CVMFS Experience at Canadian HPC Sites

Bart Oldeman, Maxime Boissonneault, Ryan Taylor

McGill University, Université Laval, University of Victoria

Calcul Québec, BC DRI Group, Digital Research Alliance of Canada

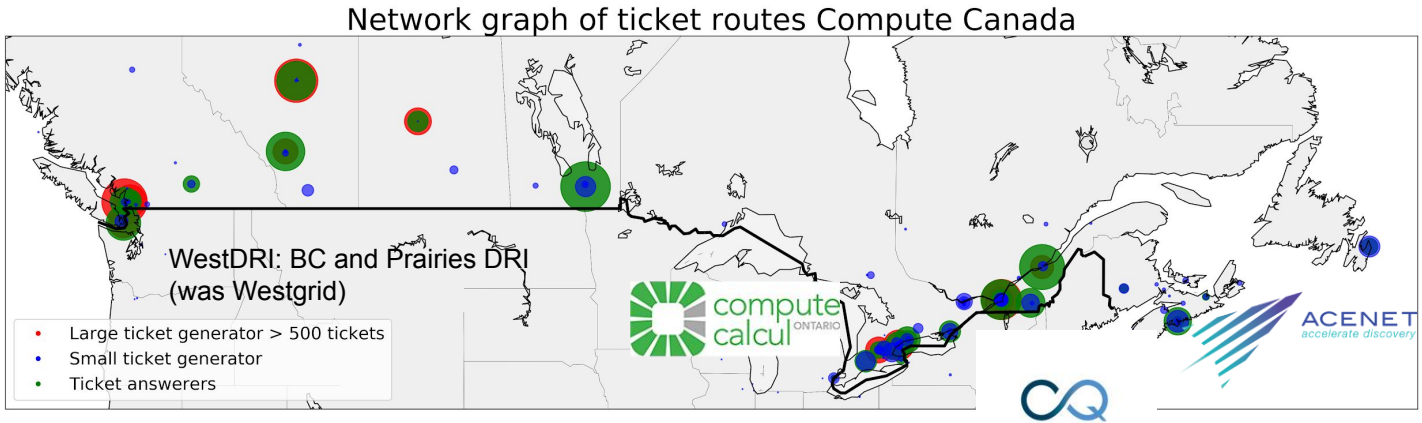
Compute Canada is now The Digital Research Alliance of Canada (The Alliance, National coordinating office, non-profit funded by Government of Canada).
The Federation = The Alliance + 38 partner universities + 5 regional organizations

Agenda

Software stack motivation
CVMFS Introduction
Site Operational Experience & Deployment Considerations
Security
- Confidentiality
- Integrity
- Availability

CVMFS Operations & Design at Canadian Sites

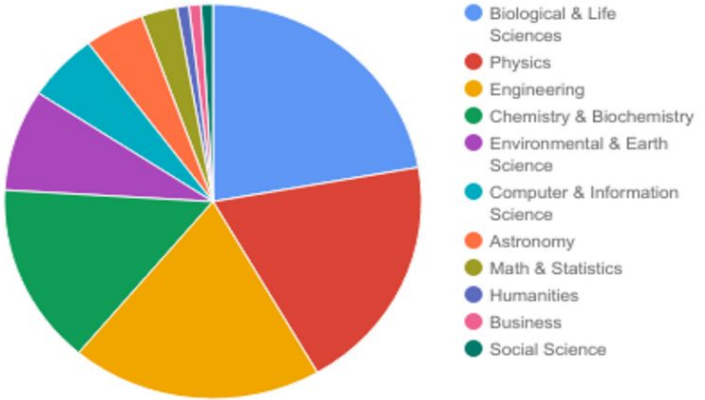
The people



All research disciplines supported

Free access for any researcher at a Canadian institution

- 5 regional consortia
- 38 member institutions
- ~250 technical staff
- ~18,000 user accounts
- 6 clusters, 4 clouds, 300k cores, 2k GPUs, 100s PB storage



The hardware

5 major national systems
300K cores, 30 PF
90 PB disk, 180 PB tape

System	Type	Network	Production
Arbutus	Cloud	10 GbE	2016 H2
Cedar	General	OPA	2017 H1
Graham	General	EDR IB	2017 H1
Niagara	Large MPI	EDR IB	2018 H1
Béluga	General	EDR IB	2019 H1
Narval	General	HDR IB	2021 H2

Starting in 2017, new bigger national systems replaced many smaller local clusters, with common software stack, scheduler (Slurm), and so on, administered by national teams.
Many sites have no physical cluster but still support.

Background

Most HPC clusters use enterprise Linux distributions for good reasons (vendor support for network, parallel filesystems, etc)

CentOS/RHEL 7

Linux kernel 3.10, GCC 4.8.5, Glibc 2.17, Python 2.7.5 (+ backports of course)

CentOS/RHEL/Rocky/... 8

Linux kernel 4.18, GCC 8.4, Glibc 2.28, Python 3.9.2 (+ backports of course)

CentOS/RHEL/Rocky/... 9

Linux kernel 5.14, GCC 11.2.1, Glibc 2.34, Python 3.9.10

compare:
Fedora 40
Linux kernel 6.8.5, GCC 14.0.1, Glibc 2.39, Python 3.12.2

Goal

Users should be presented with an interface that is as **consistent** and **easy to use** as possible across **all sites**. It should also offer **optimal performance**.

1. All software should be accessible on every site, reliably and performantly.
2. Software should be independent from the underlying OS stack.
3. Software installation should be tracked and reproducible via automation.
4. The user interface should make it easy to use a large and evolving software stack.

What this means

All new national sites

1. Need a distribution mechanism
    a. CVMFS : CERN Virtual Machine File System

Consistency

2. Independent of the OS (Ubuntu, CentOS, Fedora, etc.)
    a. Gentoo Prefix
3. Automated installation
    a. EasyBuild

Easy to use

4. Needs a module interface that scale well
    a. Lmod with a hierarchical structure

Software: design overview

Easybuild layer: modules for Intel, NVHPC, OpenMPI, CUDA, MKL, high-level applications. Multiple architectures (x86-64-v[34], extensible to others)

```
/cvmfs/soft.computecanada.ca/easybuild/{modules,software}/2023/x86-64-v{3,4}
```

Compatibility: Gentoo Prefix layer: GNU libc, autotools, make, bash, cat, ls, awk, grep, etc.

```
module gentoo/2023 => $EPREFIX=
/cvmfs/soft.computecanada.ca/gentoo/2023/x86-64-v3,  $EBROOTGENTOO=$EPREFIX/usr
```

OS kernel, daemons, drivers, libcuda, anything privileged (e.g. the sudo command): always local. Some legally restricted software too

Canadian ARC Software Stack

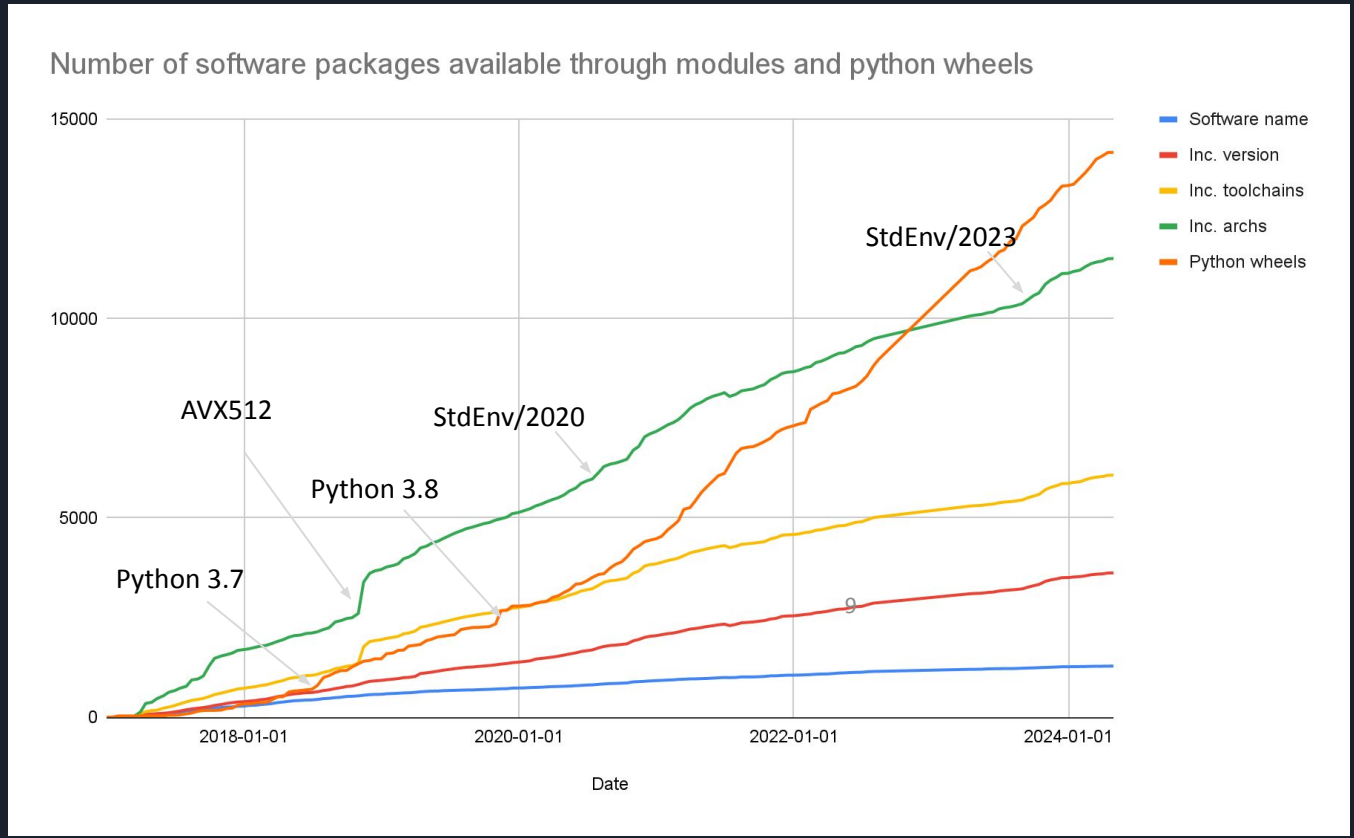
~1300 scientific applications

12000+ permutations of version/CPU/toolchain

Optimized for

- 4 major generations of CPUs (from early 2000s to recent CPUs in 2020)
- 4 major generations of NVidia GPUs
- InfiniBand, OmniPath, Ethernet

14000+ python wheels



HPC in the cloud with Magic Castle

Since the software stack is portable, it can be used in cloud environments easily

We use this for hundreds of training sessions throughout the year

https://github.com/ComputeCanada/magic_castle

Even on lab computers/laptops

- Mounting our software stack
  - https://docs.alliancecan.ca/wiki/Accessing_CVMFS

- Note:
  - It also works on Windows with WSLv2

Quick Introduction of CVMFS

What is CVMFS?

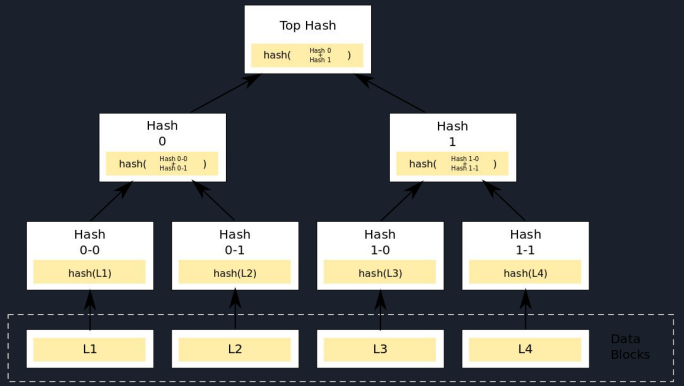
- A filesystem
  - in userspace (FUSE)
  - automatic file chunking, de-duplication, compression
  - on-demand access and caching

- Using HTTP data transport
  - like basic web access
  - read-only by default
  - geographic distribution
  - fault-tolerance, redundant servers

CVMFS Implementation

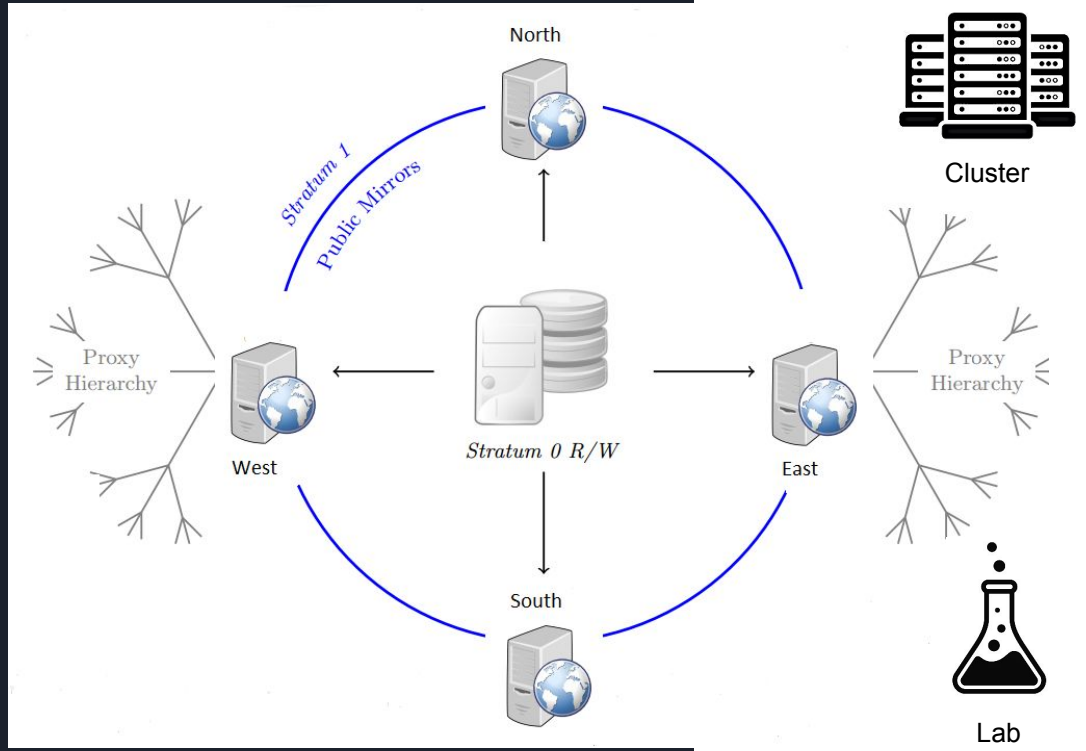
- No stateful protocols or RPCs - just HTTP (REST)
- Data structure: CAS and Merkle tree
- metadata encoded as catalog files
  - metadata queries are resolved locally
- All metadata is data
- All data is immutable and cacheable
- Good for millions of files, random IOPS
- Designed for software
  - But also good for data sets (WORM)

Content Addressable Storage



Merkle Tree

CVMFS Deployment Model



Cluster

Campus

Lab

- hierarchical caching
  - scalable performance
  - minimizes network traffic
  - global distribution

Site Operational Experience and Deployment Considerations

Site Operational Experience

General trends

- CVMFS is generally robust and mature
  - Development informed by ~ 15 years of operation in HPC (HTC) environments
- Much simpler than a cluster filesystem ("trivial")
  - Read-only, no locking, no metadata server
- Offload IOPS, metadata operations from cluster filesystem as much as possible
  - Good way to leverage underused local drives
  - Even with diskless nodes, alien cache on cluster filesystem helps
  - CVMFS is optimized for the use cases that typically challenge cluster filesystems
- Hierarchical caching is very effective
  - Only need ~2-3 caching proxies for ~2000 nodes
    - *If client disk caches are persistent and large enough*
  - Squid, frontier-squid, nginx, ~~ATS~~
- Most CVMFS problems are caused by node-level problems
  - OOM, PID/fd exhaustion, etc.

Site Operational Experience
Things to note

- “Why does CVMFS (sometimes) use so much CPU?”

```
$ top -p `pgrep -d , cvmfs2`
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2923	cvmfs	20	0	25.6g	1.2g	232	S	66.4	0.3	8969:14	cvmfs2

  - Still only 0.8% of total node CPU, on a fully loaded 80-core node
  - Probably locally resolving metadata queries (no metadata server)
  - Often a sign of users doing “interesting” (inefficient) things
  - More visibility because it runs in userspace (FUSE)
  - Cluster filesystems use CPU too, but typically hidden in kernel space
- Beware of “zombie mounts”
  - `cvmfs_config fuser`
- Watch out for cache thrashing
  - Ensure client disk caches are sized appropriately (try ~> 50 GB to start)

Site Deployment Considerations

How to ensure large HPCs can continue running independently even if WAN is down

- Large proxy caches 😬
  - Squid wipes cache on startup
  - Default cache-control header expires content in proxies after 3 days
- Alien cache (shared cache on cluster filesystem)
  - Client-writeable: good enough, easy, cleanable 😀
    - Requirements: nodes with HTTP (proxy) access, a directory `cvmfs` can write to
  - Preloaded: if you need strict guarantee, but trickier to work with 😐
  - Can be used in conjunction with local disk cache (tiered cache)
- Local private stratum 1 server 😀
- Public stratum 1 server at every location 😕
  - Too many stratum servers dilutes proxy caching efficiency
  - Probability $\propto$ 1/N of a given chunk being cached in a proxy
  - But GeoAPI should mitigate this

War Story

https://sft.its.cern.ch/jira/browse/CVM-2001

May 2021: In-place update of glibc caused widespread corruption.

- Started as fairly innocent looking change to enable memusage and memusagestat for memory profiling.
- After pushing $EPREFIX/lib64/libc-2.30.so into cvmfs, processes already started randomly crashing (they all mmap this file). Newly started programs on clean nodes were fine.
- Reason: “cache poisoning”
- sha256sum would give different results every time you ran it.
- Similar things happen when you “cp” a shared library, but not if you unlink it first and then place a new one on a local file system (two different inodes)
- In-place updates work fine if you use symbolic links (files or directories) instead.
- Fixed in CVMFS 2.10.0 (https://github.com/cvmfs/cvmfs/pull/3043)

Security

Security: Confidentiality

- Typically not very applicable to CVMFS
- But what about licensed software?
  - `restricted.computecanada.ca`
- Distributed with combination of network and POSIX ACLs
  - Requires agreeing to a policy
  - LDAP for identity synchronization
  - `CVMFS_CLAIM_OWNERSHIP=no` to preserve file owner/group

Security: Integrity
The chain of trust

- How do we know cryptominers aren't being injected onto the cluster?
- CVMFS data is delivered over HTTP
  - That's a good thing! Required for caching in forward proxies
  - HTTPS only secures the channel. Cryptographic signing secures the content.
- A chain is only as strong as the weakest link

Wikimedia Commons

Security: Integrity
The chain of trust

- Install a ‘release’ RPM which configures a yum repository
  - dnf install https://package.computecanada.ca/[...]/computecanada-release.rpm
    - Importing GPG key 0xCF214CFC:
    - Userid : "Ryan Taylor (Compute Canada CVMFS)"
    - Fingerprint: C0C4 0F04 70A3 6AF2 7CC4 4D5A 3B9F C55A CF21 4CFC
    - From : /etc/pki/rpm-gpg/RPM-GPG-KEY-CC-CVMFS-1
    - Is this ok [y/N]:

Security: Integrity
The chain of trust

- Install a ‘release’ RPM which configures a yum repository
  - dnf install https://package.computecanada.ca/[...]/computecanada-release.rpm
    - Importing GPG key 0xCF214CFC:
    - Userid : "Ryan Taylor (Compute Canada CVMFS)"
    - Fingerprint: C0C4 0F04 70A3 6AF2 7CC4 4D5A 3B9F C55A CF21 4CFC
    - From : /etc/pki/rpm-gpg/RPM-GPG-KEY-CC-CVMFS-1
    - Is this ok [y/N]:

Security: Integrity
The chain of trust

- Wiki documentation attests the expected fingerprints of GPG signing keys
- Install a ‘release’ RPM which configures a yum repository
  - dnf install https://package.computecanada.ca/[...]/computecanada-release.rpm
    - Importing GPG key 0xCF214CFC:
    - Userid : "Ryan Taylor (Compute Canada CVMFS)"
    - Fingerprint: C0C4 0F04 70A3 6AF2 7CC4 4D5A 3B9F C55A CF21 4CFC
    - From : /etc/pki/rpm-gpg/RPM-GPG-KEY-CC-CVMFS-1
    - Is this ok [y/N]:

Security: Integrity
The chain of trust

- Wiki provides change history, notification emails on changes, served via HTTPS
- Wiki documentation attests the expected fingerprints of GPG signing keys
- Install a ‘release’ RPM which configures a yum repository
  - dnf install https://package.computecanada.ca/[...]/computecanada-release.rpm
    - Importing GPG key 0xCF214CFC:
    - Userid : "Ryan Taylor (Compute Canada CVMFS)"
    - Fingerprint: C0C4 0F04 70A3 6AF2 7CC4 4D5A 3B9F C55A CF21 4CFC
    - From : /etc/pki/rpm-gpg/RPM-GPG-KEY-CC-CVMFS-1
    - Is this ok [y/N]:

Security: Integrity
The chain of trust

- Wiki provides change history, notification emails on changes, served via HTTPS
- Wiki documentation attests the expected fingerprints of GPG signing keys
- Install a ‘release’ RPM which configures a yum repository
  - dnf install https://package.computecanada.ca/[...]/computecanada-release.rpm
    - Importing GPG key 0xCF214CFC:
    - Userid : "Ryan Taylor (Compute Canada CVMFS)"
    - Fingerprint: C0C4 0F04 70A3 6AF2 7CC4 4D5A 3B9F C55A CF21 4CFC
    - From : /etc/pki/rpm-gpg/RPM-GPG-KEY-CC-CVMFS-1
    - Is this ok [y/N]:
- Install a ‘config’ RPM from that yum repository

Security: Integrity
The chain of trust

- Wiki provides change history, notification emails on changes, served via HTTPS
- Wiki documentation attests the expected fingerprints of GPG signing keys
- Install a ‘release’ RPM which configures a yum repository
  - dnf install `https://package.computecanada.ca/[...]/computecanada-release.rpm`
    - Importing GPG key 0xCF214CFC:
    - Userid : "Ryan Taylor (Compute Canada CVMFS)"
    - Fingerprint: C0C4 0F04 70A3 6AF2 7CC4 4D5A 3B9F C55A CF21 4CFC
    - From : /etc/pki/rpm-gpg/RPM-GPG-KEY-CC-CVMFS-1
    - Is this ok [y/N]:
- Install a ‘config’ RPM from that yum repository
- The config RPM contains the public key of the CVMFS configuration repository
  - Technically this is a master key that signs a list of fingerprints of other signing keys (in X.509 format) which are allowed to sign the repository manifest, which contains the hash of the Merkle tree root
  - CVMFS Security Considerations

Security: Integrity
The chain of trust

- Wiki provides change history, notification emails on changes, served via HTTPS
- Wiki documentation attests the expected fingerprints of GPG signing keys
- Install a ‘release’ RPM which configures a yum repository
  - dnf install https://package.computecanada.ca/[...]/computecanada-release.rpm
    - Importing GPG key 0xCF214CFC:
    - Userid : "Ryan Taylor (Compute Canada CVMFS)"
    - Fingerprint: C0C4 0F04 70A3 6AF2 7CC4 4D5A 3B9F C55A CF21 4CFC
    - From : /etc/pki/rpm-gpg/RPM-GPG-KEY-CC-CVMFS-1
    - Is this ok [y/N]:
- Install a ‘config’ RPM from that yum repository
- The config RPM contains the public key of the CVMFS configuration repository
  - Technically this is a master key that signs a list of fingerprints of other signing keys (in X.509 format) which are allowed to sign the repository manifest, which contains the hash of the Merkle tree root
  - CVMFS Security Considerations
- The CVMFS configuration repository distributes public keys of all other CVMFS repositories

Security: Integrity
The chain of trust

- Wiki provides change history, notification emails on changes, served via HTTPS
- Wiki documentation attests the expected fingerprints of GPG signing keys
- Install a ‘release’ RPM which configures a yum repository
  - dnf install https://package.computecanada.ca/[...]/computecanada-release.rpm
    - Importing GPG key 0xCF214CFC:
    - Userid : "Ryan Taylor (Compute Canada CVMFS)"
    - Fingerprint: C0C4 0F04 70A3 6AF2 7CC4 4D5A 3B9F C55A CF21 4CFC
    - From : /etc/pki/rpm-gpg/RPM-GPG-KEY-CC-CVMFS-1
    - Is this ok [y/N]:
- Install a ‘config’ RPM from that yum repository
- The config RPM contains the public key of the CVMFS configuration repository
  - Technically this is a master key that signs a list of fingerprints of other signing keys (in X.509 format) which are allowed to sign the repository manifest, which contains the hash of the Merkle tree root
  - CVMFS Security Considerations
- The CVMFS configuration repository distributes public keys of all other CVMFS repositories
- The other CVMFS repositories deliver content to clusters for end users

Security: Availability

Open access and resiliency

- HPC & AI centers are critically important national strategic resources
- The risk of large scale DDoS attacks is increasingly relevant in the current geopolitical/cybersecurity landscape
- But - openness is essential for scientific collaboration and innovation

Large DDoS Attacks

Date	Defender	Scale
Oct 2016	Dyn	1.2 Tb/s
Feb 2017	Github	1.3 Tb/s
Sep 2017	Google	2.5 Tb/s
Feb 2018	Github	1.3 Tb/s
Feb 2020	AWS	2.3 Tb/s
Nov 2021	Microsoft	3.5 Tb/s
Jun 2022	Google	46M RPS
Jun 2022	Cloudflare	26M RPS
Oct 2023	Google	398M RPS

Security: Availability

DDoS mitigation with commercial CDN

- DDoS mitigation is only possible “in the network”
  - Once traffic reaches your server(s), it’s too late
- Anything with a public IP is exposed to potential attack
- On our own we could ~ 2x - 5x capacity, but Cloudflare provides ~1000x
- Using a commercial CDN solves multiple problems
  - Cascading failover (looks similar to DDoS)
  - Simplified access for end users and small sites
    - Don’t need proxies, does not load our stratum servers as much
  - Improved access anywhere in the world
- Recommended: research network peering with CDN provider

Advanced Research Computing in Canada

CVMFS operations at national sites

- 6 major national systems
  - ~ 350K cores, 50 PF, 200 PB disk
- Nearly all research software on CVMFS
  - ~40 staff members can publish software

- CVMFS National Team: 8 members
  - Representatives from each site
  - Total FTE on CVMFS: ~1.5

Location	Site	System	Type
Victoria	UVic	**Arbutus**	Cloud
Vancouver	SFU	**Cedar**	General
Waterloo	UW	**Graham**	General
Toronto	U of T	**Niagara**	Large MPI
Montreal	ETS	**Beluga**	General
Montreal	ETS	**Narval**	General

CVMFS Deployment in Canada
Design

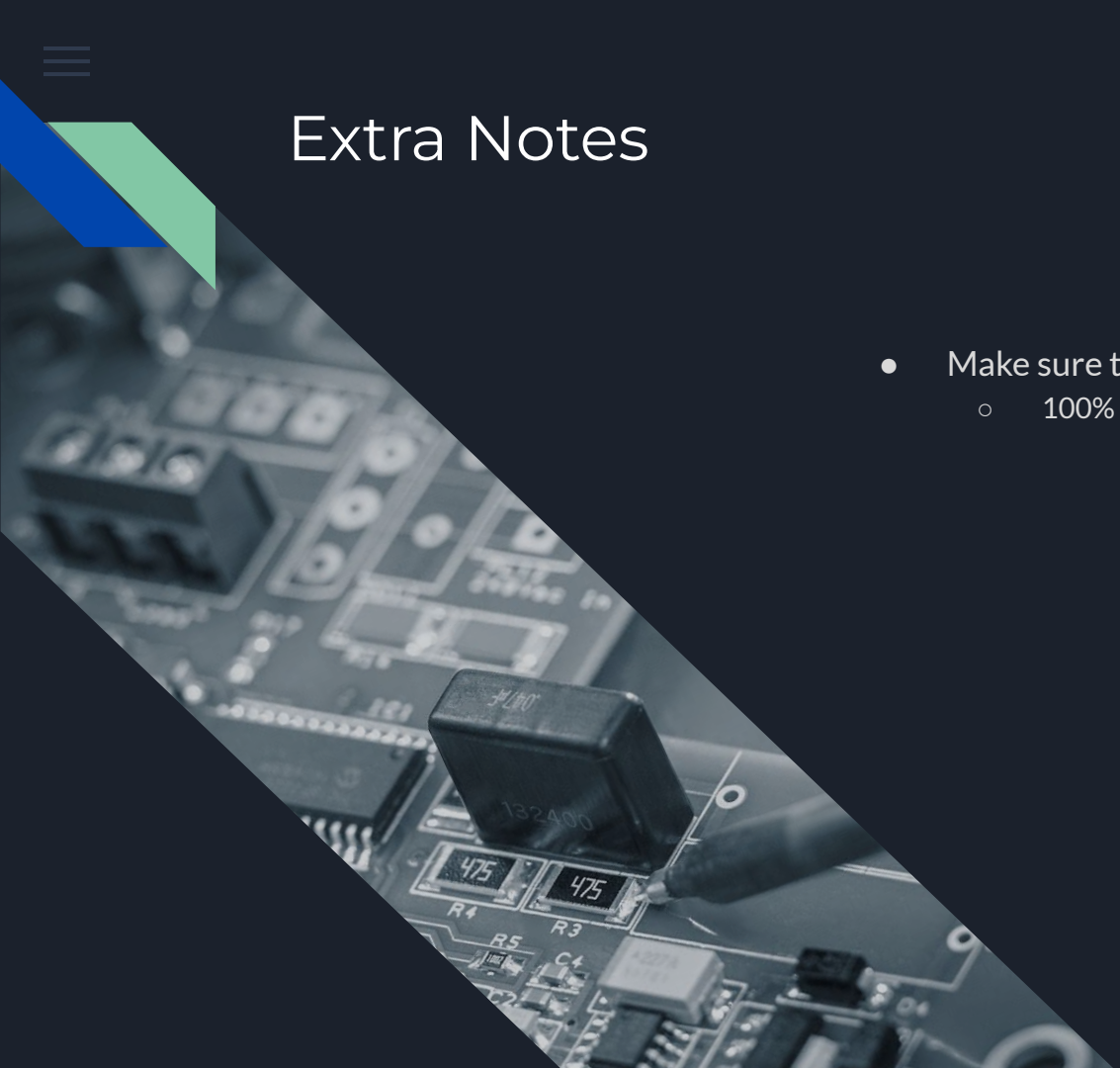
- Most sites have a local stratum 1 server
- Niagara uses alien cache for local replica
  - Diskless nodes

Location	Site	System	Type	Stratum
Victoria	UVic	**Arbutus**	Cloud	0, 1
Vancouver	SFU	**Cedar**	General	1
Waterloo	UW	**Graham**	General	1
Toronto	U of T	**Niagara**	Large MPI	
Montreal	ETS	**Beluga**	General	1
Montreal	ETS	**Narval**	General	

- Full dev/prod testing
  - Dev software repo
    - All software is published from dev to prod
  - Dev stratum 0 and 1 servers
  - Dev config repo to distribute access details to clients
  - Dev config RPMs to configure clients to use dev config repo
- Accessing CVMFS documentation

Discussion

Extra Notes



- Make sure to exempt CVMFS traffic from IPS scanning
  - 100% liability, 0% benefit

HTTP History

CERN