Digital Research Alliance of Canada site talk

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

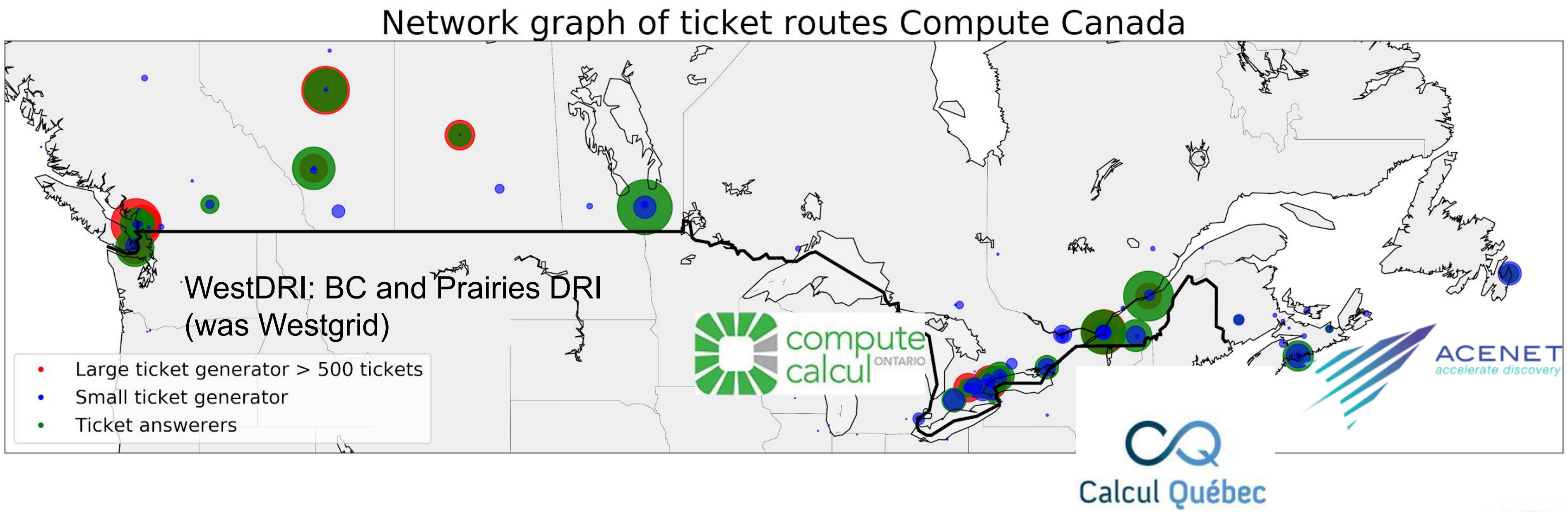
Bart Oldeman
McGill University, Calcul Québec, Digital Research Alliance of Canada
Research Support National Team Software Installation Coordinator
(with Maxime Boissonneault, Charles Coulombe, Doug Roberts (RSNT), Ryan Taylor (CVMFS))

McGill Calcul Québec

Partenaire régional de l'
Alliance de recherche numérique du Canada

A regional partner of the
Digital Research Alliance of Canada

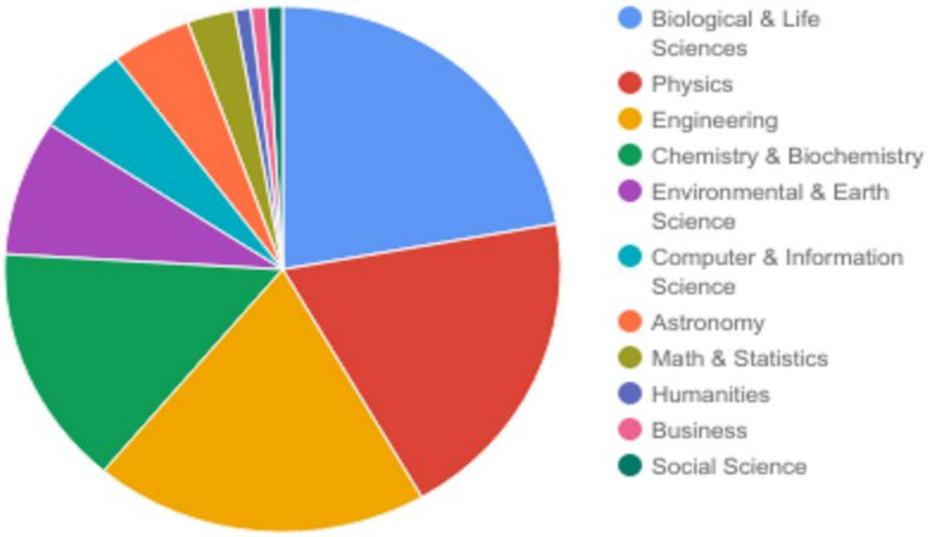
The people


Network graph of ticket routes Compute Canada
WestDRI: BC and Prairies DRI
(was Westgrid)
● Large ticket generator > 500 tickets
● Small ticket generator
● Ticket answerers
compute calcul ONTARIO
ACENET accelerate discovery
Calcul Québec

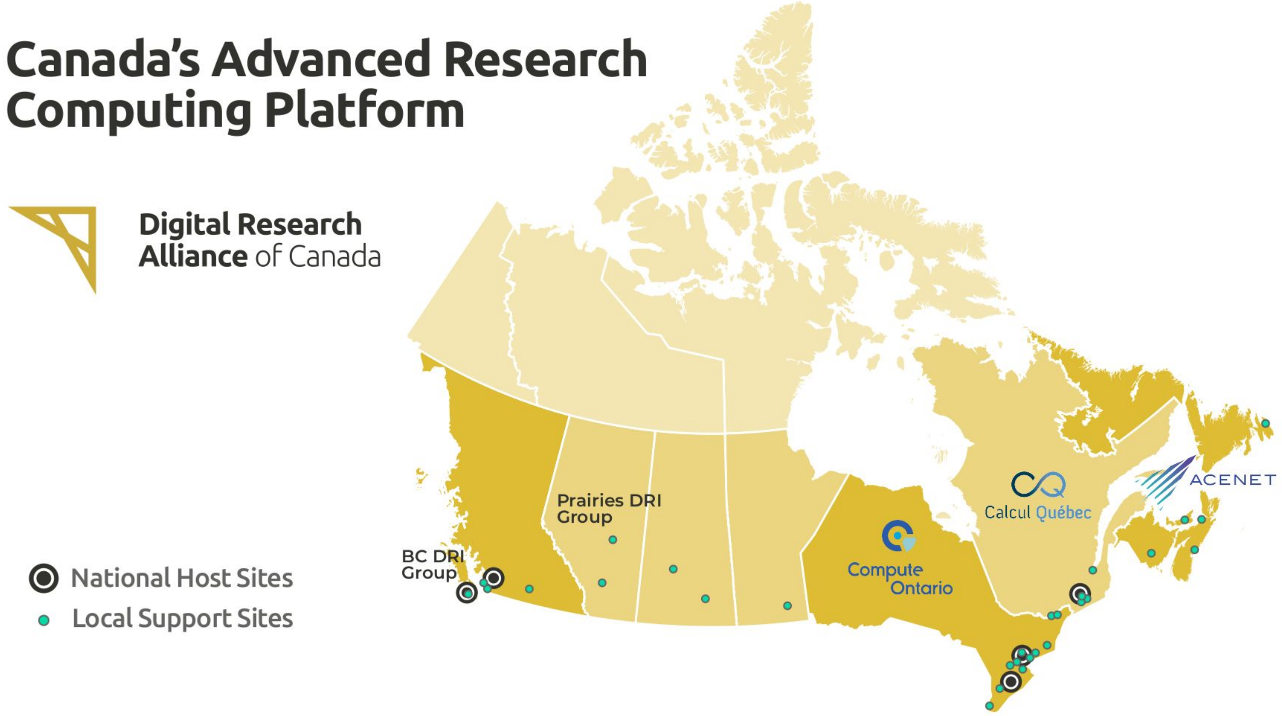
All research disciplines supported

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



6 major national systems
300K->815K cores,
90->165 PB storage

System	Type	Network	Production
Arbutus (renew)	Cloud	Ethernet	2025 H1
Cedar->Fir	General	HDR/NDR	2025 H1
Graham->Nibi	General	200 GbE	2025 H1
Niagara->Trillium	Large MPI	NDR IB	2025 H1
Béluga->Rorqual	General	HDR IB	2025 H1
Narval	General	HDR IB	2021 H2

Starting in 2017, new bigger national systems replaced many smaller local clusters, now common software stack, scheduler (Slurm), and so on, administered by national teams. Sites without physical cluster still support.

Background

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

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 41

Linux kernel 6.11.4, GCC 14.2.1, Glibc 2.40, Python 3.13.0

Background

But users on those clusters want shiny new things and install them as if it were a local Linux computer (following documentation):

```
$ sudo apt-get install python3.9-dev
We trust you have received the usual lecture from the local
System Administrator. It usually boils down to these three
things:
    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.
[sudo] password for jsmith:
sudo: apt-get: command not found
$ sudo yum install python39-devel
[sudo] password for jsmith:
Sorry, try again.
[sudo] password for jsmith:
Sorry, user jsmith is not allowed to execute ...
```

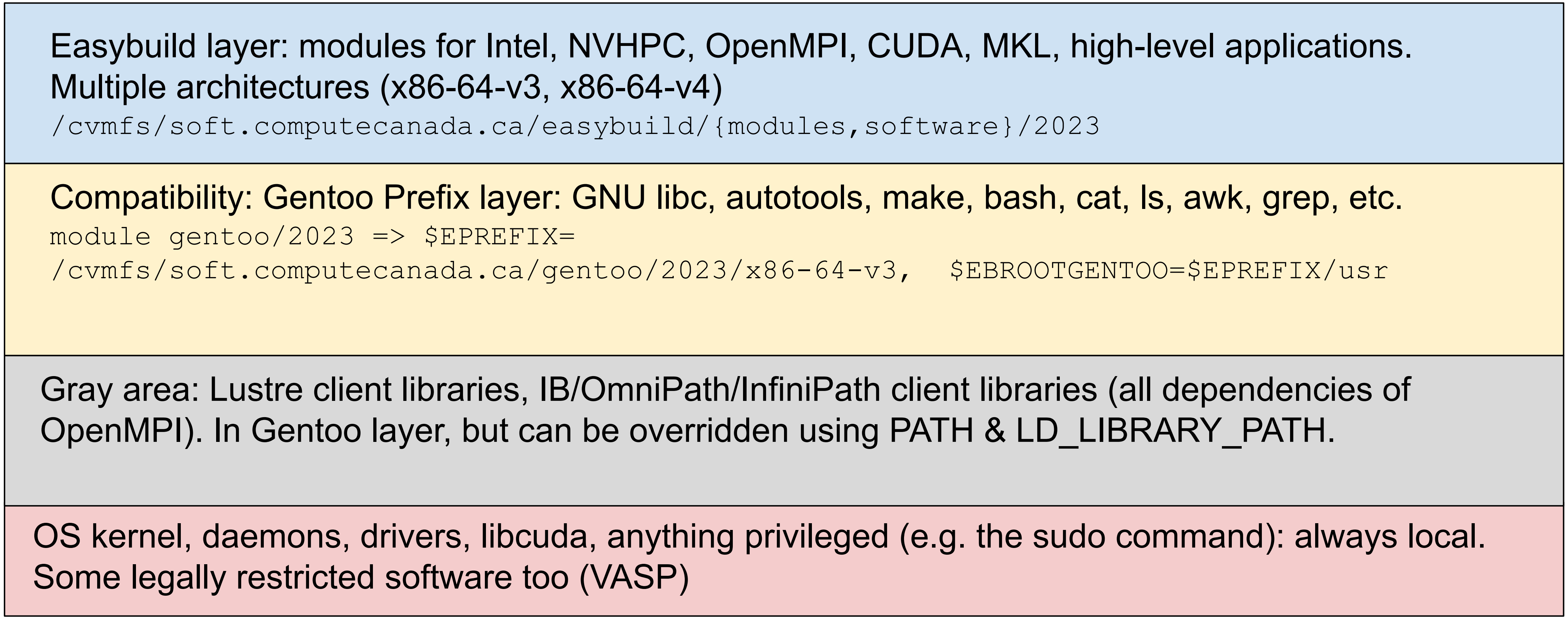
Solution: modules

Create a “modulefile” named “python/2.7.9” somewhere in $MODULEPATH

```
#%Module1.0###########################
proc ModulesHelp { } {
       puts stderr "\tAdds Python 2.7.9 to your environment"
}
module-whatis   "Adds Python 2.7 to your environment"
set  root /software/CentOS-6/tools/python-2.7.9
prepend-path MANPATH         $root/share/man
prepend-path PATH            $root/bin
prepend-path LD_LIBRARY_PATH $root/lib
prepend-path CPATH           $root/include
```

Users do “module load python/2.7.9”, which modifies their environment. “module unload python” restores it then.

Software: design overview



Easybuild layer: modules for Intel, NVHPC, OpenMPI, CUDA, MKL, high-level applications. Multiple architectures (x86-64-v3, x86-64-v4)

```
/cvmfs/soft.computecanada.ca/easybuild/{modules,software}/2023
```

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

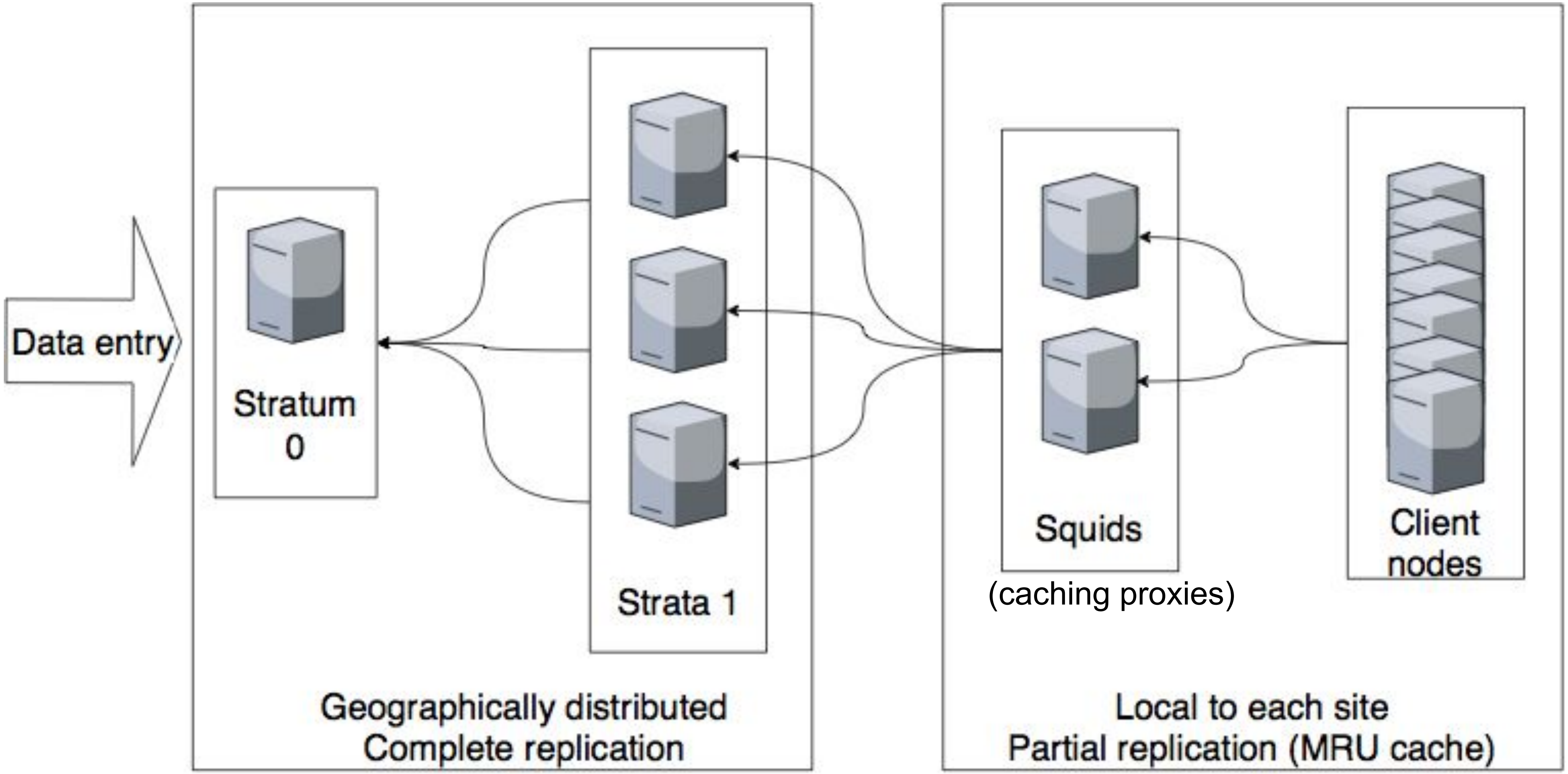
```
module gentoo/2023 => $EPREFIX=
/cvmfs/soft.computecanada.ca/gentoo/2023/x86-64-v3,  $EBROOTGENTOO=$EPREFIX/usr
```

Gray area: Lustre client libraries, IB/OmniPath/InfiniPath client libraries (all dependencies of OpenMPI). In Gentoo layer, but can be overridden using PATH & LD_LIBRARY_PATH.

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

CVMFS content delivery

CernVM File system



(caching proxies)

Compatibility layer: Gentoo Prefix

- Package, dependency & environment management system
- Builds using bash-like “ebuilds” (most upstream, small custom overlay).
- Used to provide dependencies for scientific applications, themselves of little scientific interest
  - Glibc, coreutils, awk, grep, Bash, Bison, Flex, GNU Make, ncurses, readline, libxml2, zlib, bzip2, XZ, Autotools, binutils, OpenSSL, libpng, Emacs, vim, X11, texlive, etc., etc.
  - Newer versions of those than found in enterprise distributions, e.g. Bash 5.1, Git 2.41.0, Vim 9.0, Emacs 26.2
- Abstraction layer between the OS and the scientific software stack, using gentoo/2023 module
- Carries all* the dependencies of scientific software stack

* Exceptions: drivers, kernel modules, etc.

Tools used : EasyBuild

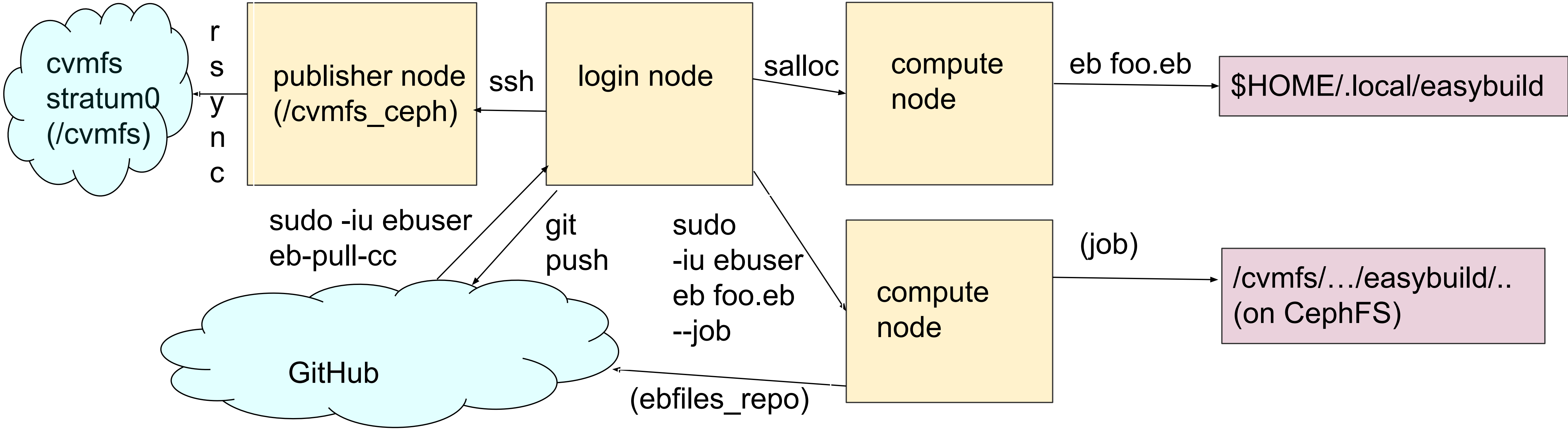
- Automates installation of (mostly) scientifically oriented software and generation of modulefiles.

Tools used : Lmod

- Lua based module system
- Makes it easy to setup a software module hierarchy
  - e.g. modules that depend on MPI implementation X are only visible if you first “module load X”.
- https://lmod.readthedocs.io/en/latest/

Archimedes, our build cluster

- Magic Castle cluster-in-the-cloud that lives on Arbutus
- login, management, test, Jupyter nodes, a publisher node, compute nodes via Slurm and auto-scaling; sharing /home and /cvmfs (rw-copy) on CephFS
- All managed via puppet, easy to bring back up.



Installing software, step by step

1. Figure out if it should be in Gentoo or EasyBuild
    - Is the software performance critical or depends on MPI?
        - Yes => EasyBuild
        - Multiple versions needed via modules ?
            - Yes => EasyBuild
            - No => Gentoo
    - Is it a Python package not directly connected to a module?
        - Build a wheel and add it to the wheelhouse
2. Install on Archimedes with the appropriate package manager, Portage (emerge) or eb: plain `eb` installs in home dir, then with `sudo -iu ebuser`
3. Test on Archimedes
4. Deploy on CVMFS dev repository
5. Test on cvmfs development client node or with `proot`
6. Deploy on CVMFS production repository
7. Final testing on the production cluster

“Compute Canada” Software Stack

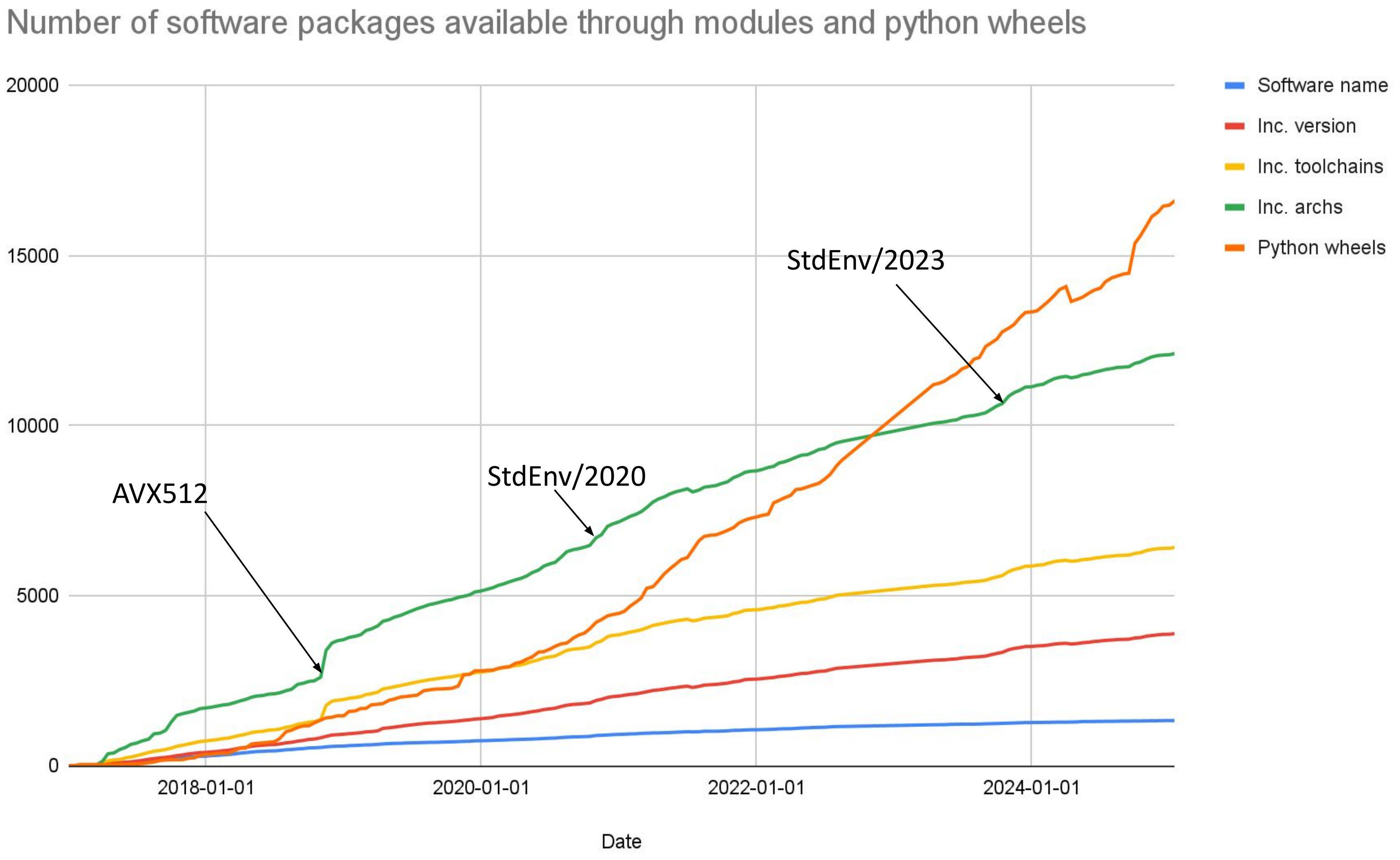
~1000 scientific applications

10000+ 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

17000+ python wheels



Current environment: StdEnv/2023

- Gentoo Prefix from August 2023
- Uses ~foss/2023a toolchain and subtoolchains for most software
  - GCCcore 12.3.0, Open MPI 4.1.5
  - 2024a and Intel/NVHPC-based toolchains also available for building your own software
  - FlexiBLAS uses MKL backend on Intel CPUs, (upstream) BLIS on AMD
- Made the default in April 2024
- Older stacks still available using
  module load StdEnv/2016.4, 2018.3, 2020
  (oldest stacks with security disclaimer)

Software challenges caused by custom prefix

Using `$EPREFIX/usr` instead of `/usr`

- `Uses Gentoo Prefix` custom loader (ELF interpreter in `$EPREFIX/lib/ld-linux-x86-64.so.2`)
- Custom `setrpaths.sh` script patches (using `patchelf --set-interpreter`) downloaded binaries so they can work with this prefix
- Some users set `LD_LIBRARY_PATH` to `/usr/lib64`, mostly by accident in old `.bashrc` files, which breaks most tools
- Some commercial packages use wrapper scripts which needed to patched
- Anaconda & Julia
  - provide binaries that are not always compatible
  - we provide pip-installable Python wheels and actively discourage Anaconda
    - Anaconda users can even end up with custom, non-optimal, installations of Open MPI or R.
- If all else fails, use `module --force purge`, or Singularity/Apptainer; we also provide a container repository for some of those situations.

Challenge: host OS/compatibility layer boundary

- Various host OS daemons write to files under /var such as /var/run/utmp, but compatibility layer utilities (`last`, `w`, `who`, etc.) read from $EPREFIX/var.
- But (for example) Gentoo Prefix’ `$EPREFIX/usr/include/paths.h` sets `_PATH_UTMP` to `$EPREFIX/var/run/utmp`. who then reads from that file. Two solutions:
  - strategic symlinks, e.g. `$EPREFIX/var/run -> /var/run` (old)
  - change `paths.h` (current)
- Rest works remarkably well:
  - `libnss_ldap/sss` libraries can be compiled from source, also read from locations under `/var`
  - Whole Mate and XFCE desktops can be compiled and work with VNC (including via JupyterHub, and with VirtualGL)
  - Even some selinux support (for filesystem labels).

Other peculiarities

- We use central libstdc++ (+libgfortran etc) at runtime from compat layer, presently:

`/cvmfs/soft.computecanada.ca/gentoo/2023/x86-64-v3/usr/lib/gcc/x86_64-pc-linux-gnu/13/libstdc++.so.6`

  - Advantages
    - we can collapse GCCcore EasyBuilds for different versions to “SYSTEM”
      - `foo-1.0-GCCcore-13.3.0.eb` -> `foo-1.0.eb` in `ebfiles_repo` via hook.
      - Large repository of compatible modules at Core level.
    - Python wheels compiled with g++13 compatible with compiled-with-g++12
  - Disadvantages
    - A little messy, not 100% compatible with upstream

Opportunities, collaboration with EESSI

- Public part of the stack is available *everywhere*
    - https://docs.alliancecan.ca/wiki/Accessing_CVMFS
    - `source /cvmfs/soft.computecanada.ca/config/profile/bash.sh`
    - The container repository for e.g. QIIME2 and a data repository for multi-GB datasets (e.g. InterProScan_data) are also public.
    - Proprietary packages, e.g. Intel compilers, MATLAB, are in a restricted repository.
- Gentoo Prefix Ansible-based bootstrap based on EESSI’s bootstrap
- `%(installdir)s` on CephFS can be a bottleneck for certain software installations; an EESSI inspired method via local storage, overlayfs, and Apptainer or bubblewrap could speed this up