HIFIS transfer service: FTS for Helmholtz

Tim Wetzel¹, Paul Millar¹, Uwe Jandt¹, Patrick Fuhrmann¹

¹ Deutsches Elektronen-Synchrotron DESY

International Symposium on Clouds and Grids
Mar 24, 2021
Helmholtz Association

- Founded in 1995 to formalise relationships between research centres
- **Members**: 19 autonomous research centres in Germany
- **Mission**: Contributions to grand challenges facing society, science and industry
- **Fields**: energy, earth & environment, health, aeronautics, space & transport, matter and key technologies

- Growing importance of cloud access to **common data treasure** and -services
- Rapidly growing **data exchange** from research instruments requires excellent data networking
- Growing connections between Helmholtz, EOSC and FAIR
Helmholtz aims for **joint research & information environment** for all Research Fields
Why data transfers?

- Large data sets in **collaborative** research projects
- Data analysis often sensitive to **latency**
- **Data locality** is important!
- **HIP** and **HelmholtzAI** projects use data storage at the computing sites
- Collaborating centres **distributed** over Germany
  - Reliable, comfortable and robust transfer endpoints needed
Transfer service

- Core service in HIFIS backbone
- CERN’s FTS3 as backend
- webFTS as comfortable WebUI
- FTS3-REST as CLI for scripted transfers
- Planned: Rucio for policy driven transfers

- Apache httpd as passive endpoint
- Lightweight solution for ad-hoc transfers
Passive endpoint components

User

WebUI

Policies

WebFTS

API

AAI

http://

SE

Disk

want-digest

Checksum

File

Local user mapping

mpm- itk

OpenSSL

APACHE

HTTP SERVER PROJECT

RUCIO

FTS

WebFTS

Disk
Instance digests

- mod.want_digest (github.com/wetzel-desy/mod.want_digest):
  - Implements instance digests in accordance with RFC 3230 (HTTP headers “Want-Digest” and “Digest“)
  - Supports ADLER32, MD5 and SHA digests
  - Alpha version until now
    - Digest caching mechanism or on-demand calculation
    - Cached digests return faster for large files than on-demand calculation
    - Coupled with inotifywait daemon for non-HTTP transfers
  ➔ Code cleanup next
Openstack VM

Confis/sources available on
https://gitlab.hzdr.de/hifis/hifis-transfer-service
Kubernetes/Helm

Repo A
- Dockerfile + X
- helm chart

Repo B
- helm chart

Chartmuseum
- helm chart

Registry
- Docker image

CI/CD

Node 1
- Pod

Node 2
- Pod

Node n
- Pod

Admin
- helm install values.yaml

Developer

Configs/sources available on https://gitlab.hzdr.de/hifis/fts-apache-k8s
Performance/Learnings

- Performance
  - Transfer rates of **40-120 MiB/s** (overall) reached in tests
  - **Negligible** overhead from virtualization
  - Fast retrieval of instance digests thanks to **caching**

- Learnings:
  - Transfers can interfere with **k8s readiness/liveness** probes
  - Data access must be regulated **internally**
  - Need to disable **mod_deflate**, prevents sending **content-length** in **http responses**
  - Custom patch of **mpm-itk** needs to be brought **upstream**
Summary & Outlook

- Apache **httpd** in VM/Docker container
  - Easily deployable, **lightweight** storage endpoint
  - **Comprehensible** setup
  - Suitable for **ad-hoc transfers** at smaller sites
- More testing, **optimization** and **enhancements** to happen
  - Use of **TPC** planned and in development (KIT)
  - Pilots with **EGI & WLCG**

All information (including git repository links) available on

https://www.hifis.net/doc/core-services/fts-endpoint/
Thank you! Questions?

Grateful acknowledgements:
Jan Erik Sundermann (KIT)
Mihai Patrascoiu (CERN)
Andrea Manzi (EGI)