DE LA RECHERCHE À L'INDUSTRIE

www.cea.fr

Easybuild on CEA/TGCC

3rd Easybuild User Meeting

Laurent Nguyen -
laurent.nguyen@cea.fr

January 31th 2018

CEA (Alternative Energies and Atomic Energy Commission) is a french organization for research, development and innovation in four main areas:

- defence and security,
- nuclear and renewable energies,
- technological research for industry,
- fundamental research in the physical sciences and life sciences.

http://www.cea.fr/

Tres Grand Centre de Calcul du CEA

TGCC for “CEA very big computing center” is a facility which hosts supercomputers.

PRACE and GENCI

- Curie (Intel SNB): 1.7 Pflop/s
- Irene Joliot-Curie :
  - Intel SKX : 6.8 Pflop/s
  - Intel KNL : 2.1 Pflop/s

CCRT

Cobalt (Intel BDW) : 1.4 Pflop/s

http://www-hpc.cea.fr
http://www.prace-ri.eu
http://www.genci.fr
http://www-ccrt.cea.fr

Usage

Users

More than 2000 users run jobs on TGCC's supercomputers.

Scientific Domains

Climate, Astrophysic, CFD, Chemistry, Molecular Dynamic, Bioinformatic, Genomic, etc ...

User support team

The user support team (8 peoples) helps users for their software problem on our clusters (debugging, optimization, ...). Product installation is part of their work.

History

2012 15 installations for the opening of Curie

2015 520 installation on Curie and 93 installations on a cluster dedicated to compilation => **Introduction of Easybuild-2.0.0**

2016 **Introduction of Easybuild-2.7.0**

2016 First success: almost 100 products were recompiled in Intel 15 in 3 days

2017 900 installations for the end of Curie and 580 installations on Cobalt (and Irene)

2017 Multi-toolchain and multi-architecture / Optimization of process for installations on many clusters

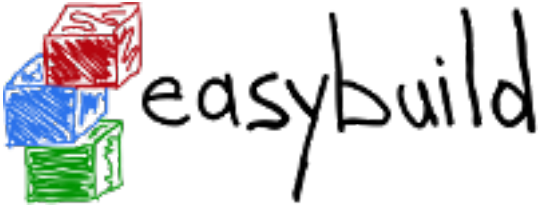
Easybuild

Before Easybuild

- No rules for the installation of products (RPM, binaries or compilation)
- Documentation written in a Word document

Since Easybuild

- Standardization of processes for installing products
- Easyblocks, easyconfigs and installation logs are part of documentation for installations
- Usage of easyblocks and easyconfigs from community



Easybuild

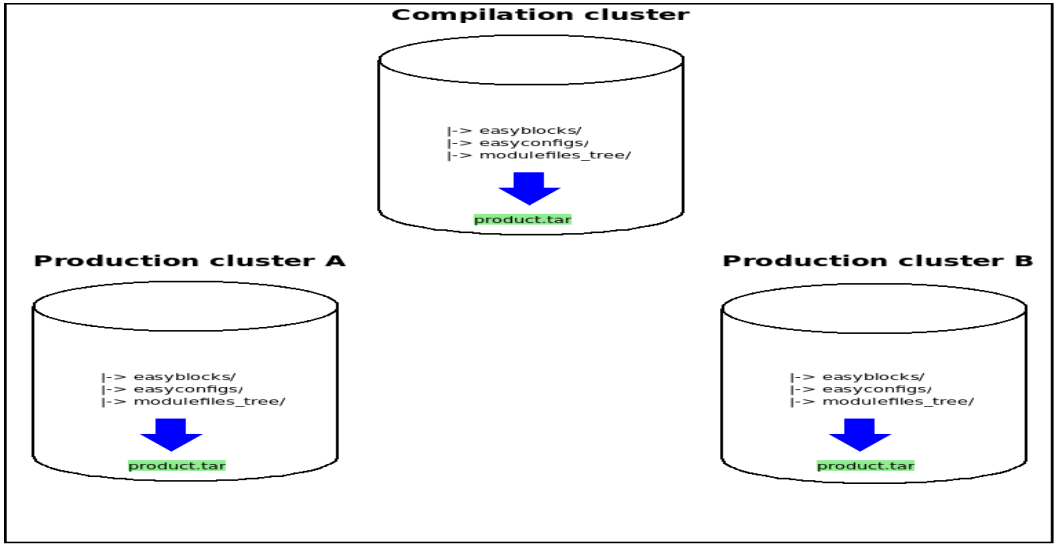
Developments in Easybuild

- Naming scheme for installation directories and for modulefiles
- Script to generate an easyconfig for a given toolchain
- eb_pack.py script to generate tar file to copy a product to other clusters

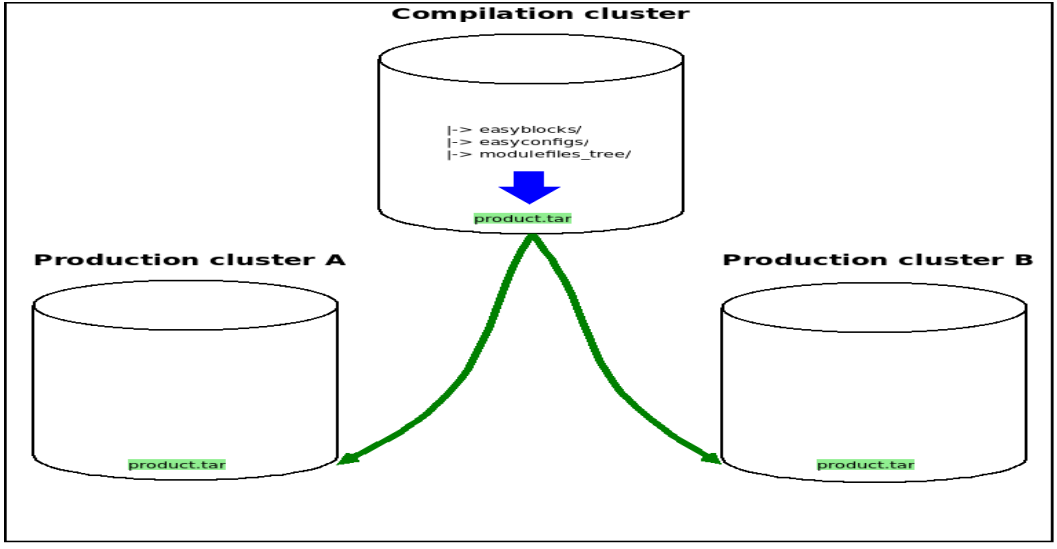
Troubles with Easybuild

- Offline installation
- Learning to write a new easyblock
- Toolchain modification / customization
- Need to create an easyconfig for each toolchain
- At first, we did have one Easybuild installation and its specific easyblocks/easyconfigs files on each cluster and at the end all installation diverged. Some products had different behaviour depending where there were compiled or who compiled it.

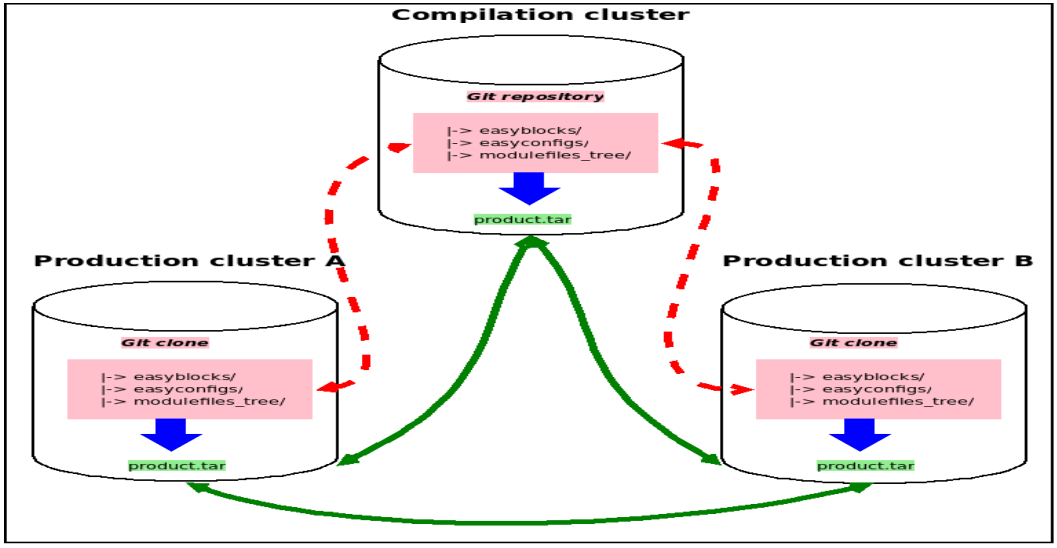
Installation process for many clusters



Installation process for many clusters



Installation process for many clusters



Notions

To lessen and to simplify the output of “module avail” command, we need to define different notions.

Flavor

A **flavor** of a product is when the product needs to be compiled in a different way than the standard installation like special patch, different options for configure command, etc...
Modules lets us create a flavor modulefile to use the product in the right installation directory.
Example: Gromacs with plumed
$ module list flavor/gromacs
flavor/gromacs/standard flavor/gromacs/plumed
$ module load flavor/gromacs/plumed gromacs

Feature

A **feature** for a product is a module which modifies the behaviour of a product.
Example: setting a OMPI_MCA_mpi_leave_pinned environment variable for OpenMPI library
$ module load feature/openmpi/enable_leave_pinned openmpi

Multi-architecture

On our cluster, we have both x86_64 and KNL architectures. Some products need to be compiled for each architecture. We provide two modules “compil/x86_64” and “compil/mic” which let users choose the product they need. (Still experimental)

Multi-toolchain

Easybuild provides a simple way to compile the same product with different toolchains. The toolchains are the combination of compilers (Intel 15.x.x.x and 17.x.x.x) and MPI libraries (OpenMPI-1.8.8, OpenMPI-2.0.2 and Wi4MPI). 6 versions of a product are installed (without counting the architecture).
Instead of providing 6 modules, we choose to provide only one module. Depending on the compiler and the MPI library loaded, the loading of the module will choose the right installation of the product.

Installation directory

/<Central installation directory>/<architecture>/
<product name>-<version>/<compiler+version>__<mpi+version>/<flavor>

Modules in Easybuild

Easybuild uses module command to build dependency tree for the installation. For each installation, a modulefile is created and it can be provided to users. There are two inconvenient for our users:

- The default name for a module contains the name of toolchain and it can be incomprehensible or too long.
- If we want a product to be compiled with different toolchains, the number of modules provided can be important.

Environment modules

On TGCC, we use Modules (aka modules-tcl) because it is a product we know well and it is highly customizable for our needs. Thanks to modules, we are able to manage multi-toolchain, multi-architecture, flavor and feature for users environment.

`https://github.com/cea-hpc/modules`

Features needed

- Find something to make the system less strict about toolchain dependency. Ex: A new minor version of Intel compiler, we don't want to compile again all products ...
- rpath (and we'll propose it as a flavor for our users)
- Remove safely a product: need to track its dependencies and products which has the product as dependency. Product life policy ?
- Packaging a self installation to install everywhere without installing Easybuild

Conclusions

Easybuild brings to our installation work

- A standardization of our process
- A strict and reproductible way for installations
- A way to provide documentation about installations
- A gain of time for our multi-toolchain and our multi-architecture installations

Futur works with Easybuild

- Try the ”–jobs” option!
- Get information about installation for users
- Automate the installation process (Jenkins)
- Test the installations (Jenkins)
- Virtualization with PCOCC (`https://github.com/cea-hpc/pcocc`) to manage different OS or versions of OS
- Try to contribute ...

Questions ?

Visit our GitHub!

https://github.com/cea-hpc

Commissariat à l'énergie atomique et aux énergies alternatives
Centre de Bruyères-le-Châtel | 91297 Arpajon Cedex
CEA
T. +33 (0)1 69 26 40 00 | F. +33 (0)1 69 26 40 00
Établissement public à caractère industriel et commercial
RCS Paris B 775 685 019