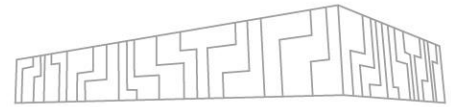
VSB TECHNICAL UNIVERSITY OF OSTRAVA | IT4INNOVATIONS NATIONAL SUPERCOMPUTING CENTER

EASYBUILD MODULES

Milan Jaros, Ondrej Vysocky, Lukas Krupcik

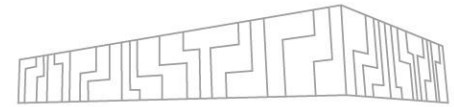
EASYBUILD & MODULES



Scientific Software Installation

- User request / critical updates / new software
- All software on shared storage
- Long-term support of installed software
- Optimized for HPC cluster
  - Built from source codes
  - Linked with optimized libraries
  - Build Optimization
  - Host instruction set

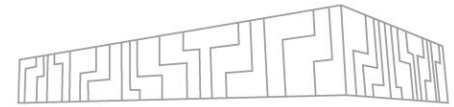
EASYBUILD & MODULES



Common Issues with Scientific Software

- Incorrect use of build tools (make, cmake…)
- Non-standard installation scripts
- Incomplete build procedure (no configure)
- Interactive installation scripts
- Hardcoded paths and parameters
- Dependent on specific version of library
- …

EASYBUILD & MODULES



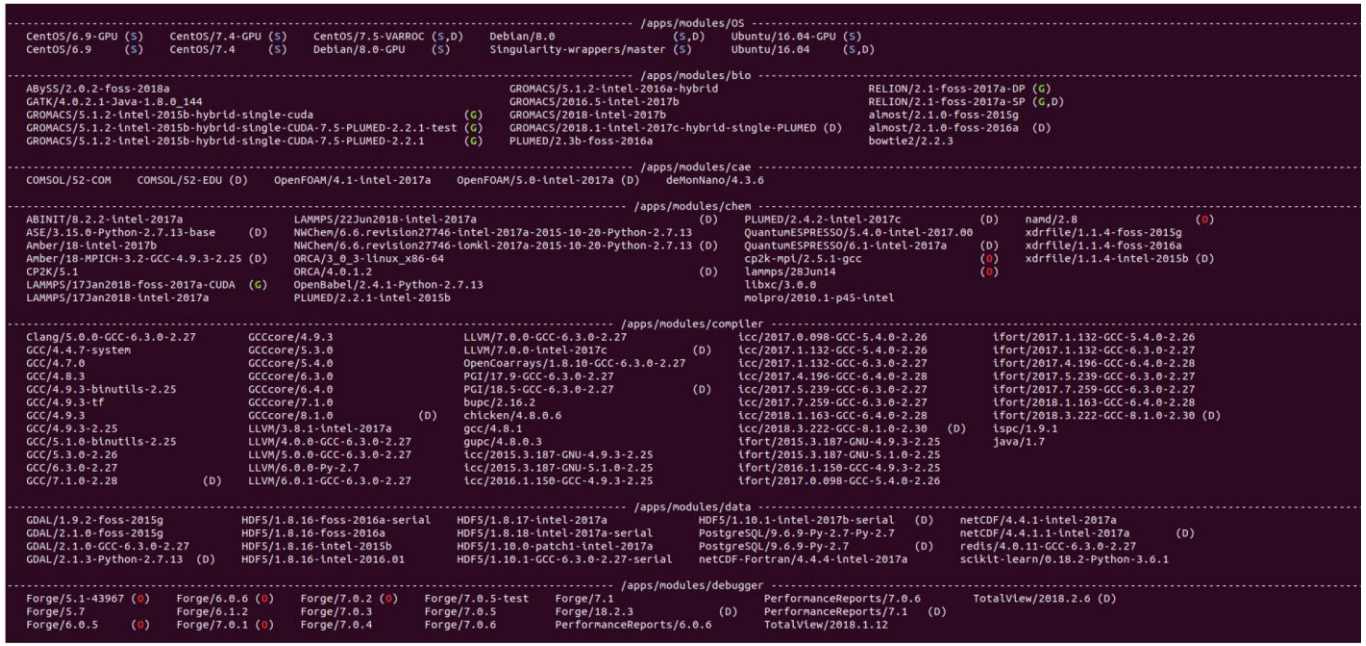
Compilation Challenges

- cmake, cmake --build
- cmake, make, make install
- make, make install
- make
- configure, make
- configure, make, make install
- autogen, configure, make, make install
- install.sh
- setup.py
- and many other compilation procedures...

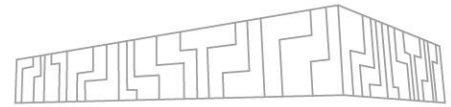
EASYBUILD & MODULES

Environment Modules

- Canonical way to access installed software
- Module files specify user environment settings
- LMOD 8.7.37 is installed on Karolina



- https://docs.it4i.cz/en/docs/software/modules/lmod



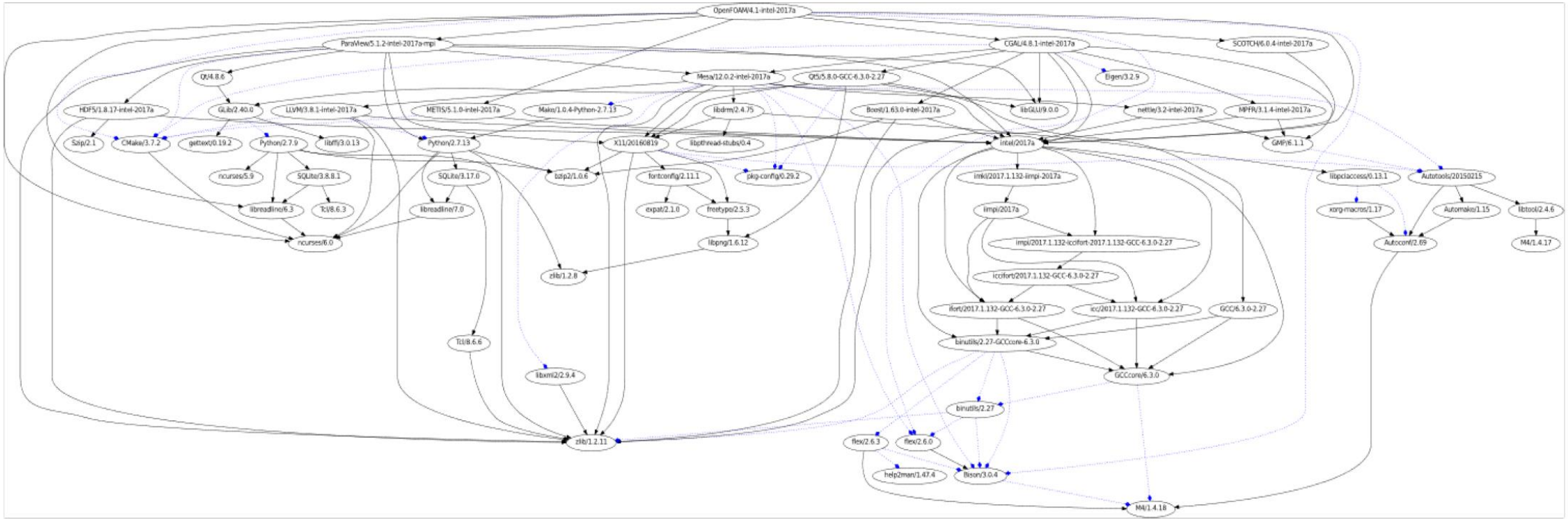
Challenges in Dependency Management

- OpenFOAM/4.1

EASYBUILD & MODULES

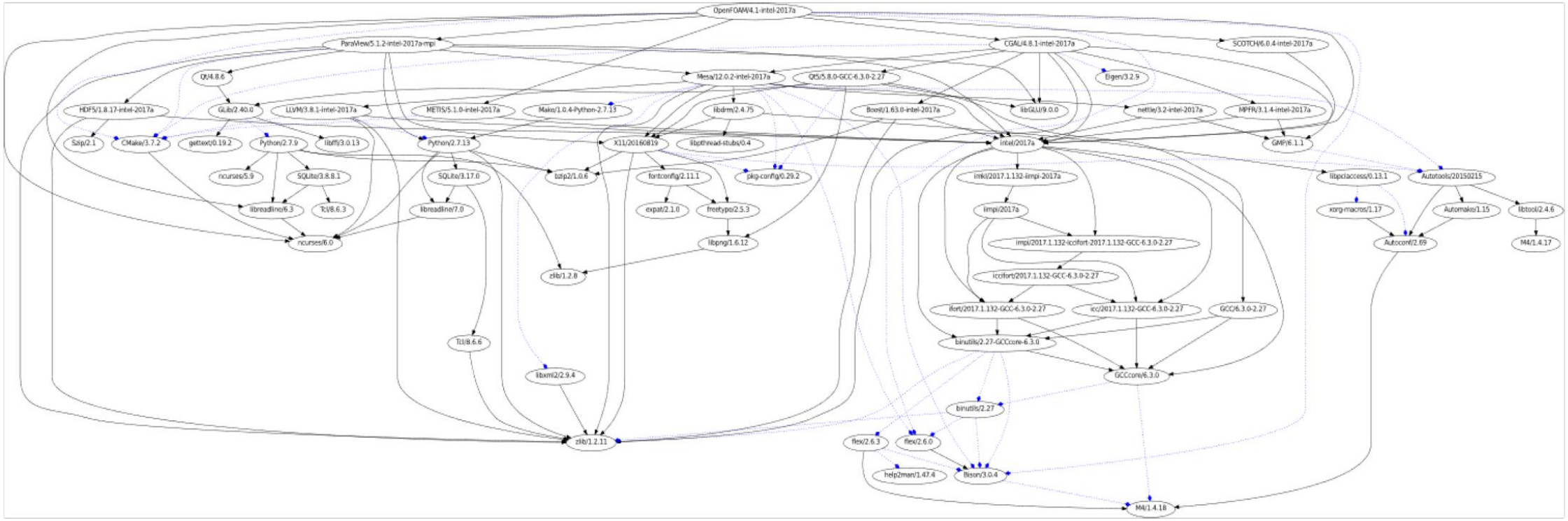
Challenges in Dependency Management

- OpenFOAM/4.1 - 67 dependencies



Challenges in Dependency Management

- OpenFOAM/4.1 - 67 dependencies

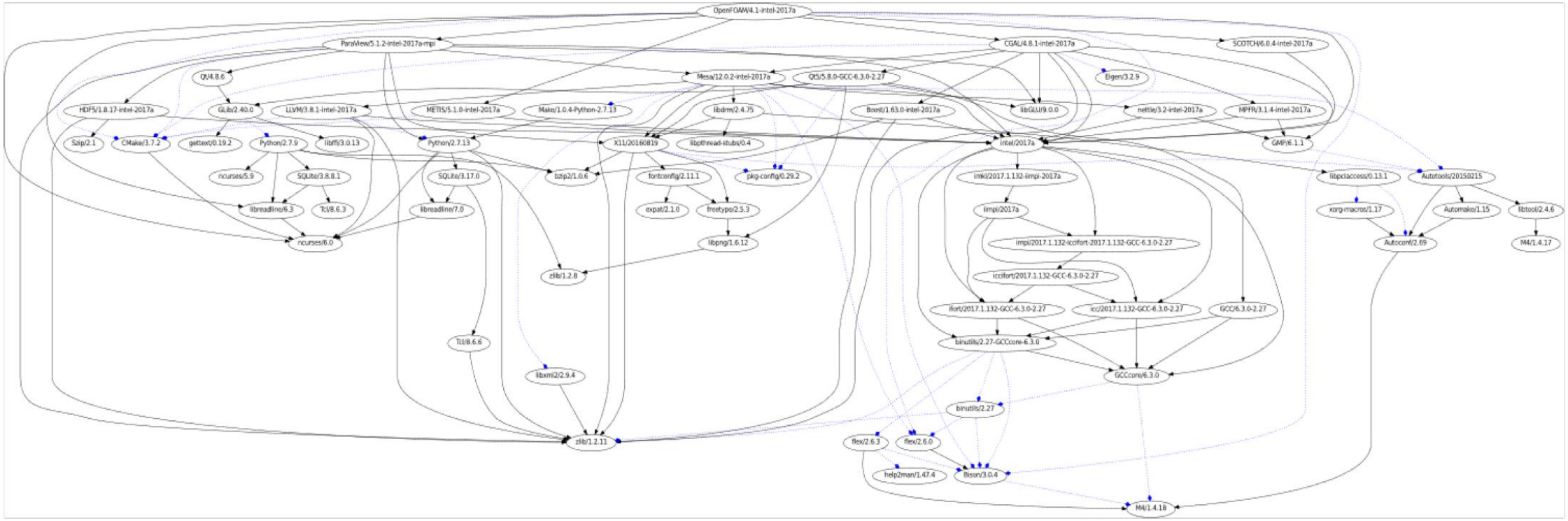


- Manual installation?

EASYBUILD & MODULES

Challenges in Dependency Management

- OpenFOAM/4.1 - 67 dependencies



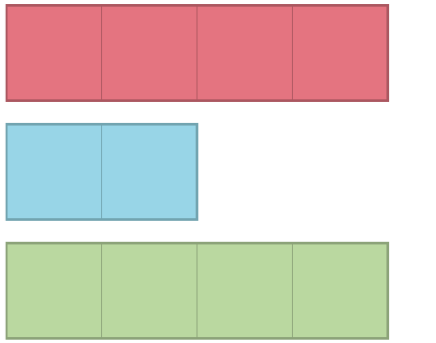
- Only one command…
- **eb OpenFOAM-4.1-intel-2017a.eb -r**

EASYBUILD & MODULES

EasyBuild – Building Software With Ease

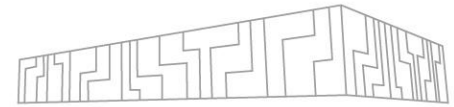
- Framework for installing software
- Collection of Python packages and modules
- In development from 2009 at The Ghent University
- Public from 2012

- Massive community
- https://docs.easybuild.io/



EASYBUILD.io
building software with ease

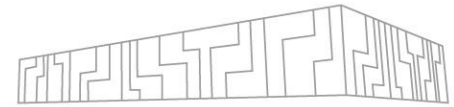
EASYBUILD & MODULES



Features

- Fully autonomous build and installation of SW
  - Automatic dependency resolution
  - Automatic module file generation
- Logging of build/install procedure
- Archiving of build specification
- Extendable by user with additional easyblocks, toolchain, easyconfigs…
- Actively developed
- Each release comprehensively tested
- Worldwide community

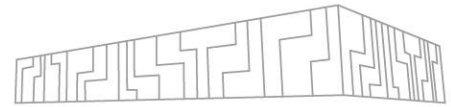
EASYBUILD & MODULES



Terminology

- EasyBuild framework
  - Core of EasyBuild, python modules and packages
  - Unpacking tarballs, applying patches…
- Easyblock
  - Implements build/install procedure
  - Python module, “plugin” for framework
- Easyconfig
  - Build specification (software name, compiler…)
- Compiler toolchain
  - Compiler with accompanying libraries

EASYBUILD & MODULES

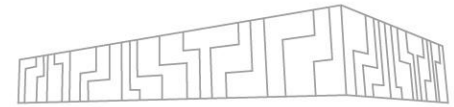


Installation Procedure

1. Parse easyconfig – process config, check dependencies
2. Fetch sources – download or search files
3. Check readiness – verify files, create build folder
4. Unpack sources – unpack tar/zip/rpm…
5. Apply patches – if any
6. Prepare – setup environment, dependencies
7. Configure – environment, libraries, optimization
8. Build – compile software
9. Test – check binaries/libraries
10. Install – to /apps/all/name/version…
11. Extensions – e.g. Python
12. Sanity checks – check folders/files/commands
13. Cleanup – remove temporary files
14. Environment module – create module file
15. Permissions – adjust access
16. Packaging – generate packages (FPM → RPM)
17. Test cases – run only for specific software

```
== processing EasyBuild easyconfig /apps/easybuild/easyconfigs-it4i/g/git/git-2.19.0.eb
== building and installing git/2.19.0...
== fetching files...
== creating build dir, resetting environment...
== unpacking...
== patching...
== preparing...
== configuring...
== building...
== testing...
== installing...
== taking care of extensions...
== postprocessing...
== sanity checking...
== cleaning up...
== creating module...
== permissions...
== packaging...
== COMPLETED: Installation ended successfully
```

EASYBUILD & MODULES

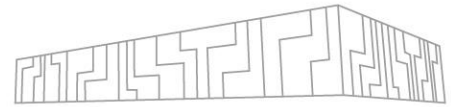


Commands:

```
$ eb git-2.19.0.eb
$ eb ~/easyconfigs/git-2.19.0.eb
$ eb --software=git,2.19.0 --toolchain=GCC,14.3.0
```

Options:

```
--robot/-r              #dependency
--debug/-d              #debug
--search/-S             #search
--dry-run/-D            #dry run
--extended-dry-run/-x   #extended dry run
```



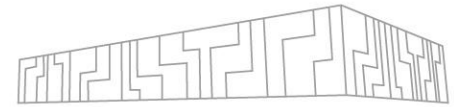
Easyconfig Structure

```
easyblock = 'ConfigureMake'
name = 'git'
version = '2.19.0'
homepage = 'http://git-scm.com/'
description = """Git is a free and open source distributed version control system..."""
toolchain = {'name': 'dummy', 'version': ''}
sources = ['v%(version)s.tar.gz']
source_urls = ['https://github.com/git/git/archive‘]
```

- Other parameters: e.g. preconfigopts, sanity_check_paths, moduleclass

- **Use:** ml git/2.19.0

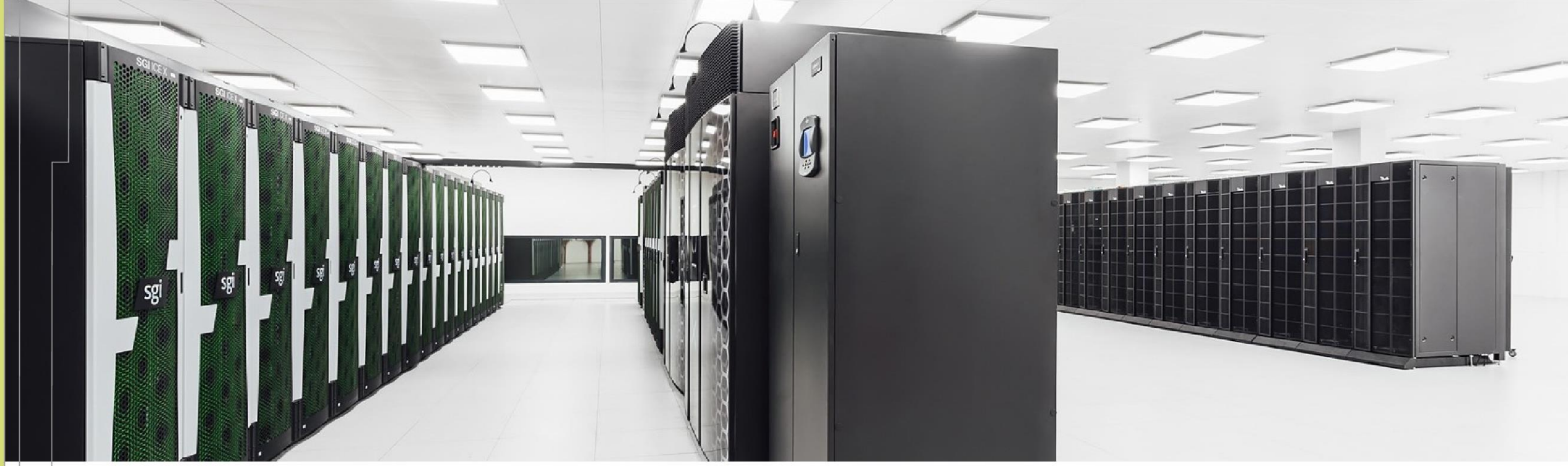
MODULES



"The Environment Modules package provides for the dynamic modification of a user's environment via module files."

- Preinstalled software ready to use for you

- **`ml av module_name`** – gives list of available modules
- **`ml module_name`** – loads the module
- **`ml`** – list of loaded modules
- **`ml unload module_name`** – remove of loaded module
- **`ml purge`** – removes all loaded modules



Milan Jaros
milan.jaros@vsb.cz

IT4Innovations National Supercomputing Center
VSB – Technical University of Ostrava
Studentská 6231/1B
708 00 Ostrava-Poruba, Czech Republic
www.it4i.cz

VSB TECHNICAL UNIVERSITY OF OSTRAVA | IT4INNOVATIONS NATIONAL SUPERCOMPUTING CENTER