17th dCache User Workshop
Logistic

• The first in-person workshop since 2019
  - Hosted by: *HTW Berlin - University of Applied Sciences*
  - May 31 – June 1
  - ½ + 1 day
  - 35 participants (3 remote)
  - 12 contributions (9 from sites)
High Speed Data Ingest

Data management & workflow control

Batch processing

Interactive analysis

Wide Area Transfers
The Challenges

- Data is going to grow… A lot…
  - High ingest data rates
  - More movements between sites

- Shared Computing Resources
  - Analysis Facilities
  - Grid Farms
  - HPC
  - Cloud resources (CPU&Storage)

- Standard analysis tools
  - ROOT
  - Jupyter Notebooks, non-ROOT analysis

- Competing Tape Operations

The pie chart shows the breakdown of the power consumption at the CERN data center. Most of the power is consumed for data processing (CPUs). Large part of the “services” are in fact CPUs. In this study we will focus on the energy needs for CPUs.
Technical Directions

- Scaleout
  - Namespace
  - Number of pools (cells)
- Token-based Authentication
- Better *Analysis Facility* support
  - POSIX access and compliance
  - HPC workload support
- QoS
- Tape integration
Some Numbers

- **XFEL**
  - Total capacity ~120 PB
  - ~400 physical hosts (~4000 dCache pools)
  - 20-40 GB/s ingest

- **Photon**
  - DB size – 2.5TB
  - ACL table 600GB
  - Directories with 3 $10^6$ files
  - 1.2 $10^9$ file system objects
  - 100K files in the flush queue
  - Two tape copies, different media type

- **ATLAS**
  - dir/file → 1/3

- **NextCloud**
  - File lifetime < 1s
DPM Migration

- Spike of new users
- Series of tutorials
- Help from EGI
  - Thanks to Petr Vokac

https://docs.google.com/spreadsheets/d/1KDVAJ9JzlycA3Wrz1iY2fQxZndWdAezFnLaDAXIipUs/edit
Re-cap
Prominent Changes

- BULK Service
- TPC improvements
- NFSv4.1/pNFS improvements
- XROOT evolution (TLS, tokens, TPC, proxy-IO)
- Namespace performance improvements
- HSM connectivity
**dCache Quiz: What Going On?!**

```
<table>
<thead>
<tr>
<th>PID</th>
<th>USER</th>
<th>PR</th>
<th>NI</th>
<th>VIRT</th>
<th>RES</th>
<th>SHR</th>
<th>S</th>
<th>%CPU</th>
<th>%MEM</th>
<th>TIME+</th>
<th>COMMAND</th>
</tr>
</thead>
<tbody>
<tr>
<td>32512</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>22384</td>
<td>19000</td>
<td>R</td>
<td>14.3</td>
<td>0.1</td>
<td>0:05.10</td>
<td>postgres:</td>
<td>postgres chimera-test 127.0.0.1(42362)</td>
</tr>
<tr>
<td>32481</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>22956</td>
<td>19580</td>
<td>S</td>
<td>13.5</td>
<td>0.1</td>
<td>0:05.50</td>
<td>postgres:</td>
<td>postgres chimera-test 127.0.0.1(42304)</td>
</tr>
<tr>
<td>32500</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>22552</td>
<td>19126</td>
<td>R</td>
<td>12.0</td>
<td>0.1</td>
<td>0:04.79</td>
<td>postgres:</td>
<td>postgres chimera-test 127.0.0.1(42338)</td>
</tr>
<tr>
<td>32585</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>22648</td>
<td>19744</td>
<td>S</td>
<td>11.6</td>
<td>0.1</td>
<td>0:05.31</td>
<td>postgres:</td>
<td>postgres chimera-test 127.0.0.1(42368)</td>
</tr>
<tr>
<td>32532</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>23128</td>
<td>19868</td>
<td>S</td>
<td>11.0</td>
<td>0.1</td>
<td>0:05.57</td>
<td>postgres:</td>
<td>postgres chimera-test 127.0.0.1(42376)</td>
</tr>
<tr>
<td>32496</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>22972</td>
<td>19588</td>
<td>S</td>
<td>10.5</td>
<td>0.1</td>
<td>0:05.35</td>
<td>postgres:</td>
<td>postgres chimera-test 127.0.0.1(42400)</td>
</tr>
<tr>
<td>32501</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>22848</td>
<td>19468</td>
<td>S</td>
<td>10.3</td>
<td>0.1</td>
<td>0:05.30</td>
<td>postgres:</td>
<td>postgres chimera-test 127.0.0.1(42340)</td>
</tr>
<tr>
<td>32519</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>23132</td>
<td>19752</td>
<td>S</td>
<td>10.3</td>
<td>0.1</td>
<td>0:05.88</td>
<td>postgres:</td>
<td>postgres chimera-test 127.0.0.1(42376)</td>
</tr>
<tr>
<td>32523</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>22760</td>
<td>19372</td>
<td>R</td>
<td>10.3</td>
<td>0.1</td>
<td>0:05.34</td>
<td>postgres:</td>
<td>postgres chimera-test 127.0.0.1(42384)</td>
</tr>
<tr>
<td>32483</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>22620</td>
<td>19312</td>
<td>R</td>
<td>10.0</td>
<td>0.1</td>
<td>0:05.11</td>
<td>postgres:</td>
<td>postgres chimera-test 127.0.0.1(42300)</td>
</tr>
<tr>
<td>32493</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>22608</td>
<td>19232</td>
<td>S</td>
<td>9.6</td>
<td>0.1</td>
<td>0:05.07</td>
<td>postgres:</td>
<td>postgres chimera-test 127.0.0.1(42324)</td>
</tr>
<tr>
<td>32511</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>22812</td>
<td>19476</td>
<td>S</td>
<td>9.6</td>
<td>0.1</td>
<td>0:05.38</td>
<td>postgres:</td>
<td>postgres chimera-test 127.0.0.1(42360)</td>
</tr>
<tr>
<td>32518</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>22896</td>
<td>19508</td>
<td>S</td>
<td>9.6</td>
<td>0.1</td>
<td>0:05.40</td>
<td>postgres:</td>
<td>postgres chimera-test 127.0.0.1(42374)</td>
</tr>
<tr>
<td>32516</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>23028</td>
<td>19636</td>
<td>S</td>
<td>9.3</td>
<td>0.1</td>
<td>0:05.55</td>
<td>postgres:</td>
<td>postgres chimera-test 127.0.0.1(42376)</td>
</tr>
<tr>
<td>32473</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>23212</td>
<td>19660</td>
<td>S</td>
<td>9.0</td>
<td>0.1</td>
<td>0:05.48</td>
<td>postgres:</td>
<td>postgres chimera-test 127.0.0.1(42288)</td>
</tr>
<tr>
<td>32491</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>22868</td>
<td>19484</td>
<td>S</td>
<td>9.0</td>
<td>0.1</td>
<td>0:05.35</td>
<td>postgres:</td>
<td>postgres chimera-test 127.0.0.1(42320)</td>
</tr>
<tr>
<td>32494</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>22820</td>
<td>19432</td>
<td>S</td>
<td>9.0</td>
<td>0.1</td>
<td>0:05.18</td>
<td>postgres:</td>
<td>postgres chimera-test 127.0.0.1(42326)</td>
</tr>
<tr>
<td>32502</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>23160</td>
<td>19780</td>
<td>S</td>
<td>9.0</td>
<td>0.1</td>
<td>0:05.73</td>
<td>postgres:</td>
<td>postgres chimera-test 127.0.0.1(4242)</td>
</tr>
<tr>
<td>32517</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>22736</td>
<td>19352</td>
<td>S</td>
<td>9.0</td>
<td>0.1</td>
<td>0:05.26</td>
<td>postgres:</td>
<td>postgres chimera-test 127.0.0.1(42376)</td>
</tr>
<tr>
<td>32531</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>23176</td>
<td>19780</td>
<td>S</td>
<td>9.0</td>
<td>0.1</td>
<td>0:05.28</td>
<td>postgres:</td>
<td>postgres chimera-test 127.0.0.1(4242)</td>
</tr>
<tr>
<td>32586</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>22664</td>
<td>19336</td>
<td>S</td>
<td>8.6</td>
<td>0.1</td>
<td>0:05.09</td>
<td>postgres:</td>
<td>postgres chimera-test 127.0.0.1(42350)</td>
</tr>
<tr>
<td>32521</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>23004</td>
<td>19724</td>
<td>S</td>
<td>8.6</td>
<td>0.1</td>
<td>0:05.32</td>
<td>postgres:</td>
<td>postgres chimera-test 127.0.0.1(42380)</td>
</tr>
<tr>
<td>32527</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>22620</td>
<td>19240</td>
<td>S</td>
<td>8.3</td>
<td>0.1</td>
<td>0:04.89</td>
<td>postgres:</td>
<td>postgres chimera-test 127.0.0.1(42392)</td>
</tr>
<tr>
<td>32470</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>23048</td>
<td>19504</td>
<td>S</td>
<td>8.0</td>
<td>0.1</td>
<td>0:05.07</td>
<td>postgres:</td>
<td>postgres chimera-test 127.0.0.1(42282)</td>
</tr>
</tbody>
</table>
```
POSIX Constraints

- According to POSIX standard, on new file system object creation the parent directories *modification time* should be updated.

- To track the directory changes that happen at a higher rate than the precision of mtime attribute Linux kernel has an additional attribute *iversion* that is incremented whenever the inode's data is changed.

- To reduce unnecessary directory listing requests to the servers, the NFSv4 clients utilize the *iversion* attribute to identify the directory content changes and use the locally cached copy of the directory entry list as long as last known *iversion* attribute value matches the remote one.
Near-POSIX Behavior

top - 23:10:33 up 52 days, 12:21, 3 users, load average: 37.60,
Tasks: 356 total, 28 running, 328 sleeping, 0 stopped, 0 zombie
%CPU(s): 62.3 us, 29.1 sy, 0.0 ni, 4.7 id, 0.0 wa, 0.0 hi, 3.9 si, 0.0 st
KiB Mem: 32548896 total, 2054804 used, 32506900 buff/cache
KiB Swap: 8191996 total, 8188148 free, 3848 used, 18183296 avail Mem

<table>
<thead>
<tr>
<th>PID</th>
<th>USER</th>
<th>PR</th>
<th>NI</th>
<th>VIRT</th>
<th>SHR</th>
<th>S</th>
<th>%CPU</th>
<th>%MEM</th>
<th>TIME+</th>
<th>COMMAND</th>
</tr>
</thead>
<tbody>
<tr>
<td>30933</td>
<td>root</td>
<td>20</td>
<td>0</td>
<td>1610</td>
<td>22580</td>
<td>S</td>
<td>246.5</td>
<td>2.0</td>
<td>3:54.37</td>
<td>/usr/lib64/jvm/java-11-openjdk-11.0.15.0.8-2.el7_9.x86_64/bin/java -XX:MaxPermSize=512m</td>
</tr>
<tr>
<td>7011</td>
<td>root</td>
<td>20</td>
<td>0</td>
<td>262056</td>
<td>2692</td>
<td>S</td>
<td>1188</td>
<td>0.0</td>
<td>31:26.71</td>
<td>postgres: logger</td>
</tr>
<tr>
<td>31092</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>7792496</td>
<td>352548</td>
<td>349380</td>
<td>R</td>
<td>19.6</td>
<td>1.1</td>
<td>0:15.57</td>
</tr>
<tr>
<td>31045</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>7793104</td>
<td>356080</td>
<td>328680</td>
<td>S</td>
<td>19.3</td>
<td>1.1</td>
<td>0:15.71</td>
</tr>
<tr>
<td>31048</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>7793104</td>
<td>351708</td>
<td>348552</td>
<td>S</td>
<td>19.3</td>
<td>1.1</td>
<td>0:15.57</td>
</tr>
<tr>
<td>31052</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>7793104</td>
<td>350712</td>
<td>347568</td>
<td>S</td>
<td>19.3</td>
<td>1.1</td>
<td>0:15.71</td>
</tr>
<tr>
<td>31059</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>7792548</td>
<td>349620</td>
<td>346624</td>
<td>S</td>
<td>19.3</td>
<td>1.1</td>
<td>0:15.64</td>
</tr>
<tr>
<td>31062</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>7792548</td>
<td>352516</td>
<td>349528</td>
<td>R</td>
<td>19.3</td>
<td>1.1</td>
<td>0:15.65</td>
</tr>
<tr>
<td>31064</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>7792548</td>
<td>352556</td>
<td>349588</td>
<td>S</td>
<td>19.3</td>
<td>1.1</td>
<td>0:15.58</td>
</tr>
<tr>
<td>31066</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>7792496</td>
<td>351912</td>
<td>348932</td>
<td>S</td>
<td>19.3</td>
<td>1.1</td>
<td>0:15.62</td>
</tr>
<tr>
<td>31068</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>7792548</td>
<td>351736</td>
<td>348768</td>
<td>S</td>
<td>19.3</td>
<td>1.1</td>
<td>0:15.63</td>
</tr>
<tr>
<td>31076</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>7792948</td>
<td>354100</td>
<td>351120</td>
<td>S</td>
<td>19.3</td>
<td>1.1</td>
<td>0:15.65</td>
</tr>
<tr>
<td>31082</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>7792548</td>
<td>358060</td>
<td>355976</td>
<td>S</td>
<td>19.3</td>
<td>1.1</td>
<td>0:15.67</td>
</tr>
<tr>
<td>31085</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>7792548</td>
<td>354636</td>
<td>351660</td>
<td>S</td>
<td>19.3</td>
<td>1.1</td>
<td>0:15.65</td>
</tr>
<tr>
<td>31086</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>7792548</td>
<td>356200</td>
<td>353320</td>
<td>S</td>
<td>19.3</td>
<td>1.1</td>
<td>0:15.58</td>
</tr>
<tr>
<td>31089</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>7792548</td>
<td>351996</td>
<td>349920</td>
<td>R</td>
<td>19.3</td>
<td>1.1</td>
<td>0:15.59</td>
</tr>
<tr>
<td>31100</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>7792548</td>
<td>355064</td>
<td>352084</td>
<td>S</td>
<td>19.3</td>
<td>1.1</td>
<td>0:15.63</td>
</tr>
<tr>
<td>31039</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>7793104</td>
<td>354112</td>
<td>350964</td>
<td>R</td>
<td>18.9</td>
<td>1.1</td>
<td>0:15.51</td>
</tr>
<tr>
<td>31041</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>7793104</td>
<td>348012</td>
<td>344864</td>
<td>R</td>
<td>18.9</td>
<td>1.1</td>
<td>0:15.65</td>
</tr>
<tr>
<td>31043</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>7793104</td>
<td>359088</td>
<td>346942</td>
<td>S</td>
<td>18.9</td>
<td>1.1</td>
<td>0:15.61</td>
</tr>
<tr>
<td>31044</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>7793104</td>
<td>353590</td>
<td>350348</td>
<td>S</td>
<td>18.9</td>
<td>1.1</td>
<td>0:15.63</td>
</tr>
<tr>
<td>31046</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>7793104</td>
<td>350336</td>
<td>347212</td>
<td>R</td>
<td>18.9</td>
<td>1.1</td>
<td>0:15.56</td>
</tr>
<tr>
<td>31047</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>7793104</td>
<td>356932</td>
<td>353788</td>
<td>R</td>
<td>18.9</td>
<td>1.1</td>
<td>0:15.60</td>
</tr>
<tr>
<td>31050</td>
<td>postgres</td>
<td>20</td>
<td>0</td>
<td>7793104</td>
<td>362412</td>
<td>359252</td>
<td>S</td>
<td>18.9</td>
<td>1.1</td>
<td>0:15.59</td>
</tr>
</tbody>
</table>
SRM Still Here ... *but not too long*

- Two main gaps to fill
  - Space allocation
  - Tape operation
- Two alternatives to replace
  - User and Group based Quota system
  - WLCG tape recall API
User/Group Quotas

- **Quota ≠ Space reservation**
- Lazy, based on periodic scans
  - Users might overrun
  - Removed space not reclaimed immediately
- Global per file system
  - No quota per directories
- Respects Files Retention policy
  - Separate for 'disk' and 'tape' files
- Available since 7.2, enabled by default since 8.2
The Renaissance of Tape?

WLCG data centers power consumption

The pie chart shows the breakdown of the power consumption at the CERN data center.

Most of the power is consumed for data processing (CPUs). Large part of the “services” are in fact CPUs.

In this study we will focus on the energy needs for CPUs.

Shameless stolen from Simone Campana

Use Tape Gateway to replace physical tapes or premises with virtual tapes on AWS—reducing your data storage costs without changing your tape-based backup workflows. Tape Gateway supports all leading backup applications and caches virtual tapes on premises for low-latency data access. It compresses your tape data, encrypts it, and stores it in a virtual tape library in Amazon Simple Storage Service (Amazon S3). From there, you can transfer it to either Amazon S3 Glacier Flexible Retrieval or Amazon S3 Glacier Deep Archive to help minimize your long-term storage costs.

Tape Gateway  AWS Snowball with Tape Gateway

Back up and archive on-premises data to virtual tapes on AWS using your network.
Production Deployment at DESY

- CTA 5.7.14
- dcache-cta 0.9.0
- dCache 8.2.7
- IBM TS 4500
- 4xIBM TS1160
- 4xLTO9
- PostgreSQL 14.1
dCache+CTA Status

- Seamless integration with dCache is merged into upstream CTA code at CERN
  - The latest official CERN releases starting {4,5}.7.12 provide dCache required functionality
  - The proposed dCache interface is under adoption by EOS.

- The existing ENSTORE/OSM tape format is supported for READ
  - The ENSTORE/OSM tape catalog conversion procedures are successfully tested at DESY and Fermilab.
  - All HERA experiments and BELLE-II at DESY are migrated to CTA (5.4 PB)
  - EuXFEL migration will take place next week (Jul 17-21) (99 PB)

- dCache+CTA deployment replicate to by other HEP sites
  - Fermilab and PIC Barcelona have successfully replicated our setup (currently dCache + ENSTORE).
  - RAL in UK plans to migrate to PostgreSQL from ORACLE based on our experience
dCache Bulk Service
and WLCG TAPE API:
The Demo, Redux

Albert L. Rossi (FNAL)
dCache and WLCG TAPE

The Bulk Service

- Introduced last year.
- Since then, many improvements, especially a substantial reworking of the data storage layer.
- Will not describe, but simply demo the capabilities.
Big-Data Tools for Log Processing

Processing build on top of widely used tools.

By Christian Voß
Distributed dCache (datalake) Operation

Services: Storage

- 32 PB disk installed
  - Tier-1 pledged disk +
    - (Including some purchased disk for future pledges)
  - Swedish T2 +
  - Swiss (Bern) T2 +
  - Slovenian T2
    - (Including 3PB temporary commitment due to war)
- 19 PB tape installed
  - All in the Nordics

By: Mattias Wadenstein
Distributed dCache operation

Services: Storage

- 32 PB disk installed
  - Tier-1 pledged disk +
    - Including some purchased disk for future pledges
  - Swedish T2 +
  - Swiss (Bern) T2 +
  - Slovenian T2
    - Including 3PB temporary commitment due to war
- 19 PB tape installed
  - All in the Nordics

Challenges in federated storage

- The local funding agencies would like to see their contributions in WLCG storage accounting
- dCache supports SRR to publish partitioned storage accounting
- Apparently there is development needed in WLCG accounting in order to make this visible
  - Somewhat surprising to us
Distributed dCache operation

Services: Storage

- 32 PB disk installed
  - Tier-1 pledged disk +
    - (Including some purchased disk for future pledges)
  - Swedish T2 +
  - Swiss (Bern) T2 +
  - Slovenian T2
    - (Including 3PB temporary commitment due to war)
- 19 PB tape installed
  - All in the Nordics

Challenges in federated storage

- The local funding agencies contribute in WLCG?
- dCache supports SRR, storage accounting
- Apparently there is dedicated accounting in order to:
  - Somewhat surprising to us

Storage operations

- Local site admins maintain hardware, filesystem, operating system, networking, kernel tuning
  - Provides one unprivileged account with lots of storage to the central ops team
- Central ops team runs dCache pools
  - Install java + dCache
  - Configure, upgrade, restart dCache
- Investigating issues sometimes takes cooperation
  - Pool shutdown (central ops notice) due to IO error (investigation by both) because of raid controller issue (local ops fix)
**Collaboration**

- **A successful data lake is a successful collaboration between:**
  - Funding agencies - usually one in each participating country
  - Sysadmins - NeID central team and site admins at each site
  - Physics projects and their PIs - one to two per country for us
  - Networking providers - NORDUnet, GEANT, CERN, plus all NRENs
  - Researchers - the entire purpose of research infrastructure
  - Experiment coordinators - ALICE and ATLAS currently
  - Scientific computing centers - Nine currently participating
  - Coordinating body - Nordic e-Infrastructure Collaboration, NeID
  - etc
  - etc
HISTORY OF dCache AT CC-IN2P3

- Started getting familiar with dCache v1.6.5 in 2004
- Currently operating several instances for different projects
  - LHC [v8.2.16]
    - shared by ATLAS, CMS, LHCb
    - 38 PB, 157 servers, 155 M objects
  - EGEE [v7.2.27]
    - shared by CTA, Juno, Belle II, Calice, Dune, Xenon
    - 1.4 PB, 7 servers, 50 M objects
  - NESSIE [v8.2.16]
    - for R&D purposes, currently mainly ESCAPE and DOMA
    - 3 servers, 300 TB
  - Rubin LSST
    - the subject of this talk

By: Fabio Hernandez
dCache for Rubin LSST

**HISTORY OF dCache AT CC-IN2P3**

- Started getting familiar
- Currently operating systems
  - LHC [v8.2.16]
    - shared by ATLAS, CMS, LH
  - 38 PB, 157 servers, 155 M objects
  - EGEE [v7.2.27]
    - shared by CTA, Juno, Belle
    - 1.4 PB, 7 servers, 50 M objects
  - NESSIE [v8.2.16]
    - for R&D purposes, currently
    - 3 servers, 300 TB
- Rubin LSST
  - the subject of this talk

---

**Cloud**
- EPDC Data Center

**US Data Facility**
- SLAC, California, USA
  - Archive Center
  - Alert Production
  - Data Release Production (35%)
  - Calibration Products Production
  - Long-term storage
  - Data Access Center
  - Data Access and User Services

**Dedicated Long Haul Networks**
- Two redundant 100 Gb/s links from Santiago to Florida (existing fiber)
- Additional 100 Gb/s link (spectrum on new fiber) from Santiago-Florida (Chile and US national links not shown)

**UK Data Facility**
- IRIS Network, UK
  - Data Release Production (25%)

**France Data Facility**
- CC-IN2P3, Lyon, France
  - Data Release Production (40%)
  - Long-term storage

**Summit and Base Sites**
- Observatory Operations Telescope and Camera
- Data Acquisition
- Long-term storage
- Chilean Data Access Center

---

2023-07-12

dCache workshop summary

28/43
dCache for Rubin LSST

**Dedicated Long Haul Networks**
- Two redundant 100 Gb/s links from Santiago to Florida (existing fiber)

**UK Data Facility**
- IRIS Network, UK
  - Data Release Production (23%)

**Rubin Data Preview: Data Products Sizes**

- **US Data Facility**
  - SLAC, California, USA
    - Archive Center
    - Alert Production
    - Data Release Production (35%)
    - Calibration Products Production
    - Long-term storage
    - Data Access Center
    - Data Access and User Services

- **HQ Site**
  - AURA, Tucson, USA
    - Observatory Management
    - Data Production
    - System Performance
    - Education and Public Outreach

High number of (very) small files resulting from processing of raw images
dCache for Rubin LSST

Activity Profile: LCG vs Rubin LSST

LCG instance
5M events
(HTTP + XRootD)
15k jobs
157 pools

Rubin LSST instance
21M events (webDAV)
5k jobs in execution
19 pools

Observed activity over a representative period of 48h

HISTORY of dCache at CERN
- Started getting data from LHC in 2012
- Currently operating
- LHC [v8.2.16]
  - shared by ATLAS and CMS
  - 38 PB, 157 servers
- EGEE [v7.2.27]
  - shared by CTA and others
  - 1.4 PB, 7 servers
- NEST [v8.2.1]
  - for R&D purposes
  - 3 servers, 300 TB
- Rubin LSST
  - the subject of this workshop

2023-07-12

dCache workshop summary
dCache instances are isolated per SC

- SC diverge in their requirements
- Procurement and resource control
- Infrastructure supported on physical and virtual Machines

Storage technology flexible to support SC individual requirements
dCache at BNL

Towards an Improved dCache Operation

Areas of work:

- Enhancing software for interaction among dCache and TAPE HPSS systems
  - ENDIT archiver/retriever
- Improving dCache data access workflows for client access
  - Non firewalled Xrootd client access for write/read
  - DUAL IPv4/IPv6 dCache application stack configuration
- Extending monitoring for dCache operations
- Evolving dCache along with infrastructure
Towards an Improved dCache

**dCache and ELK Stack**

Started to be Used in Operations

Filebeat / Logstash pipelines enabled for domain logs and billing logs

ELK use to mine the billing logs with arbitrary queries

Monitoring Enhancement

Grafana based monitor using the dCache billing/chimera/srm databases to provide information use in operations

Allows aggregate information from different dCache events by entering the PNFSID (dCache file ID)

Performance of dCache
Performance evaluation & comparison: Lustre, dCache, Xrootd, zfs, etc
OIDC and all that Jazz

By: Marina Sahakyan
OIDC and Token-based Access

dCache how does access token work

browser

oidc-agent

Relying Party

client_id
client_secret

authorization_code

OAuth 2 Authorization Server

authorization_code + client_secret

access_token

OAuth 2 Resource Server

client_id + desired scopes

authorization_code

IAM

Google

client/RP

06/01/2023

06/01/2023
Identifying Token Types

- It can be confusing sometimes to distinguish between the different token types.
  - **ID tokens**
    - carry identity information encoded in the token itself, which must be a JWT
  - **Access tokens**
    - used to gain access to resources by using them as bearer tokens
  - **Refresh tokens**
    - exist solely to get more access tokens
Sustainability

By: Onno Zweers
Sustainability

By: Onno Zweers
Our dCache uses ~5W per stored TB.

At the moment a little bit less because of decommissioned hardware

Reading & writing does not make a big difference

By: Onno Zweers
Wrap-up

• Moving data from old, small, inefficient servers to new, large, efficient servers and switching them off early is an easy way to reduce energy consumption.

• Still investigating environmental impact, especially production of the hardware. When we have time.

• Haven’t looked at hardware energy settings yet.

• “Green” electricity often based on certificates – doesn’t help. Buying locally produced green electricity may stimulate energy suppliers to become more sustainable.

• All this pales in comparison to the effect you can have as a climate activist.
Thank You!

More info:
https://dcache.org

To steal and contribute:
https://github.com/dCache/dcache

Help and support:
support@dcache.org, user-forum@dcache.org

Developers:
dev@dcache.org
Useful Links

- **Workshop Indico:**
  - https://indico.desy.de/e/dcache-ws17

- **dCache documentation:**
  - https://www.dcache.org/documentation/

- **Mini hands on:**