11th EasyBuild User meeting

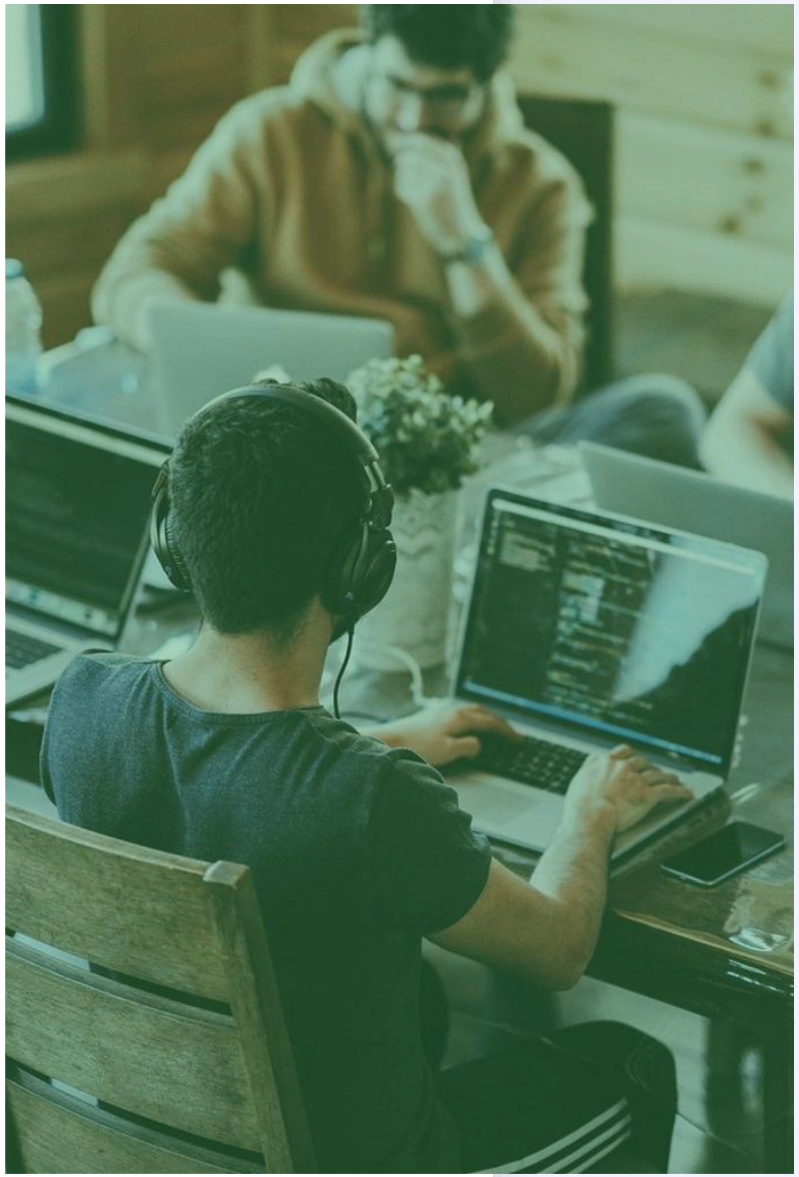
Site presentation – BASF

By Thomas Eylenbosch – Gluo

1

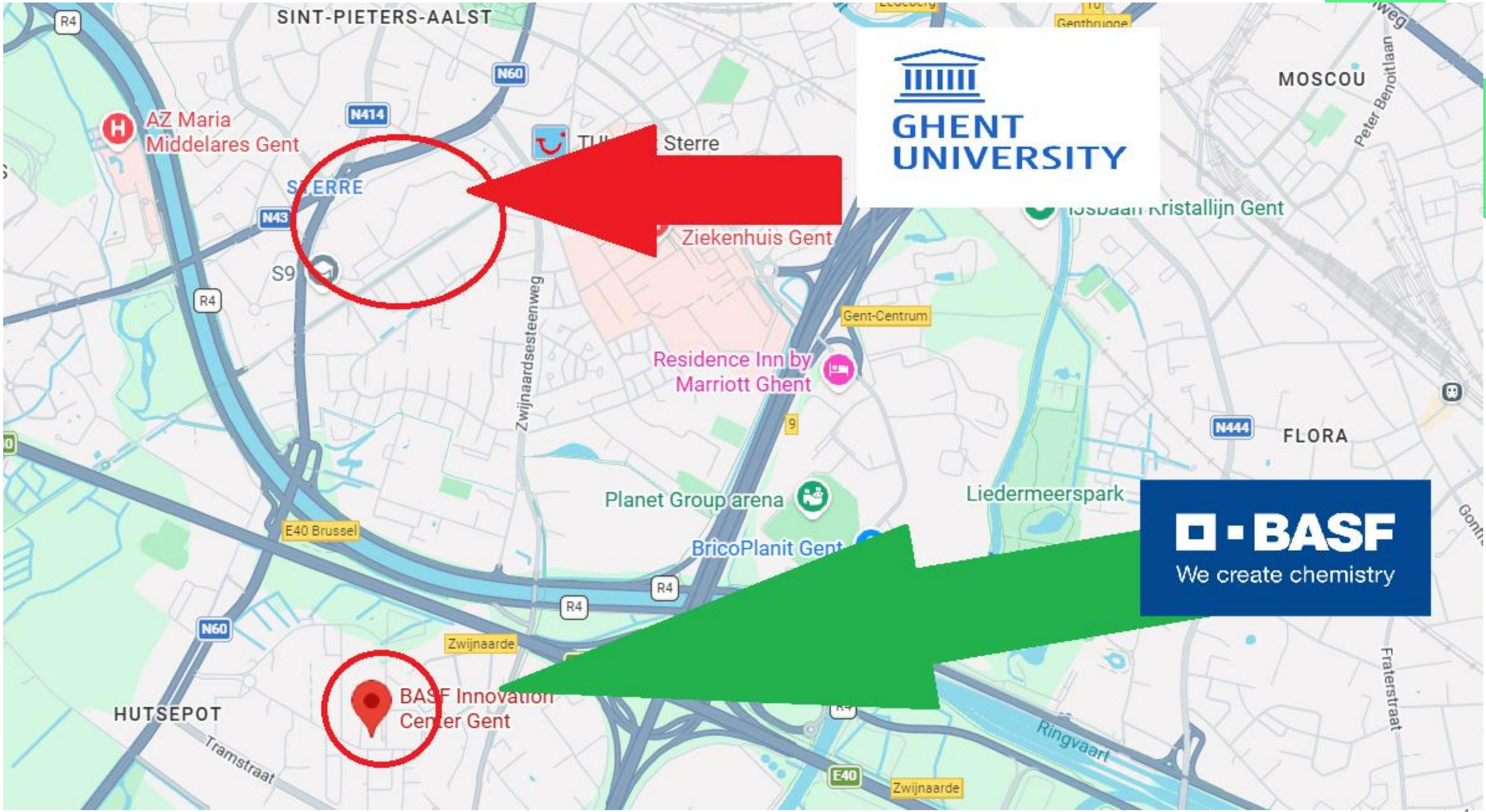
Introduction

Gluo & BASF



Who am I?

- Thomas Eylenbosch
  - Github - @eylenth
- Gluo (Belgium)
  - Consultant @ BASF in Ghent since 2017
- Degree: Applied Computer science
  - University of Applied Sciences and Arts Ghent (HOGENT) ‘2017


SINT-PIETERS-AALST
R4
N60
N414
AZ Maria Middelares Gent
Sterre
N43
S9
Ziekenhuis Gent
Gent-Centrum
Residence Inn by Marriott Ghent
Zwijnaardsesteenweg
Planet Group arena
BricoPlanit Gent
Liedermeerspark
MOSCOU
Peter Benoitlaan
FLORA
N444
GHENT UNIVERSITY
BASF
We create chemistry
E40 Brussel
Zwijnaarde
BASF Innovation Center Gent
HUTSEPOT
Tramstraat
Ringvaart
E40
Fraterstraat

Evolution HPC clusters – BASF AP

...-2017

- Install all modules manually
- RHEL6
- TCL modules

2017-2023

- Install all modules with Easybuild
- RHEL7
- Lmod modules
- Rebuild all modules every 2 years in recent generation

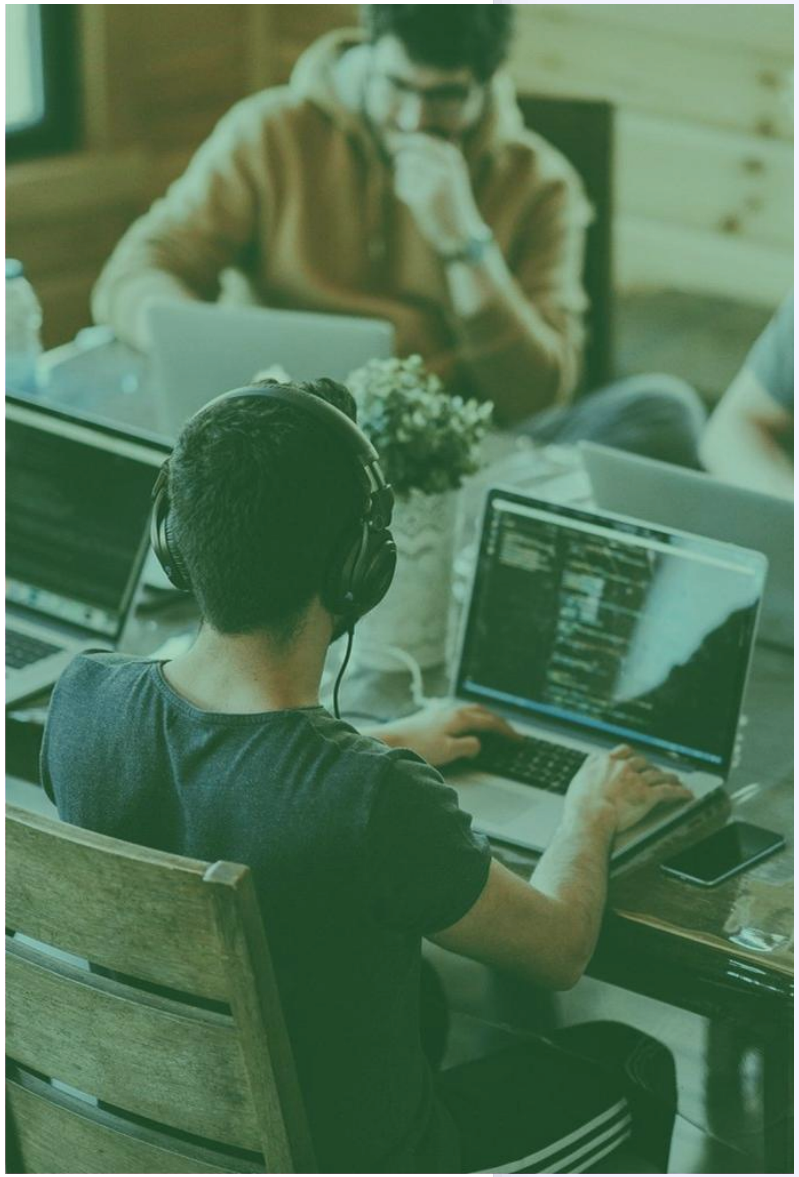
2023-... (Current situation)

- Install all modules with Easybuild on CVMFS
- RHEL9
- Lmod modules
- Rebuild all modules every 2-3 years in recent generation

2

EasyBuild @ BASF

How we are using EasyBuild at BASF AP



Custom Module naming scheme

- GenerationModuleNaming Scheme
  - Foss focused
  - Intel & foss should be possible
- PR open in easybuild framework since 2021: #3547
- An implementation of different generation modules using release dates
- (EasyBuildMNS is also available at our platform)

```
[eylenbt@by3act] ~ $ ml modules/2025b

The following have been reloaded with a version change:
  1) modules/2022b => modules/2025b

[eylenbt@by3act] ~ $ module avail

---- /cvmfs/biogrid-software.rnd.seeds.basf.net/tools/eb/x86_64/generic/gmns/modu
les/all/releases/2025b ----
   ABySS/2.3.10                          RSeQC/5.0.4
   AGAT/1.6.1                            Racon/1.5.0
   ART/2016.06.05                        RagTag/2.1.0
   AUGUSTUS/3.5.0                        RepeatMasker-open/4.0.7
   BBMap/39.80                           RepeatMasker/4.2.1
   BCFtools/1.22                         Ruby/3.4.7                    (D)
   BEDTools/2.31.1                       SAMtools/1.22.1
   BLAST+/2.17.0                         SPAdes/4.2.0
   BUSCO/6.0.0                           SQLAlchemy/2.0.41
   BWA/0.7.19                            STAR/2.7.11b
   BamTools/2.5.3                        SVIM/2.0.0
   Boost/1.88.0                          Salmon/1.11.4
   Bowtie/1.3.1                          Sniffles/2.7.1
   Bowtie2/2.5.4                         StringTie/3.0.3
   CCMpred/0.3.3                         SyRI/1.7.1
   CD-HIT/4.8.1                          TWILIGHT/0.2.3
   CNVpytor/1.3.2                        TransDecoder/6.0.0
   CRISPResso2/2.3.3                     VCFtools/0.1.17
   Circos/0.69-9                         Vegetables_R/dev-R-4.5.2
   Clustal-Omega/1.2.4                   Vegetables_R/prod-R-4.5.2     (D)
   Cluster-Buster/20240927               Velvet/1.2.10-mt-kmer_191
   DBD-Pg/3.18.0                         awscli/2.32.16
   DIA-NN/2.3.2-Enterprise       (D)     bcbio/1.2.9
   DRAGMAP/1.3.0                         biofacet/14.1.D027
   EMBOSS/6.6.0                          biogrid_R/prod-R-4.5.2
   EVM/2.1.0                     (D)     biogrid_python/3.13.5
```

Example:
Modules from GCCcore-12.2.0, GCC-12.2.0 and foss-2022b, … will be installed in one specific modulepath: releases/2022b

```
$ ml modules/2022b
```

Modules from GCCcore-14.3.0, GCC-14.3.0, foss-2025b, gompi-2025b, … will be installed in another modulepath: releases/2025b

```
$ ml modules/2025b
```

Custom hooks #1

- Helps our team avoid forgetting to include custom configurations in default easyconfig recipes.
- To inject customs cacerts files, licenses, ...

```
        eb.cfg.update('postinstallcmds', x)
if eb.name == 'Java' and eb.cfg['easyblock'] != 'ModuleRC':
    # Determine keystore path based on Java version
    is_java8 = re.match(r"(1\.8\.0_.*|8\.\d+)", eb.version)
    is_java11plus = re.match(r"^(1[1-9]|[2-9]\d+)(\.\d+)*$", eb.version)

    if is_java8:
        keystore_path = '%(installdir)s/jre/lib/security/cacerts'
    elif is_java11plus:
        keystore_path = '%(installdir)s/lib/security/cacerts'
    else:
        keystore_path = None

    if keystore_path:
        cacerts = [
           f'%(installdir)s/bin/keytool -import -alias [redacted] /etc/pki/CA/certs [redacted] keystore_path} -storepass changeit [redacted]
           f'%(installdir)s/bin/keytool -import -alias [redacted] /etc/pki/CA/cert [redacted] {keystore_path} -storepass change
           f'%(installdir)s/bin/keytool -import -alias [redacted] /etc/pki/CA/certs [redacted] keystore_path} -storepass changeit
           f'%(installdir)s/bin/keytool -import -alias [redacted] etc/pki/CA/certs/ [redacted] ystore_path} -storepass changeit -
           f'%(installdir)s/bin/keytool -import -alias [redacted] -file /etc/pki/CA [redacted] -keystore {keystore_path} -storep
           f'%(installdir)s/bin/keytool -import -alias [redacted] file /etc/pki/CA/ [redacted] keystore {keystore_path} -storepas
           f'%(installdir)s/bin/keytool -import -alias [redacted] g -file /etc/pki/ [redacted] .pem -keystore {keystore_path} -st
        ]
        for x in cacerts:
          eb.cfg.update('postinstallcmds', x)
```

Custom hooks #2

- To add dependencies which are not defined in default Easyconfig recipes

- Only working for fixed version dependencies
  - New Generation => new version of the dependency
  - E.g., MariaDB

```
def add_extra_dependencies(ec, dep_type, extra_deps):
    """dep_type: must be in DEPENDENCY_PARAMETERS or 'osdependencies'"""
    ec.log.info("[parse hook] Adding %s: %s" % (dep_type, extra_deps))

    if dep_type in DEPENDENCY_PARAMETERS:
        for dep in extra_deps:
            ec[dep_type].append(dep)
    elif dep_type == 'osdependencies':
        if isinstance(extra_deps, tuple):
            ec[dep_type].append(extra_deps)
        else:
            raise EasyBuildError("parse_hook: Type of extra_deps argument (%s), for 'osdependencies' must be "
                                 "tuple, found %s" % (extra_deps, type(extra_deps)))
    else:
        raise EasyBuildError("parse_hook: Incorrect dependency type in add_extra_dependencies: %s" % dep_type)

extra_deps = []

if ec.name == 'R':
    extra_deps.append(('odbcinst', '1'))

if ec.name == 'GATK':
    extra_deps.append(('GATK_python', '0.2'))

if extra_deps:
    add_extra_dependencies(ec, 'dependencies', extra_deps)
```

CVMFS @ BASF

- One build server: Intel Sapphire Rapids
  - RHEL9
  - No need for Gentoo
  - Entire platform is running on RHEL9/UBI9 or higher
- Everything is built on the GENERIC optarch + optimized Sapphire Rapids
- Verify if default loading for Sapphire Rapids would be beneficial (benchmark)
- 3 regions:
  - Europe, North Carolina and AWS
  - Each region has 2 stratum servers
    - No Geo API
- CVMFS clients: HPC + K8S (CVMFS CSI plugin)

Eb wrapper script (in-house)

- Builds easyconfig recipe as a module in several use cases
  - GENERIC archspec
    - EasyBuild module naming scheme (used as INSTALLPATH)
    - Generation module naming scheme (--module-only --fixed-installdir-naming-scheme)
    - EASYBUILD_OPTARCH: GENERIC
  - optimized intel sapphirerapids archspec
    - EasyBuild module naming scheme (used as INSTALLPATH)
    - Generation module naming scheme (--module-only --fixed-installdir-naming-scheme)
    - EASYBUILD_OPTARCH: /
      - CPU type of the build node: intel sapphirerapids

```
[easybuild@by3adm] sandbox_2025b $ ebmmns_biogrid-software.rnd.seeds.basf.net Circos-0.69-9-GCCcore-14.3.0.eb 2025b

building on x86_64/generic

building the software in the default modulenamingscheme

INFO: Following module is loaded: modules/defaultmns

INFO: Following Variables are defined
  - EASYBUILD_INSTALLPATH: /cvmfs/biogrid-software.rnd.seeds.basf.net/tools/eb/x86_64/generic
  - EASYBUILD_INSTALLPATH_MODULES: /cvmfs/biogrid-software.rnd.seeds.basf.net/tools/eb/x86_64/generic/modules
  - EASYBUILD_MODULE_NAMING_SCHEME: EasyBuildMNS
  - EASYBUILD_OPTARCH: GENERIC
INFO: building with the following command: eb Circos-0.69-9-GCCcore-14.3.0.eb --robot

== Temporary log file in case of crash /tmp/eb-qqgddq1i/easybuild-8n3brutd.log
== found valid index for /opt/ppt/eb/easybuild/easyconfigs, so using it...
== resolving dependencies ...
== processing EasyBuild easyconfig /tools/eb/sandbox_2025b/Circos-0.69-9-GCCcore-14.3.0.eb
== building and installing Circos/0.69-9-GCCcore-14.3.0...
  >> installation prefix: /cvmfs/biogrid-software.rnd.seeds.basf.net/tools/eb/x86_64/generic/software/Circos/0.69-9-GCCcore-14.3.0
== fetching files and verifying checksums...
  >> sources:
  >> /gpfs/gssgpfs1/biogrid/tools/eb/sources/c/Circos/circos-0.69-9.t
== ... (took < 1 sec)
== creating build dir, resetting environment...
  >> build dir: /tools/eb/build/Circos/0.69-9/GCCcore-14.3.0
== ... (took < 1 sec)
== unpacking...
  >> running shell command:
        tar xzf /gpfs/gssgpfs1/biogrid/tools/eb/sources/c/Circos/circ
        [started at: 2026-03-30 17:35:36]
        [working dir: /tools/eb/build/Circos/0.69-9/GCCcore-14.3.0]
        [output and state saved to /tmp/eb-qqgddq1i/run-shell-cmd-out
  >> command completed: exit 0, ran in < 1s
== ... (took < 1 sec)
== patching...
== ... (took < 1 sec)
== preparing...
  >> loading toolchain module: GCCcore/14.3.0
  >> loading modules for build dependencies:
  >>  * binutils/2.44-GCCcore-14.3.0
  >> loading modules for (runtime) dependencies:
  >>  * Perl/5.40.2-GCCcore-14.3.0
  >>  * GD/2.77-GCCcore-14.3.0
```

```
INFO: Following module is loaded: modules/2025b

INFO: Following Variables are defined
  - EASYBUILD_INSTALLPATH_MODULES: /cvmfs/biogrid-software.rnd.seeds.basf.net/tools/eb/x86_64/generic/gmns/modules
  - EASYBUILD_MODULE_NAMING_SCHEME: BASFModuleNamingScheme
INFO: building with the following command: eb Circos-0.69-9-GCCcore-14.3.0.eb --robot --module-only --fixed-installdir-naming-scheme

== Temporary log file in case of crash /tmp/eb-ouk_h6u_/easybuild-_uotdugy.log
== found valid index for /opt/ppt/eb/easybuild/easyconfigs, so using it...
== resolving dependencies ...
== processing EasyBuild easyconfig /tools/eb/sandbox_2025b/Circos-0.69-9-GCCcore-14.3.0.eb
== building and installing releases/2025b/Circos/0.69-9...
  >> installation prefix: /cvmfs/biogrid-software.rnd.seeds.basf.net/tools/eb/x86_64/generic/software/Circos/0.69-9-GCCcore-14.3.0
== fetching files and verifying checksums [skipped]
== creating build dir, resetting environment...
  >> build dir: /tools/eb/build/Circos/0.69-9/GCCcore-14.3.0
== ... (took < 1 sec)
== unpacking [skipped]
== patching [skipped]
== preparing...
  >> loading toolchain module: GCCcore/14.3.0
  >> loading modules for build dependencies:
  >>  * binutils/2.44
  >> loading modules for (runtime) dependencies:
  >>  * Perl/5.40.2
  >>  * GD/2.77
  >> defining build environment for GCCcore/14.3.0 toolchain
== ... (took 2 secs)
== configuring [skipped]
== building [skipped]
```

GPU software

- Build on one GPU server
- cuda_compute_capabilities
  - 6.1 (*P40*)
  - 8.9 (*L40, L40S*)
  - 9.0 (*H100*)
- Easyconfig recipes (if supported)
  - Jaxlib
  - PyTorch
    - Other GPU modules depends on PyTorch
  - TensorFlow

Lmod tracking

- Tracking module Usage from Lmod
- Only track “top loaded” modules in SitePackage.lua
  - No need to track dependencies

How many times a module is loaded

How many times loaded	module	path
9	CRISPResso2/2.3.2	/cvmfs/biogrid-software.rnd.seeds.basf.net/tools/eb/x86_64/generic/gmns/modules/all/releases/2022b/CRISPResso2/2.3.2.lua
1	CRISPResso2/20251222	/cvmfs/biogrid-software.rnd.seeds.basf.net/tools/eb/x86_64/generic/gmns/modules/all/releases/2022b/CRISPResso2/20251222.lua
7	cutadapt/4.4	/cvmfs/biogrid-software.rnd.seeds.basf.net/tools/eb/x86_64/generic/gmns/modules/all/releases/2022b/cutadapt/4.4.lua
1.07 K	DBD-Pg/3.16.3	/cvmfs/biogrid-software.rnd.seeds.basf.net/tools/eb/x86_64/generic/gmns/modules/all/releases/2022b/DBD-Pg/3.16.3.lua
27	DIA-NN/2.2.0-Enterprise	/cvmfs/biogrid-software.rnd.seeds.basf.net/tools/eb/x86_64/generic/gmns/modules/all/releases/2022b/DIA-NN/2.2.0-Enterprise.lua
30	DIA-NN/2.3.1-Enterprise	/cvmfs/biogrid-software.rnd.seeds.basf.net/tools/eb/x86_64/generic/gmns/modules/all/releases/2022b/DIA-NN/2.3.1-Enterprise.lua
2	DIA-NN/2.3.2-Enterprise	/cvmfs/biogrid-software.rnd.seeds.basf.net/tools/eb/x86_64/generic/gmns/modules/all/releases/2022b/DIA-NN/2.3.2-Enterprise.lua
9	Diamond2GO/20241025	/cvmfs/biogrid-software.rnd.seeds.basf.net/tools/eb/x86_64/generic/gmns/modules/all/releases/2022b/Diamond2GO/20241025.lua
16	DRAGMAP/1.3.0	/cvmfs/biogrid-software.rnd.seeds.basf.net/tools/eb/x86_64/generic/gmns/modules/all/releases/2022b/DRAGMAP/1.3.0.lua
5	dwreml/1.0.7-R-4.2.2	/cvmfs/biogrid-software.rnd.seeds.basf.net/tools/eb/x86_64/generic/gmns/modules/all/releases/2022b/dwreml/1.0.7-R-4.2.2.lua
5	Emacs/28.2	/cvmfs/biogrid-software.rnd.seeds.basf.net/tools/eb/x86_64/generic/gmns/modules/all/releases/2022b/Emacs/28.2.lua

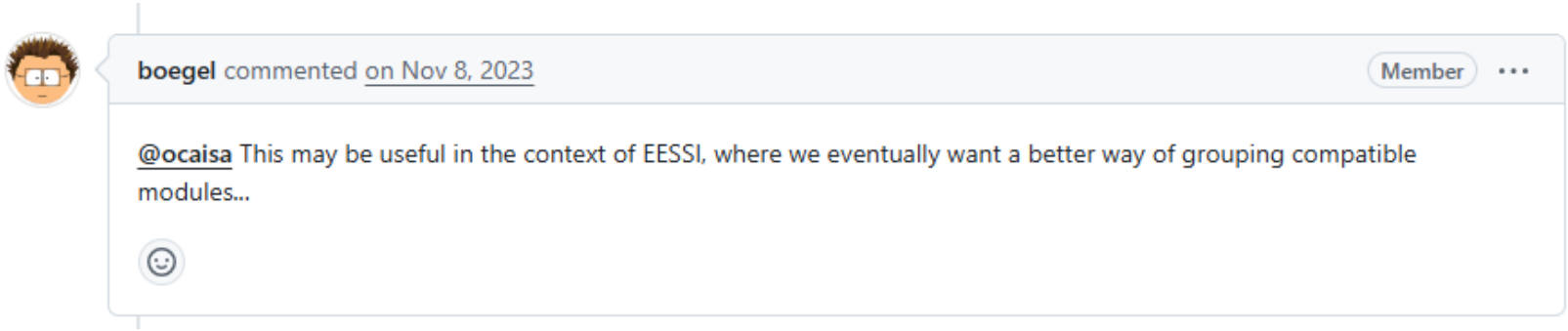
```
ExitHookA.register(report_loads)

function l_registerTopLoaded(fullName, frameStk)
   local varT     = frameStk:varT()
   local modList  = "LOADEDTOPMODULES"
   local nodups   = true
   local priority = 0
   local delim    = ":"
   if (varT[modList] == nil) then
      varT[modList] = Var:new(modList, nil, nodups, delim)
   end

   varT[modList]:append(fullName, nodups, priority)

end

function unload_hook(t)

   if (mode() ~= "unload") then return end
   local frameStk    = FrameStk:singleton()

   l_registerUnTopLoaded(t.modFullName, frameStk)

end

function l_registerUnTopLoaded(fullName, frameStk)
   local varT     = frameStk:varT()
   local modList  = "LOADEDTOPMODULES"
   local nodups   = true
   local priority = 0
   local delim    = ":"
   local where    = "all"

   if (varT[modList] == nil) then
      varT[modList] = Var:new(modList, nil, nodups, delim)
   end

   varT[modList]:remove(fullName, where, priority)

end

hook.register("unload",unload_hook)
```

Common issues and fixes in 2025b

- Issues:
  - GCTA & bcl2fastq2:
    - Using a Boost version, as dependency, which is too old
  - 2025b is using numpy version 2.3.2 (Python 3.13.5)
    - PyMOL: pymol 3.1.0 has requirement numpy<2,>=1.26.4
    - omamer: requires numpy<2
  - PyRosetta
    - Linux variant is not supported yet for python 3.13
- Fixes:
  - RMBlast: Boost 1.88 issue
    - ::boost::test_tools::tt_detail::dummy_cond() in test_boost.hpp
      - Replacing by “false”
  - Argtable: include ctype header
  - DRAGMAP
    - include <boost/range/iterator_range_core.hpp> to fix 'make_iterator_range'

Future plans

- Experiment to build our CVMFS stack on top of the EESSI stack
  - Use the EESSI stack as “base layer”
  - Would helps if PR would be merged: https://github.com/easybuilders/easybuild-framework/pull/3547☺



  - Maybe no longer relevant with lfoss, rfoss, …
- Setup open OnDemand (as PoC)
- Optimize eb wrapper script

Thank you!

Questions?