a third fat VM
Our production HA dCache

<table>
<thead>
<tr>
<th>Role</th>
<th>Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>PnfsManager</td>
<td>kermitDomain</td>
</tr>
<tr>
<td>PnfsManager</td>
<td>piggyDomain</td>
</tr>
<tr>
<td>PoolManager</td>
<td>kermitDomain</td>
</tr>
<tr>
<td>PoolManager</td>
<td>piggyDomain</td>
</tr>
<tr>
<td>SRM-kermit</td>
<td>kermitDomain</td>
</tr>
<tr>
<td>SRM-piggy</td>
<td>piggyDomain</td>
</tr>
<tr>
<td>SpaceManager</td>
<td>kermitDomain</td>
</tr>
<tr>
<td>SpaceManager</td>
<td>piggyDomain</td>
</tr>
<tr>
<td>SrmManager</td>
<td>kermitDomain</td>
</tr>
<tr>
<td>SrmManager</td>
<td>piggyDomain</td>
</tr>
<tr>
<td>WebDAV-http-kermit</td>
<td>kermitDomain</td>
</tr>
<tr>
<td>WebDAV-http-piggy</td>
<td>piggyDomain</td>
</tr>
<tr>
<td>WebDAV-https-kermit</td>
<td>kermitDomain</td>
</tr>
<tr>
<td>WebDAV-https-piggy</td>
<td>piggyDomain</td>
</tr>
<tr>
<td>WebDAV-srm-kermit</td>
<td>kermitDomain</td>
</tr>
<tr>
<td>WebDAV-srm-piggy</td>
<td>piggyDomain</td>
</tr>
</tbody>
</table>
Our production HA dCache

- On the hardware nodes we run haproxy and ucarp
  - Except for gsixrootd due to client certificate validation stupidity, points directly to one of the fat muppets
  - Configuration pretty straight-forward and short, but might take some testing before deployment so that it works reliably
- IP failover tested well in our preproduction setup, but only seen light testing in production
- Working on documented procedures for rolling upgrades etc
  - Also haven’t decomissioned some old services, so SRM has four gsiftp doors etc right now, but we only need two
Is it perfect yet?

• Move of IP kills existing connections
  - Not a big worry for short-lived connections, like SRM and webdav, but could be an issue for gsiftp

• Expect some blips in logs and monitoring during failver

• Forensics more difficult to partitioning of logs

• No or only limited and manual “draining” of nodes for rolling upgrade or other maintenance
  - Hope for improvement here in dCache in the future
11th International dCache workshop in Umeå
2017-05-29 - 2017-05-30

• Should just about hit first 2 days of summer
  – Means weather roughly like here, now
  – But far less dark, due to summer in the north
• Looking forward to interesting discussions on HA dCache and future technologies
• Followed by 2 days of NeIC 2017 conference
  – For those interested in other parts of computing, storage, etc
• Excellent location :)
• More info on: http://neic2017.nordforsk.org
Questions?