Building a global file system for data access using Large Scale CVMFS and DynaFed

Alastair Dewhurst, Catalin Condurache, Louise Davies, Sam Skipsey

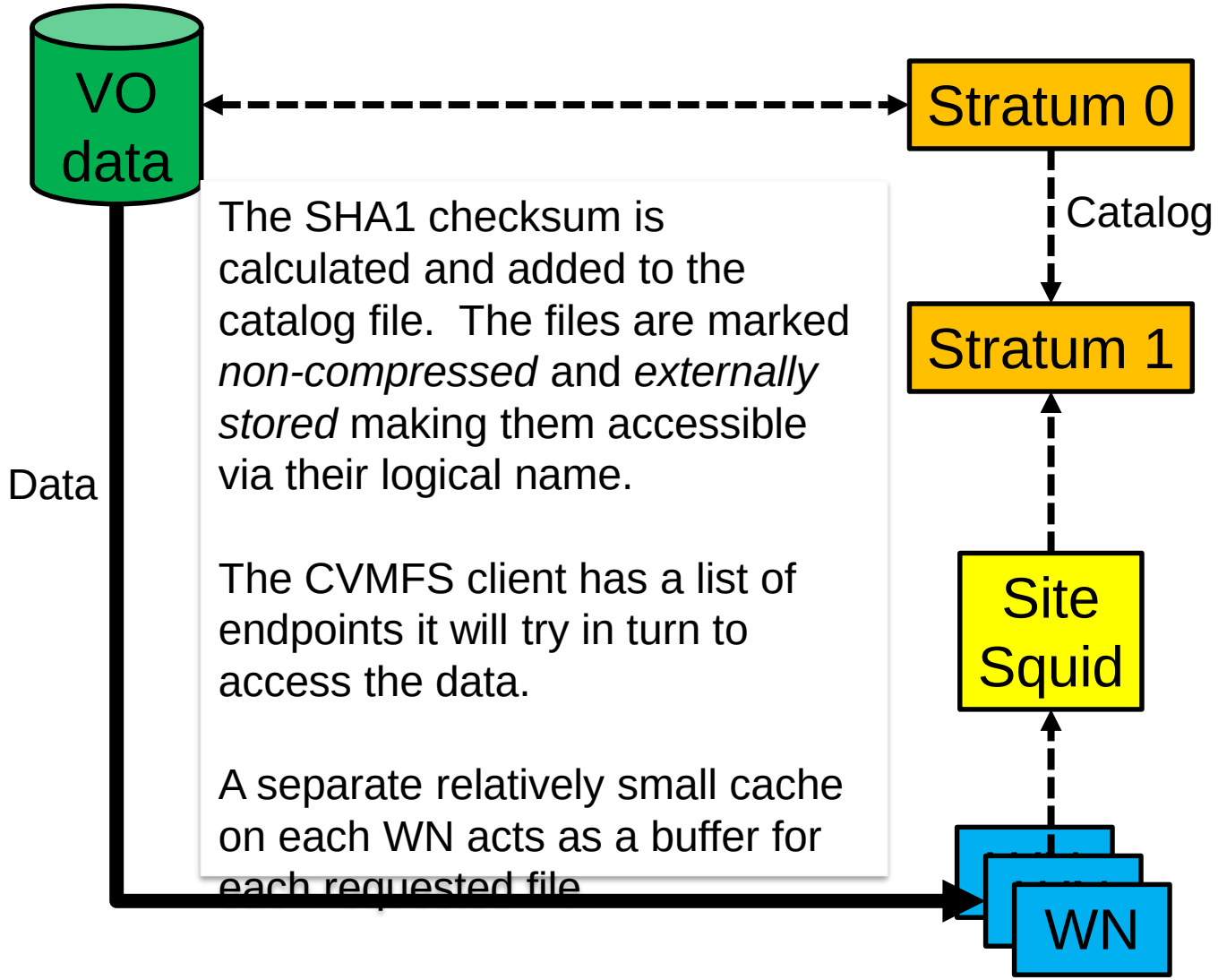
Motivation

- Effective Distributed Data Management has proven to be an extremely difficult problem to solve.
  - Large experiments have managed it but at significant effort costs.
- For most experiments, difficulty accessing their data limits where they can run their work.
  - Experiments are producing ever more data
  - Experiments have less effort and expertise in data management.
- We need to make it significantly easier for experiments to access their data.
  - POSIX and Web browser access.
  - No jump in difficulty as the amount of data grows.

CVMFS

- CVMFS has been primarily developed for distributing large software stacks.
  - Provides read only POSIX access.
  - Extremely effective and popular.
- Large Scale CVMFS is an extension to the base software which allows it to distribute large, non-public datasets.
  - Presented at CHEP16 by Derek Weitzel et al

Large Scale CVMFS



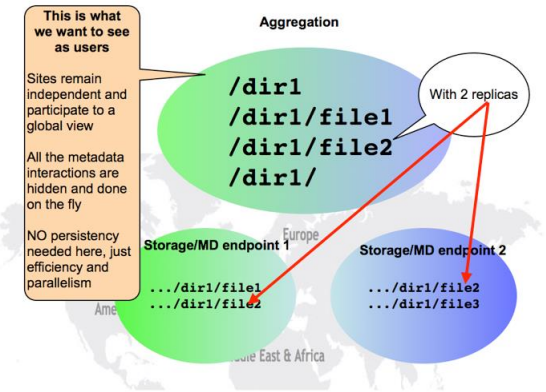
The SHA1 checksum is calculated and added to the catalog file. The files are marked *non-compressed* and *externally stored* making them accessible via their logical name.

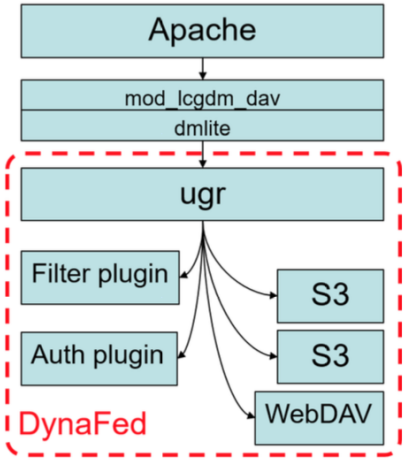
The CVMFS client has a list of endpoints it will try in turn to access the data.

A separate relatively small cache on each WN acts as a buffer for each requested file

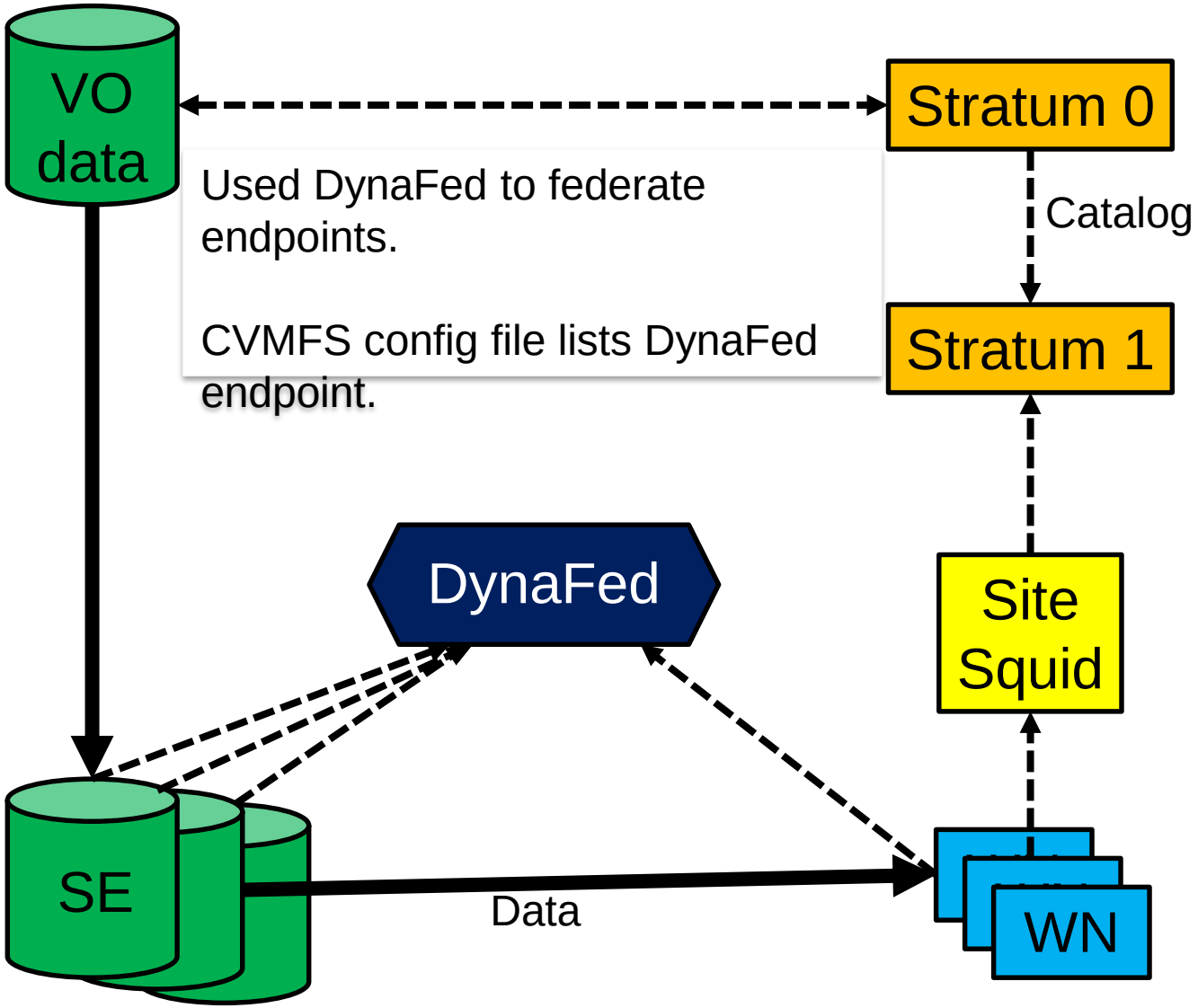
DynaFed

- DynaFed aims to provide a “Dynamic HTTP storage federation”.
  - Provides single namespace.
  - Redirects clients to closest file replica.
- DynaFed also provides access to Cloud storage.





Setup


VO data
Stratum 0
Catalog
Stratum 1
Used DynaFed to federate endpoints.
CVMFS config file lists DynaFed endpoint.
DynaFed
Site Squid
SE
Data
WN

What does DynaFed add?

- Allows access to data stored on RAL's S3 storage.
- Provides a single name space so that CVMFS doesn't have to try each endpoint in turn to access the data.
- If data is available at multiple sites, it will select the closest.
- Simplifies CVMFS configuration (which is needed on every client e.g. WN).
  - Makes it much easier to use opportunistic storage.
- Bonus: Provides access to data via a web browser.
  - We have written LDAP plugin for DynaFed

LIGO

- LIGO were the original user of Large Scale CVMFS for some of their work on the Grid.
  - The PyCBC workflow which needs ~10TB data.
- Number of UK LIGO collaborators rapidly growing.
- Data was being exported to Cardiff.
  - Rest of UK still accessing data direct from US.
- Data has been placed at RAL and started to be replicated to Glasgow.
  - Using DynaFed just works!
- Relatively small numbers of LIGO jobs running at RAL to make any performance/scalability claims.

Filter Plugin

- The Filter plugin is normally referred to as the geolocation plugin.
  - Because that is the default plugin available.
- Plugin will take list of endpoints and return one.
  - Is the same for both reads and writes.
- We looked at using “Rucio closeness” to select best replica.
  - Still need geolocation to map host to site.
  - Geolocation works pretty well.
  - Useful to add exceptions but unnecessary optimization most of the time.

Limitations?

- LS-CVMFS + DynaFed provides a solution for the problem of how to access data but:
  - Can't upload files by just "cp" to a CVMFS directory.
  - Uploaded files need to be grafted before they are available.
    - Workflows that have the output of one job being the input of the next job would need to wait if relying on LS-CVMFS
- Data distribution is a separate problem.
  - At RAL we are testing Rucio which would handle this half of the problem.

Next steps

- Euclid is a visible to near-infrared space telescope currently under development by the ESA.
- In August they are planning on running a MC campaign across the UK requiring ~10 Million CPU hours.
- Euclid have one person doing this who has developed his workflow on a local batch farm with a shared file system.
  - Output from some jobs are the input to the next.
- Challenge is to see how quickly we can graft files into LS-CVMFS

Conclusions

- Adding DynaFed to LS-CVMFS was a simple improvement that added many benefits.
  - All the components were already setup at RAL.
- I believe that LS-CVMFS + DynaFed will avoid problems as an experiments data grows.

References

- Large Scale CVMFS documentation: http://cvmfs.readthedocs.io/en/stable/cpt-large-scale.html
- Accessing data federations through CVMFS (2017): https://drive.google.com/file/d/0B_RVv_OjWcURUi15cmtUaXotVkU/view
- CVMFS for data federations (2016): https://indico.fnal.gov/event/10571/session/7/contribution/34/material/slides/0.pdf
- DynaFed whitepaper (May 2017): http://svnweb.cern.ch/world/wsvn/lcgdm/ugr/trunk/doc/whitepaper/Doc_DynaFeds.pdf
- Data Access for LIGO on the OSG (May 2017): https://arxiv.org/pdf/1705.06202.pdf
- StashCache documentation: http://opensciencegrid.github.io/StashCache/