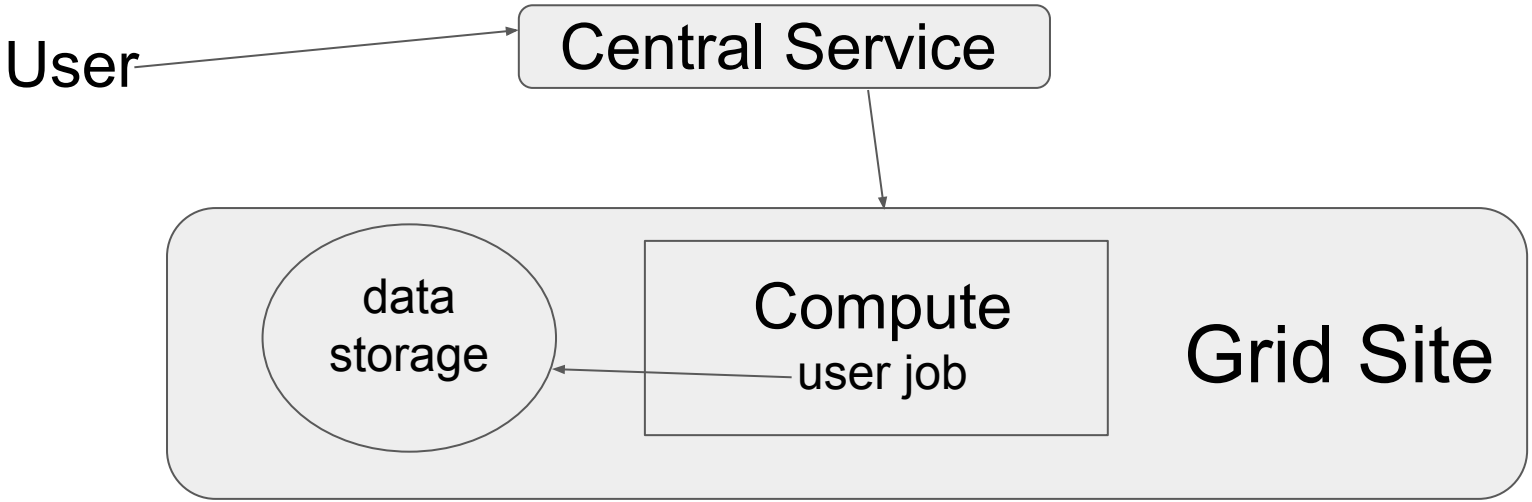
Using a dynamic data federation for running Belle-II simulation applications in a distributed cloud environment

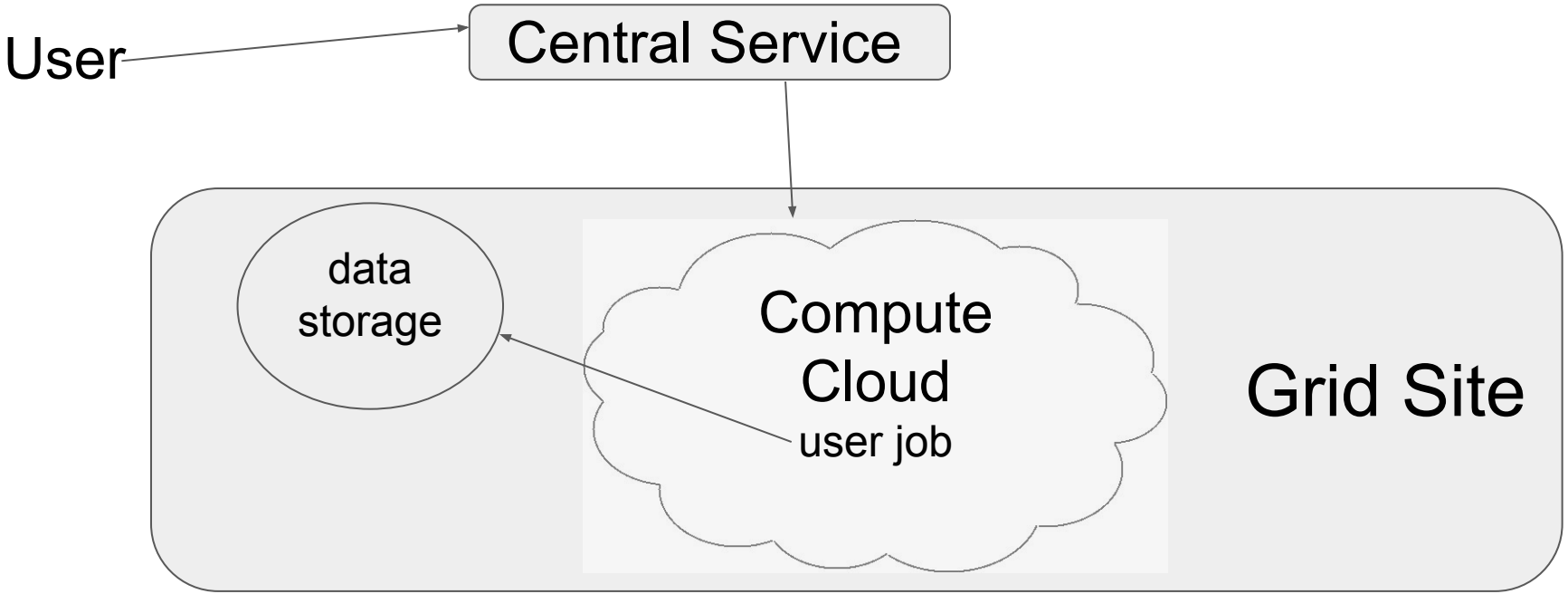
Marcus Ebert
mebert@uvic.ca
on behalf of the HEP-RC UVic group:
Frank Berghaus, Kevin Casteels, Colson Driemel, Colin Leavett-Brown, Michael Paterson, Rolf Seuster, Randall Sobie, Ryan Taylor (University of Victoria)
Fernando Fernandez Galindo, Reda Tafirout (TRIUMF)

- Why we use a dynamic data federation (Dynafed)
- What is Dynafed
- Dynafed for Belle-II jobs at UVic
- Performance comparisons

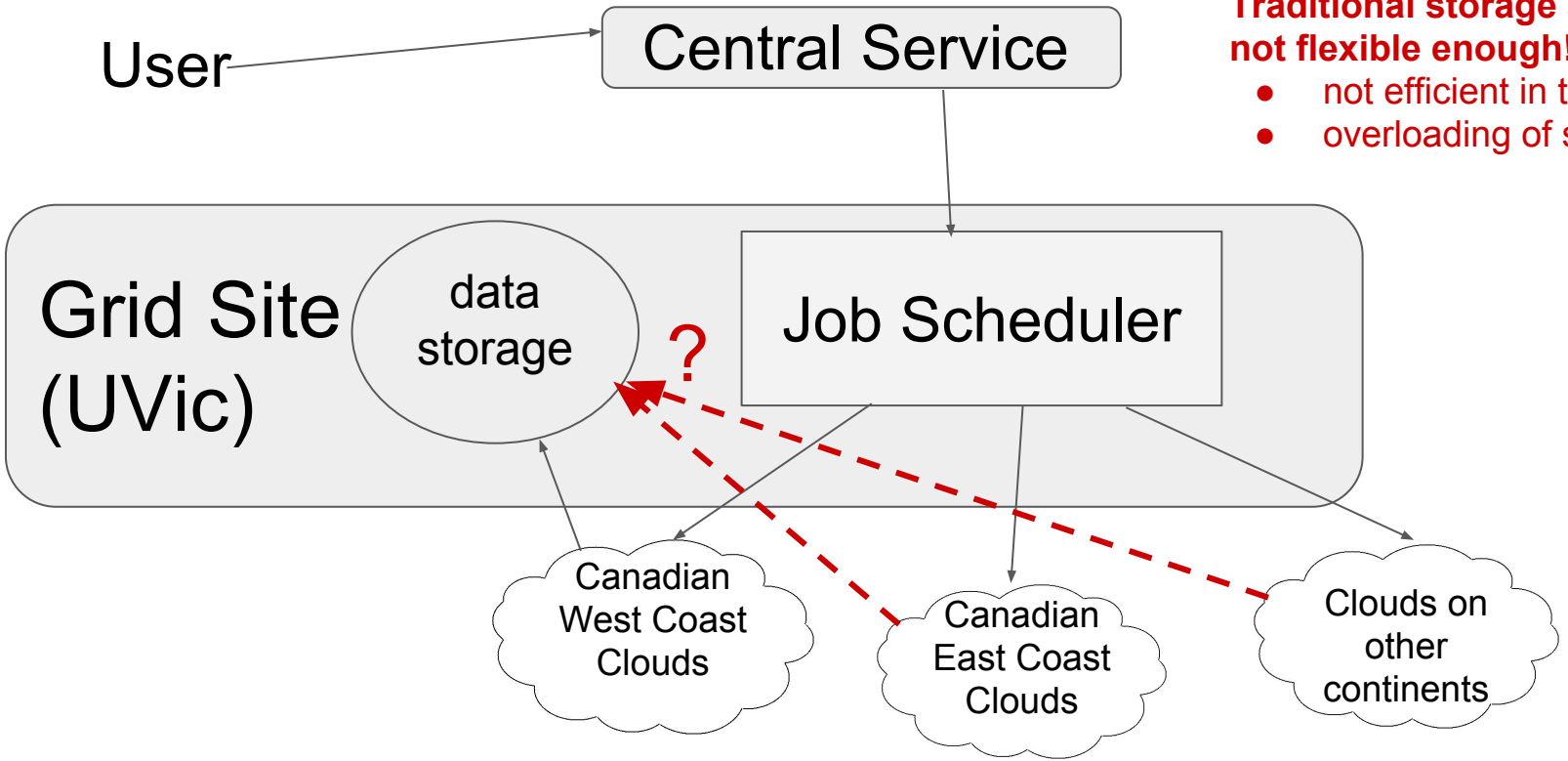
Traditional GRID site



Cloud computing for the GRID



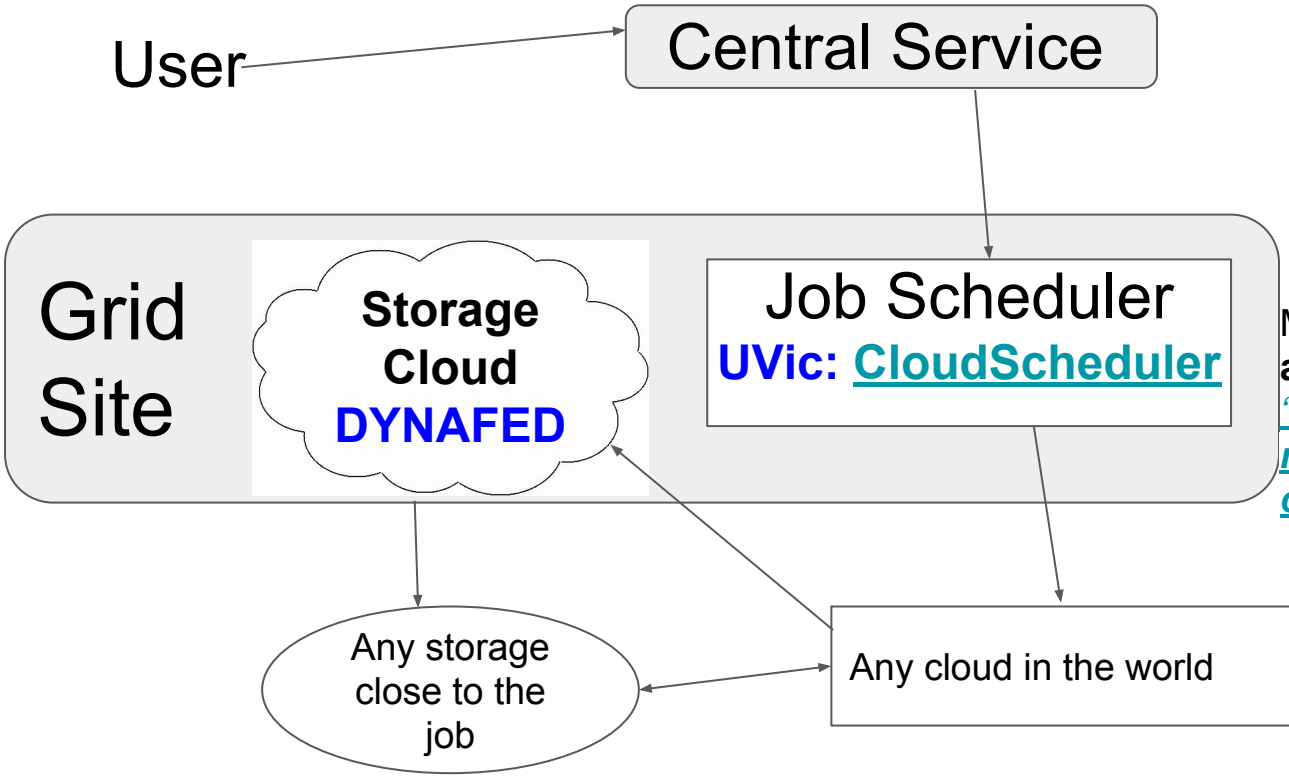
Cloud computing for the GRID



Problem:
Traditional storage systems are not flexible enough!

- not efficient in that situation
- overloading of single SE

Distributed cloud-storage and cloud-compute for the GRID



Storage cloud not only **useful** for cloud jobs but **also for compute-only sites and experiments as a whole**!

More about the **compute and accounting** in:

"Quasi-online accounting and monitoring system for distributed clouds"

(Wednesday, 11:30am T7)

"Sim@P1: Using Cloudscheduler for offline processing on the ATLAS HLT farm"

(Tuesday 11:30 T7)

Dynafed

- redirector for a dynamic data federation, developed by CERN IT
  - for data transfers, client is redirected to a storage element with the data
- access through http(s)/dav(s) protocols
- can federate existing sites without configuration changes at sites
  - site needs to be accessible by http(s)/dav(s)
  - world wide distributed data can be made accessible under common name space and from single endpoint
- in addition, can also directly access S3 and Azure based storage
  - no credentials visible to the client
  - preauthorized URL with limited lifetime is used to access files on the storage
  - no large Grid storage installation needed (DPM, dCache,...)
- X509/VOMS based authentication/access authorization can be used with dynafed
  - http://heprc.blogspot.com for grid-mapfile based authentication/authorization
    - different posts have also links to dynafed installation instructions in our TWiki

more details in poster #69 by Fabrizio Furano and Oliver Keeble

Some features using Dynafed

- redirecting client to nearest site that has data
  - in the future other characteristics could be added, like latency, bandwidth, storage space, cost,...
- client tools can get new redirect to another site if anything happens with an already established connection
  - site outage, network problems at a site,....
- root based tools can speak webdav and access data over network using dynafed
  - *TFile *f=TFile::Open("davs://dynafed.server:PORT/belle/path/to/file/file.root")*
  - uses external davix libraries
- writing into Dynafed also goes to one of the connected sites
  - uses also location based redirect

Dynafed for Belle-II at UVic

- Dynafed uses http(s)/dav(s) protocol which is supported by the gfal2 tools
  - currently Belle-II only supports srm and locally available data

- gfalFS can be used to mount endpoint
  - use Dynafed as endpoint for gfalFS
  - everything behind Dynafed appears as “local” data
  - still benefits from all Dynafed features
    - location based redirect
    - fail-over when endpoint goes down

- workflow: copy data from SE or local directory to workdir of the job
  - no streaming over network

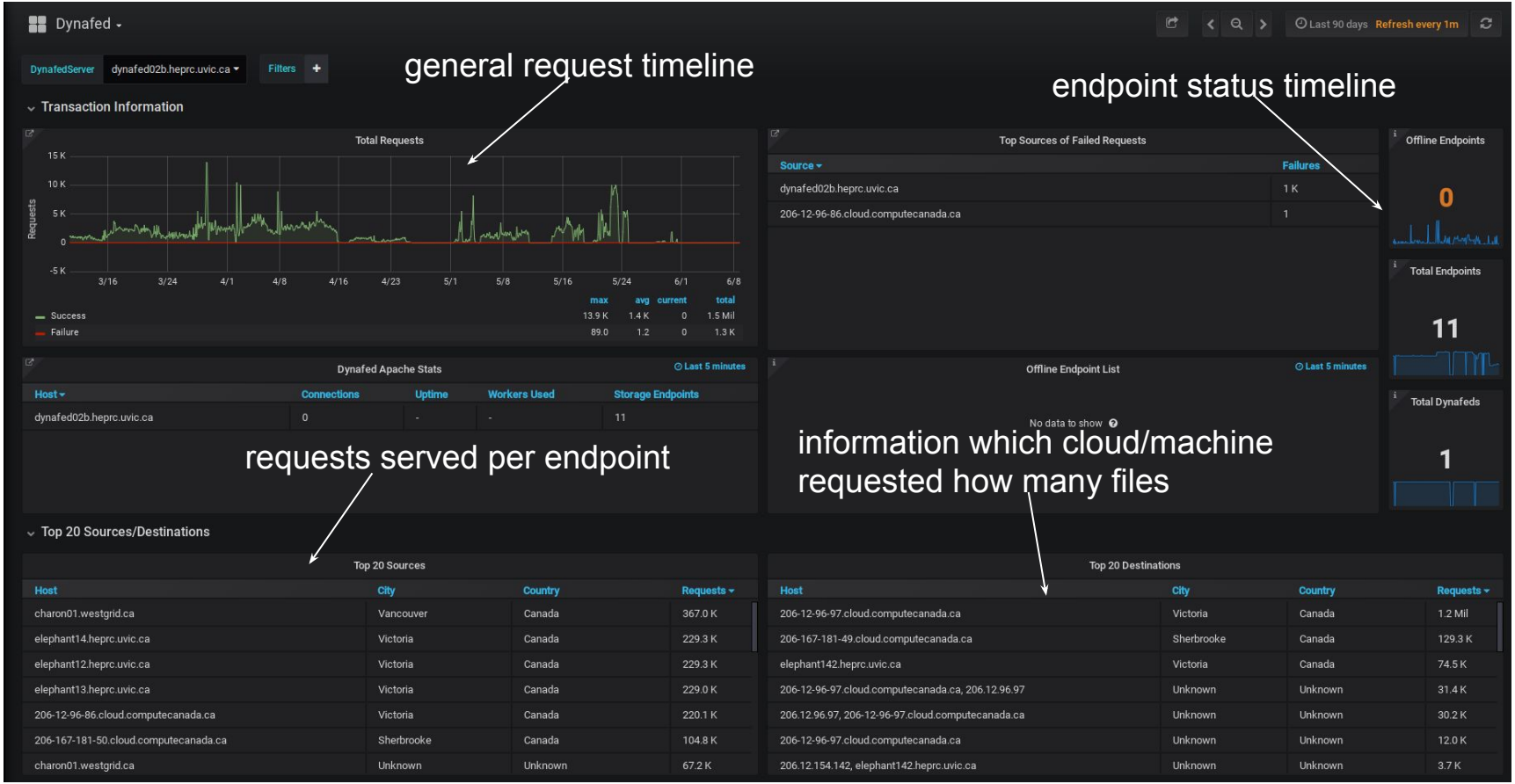
- ~3000 cores used in parallel for Belle-II
  - single core jobs
  - each job needs input file
  - ~30TB data per day transferred to jobs
    - very inefficient if using long-distance transfers (site SE -> cloud)
    - can also easily overload a single site SE

Dynafed for Belle-II at UVic

- Dynafed uses http(s)/dav(s) protocol which is supported by the gfal2 tools
  - currently Belle-II only supports srm and locally available data
- gfalFS can be used to mount endpoint
  - use Dynafed as endpoint for gfalFS
  - everything behind Dynafed appears as “local” data
  - still benefits from all Dynafed features
    - location based redirect
    - fail-over when endpoint goes down
- workflow: copy data from SE or local directory to workdir of the job
  - no streaming over network
- ~3000 cores used in parallel for Belle-II
  - single core jobs
  - each job needs input file
  - ~30TB data per day transferred to jobs
    - very inefficient if using long-distance transfers (site SE -> cloud)
    - can also easily overload a single site SE

see Frank’s Poster (#161) about Dynafed for Atlas

Dynafed monitoring



Testing Dynafed usage

- to make test comparable: using gfal-copy
  - dynafed instance at Victoria (Canadian West Coast)

- Grid and cloud storage endpoints behind Dynafed
  - Grid storage: UVic SE, BNL SE
  - cloud storage: Ceph at UVic, minio at Compute Canada Westcloud and Eastcloud VMs using volumes, minio at group's own cloud using large local root disk

- 500 3GB files copied to a worker node VM
  - copied to /dev/shm to not rely on virtual disk access

- gfal2 usage with:
  - srm access to UVic SE (site SE)
  - http access to UVic SE
  - dynafed access using grid sites behind it
  - dynafed access using cloud storage endpoints behind it

- VM on Compute Canada Westcloud (Victoria,BC), Eastcloud (Sherbrooke,QC), group's own development cloud (Victoria, BC)
  - do all copy processes on all clouds and compare results
  - cloud shared with other users
  - endpoint used by production jobs and other VOs

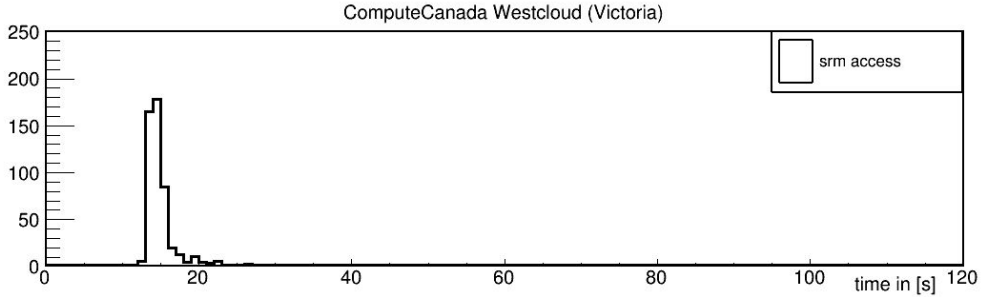
Testing gfalFS/dynafed

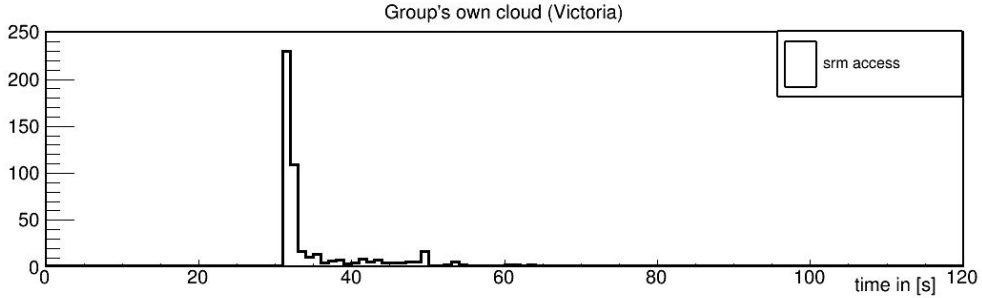
Victoria - Sherbrooke: 3853km

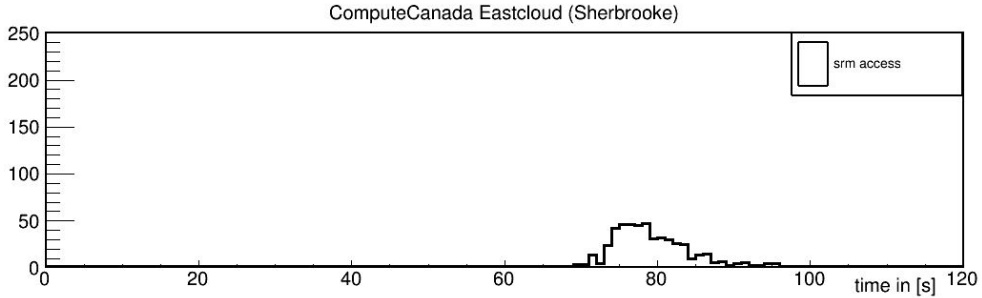
Sherbrooke - BNL: 515km

Victoria : 1xminio, 1xown Ceph, 1 Belle-II SE, 1xshared Openstack, 1xown Openstack
Sherbrooke : 1xminio, 1xshared Openstack
BNL : 1x Belle-II SE

Direct access to site SE (srm)

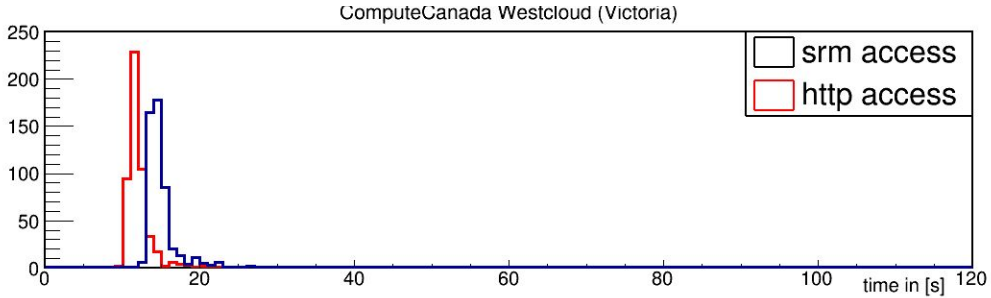


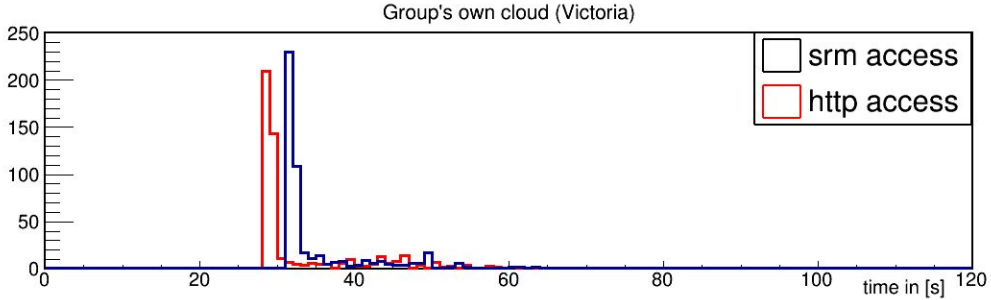


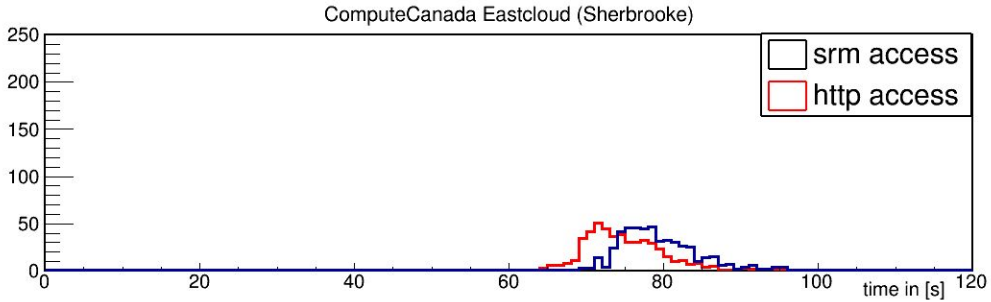


- direct access to site SE using srm protocol
- Westcloud nodes have direct access to our site SE
  - other clouds need to go through cloud router first

Direct access to site SE (http)

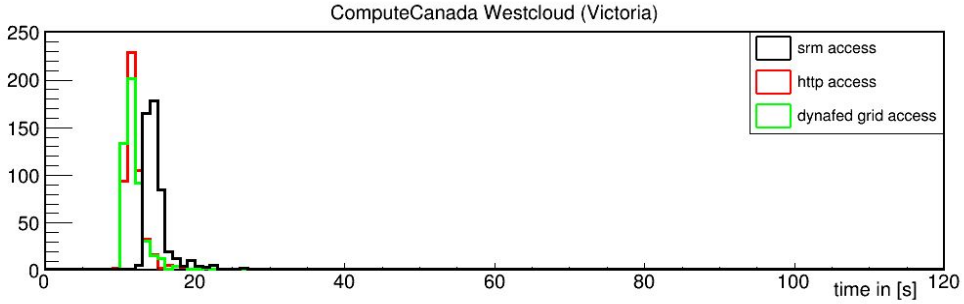


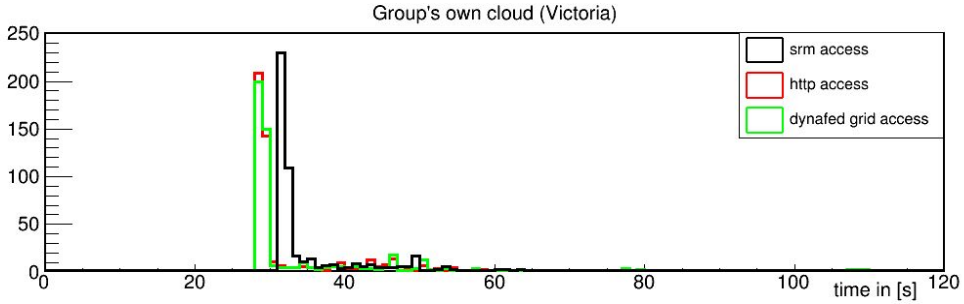


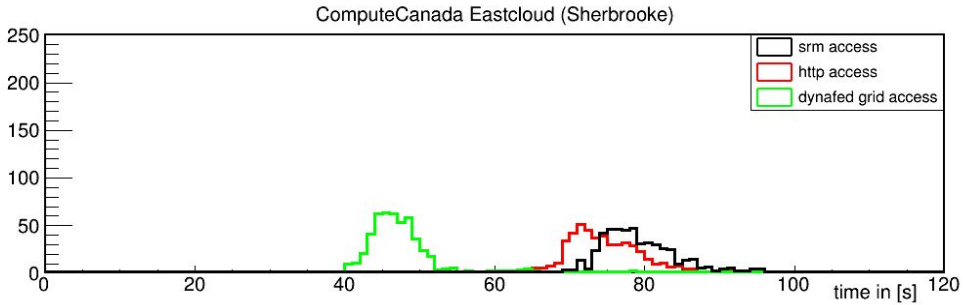


- direct access to site SE using http protocol
- performs better than srm access
  - consistent across all clouds

Access to grid sites (dynafed)

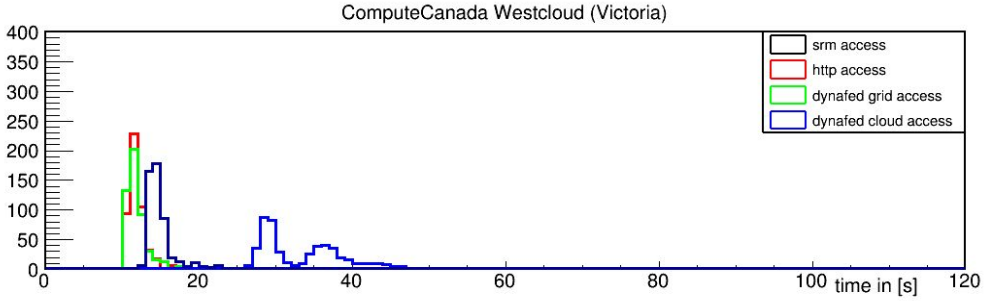


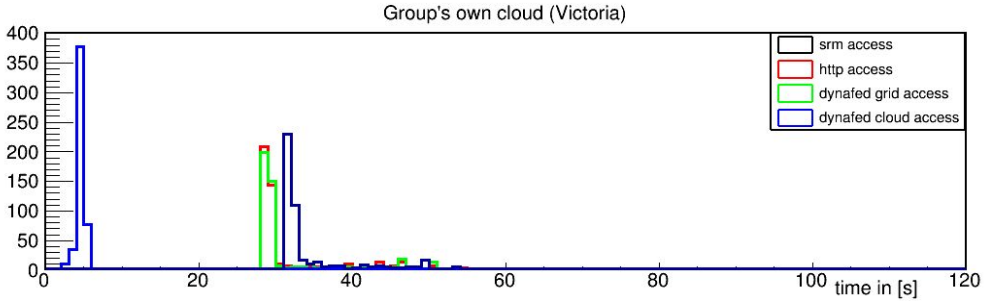


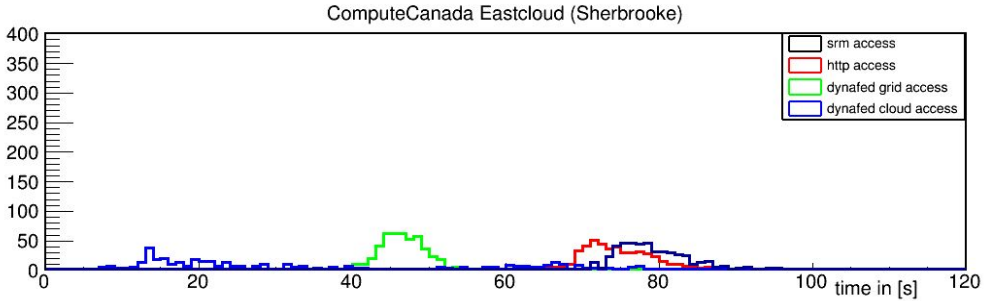


- using dynafed to access grid endpoints only
  - all 3 VMs access the same dynafed URL
- using dynafed not slower than direct http access
  - consistent between Westcloud and own cloud
- Eastcloud data access benefits from dynafed
  - dynafed chooses nearest endpoint
  - can use BNL grid site instead of site SE

Access to cloud sites (dynafed)







- using dynafed to access cloud endpoints only
  - all 3 VMs access the same dynafed URL
- cloud storage on Westcloud performs not as good as grid storage
  - data on volume mounted in minio VM (network based)
  - Ceph runs outside of cloud
    - many more layers than accessing site SE
    - needs to go through cloud network to access data on minio/Ceph
- on own cloud, minio performs best
  - data access within cloud
  - data stored on local “disk” of minio VM
- on Eastcloud setup like on Westcloud
  - smaller cloud
- on ComputeCanada clouds other users run VMs/data transfers too

Summary

- **Dynafed works very well for our distributed cloud computing when reading input data**
  - “instant” fail-over to other sites kept jobs running
  - relieves pressure on site SE
  - much more efficient to use than just site SE

- **gfalFS is a great way to use Dynafed when only “local” data access tools are supported**
  - all features of dynafed can be used

- **no performance issues when using dynafed compared to direct http access to site SE**

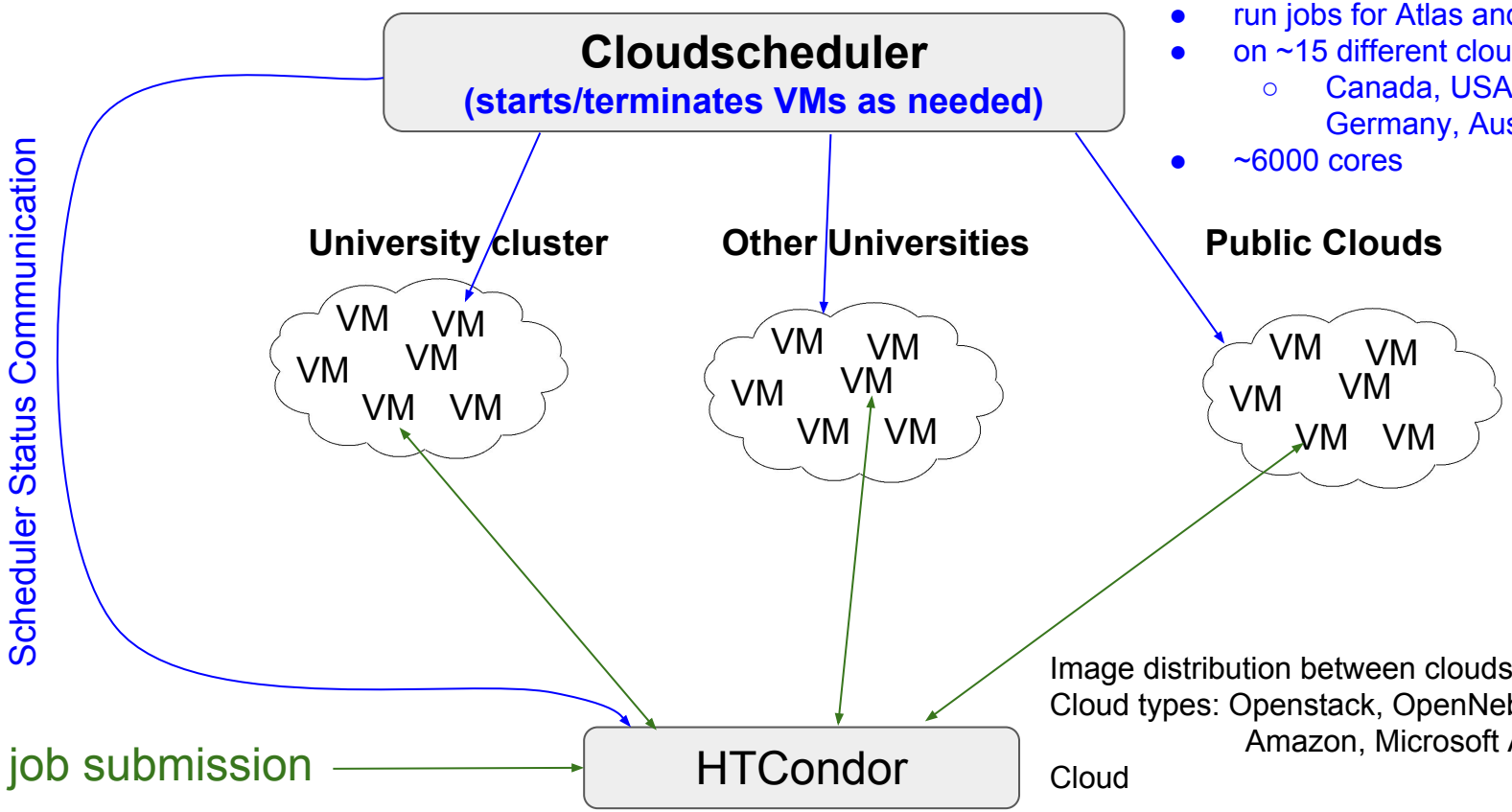
- **http access performs better than srm access**

- **dynafed makes it easy to add storage to the federation**
  - using dynafed as site SE will make the setup of grid storage much simpler
  - can natively use object storage which eliminates the whole grid storage infrastructure setup

Very positive experience in using it over the last year!

Backup

Cloud compute at UVic



- run jobs for Atlas and Belle-II
- on ~15 different clouds
  - Canada, USA, CERN, UK, Germany, Australia
- ~6000 cores

Image distribution between clouds: Glint
Cloud types: Openstack, OpenNebula, Amazon, Microsoft Azure, Google Cloud

gfalFS for Belle-II at UVic

- mount:
  gfalFS -s ${HOME}/b2data/belle davs://dynafed02.heprc.uvic.ca:8443/belle
- usage: *export VO_BELLE_DATA_DIR=$HOME/b2data*
- VOMS proxy of the job is used for the authentication and access authorization

Worked very well for about a year so far.

Problem occured recently:

- segfaults of gfalFS happen
  - ~since beginning of year
  - in libcrypto.so as part of openssl
  - ticket with CERN IT open
    - hard to debug since not reproducible manually
  - cronjob makes sure dynafed gets remounted when that happens

Advantages of using S3 based storage

- easy to manage
  - no extra servers needed, no need for the whole Grid infrastructure on site (DPM, mysql, apache, gridftp, xrootd, VOMS information, grid-mapfile, accounting, ...)
  - just use private/public access key in central Dynafed installation

- no need for extra manpower to manage a grid storage site
  - small group with budget to provide storage but no manpower for it: Just buy S3 based xTB for y years and put the information into dynafed ---> instantly available to the Grid,
    no need to buy/manage/update extra hardware
  - if university/lab has already large Ceph installation --> just ask for/create a bucket, and put credentials in dynafed

- industry standard
  - adopted from Amazon by Open Source and commercial cloud and storage solutions
    - HPC, Openstack, Ceph, Google, Rackspace cloud storage, NetApp, IBM,...

- scalable
  - traditional local file storage servers based on traditional filesystems will become harder to manage/use with growing capacity needs, same for other “bundle” solutions (DPM,...)
  - raid5 dead, raid6 basically dead too, ZFS will get problems with network performance