Best Practices in Building & Installing Scientific Software

Jens Timmerman Stijn De Weirdt

Department ICT
Ghent University
easybuild@lists.ugent.be

International Supercomputing Conference, 2013

Who Are We

- Central contact for HPC at Ghent University, Belgium
- Part of the central ICT department (DICT)
- Member of the VSC (Flemish Supercomputing Centre)
- We support researchers using our infrastructure
  - A lot of software installations

UNIVERSITEIT GENT

Outline

- Problem description
- Show of hands
- Best Practices for Software installation tools
- EasyBuild
- Discussion

UNIVERSITEIT GENT

Problem: Building and Installing Scientific software

- Compilation requires lots of expertise and specific knowledge.
- Not all documentation is created equal.
- Collaboration and sharing the knowledge is problematic at best.
- Not all software has a straightforward build procedure. e.g.:
  - ALADIN, CP2K, dolfin, NCL, NWChem, OpenFOAM, PETSc, QuantumESPRESSO, WIEN2k, WRF

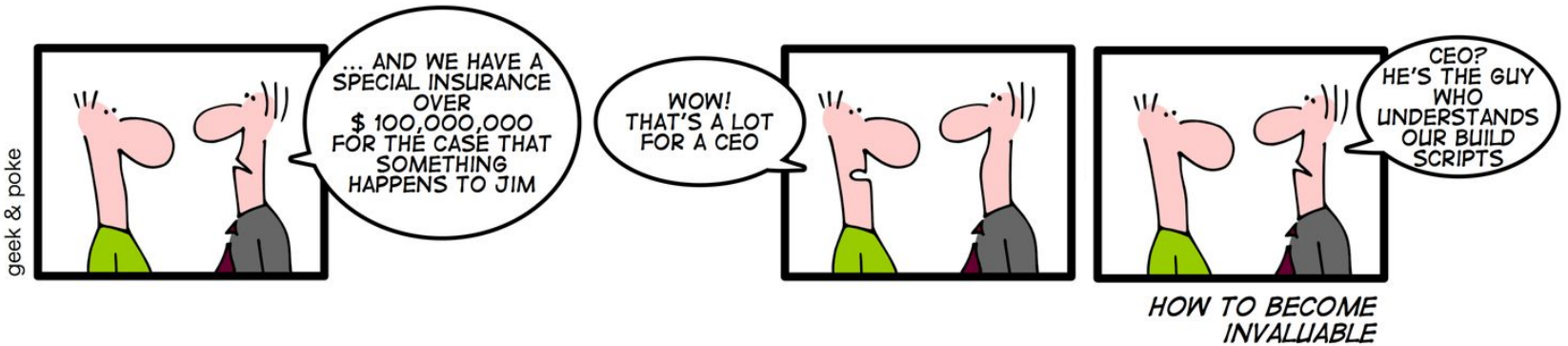


Figure : http://geekandpoke.typepad.com/

Who Are You?

Show of hands

- Scientific software Developers
- Researchers / Users
- System Administrators
  - Who do user support
  - Who install scientific software
- Others ?

UNIVERSITEIT GENT

Tools For Software Management

- Environment Modules
- Lmod
  - Recent version?
- Naming Schemes?
  - flat vs tree?

UNIVERSITEIT GENT

Tools For Software Installation

- Bash Scripts / Makefiles
- Wrappers
  - portage / ports / homebrew
- RPM's / deb's
- Wiki
- 'That guy' (Jim)
- Others?

UNIVERSITEIT GENT

Best Practices

- Automation
  - Reproducability
    - Installations on different systems
    - Colaboration: Be able to easily share your work
  - Recompile using different compilers / libraries (math / mpi)
  - Autogenerate the module files
- Provide multiple versions of the same packages
- Verification of the installation
  - All libraries and binaries are available
  - And they produce the correct results

UNIVERSITEIT GENT

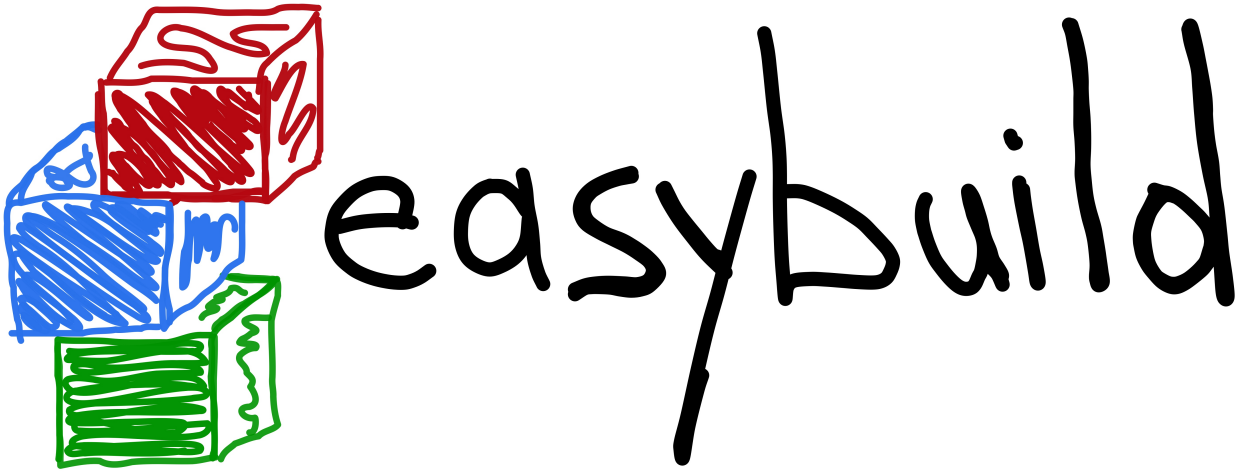
Best Practices (2)

- Keep track of :
  - build procedure
  - patches
  - output
  - Metadata: build time, build by, ...
  - Dependencies (automatically build these)
- Encapsulate upstream build tools where they are provided
  - Use make, cmake, setuptools
- Good to have:
  - Automatic downloads of sources
- More?

UNIVERSITEIT GENT

EasyBuild: A Software installation framework

- Open source (GPLv2) since April 2012
  - developed in-house (HPC-UGent) for 2.5 years
  - stable API since Nov. 2012 (v1.0.0)
- Written in Python



UNIVERSITEIT GENT

Features

- Uses specification files/recipes called easyconfigs which describe a software package.
  - name, version, website, description
  - installation files (sources), patches, download location
  - compiler toolkit to build with
  - optional build parameters
  - versions of the package that are know to work this way
  - ...
- Allows for diverting from default 'configure/ make' through extensions called *easyblocks*.
- Has all 'features' reported above

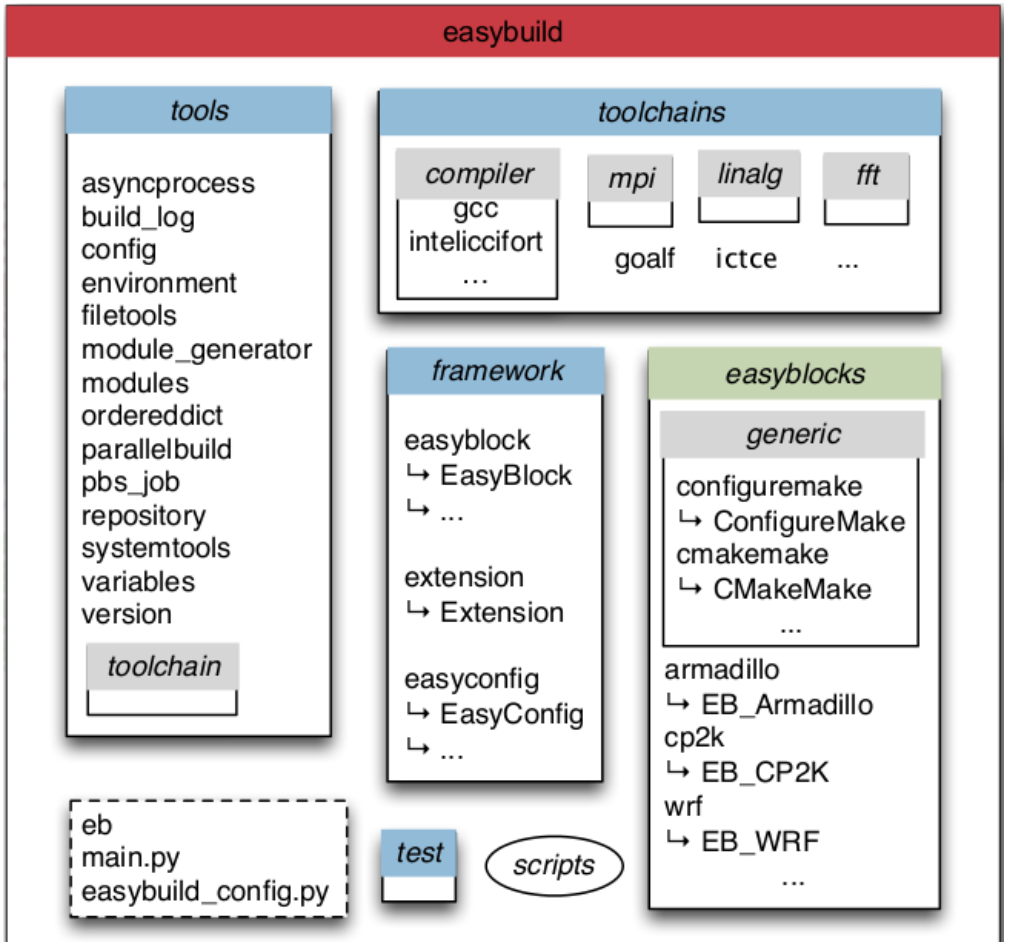
UNIVERSITEIT GENT

Dependencies

- Linux / OS X
  - used daily on Scientific Linux 5.x/6.x (Red Hat-based)
  - also tested on Fedora, Debian, Ubuntu, CentOS, SLES, ...
  - Recently: OS X support (but not turned on for all software packages yet)
- Python 2.4 or more recent 2.x
- environment modules or lmod (support added in develop, will be available in v1.6, due 1st of July)
- C/C++ compiler
  - to bootstrap GCC or llvm
  - or icc

UNIVERSITEIT GENT

Architecture



Supported Out of the box

a2ps ABAQUS ABINIT ABySS ACML ALADIN AMOS AnalyzeFMRI ant aria2 Armadillo ASE ATLAS Autoconf bam2fastq Bash bbcp bbFTP bbftpPRO BEAGLE BFAST binutils BioPerl BiSearch Bison BLACS BLAST Bonnie++ Boost Bowtie Bowtie2 BWA byacc bzip2 CBLAS ccache cflow CGAL cgdb cgmpich cgmpolf cgmvapich2 cgmvolf cgompi cgoolf Chapel Clang ClangGCC CLHEP ClustalW2 CMake Corkscrew CP2K CPLEX CRF++ CUDA Cufflinks cURL CVXOPT Cython DL_POLY_Classic Docutils DOLFIN Doxygen EasyBuild ECore Eigen ELinks EMBOSS EPD ESMF ESPResSo expat FASTX-Toolkit FCM Ferret FFC FFTW FIAT flex FLUENT fmri FRC_align freeglut FreeSurfer freetype FSL g2clib g2lib GATE GATK GCC GDAL GDB Geant4 GEOS GHC git glproto gmacml GMP gmvapich2 gmvolf gnuplot gnutls goalf gompi google-sparsehash goolf goolfc GPAW gperf Greenlet grib_api GROMACS GSL guile gzip h5py h5utils Harminv HDF HDF5 HH-suite HMMER HPL hwloc Hypre icc iccifort ictce ifort iiqmpi imkl impi Infernal Inspector Instant iomkl Iperf ipp iqacml itac JasPer Java Jinja2 JUnit LAPACK lftp libctl libdrm libffi libgtextutils libharu libibmad libibumad libibverbs Libint libmatheval libpciaccess libpng libpthread-stubs libreadline libsmm libtool libunistring libxc libxcb libxml2 libxslt libyaml likwid lxml LZO M4 makedepend Maple MariaDB MATLAB matplotlib mc MCL MDP Meep MEME Mercurial Mesa MetaVelvet METIS Mothur MPFR mpi4py mpiBLAST MPICH MrBayes MTL4 MUMmer MVAPICH2 MySQLdb nano NASM NCBI-Toolkit NCL ncurses netCDF netCDF-C++ netCDF-Fortran nettle NEURON ns numactl numexpr numpy NWChem Oger OpenBLAS OpenFOAM OpenIFS OpenMPI OpenPGM OpenSSL ORCA orthomcl otcl PAML pandas PAPI parallel ParMETIS Pasha paycheck PCRE Perl PETSc petsc4py phonopy pkg-config Primer3 problog PSI pyTables Python python-meep PyYAML PyZMQ QLogicMPI QuantumESPRESSO R RAxML RNAz ROOT SAMtools ScaLAPACK ScientificPython scikit-learn scipy SCOOP SCOTCH setuptools Shapely SHRiMP SLEPc SOAPdenovo Sphinx Stow SuiteSparse SWIG Szip Tar tbb Tcl tclcl tcsh test.pl Theano TiCCutils TiMBL TinySVM Tk TopHat Tornado TotalView Trilinos Trinity UDUNITS UFC UFL util-linux Valgrind Velvet ViennaRNA Viper VSC-tools VTK VTune WIEN2k wiki2beamer WPS WRF xcb-proto XCrySDen XML xorg-macros xproto YamCha Yasm ZeroMQ zlib zsync

UNIVERSITEIT GENT

Call For Contributions

- Feedback
  - give it a spin, let us know how it turns out
  - what features are missing that you require?
- report problems
  - via mail, GitHub issue tracker, IRC, ...
- Send us a pull request
  - Better than reporting problems is fixing them!
  - Contribute back features in Framework
  - Add support for aditional compilers/libraries
    - Intel Xeon Phi (look at the cuda toolchain)
  - Add easyconfigs / easyblocks for new software
- help verify correctness of easyblocks/easyconfigs

UNIVERSITEIT GENT

Further Reading

- Website: http://hpcugent.github.io/easybuild
- Github: http://github.com/hpcugent/easybuild[-framework|-easyblocks|-easyconfigs]
- PyPi: http://pypi.python.org/pypi/easybuild[-framework|-easyblocks|-easyconfigs]
- Mailing list: easybuild@lists.ugent.be
- Twitter: @easy_build
- IRC: #easybuild @ freenode
- Paper: Hoste, K., Timmerman, J., Georges, A., & De Weirdt, S. EasyBuild: Building Software With Ease.

UNIVERSITEIT GENT

Open Problems

- Naming schemes
- Tools for testing
  - Testcases to validate installations
  - Example jobscripts
    - require domain specific knowledge
- -rpath vs $LD_LIBRARY_PATH
- Source being removed from the web
  - Set up an own mirror for software with a license that allows redistribution
- Others?

UNIVERSITEIT GENT