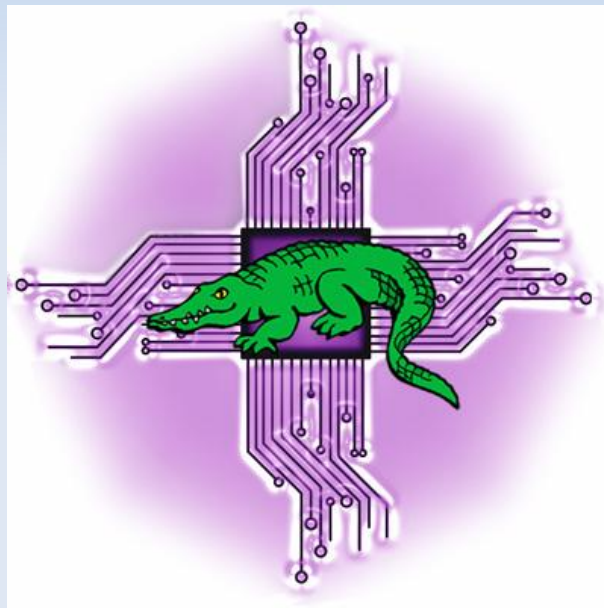


EasyBuild: Building Software with Ease

Presentation to Linux Users of Victoria

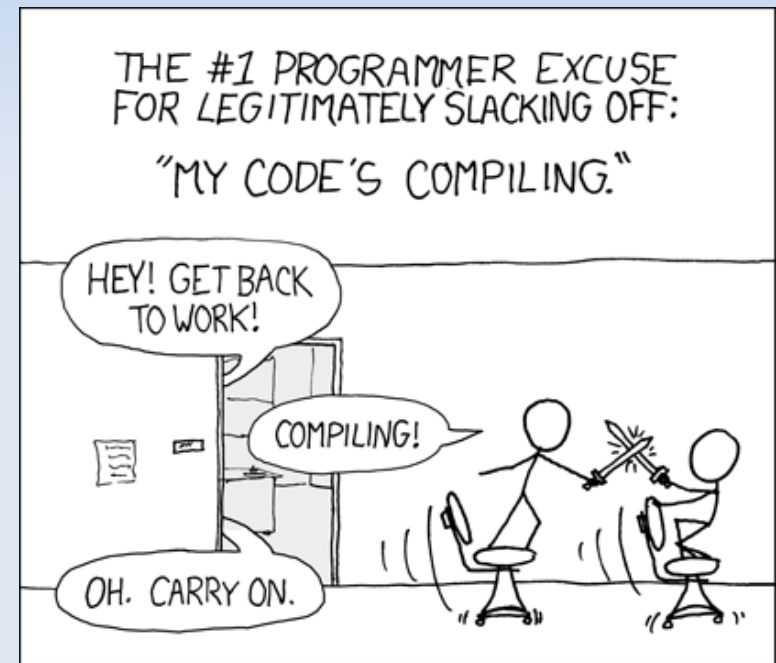


Melbourne, April 2, 2019

<http://levlafayette.com>

Building Software from Source

- It is much easier to install packaged software in almost all cases, where available (too old? too new?).
- Software sources are required if you want to do any development on an application or you need multiple versions of the same software.
- Performance of compiled software is often much faster than packaged software.
- See **Compiling from Source in Linux. Presentation to Linux Users of Victoria, April 22, 2017**



The Problem with Dependencies and Builds

- Even on systems where a single version of an application is required dependency issues arise with builds.
- Quantity or chain of dependencies can be overwhelming, conflicting or circular dependencies etc.
- Need exports of library paths and/or use of an environment modules system and/or build shell scripts.
- Several build systems (make, GNU Autoconf, CMake, scons, binaries etc).
- Many people who are building software (e.g., in a HPC space) are spending a *lot* of time on this!



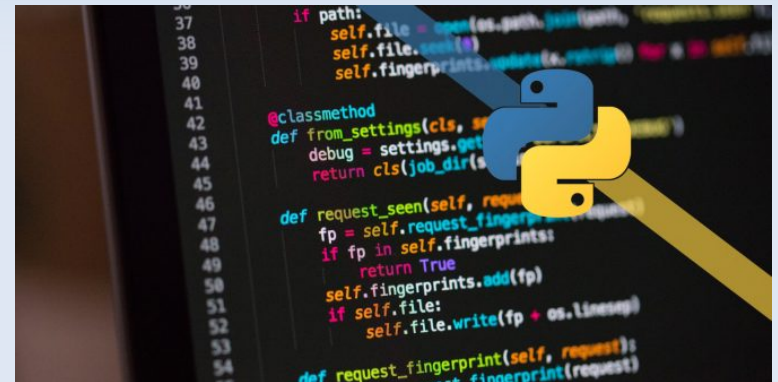
Introducing EasyBuild

- The EasyBuild project was started in 2009 by the HPC team of Ghent University (Belgium).
- First public release, v0.5, in April 2012 with v1.0 in November 2012.
- Provides automated and reproducible software builds with differentiation between software, version, compiler etc.
- Integrated with the Lmod advanced modules system.
- Built around principle of open collaboration between HPC sites.



Installing Easybuild

- Ironically, one of the more challenging things about Easybuild used to be installing Easybuild.
- Requires Linux (or OS X), Python 2.6+ with setuptools (0.6+), vsc-install, vsc-base, etc, plus a few shell tools.
- Requires a modules system. Lmod \geq 5.6.3 recommended.
- Bootstrap script makes installation a lot easier, but alternative installation methods documented.
- Run sanity check and unit tests after installation.



Easybuild Recipes, Now We're Cooking

- Easybuild builds software packages with an easybuild user through easyconfig files in the current working directory and with command options. For an overview of known toolchains, use ``eb --list-toolchains`` (e.g., foss, intel). Toolchains can become deprecated.
- Easybuild easyconfig requires name, version, easyblock, homepage, description, toolchain, sanity-check.
- Easybuild will detect if modules are already load and will issue a warning. Modules can be white-listed.
- Produces a temporary build directory and log for debugging and includes build log in the .eb recipe.



Easybuild Recipes, Professional Chef

- More advanced recipe options include patches, configopts, buildopts, installopts etc.
- Customised steps and values available for specific easyblocks.
- Easybuild can be used to generate (singularity, docker) container images with ``eb --containerize``
- Easybuild can install software and dependencies with ``--robot`` option e.g., ``eb HPL-2.0-goolf-1.4.10.eb --robot --dry-run``
- Collaboration is important. Collection of easybuild recipes available. Bugs, pull requests, yaml syntax etc.



Easybuild Recipes, Master Chef

- Additional Python modules can be added with the CSV listed of paths for non-supported software, different naming schemes, new compilers, additional EasyBlocks etc.
- Easybuild can integrate with github for pull requests (testing, merging, submitting), uploading test reports, etc e.g., ``eb --new-pr example.eb``
- With FPM can be used to create binary packages (RPMs, Debian files, etc.) e.g., ``eb --package Perl-5.20.1-GCC-4.9.2-bare.eb``
- With ``--job`` Easybuild can submit jobs for the installations. Default is GC3Pie (works with TORQUE, PBS, Slurm) with usual paramters (cores, walltime, dependencies etc) `--robot``.



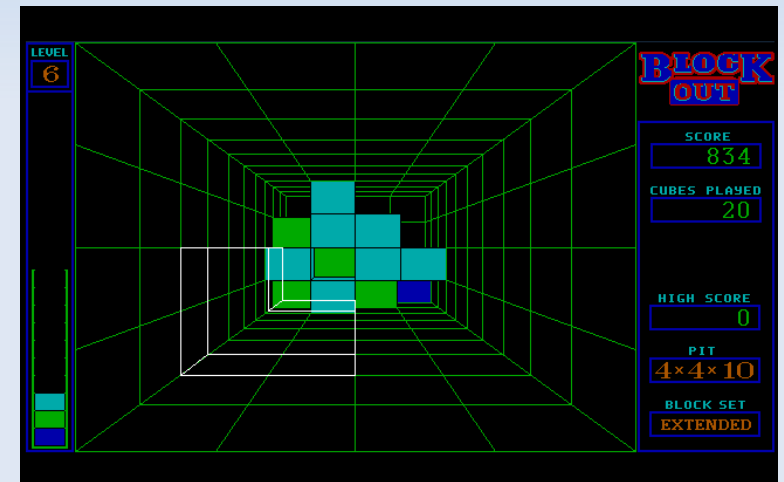
Easybuild Recipe Madness, Icelandic Cuisine

- **Written with PEP 8 for Python (e.g., spaces not tabs, underscores not CamelCase) etc. Can check with ``eb --check-contrib application.eb``**
- **Syntax for unusual dependences can be interesting (i.e., those that don't follow default toolchain).**
- **Which path to insanity for applications with extensions? e.g., R, Python, Perl. Use of ``--skip --rebuild``.**
- **Partial builds allow for ``--stop`` and ``--module-only``, avoiding critical checks.**



Easybuild Easyblocks

- The command ``eb --list-easyblocks`` provides a list of easyblocks in a hierarchy.
- Easyblocks are Python modules that implement a software build and installation procedure.
- Easyblocks are either generic, or application oriented which derive from the generic. Include custom easyconfig parameters, class variables, reading/writing/patching files, shell commands, custom sanity checks etc.
- Easyblocks can include pre-defined functions ("hooks") to alter or augment installation procedures.



Further Reading

- **Compiling from Source in Linux**
<http://levlafayette.com/files/2017luv-compiling.pdf>
- **Easybuild Easyblocks**
https://easybuild.readthedocs.io/en/latest/version-specific/generic_easyblocks.html
<https://github.com/easybuilders/easybuild-easyblocks/tree/master/easybuild/easyblocks>
- **Easybuild Easyconfigs**
<https://github.com/easybuilders/easybuild-easyconfigs/tree/master/easybuild/easyconfigs>
- **EasyBuild Templates**
http://easybuild.readthedocs.io/en/latest/version-specific/easyconfig_templates.html
- **Effing Package Management**
<https://github.com/jordansissel/fpm>
- **GC3Pie**
<https://gc3pie.readthedocs.org/>
- **Installing R with EasyBuild: Which Path to Insanity?**
<http://levlafayette.com/node/527>
- **Modern Scientific Software Management Using EasyBuild and Lmod, HUST 2014**
https://easybuilders.github.io/easybuild/files/hust14_paper.pdf
- **Python PEP 8 Style Guide**
<https://www.python.org/dev/peps/pep-0008/>

THANKS FOR WATCHING

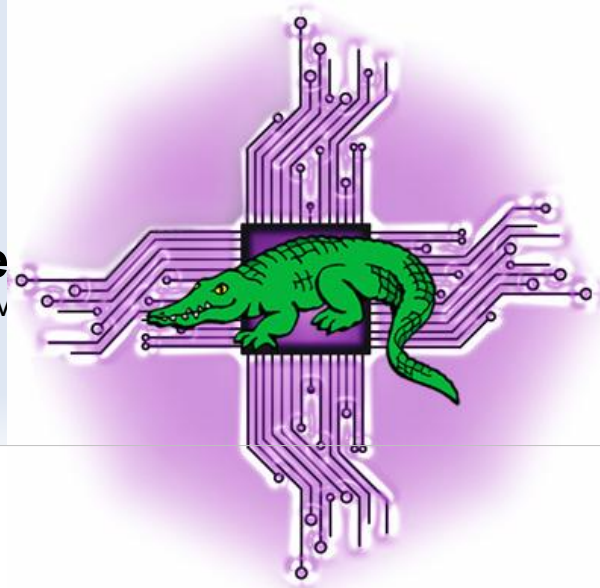


& LISTENING PATIENTLY

EasyBuild: Building Software with Ease

Presentation to Linux Users of Victoria

Melbourne
<http://lev>



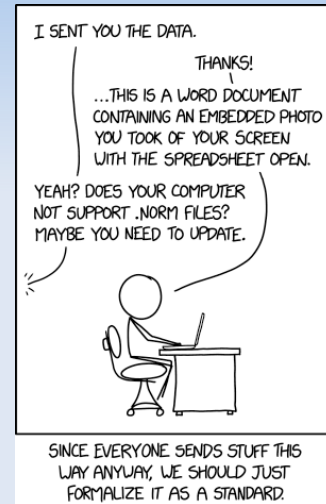
Building Software from Source

- It is much easier to install packaged software in almost all cases, where available (too old? too new?).
- Software sources are required if you want to do any development on an application or you need multiple versions of the same software.
- Performance of compiled software is often much faster than packaged software.
- See [Compiling from Source in Linux](#). Presentation to Linux Users of Victoria, April 22, 2017



The Problem with Dependencies and Builds

- Even on systems where a single version of an application is required dependency issues arise with builds.
- Quantity or chain of dependencies can be overwhelming, conflicting or circular dependencies etc.
- Need exports of library paths and/or use of an environment modules system and/or build shell scripts.
- Several build systems (make, GNU Autoconf, CMake, scons, binaries etc).
- Many people who are building software (e.g., in a HPC space) are spending a *lot* of time on this!



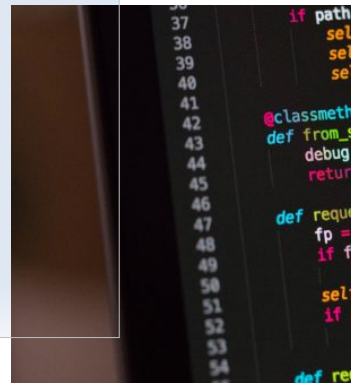
Introducing EasyBuild

- The EasyBuild project was started in 2009 by the HPC team of Ghent University (Belgium).
- First public release, v0.5, in April 2012 with v1.0 in November 2012.
- Provides automated and reproducible software builds with differentiation between software, version, compiler etc.
- Integrated with the Lmod advanced modules system.
- Built around principle of open collaboration between HPC sites.



Installing Easybuild

- Ironically, one of the more challenging things about Easybuild used to be installing Easybuild.
- Requires Linux (or OS X), Python 2.6+ with setuptools (0.6+), vsc-install, vsc-base, etc, plus a few shell tools.
- Requires a modules system. Lmod \geq 5.6.3 recommended.
- Bootstrap script makes installation a lot easier, but alternative installation methods documented.
- Run sanity check and unit tests after installation.



Easybuild Recipes, Now We're Cooking

- Easybuild builds software packages with an easybuild user through easyconfig files in the current working directory and with command options. For an overview of known toolchains, use ``eb --list-toolchains`` (e.g., foss, intel). Toolchains can become deprecated.
- Easybuild easyconfig requires name, version, easyblock, homepage, description, toolchain, sanity-check.
- Easybuild will detect if modules are already load and will issue a warning. Modules can be white-listed.
- Produces a temporary build directory and log for debugging and includes build log in the .eb recipe.



Easybuild Recipes, Professional Chef

- More advanced recipe options include patches, configopts, buildopts, installopts etc.
- Customised steps and values available for specific easyblocks.
- Easybuild can be used to generate (singularity, docker) container images with ``eb --containerize``
- Easybuild can install software and dependencies with ``--robot`` option e.g., ``eb HPL-2.0-goolf-1.4.10.eb --robot --dry-run``
- Collaboration is important. Collection of easybuild recipes available. Bugs, pull requests, yaml syntax etc.



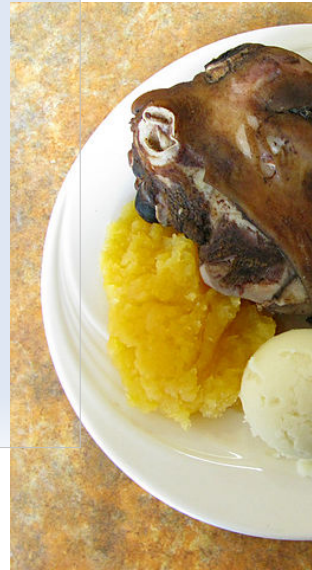
Easybuild Recipes, Master Chef

- Additional Python modules can be added with the CSV listed of paths for non-supported software, different naming schemes, new compilers, additional EasyBlocks etc.
- Easybuild can integrate with github for pull requests (testing, merging, submitting), uploading test reports, etc e.g., ``eb --new-pr example.eb``
- With FPM can be used to create binary packages (RPMs, Debian files, etc.) e.g., ``eb --package Perl-5.20.1-GCC-4.9.2-bare.eb``
- With ``--job`` Easybuild can submit jobs for the installations. Default is GC3Pie (works with TORQUE, PBS, Slurm) with usual parameters (cores, walltime, dependencies etc) `-- robot``.



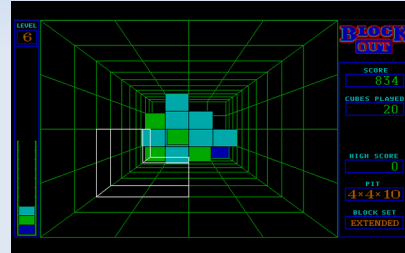
Easybuild Recipe Madness, Icelandic Cuisine

- Written with PEP 8 for Python (e.g., spaces not tabs, underscores not CamelCase) etc. Can check with ``eb --check-contrib application.eb``
- Syntax for unusual dependences can be interesting (i.e., those that don't follow default toolchain).
- Which path to insanity for applications with extensions? e.g., R, Python, Perl. Use of ``--skip --rebuild``.
- Partial builds allow for ``--stop`` and ``--module-only``, avoiding critical checks.



Easybuild Easyblocks

- The command ``eb --list-easyblocks`` provides a list of easyblocks in a hierarchy.
- Easyblocks are Python modules that implement a software build and installation procedure.
- Easyblocks are either generic, or application oriented which derive from the generic. Include custom easyconfig parameters, class variables, reading/writing/patching files, shell commands, custom sanity checks etc.
- Easyblocks can include pre-defined functions ("hooks") to alter or augment installation procedures.



Further Reading

- **Compiling from Source in Linux**
<http://levlafayette.com/files/2017luv-compiling.pdf>
- **Easybuild Easyblocks**
https://easybuild.readthedocs.io/en/latest/version-specific/generic_easyblocks.html
<https://github.com/easybuilders/easybuild-easyblocks/tree/master/easybuild/easyblocks>
- **Easybuild Easyconfigs**
<https://github.com/easybuilders/easybuild-easyconfigs/tree/master/easybuild/easyconfigs>
- **EasyBuild Templates**
http://easybuild.readthedocs.io/en/latest/version-specific/easyconfig_templates.html
- **Effing Package Management**
<https://github.com/jordansissel/fpm>
- **GC3Pie**
<https://gc3pie.readthedocs.org/>
- **Installing R with EasyBuild: Which Path to Insanity?**
<http://levlafayette.com/node/527>
- **Modern Scientific Software Management Using EasyBuild and Lmod, HUST 2014**
https://easybuilders.github.io/easybuild/files/hust14_paper.pdf
- **Python PEP 8 Style Guide**
<https://www.python.org/dev/peps/pep-0008/>

THANKS FOR WATCH



& LISTENING PATIENT