CernVM-FS at Extreme Scales

CHEP 2023, Norfolk, VA, USA

Jakob Blomer\textsuperscript{1}, Laura Promberger\textsuperscript{1}, Valentin Völkl\textsuperscript{1} and Matt Harvey\textsuperscript{2}

May 9, 2023

\textsuperscript{1}CERN, Experimental Physics Department, Switzerland
\textsuperscript{2}Jump Trading
Motivation

Expectation for HL-LHC

Increase of all CVMFS metrics by an order of magnitude

Accumulation of (existing) data

- More versions
- More architectures/compilers
- Larger software projects

Extending use cases

- Faster release cycles
- Higher usage of containers
- More repositories
Motivation II

New versions up to 22% larger and externals are 10 - 220% larger
ALICE OCDB has on average 20 new revisions per day

LHCbDev has **on average 1067 new revisions per day**
Good News: We know already it works...

Jump Trading: Growth of data of the data archive

Archive Total Size [PiB]

Cumulative Total PiB  Trendline (exponential)

Full presentation https://indico.cern.ch/event/1079490/
Good performance achieved through multiple level of caches, data is stored in the cloud

Designed for the next 10 years

Can scale by orders of magnitude:
- Storage PB
- Network links from a colo to cloud provider
- Number of colos
- Number of fabric cache/gateway boxes
- Amount and performance of regional internal cache
CVMFS Challenges and Solutions

"Problems"

- Growth of data
- Acceptance in community means more opportunities where cvmfs is used

Solution

- Optimize performance by smarter caching in all locations
- Increase ease-of-use of end users and operators
- Optimize download bottleneck
**Improvements**

**Caching Performance**

- **2.10** Page Cache Tracker: Much better use of kernel page cache
- **2.10** Support for in-place replacement of files without crashing long-running software that use the “old” version of these files
- **2.11** Symlink caching for fuse3 (Kernel 6.2, RedHat backporting request open)
- **2.11** Statfs caching
- **WIP 2.11** Proxy sharding to allow for better caching
- **Future** Prefetching of known files clusters (Python, ROOT, etc.)

**Download Improvements**

- **WIP 2.11** Parallel file decompression during download
- **Future** Zstd as new compression algorithm
Operational Improvements

- 2.10 More extended attributes, and 2.11 protected extended attributes
- 2.10 Better publish failure handling on publishers
- 2.10 Support for unpacking container images through Harbor registry proxies
- 2.11 Telemetry exposure of internal affairs to allow better monitoring
- 2.11 Quicker garbage collections and cvmfs_server check
- Future Creation of official Helm chart for cvmfs on Kubernetes
- Future Feature parity between remote publishers (with gateway) and local publishers
ATLAS Performance: CVMFS version 2.9 vs 2.10

Many-core compilation of ATLAS Athena with having the build tools on cvmfs

Improvements due to the page cache tracker
Some First Performance Comparison - Setup

Setup

- CVMFS client: 2x AMD EPYC 7302 16-Core, 256 GB RAM, 2 TB NVMe
- Private squid proxy: 1x Intel i7-7820X 8-Core, 64 GB RAM, 1 TB HDDs

Commands: Load software from CVMFS

- CMS: Create a simulation setup script
- DD4Hep: Load detector description in ROOT
- ROOT: Load ROOT and draw a histogram
- Tensorflow: Load python and the modules numpy and tensorflow

Measurements

- Cold, warm, and hot cache on full machine (1 proc per hyper-thread)
- time, cvmfs_talk -i <repo> internal affairs
Some First Performance Comparison - version 2.9 vs 2.11 (WIP, April 23)

(Real) run time in seconds

![CMS Performance Graph]

![Tensorflow Performance Graph]
Some First Performance Comparison - Symlink Caching

CVMFS v2.11 (WIP, April 23) with and without symlink caching
(Default Client Config: Statfs Caching, Kernel Caching)
Compressing CVMFS cache file chunks

<table>
<thead>
<tr>
<th>Library</th>
<th>uncompressed</th>
<th>zlib</th>
<th>zstd</th>
</tr>
</thead>
<tbody>
<tr>
<td>#Files</td>
<td>1004</td>
<td>1004</td>
<td>1004</td>
</tr>
<tr>
<td>Size (MB)</td>
<td>2300</td>
<td>999</td>
<td>866</td>
</tr>
<tr>
<td>Time (min)</td>
<td>-</td>
<td>1:36</td>
<td>0:15</td>
</tr>
<tr>
<td>Compression Ratio</td>
<td>-</td>
<td>2.30</td>
<td>2.66</td>
</tr>
</tbody>
</table>

Zstd saves 15% in space and is 6x faster than zlib
• CVMFS expects an order of magnitude growth in all metrics for HL-LHC

• Confident that the current design sustains the expected scale

• Rich set of performance and operational improvements underway to ensure proper quality of service at HL-LHC scales
Questions?
CMSSW: Increase of Number of Files

Number of Files CMSSW

- 10_0_0
- 11_2_2
- 12_6_4
- 13_0_4

CMSSW
Externals
Some First Performance Comparison - Finding bugs

DD4hep

Tensorflow