XCache setup and support in Germany

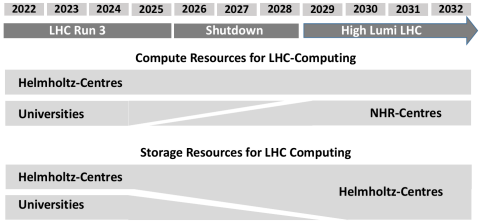
Guenter Duckeck[1] Alexander Heidelbach[1] Raphael Kleinemühl[2]

[1]Ludwig Maximilians Universität

[2]Bergische Universität Wuppertal

Motivation - NHR Centres

- Germany will do a gradual transition of university-based Tier-2 resources
  - CPU to 3 selected National High Performance Computing (NHR) centres
  - Mass storage to 2 Helmholtz-Centres (HGF)
- Local ATLAS/CMS groups keep supervising the NHR resources (funding for personnel) granted
- Not all NHR centres will be close to data
- Universities will have less storage
- dCache and XCache are investigated as caching solutions



BERGISCHE UNIVERSITÄT WUPPERTAL

Motivation - Experience so far

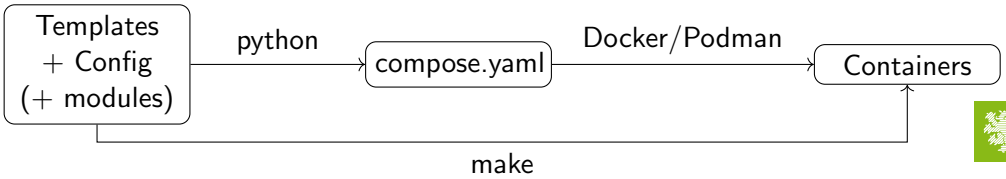
- So far, each group went through the same process
  - Research: What is XCache? Is it a dedicated software?
  - Build: How to create an XCache? The documentation is extensive, which is good, but also make it difficult for new people
  - Deploy: How do other sites do it? Every site has built from scratch. Then there is Kubernetes and SLATE
- We want to simplify this process and make a starting ground, so that people don't start from scratch

BERGISCHE UNIVERSITÄT WUPPERTAL

XCache Setup

Goal: Provide a basic but extensible XCache Setup that can be reused by several sites.

- We use a Docker/Podman Compose Setup, running the XCache as well as other services in containers
- Compose files are provided as Jinja templates, and rendered in Python
- Process is extensible with additional modules
- Everything is configured in a single `config.yaml`
- The process is defined in a `Makefile` for convenience



BERGISCHE UNIVERSITÄT WUPPERTAL

XCache Setup - modules

- Modules are housed in separate subdirectories
- They are optional and can be ignored if not needed
- Modules are activated in `config.yaml`:

```
cat config.yaml
<...>
#### Specify which modules to use
use_localmon: true
use_vp_atlas: false
#### Settings for module ‘vp-atlas‘
# Instance and address of XCache in VP
xc_instance: instance01
xc_address: xxx.xxx.xxx.xxx:1094
```

```
xcache_modular
|-- modules
|   |-- localmon
|   |   ‘-- compose.yaml.j2
|   ‘-- vp-atlas
|       ‘-- compose.yaml.j2
|-- compose.yaml.j2
|-- config.yaml
|-- Dockerfile
|-- Makefile
‘-- render_templates.py
```

BERGISCHE UNIVERSITÄT WUPPERTAL

Module localmon

- Module for a local monitoring of the XCache
- Incorporates summary data and g-Stream data
  - Summary data is collected via `https://github.com/GSI-HPC/xrootd_exporter`
  - g-Stream data is collected via our own collector
  - Both streams are exorted to a Prometheus
- Visualized in a Grafana Dashboard

BERGISCHE UNIVERSITÄT WUPPERTAL

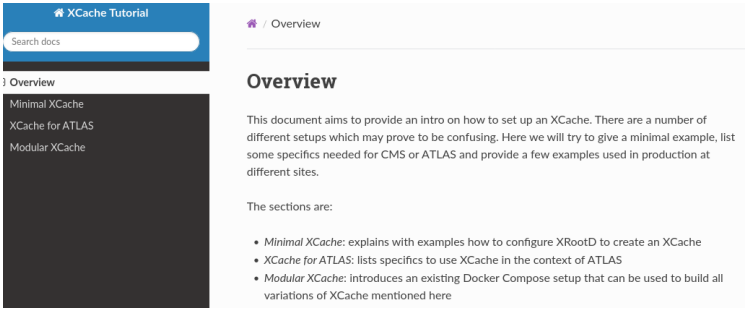
Module vp-atlas

- The module provides site-integration with ATLAS Virtual placement
  - Heartbeats to VP service and Rucio
  - Reports to external monitoring
- Similar modules can be added for other experiments

BERGISCHE UNIVERSITÄT WUPPERTAL

Documentation

- For a first documentation see our readme at `https://gitlab.com/fuse_xcache/xcache_modular`
- We are building a more extensive guide (WIP) at `https://xcache-docs.readthedocs.io/en/latest/`



BERGISCHE UNIVERSITÄT WUPPERTAL

Benchmarking

- We aim to Benchmark XCache deployments
- How do Job performance and raw storage IO translate?
- There have been several performance tests of XCache deployments, but these often lack data about machine performance
- Job performance is measured by Hammercloud (ATLAS/CMS) jobs
- IO performance is measured by the benchmarking tool Fio

BERGISCHE UNIVERSITÄT WUPPERTAL

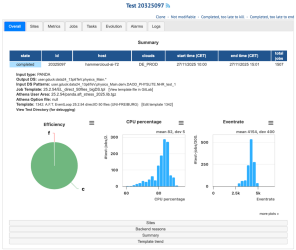
Benchmarking - Hammercloud

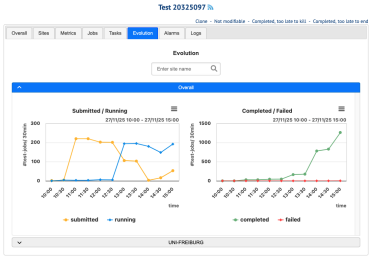
- Goal: Prepare test-suite for systematic evaluation of site performance, in particular compute - storage configurations
- Hammercloud: established service in ATLAS&CMS to systematically test sites in WLCG
  - functional tests for site validation/exclusion
  - dedicated stress tests for sites or groups of sites
- Monitoring: detailed monitoring information essential to analyze tests and compare setups
  - basic HC stress test monitoring
  - complemented by detailed job-logs from ATLAS panda WFMS system
  - jupyter notebooks for detailed analysis

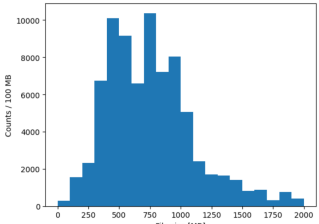
BERGISCHE UNIVERSITÄT WUPPERTAL

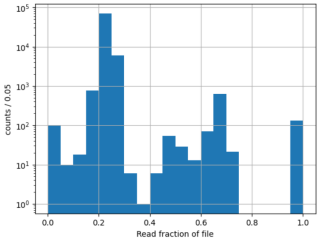
Benchmarking - Hammercloud

- We created a Stress Test Workload based on simple ATLAS analysis job with minimal CPU need
- Created reference dataset (6750 files, 5.1 TB total), replicated to several storage sites



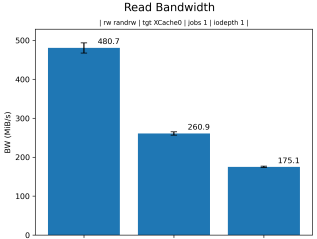


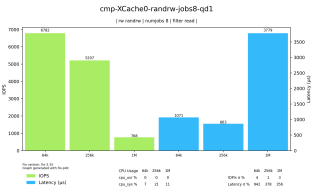


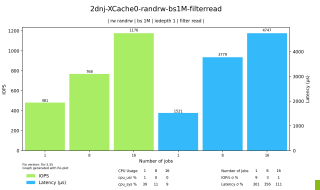


Benchmarking - Fio

- We perform local storage benchmarks with Fio (Flexible I/O tester)
- We automate running Fio and analyzing its reports with our own Python package
- Is currently rewritten due to limitations of 3rd-party packages







Future plans

- Thoroughly test our setup at Göttingen (NHR), Munich and Wuppertal
- Real benchmarks at these sites
- Adapt to possible limitations and specifics of NHR sites
- Expanding documentation and tutorials
- Investigate the possibility of using `blktrace`/`bpftrace` with fio's replay mode

BERGISCHE UNIVERSITÄT WUPPERTAL