

Gestion des environnements utilisateurs

Logiciel Cluster | Philippe GRÉGOIRE

Pile logicielle pour les utilisateurs

Visualisation

Maillage

Code Maison

Code ISV

Interface Web

Biblio.
scientifiques

Biblio.
I/O

Bibliot.
Message
Passing

Outils
de tests

Profilers,
Debuggers

Compilateurs
Intel,
GNU,
PGI

Gestion.
Ressources

Filesystem

Interconnect

Système d'exploitation

Matériel

Pile logicielle selon les nœuds

Nœud Login

User login,
development,
job launch &
monitor, result

User Interfaces

Vizu tools

Dev tools

Librairies IO et //

Ressources Mgt

Agent mesure

Librairies Dyn

OS

Nœud Calcul

No user login,
job execution,
computation &
I/O

Applications //

Dev tools

Librairies IO et //

Ressources Mgt

Agent mesure

Librairies Dyn

OS

Nœud IO

No user login,
provide IO job
service :

* network IO

* storage IO

Routeur Réseau

Routeur FS //

Agent mesure

Librairies Dyn

OS

Nœud service

No user login,
provide system
services :

Surveillance

Annuaire

Noms de domaine

Service Config

Ressources Mgt

Agent mesure

Librairies Dyn

OS

Logiciels OpenSource

Quels logiciels

- Logiciels non fournis par des ISV
 - Compilateurs
 - Librairies de programmation MPI
 - Librairies d'entrées/sorties
 - Librairies mathématiques
 - Outils de visualisation
 - Utilitaires
- Chacun vient avec son environnement
 - Variables d'environnement
 - Man pages (MANPATH)
 - Emplacement des commandes : PATH
 - Emplacement des librairies : LD_LIBRARY_PATH

Compilateurs Intel

- Génère du code optimisé pour x86_64
- Package Intel Parallel Studio
 - [C , C++, Fortran, MKL, DBG]
- Outil de profiling Vtune
- License

Linux*

Intel provides the development tools Linux developers need to deliver winning performance.



Intel® Parallel Studio XE

Develop high-performance, scalable, and reliable software for high-performance computing (HPC), enterprise, and cloud solutions. This comprehensive tool suite simplifies the creation and modernization of code with the latest techniques in vectorization, multithreading, multinode, and memory optimization. Includes industry-leading compilers, numerical libraries, performance profilers, and code analyzers.



Intel® VTune™ Amplifier

The premier performance profiler for C, C++, C#, Fortran, Assembly, and Java*. Get profiling data and analysis tools to tune performance of CPU, GPU, threading, memory, energy efficiency, cache, and storage.

Compilateurs OpenSource

- GNU :
 - Front ends C, C++, Fortran, Ada, and Go,
 - Back-ends X86, X86-64, PowerPC, PowerPC-64, ARM
 - <https://gcc.gnu.org/>
- LLVM
 - Low Level Virtual Machine
 - Infrastructure de compilation et optimisation indépendante du langage et du matériel
 - Front-ends C, C++, Java, Fortran
 - Back-ends X86, X86-64, PowerPC, PowerPC-64, ARM
 - <https://llvm.org/>

Logiciels Message Passing

- MPI : Message Passing Interface
- Bibliothèque de fonctions C, C++ et Fortran
- Standard de communication pour les programmes parallèles s'exécutant sur des clusters
- Communications point à point ou collectives, synchronisation, type de données.
- Open Source
 - MVAPICH : <http://mvapich.cse.ohio-state.edu>
 - OpenMPI : <https://www.open-mpi.org/>

Logiciels Calcul Parallèle

- OpenMP : Open MultiProcessing
- Interface de programmation pour calcul parallèle sur architecture à mémoire partagée (intra nœud)
- API prise en charge par les compilateurs C, C++ et Fortran
 - Bibliothèques, directives, variables d'environnement
 - Implémentations : GCC et Intel Compilers.

Norme	3.0	3.1	4.0
GCC	4.3.1	4.7	4.9.1
INTEL	11	12.1	15
LVM	N/A	3.7	3.7

- Site : <http://www.openmp.org/>

Logiciels de représentation de données HDF5

- Hierarchical Data Format – version 5
- Définit un ensemble de formats pour
 - Structurer des gros volumes de données,
 - Les sauvegarder dans des fichiers
- Deux types de données
 - Dataset : tableaux multi-dimensionnels de données d'un meme type
 - Groups : regroupement d'ensemble de données
- Organisation en structure hiérarchique
- Site : <https://www.hdfgroup.org/>

Logiciels de représentation de données

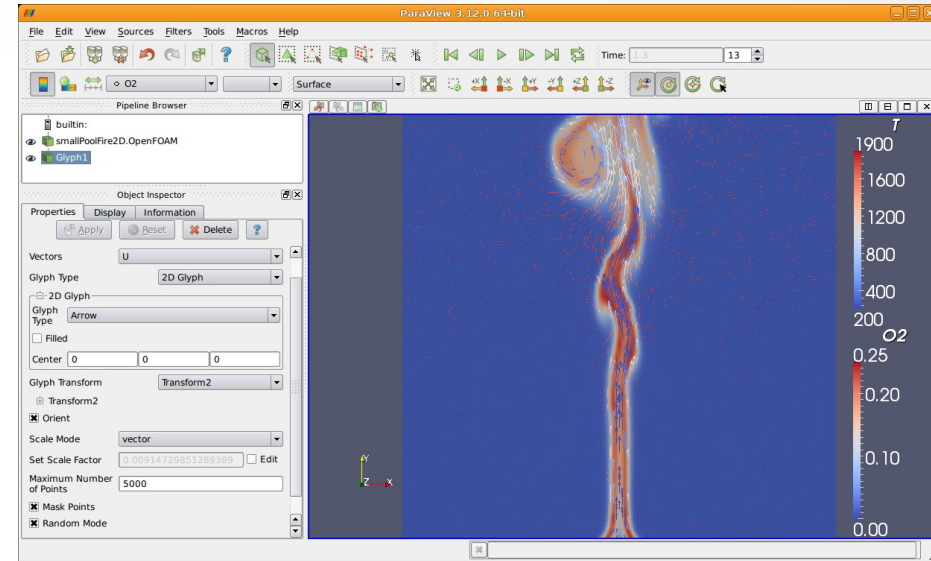
NetCDF

- Network Common Data Form Version 4
- Format de données auto-décrit
- API d'accès aux données stockées sous forme de tableaux
- Support de HDF5
- Version parallèle : PnetCDF
- Site : <https://www.unidata.ucar.edu/software/netcdf/>

Logiciels de visualisation de données

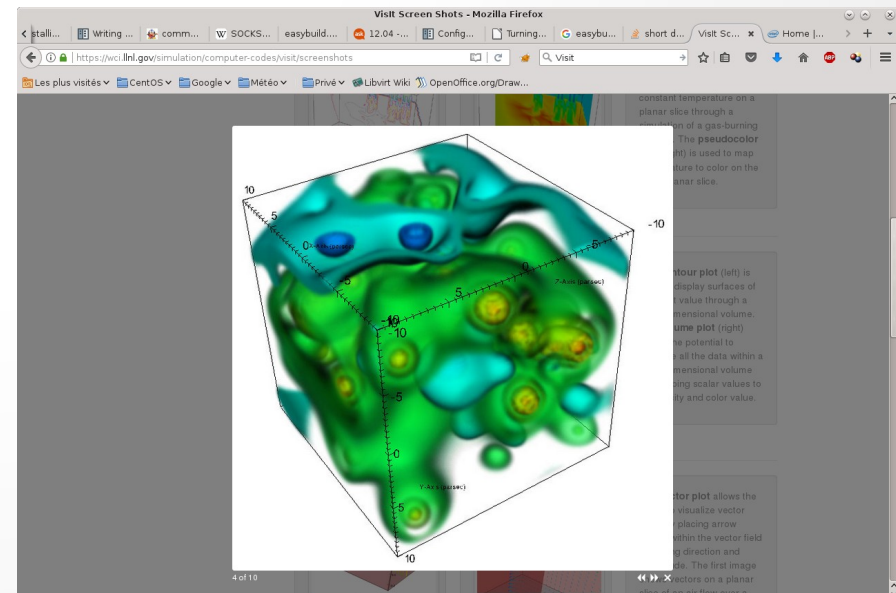
- Paraview

- Outil d'analyse et de visualisation de données
- <https://www.paraview.org/>



- Visit

- <https://wci.llnl.gov/simulation/computer-codes/visit/>
-



Applicatifs

- Codes scientifiques
- VASP : Vienna Ab initio Simulation
 - Code de simulation des matériaux à l'échelle atomique
- NAMD : Nanoscale Molecular Dynamics
- Voir : https://hpc.uni.lu/users/software/legacy_software_2017.html

Rappel de base sur l'environnement

Environnement ?

- Ensemble de variables/valeurs influençant l'exécution des commandes
- Ensemble initialisé au début d'une session utilisateur par le shell (et son père)
- Ensemble propagé de processus parent à processus enfant lors du fork/exec.

```
int execve(const char *filename, char *const argv[],  
           char *const envp[]);
```

Environnement

- Variables d'environnement positionnées par bash
 - SHELL
 - TERM
 - USER, UID
 - PWD, OLDPWD
 - MAIL
 - PATH
 - LANG
 - HOME

Voir son environnement

- /bin/env
- /bin/printenv [variable]

```
[guest22@hpc01 ~]$ /bin/env
XDG_SESSION_ID=606
HOSTNAME=hpc01.c-hpc.pedago.ensie.fr
SHELL=/bin/bash
TERM=xterm-256color
HISTSIZE=1000
QT_GRAPHICSSYSTEM_CHECKED=1
USER=guest22
LS_COLORS=rs=0:di=38;5;27:ln=38;5;51:mh=44;38;5;15:pi=40;38;5;11:so=38;5;13:do=38;5;5:bd=48;5;232;38;5;1
1:cd=48;5;232;38;5;3:or=48;5;232;38;5;9:mi=05;48;5;232;38;5;15:su=48;5;196;38;5;15:sg=48;5;11;38;5;16:ca
=48;5;196;38;5;226:tw=48;5;10;38;5;16:ow=48;5;10;38;5;21:st=48;5;21;38;5;15:ex=38;5;34:*.tar=38;5;9:*.tg
z=38;5;9:*.arc=38;5;9:*.arj=38;5;9:*.taz=38;5;9:*.lha=38;5;9:*.lz4=38;5;9:*.lzh=38;5;9:*.lzma=38;5;9:*.t
lz=38;5;9:*.txz=38;5;9:*.tzo=38;5;9:*.t7z=38;5;9:*.zip=38;5;9:*.z=38;5;9:*.Z=38;5;9:*.dz=38;5;9:*.gz=38;
5;9:*.lrz=38;5;9:*.lz=38;5;9:*.lzo=38;5;9:*.xz=38;5;9:*.bz2=38;5;9:*.bz=38;5;9:*.tbz=38;5;9:*.tbz2=38;5;
9:*.tz=38;5;9:*.deb=38;5;9:*.rpm=38;5;9:*.jar=38;5;9:*.war=38;5;9:*.ear=38;5;9:*.sar=38;5;9:*.rar=38;5;9
:*.alz=38;5;9:*.ace=38;5;9:*.zoo=38;5;9:*.cpio=38;5;9:*.7z=38;5;9:*.rz=38;5;9:*.cab=38;5;9:*.jpg=38;5;13
:*.jpeg=38;5;13:*.gif=38;5;13:*.bmp=38;5;13:*.pbm=38;5;13:*.pgm=38;5;13:*.ppm=38;5;13:*.tga=38;5;13:*.xb
m=38;5;13:*.xpm=38;5;13:*.tif=38;5;13:*.tiff=38;5;13:*.png=38;5;13:*.svg=38;5;13:*.svgz=38;5;13:*.mng=38
;5;13:*.pcx=38;5;13:*.mov=38;5;13:*.mpg=38;5;13:*.mpeg=38;5;13:*.m2v=38;5;13:*.mkv=38;5;13:*.webm=38;5;1
3:*.ogm=38;5;13:*.mp4=38;5;13:*.m4v=38;5;13:*.mp4v=38;5;13:*.vob=38;5;13:*.qt=38;5;13:*.nuv=38;5;13:*.wm
v=38;5;13:*.asf=38;5;13:*.rm=38;5;13:*.rmvb=38;5;13:*.flc=38;5;13:*.avi=38;5;13:*.fli=38;5;13:*.flv=38;5
;13:*.gl=38;5;13:*.dl=38;5;13:*.xcf=38;5;13:*.xwd=38;5;13:*.yuv=38;5;13:*.cgm=38;5;13:*.emf=38;5;13:*.ax
v=38;5;13:*.anx=38;5;13:*.ogv=38;5;13:*.ogx=38;5;13:*.aac=38;5;45:*.au=38;5;45:*.flac=38;5;45:*.mid=38;5
;45:*.midi=38;5;45:*.mka=38;5;45:*.mp3=38;5;45:*.mpc=38;5;45:*.ogg=38;5;45:*.ra=38;5;45:*.wav=38;5;45:*.
axa=38;5;45:*.oga=38;5;45:*.spx=38;5;45:*.xspf=38;5;45:
MAIL=/var/spool/mail/guest22
PATH=/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/usr/lib64/openmpi/bin:/opt/puppetlabs/bin:/
opt/dell/srvadmin/bin:/home/guest22/.local/bin:/home/guest22/bin
PWD=/home/guest22
LANG=en_US.UTF-8
```

Modifier son environnement

- Export WORKDIR=/work/

```
[guest22@hpc01 ~]$ export WORKDIR=/workdir/$USER
[guest22@hpc01 ~]$ printenv WORKDIR
/workdir/guest22
[guest22@hpc01 ~]$ WORKDIR=/scratch/$USER
[guest22@hpc01 ~]$ printenv WORKDIR
/scratch/guest22
[guest22@hpc01 ~]$ /bin/env WORKDIR=/ptmp/guest28 printenv WORKDIR
/ptmp/guest28
[guest22@hpc01 ~]$ printenv WORKDIR
/scratch/guest22
[guest22@hpc01 ~]$ export -n WORKDIR
[guest22@hpc01 ~]$ printenv WORKDIR
[guest22@hpc01 ~]$ echo $WORKDIR
[guest22@hpc01 ~]$
[guest22@hpc01 ~]$ unset WORKDIR
[guest22@hpc01 ~]$ echo $WORKDIR
[guest22@hpc01 ~]$
```

Environnement : man pages

- Man 7 environ
- Man 3 getenv
- Man 3 putenv
- Man 3 setenv
- Man 3 clearenv
- Man 3 execve

Environnement : de parent en enfant

- Trois arguments :
 - **Argument count**
 - **Argument values**
 - **Environment pointer**
- Toujours au moins un argument
 - Le nom du programme
- Arguments
 - Un tableau de chaînes de caractères

```
$ cat main.c
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char **argv, char *envp[])
{
    int i;
    printf("Number of arguments: %d\n", argc);
    for (i=0; i < argc; i++)
        printf("argv[%d]='%s'\n", i, argv[i]);
    exit(argc);
}

$ make main
cc      main.c  -o main
$ ./main
Number of arguments: 1
argv[0]='./main'
$ ./main un deux 3
Number of arguments: 4
argv[0]='./main'
argv[1]='un'
argv[2]='deux'
argv[3]='3'
$ ./main un "deux 3 4" foo
Number of arguments: 4
argv[0]='./main'
argv[1]='un'
argv[2]='deux 3 4'
argv[3]='foo'
$ ./main un "deux 3 4" foo\ bar
Number of arguments: 4
argv[0]='./main'
argv[1]='un'
argv[2]='deux 3 4'
argv[3]='foo bar'
$ echo $?
4
```

Environnement : de parent en enfant

- L'environnement :
 - Un tableau de chaînes de caractères

```
$ pygmentize envp.c
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char **argv, char *envp[])
{
    int i=0;
    while (envp[i]) {
        printf("envp[%d]: '%s'\n", i, envp[i]);
        i++;
    }
    printf("Number of environment vars = %d\n", i);
    exit(i);
}
$ ./envp | head -7 ; echo "..."; ./envp | tail -7
envp[0]: 'XDG_VTNR=1'
envp[1]: 'SSH_AGENT_PID=15539'
envp[2]: 'XDG_SESSION_ID=88'
envp[3]: 'KDE_MULTIHEAD=false'
envp[4]: 'HOSTNAME=delice'
envp[5]: 'IMSETTINGS_INTEGRATE_DESKTOP=yes'
envp[6]: 'SHELL=/bin/bash'
...
envp[60]: 'DISPLAY=:0'
envp[61]: 'QT_PLUGIN_PATH=/usr/lib64/kde4/plugins:/usr/lib/kde4
/plugins:/home/gregoirep/.kde/lib64/kde4/plugins:/usr/lib64/k
de4/plugins/'
envp[62]: 'XDG_CURRENT_DESKTOP=KDE'
envp[63]: 'GTK_IM_MODULE=gtk-im-context-simple'
envp[64]: 'XAUTHORITY=/tmp/kde-gregoirep/xauth-1000-_0'
envp[65]: '._./envp'
Number of environment vars = 66
$ █
```

Voir l'environnement d'un processus

- Le kernel donne une visibilité des processus dans un pseudo filesystem /proc

```
$ F00=bar /bin/sleep 400 &
[1] 3362
$ cat /proc/$!/cmdline | tr '\0' '\n'
/bin/sleep
400
$ cat /proc/$!/environ | tr '\0' '\n' | egrep 'PATH|F00'
F00=bar
PATH=/usr/lib64/qt-3.3/bin:/usr/local/bin:/usr/local/sbin:/usr/bin:/usr/sbin:/bin:/sbin:/home/gregoirep/scripts:/home/gregoirep/.local/bin:/home/gregoirep/bin:/home/gregoirep/scripts
WINDOWPATH=1:1
QT_PLUGIN_PATH=/usr/lib64/kde4/plugins:/usr/lib/kde4/plugins:/home/gregoirep/.kde/lib64/kde4/plugins:/usr/lib64/kde4/plugins/
$ ls /proc/$!
attr          cmdline      environ      io           mem          ns           pagemap      schedstat    stat          timers
autogroup     comm        exe          limits      mountinfo   numa_maps    personality  sessionid    statm         uid_map
auxv          coredump_filter fd           loginuid    mounts      oom_adj      projid_map   setgroups    status        wchan
cgroup        cpuset      fdinfo      map_files   mountstats  oom_score    root         smaps        syscall
clear_refs   cwd         gid_map     maps        net         oom_score_adj sched          stack        task
$ kill $!
[1]+  Terminated                  F00=bar /bin/sleep 400
$ █
```

Mécanisme de bases pour le gestion d'environnement

Variables d'environnement

- PATH (Bash & co)
 - Liste de répertoires séparés par un caractère ':'
 - Le shell parcourt cette liste de gauche à droite
- MANPATH (commande man)
 - Liste de répertoires séparés par un caractère ':'
 - La commande man cherche les pages de manuel dans chacun de ces répertoires.

Variables d'environnement applicatives

- Les applications gèrent souvent leurs propres variables d'environnement.
- <APPLI>[_]HOME
 - Donne le répertoire racine de l'installation de l'application
- <APPLI>[_]PATH :
 - PATH spécifique à l'application
- Exemple : PYTHON :
 - PYTHONPATH, PYHTONHOME

Fonctionnement du loader

Le LOADER Linux ld.so

- Un programme compilé utilise des bibliothèques dynamiques sauf si compilé avec -static.
- ld.so initie l'exécution d'un programme binaire.
- Cherche et charge les bibliothèques dynamiques utilisées par un programme et le lance.
- Résout les dépendances entre les bibliothèques dynamiques.
- Comprend le format ELF

Recherche des librairies par ld.so

- Le nom de la librairie contient un ' /', c'est un path
 - ld.so charge le fichier (relatif ou absolu)
- Le nom de la librairie ne contient pas un ' /' :
 - **DT_RPATH** présent et **DT_RUNPATH** absent :
 - Recherche dans les répertoires donnés par **DT_RPATH**
 - Recherche dans les répertoires donnés par la variable d'environnement **LD_LIBRARY_PATH**
 - **DT_RUNPATH** présent :
 - Recherche dans les répertoires donnés par **DT_RUNPATH**.
 - Recherche dans le cache : **/etc/ld.so.cache**
 - Recherche dans **/lib** puis **/usr/lib**

Recherche des librairies par ld.so

- Les paquets peuvent installer des librairies ailleurs que /lib ou /usr/lib
- La commande ldconfig génère le cache :
 - Fichier /etc/ld.so.conf
 - Répertoires /etc/ld.so.conf.d/*
- Les paquets installent un fichier dans /etc/ld.so.conf.d

```
$ cat /etc/ld.so.conf
include ld.so.conf.d/*.conf
$ ls -l /etc/ld.so.conf.d/
total 32
-rw-r--r--. 1 root root 17 Aug 3 2017 atlas-x86_64.conf
-rw-r--r--. 1 root root 19 Aug 3 2017 dyninst-x86_64.conf
-r--r--r--. 1 root root 63 Dec 5 01:07 kernel-3.10.0-693.11.1.el7.x86_64.conf
-r--r--r--. 1 root root 63 Jan 25 21:28 kernel-3.10.0-693.17.1.el7.x86_64.conf
-r--r--r--. 1 root root 63 Aug 22 2017 kernel-3.10.0-693.el7.x86_64.conf
-rw-r--r--. 1 root root 17 Nov 5 2016 libiscsi-x86_64.conf
-rw-r--r--. 1 root root 17 Aug 4 2017 mariadb-x86_64.conf
-rw-r--r--. 1 root root 22 Jun 10 2014 qt-x86_64.conf
$ cat /etc/ld.so.conf.d/qt-x86_64.conf
/usr/lib64/qt-3.3/lib
$ rpm -qf /etc/ld.so.conf.d/qt-x86_64.conf --scripts
postinstall program: /sbin/ldconfig
postuninstall program: /sbin/ldconfig
$ █
```

Édition de liens

- La commande `/bin/ld` assemble un ensemble de fichiers `*.o` en un programme binaire au format ELF
- Généralement appelée par le compilateur :
 - `Gcc foo.o bar.o -o foobar [-Wl,-rpath=...]`
- L'option `-rpath=dir` crée une section `DT_RPATH`
- L'option `-rpath=dir` utilisée avec l'option `--enable-new-dtags` crée une section `DT_RUNPATH`
- `DT_RPATH` est obsolète (deprecated) et ne devrait plus être utilisé (en théorie)

Variables d'environnement pour le loader

- `LD_LIBRARY_PATH` :
 - liste de répertoires séparés par ':'
 - **Ignoré si le programme a un `setuid/setgid` bit.**
- `LD_PRELOAD`
 - Liste de librairies séparés par ':'
 - Chargées avant toutes les autres
 - Sert à modifier le comportement d'une librairie
- `See` :
 - Man 3 `dlopen dclose dlsym`
 - Man 1 `ld ldd readelf objdump`
 - Man 8 `ld.so ldconfig`

Module

Modules

- Outil de gestion dynamique de l'environnement des utilisateurs,
- Positionne/ supprime les variables d 'environnement nécessaire d'un « logiciel »
- Permet de gérer sur un même système plusieurs versions d 'un même outil
- Un utilisateur peut créer ses propres modules
- Plusieurs générations/concurrents de "module" :
 - Module-C, Lmod, module-tcl

Modules

- Supporte les interpréteurs les plus courants :
 - bash, ksh, zsh, sh, csh, tcsh, fish,
- Intégration de l'outil module dans l'environnement utilisateur :
 - /etc/profile
 - /etc/profile.d/modules.sh
 - /etc/profile.d/modules.csh
- Offre une API pour certains langages :
 - tcl, perl, python, ruby, cmake, R.

Module : charger un module

- module load | add

```
0 ens - Konsole
File Edit View Bookmarks Settings Help
[gregoirep@hpc01 ~]$ module avail
----- /usr/share/Modules/modulefiles -----
dot          module-git  module-info modules      null          use.own
----- /etc/modulefiles -----
mpi/mpich-3.0-x86_64      mpi/mpich-x86_64      mpi/mvapich2-2.2-psm2-x86_64  mpi/mvapich2-x86_64
mpi/mpich-3.2-x86_64      mpi/mvapich2-2.0-x86_64  mpi/mvapich2-2.2-x86_64      mpi/openmpi-x86_64
[gregoirep@hpc01 ~]$ echo $PATH
/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/gregoirep/bin
[gregoirep@hpc01 ~]$ module load mpi/openmpi-x86_64
[gregoirep@hpc01 ~]$ echo $PATH
/usr/lib64/openmpi/bin:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/gregoirep/bin
[gregoirep@hpc01 ~]$
```

Module : voir les modules disponibles

- module available

```
0 ens - Konsole
File Edit View Bookmarks Settings Help
[gregoirep@hpc01 ~]$ module avail

----- /usr/share/Modules/modulefiles -----
dot          module-git  module-info  modules      null          use.own

----- /etc/modulefiles -----
mpi/mpich-3.0-x86_64      mpi/mpich-x86_64      mpi/mvapich2-2.2-psm2-x86_64  mpi/mvapich2-x86_64
mpi/mpich-3.2-x86_64      mpi/mvapich2-2.0-x86_64  mpi/mvapich2-2.2-x86_64      mpi/openmpi-x86_64
[gregoirep@hpc01 ~]$ tree -d /etc/modulefiles
/etc/modulefiles
|-- mpi

1 directory
[gregoirep@hpc01 ~]$
```

Module : afficher les modules chargés

- module list

```
File Edit View Bookmarks Settings Help
() ens - Konsole

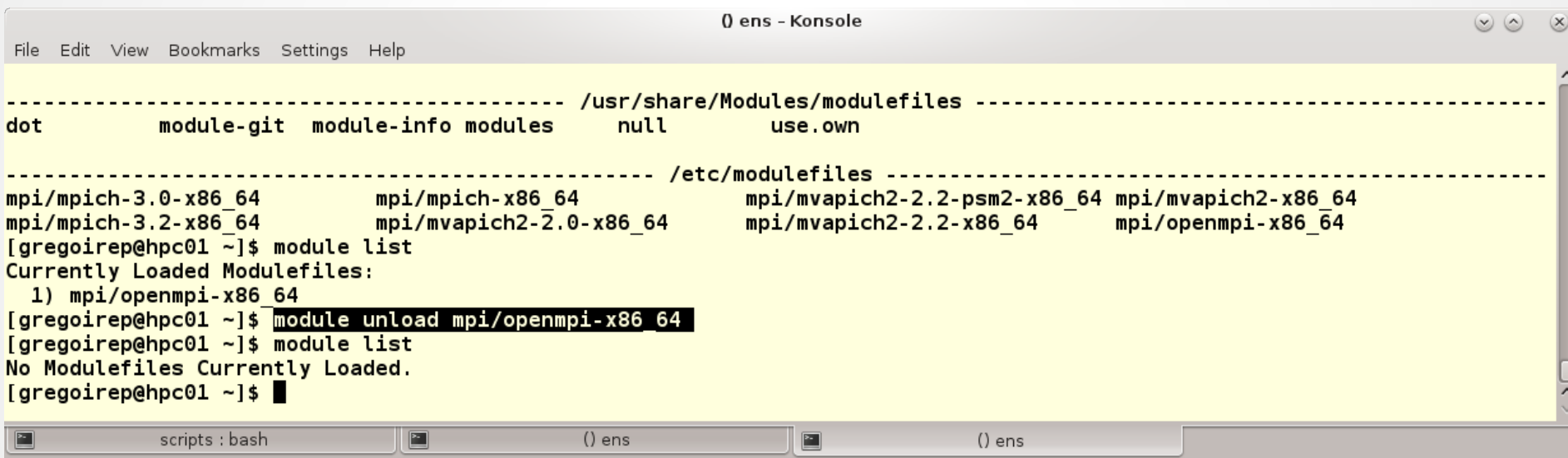
[gregoirep@hpc01 ~]$
[gregoirep@hpc01 ~]$
[gregoirep@hpc01 ~]$ module av

----- /usr/share/Modules/modulefiles -----
dot          module-git  module-info modules  null          use.own

----- /etc/modulefiles -----
mpi/mpich-3.0-x86_64      mpi/mpich-x86_64      mpi/mvapich2-2.2-psm2-x86_64  mpi/mvapich2-x86_64
mpi/mpich-3.2-x86_64      mpi/mvapich2-2.0-x86_64  mpi/mvapich2-2.2-x86_64      mpi/openmpi-x86_64
[gregoirep@hpc01 ~]$ module list
Currently Loaded Modulefiles:
  1) mpi/openmpi-x86_64
[gregoirep@hpc01 ~]$
```

Module : décharger un module

- `module unload | rm`



```
ens - Konsole
File Edit View Bookmarks Settings Help

----- /usr/share/Modules/modulefiles -----
dot          module-git  module-info modules  null          use.own

----- /etc/modulefiles -----
mpi/mpich-3.0-x86_64      mpi/mpich-x86_64      mpi/mvapich2-2.2-psm2-x86_64  mpi/mvapich2-x86_64
mpi/mpich-3.2-x86_64      mpi/mvapich2-2.0-x86_64  mpi/mvapich2-2.2-x86_64      mpi/openmpi-x86_64
[gregoirep@hpc01 ~]$ module list
Currently Loaded Modulefiles:
  1) mpi/openmpi-x86_64
[gregoirep@hpc01 ~]$ module unload mpi/openmpi-x86_64
[gregoirep@hpc01 ~]$ module list
No Modulefiles Currently Loaded.
[gregoirep@hpc01 ~]$
```

Module : échanger deux modules

- module switch | swap

```
0 ens - Konsole
File Edit View Bookmarks Settings Help

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Apr  4 18:32:15 2018 from 132.166.86.21
philippe.gregoire@pedago-ens:~$ ssh hpc
Last login: Wed Apr  4 18:40:33 2018 from 172.30.140.232
[gregoirep@hpc01 ~]$ module load mpi/openmpi-x86_64
[gregoirep@hpc01 ~]$ module unload mpi/openmpi-x86_64
[gregoirep@hpc01 ~]$ module avail

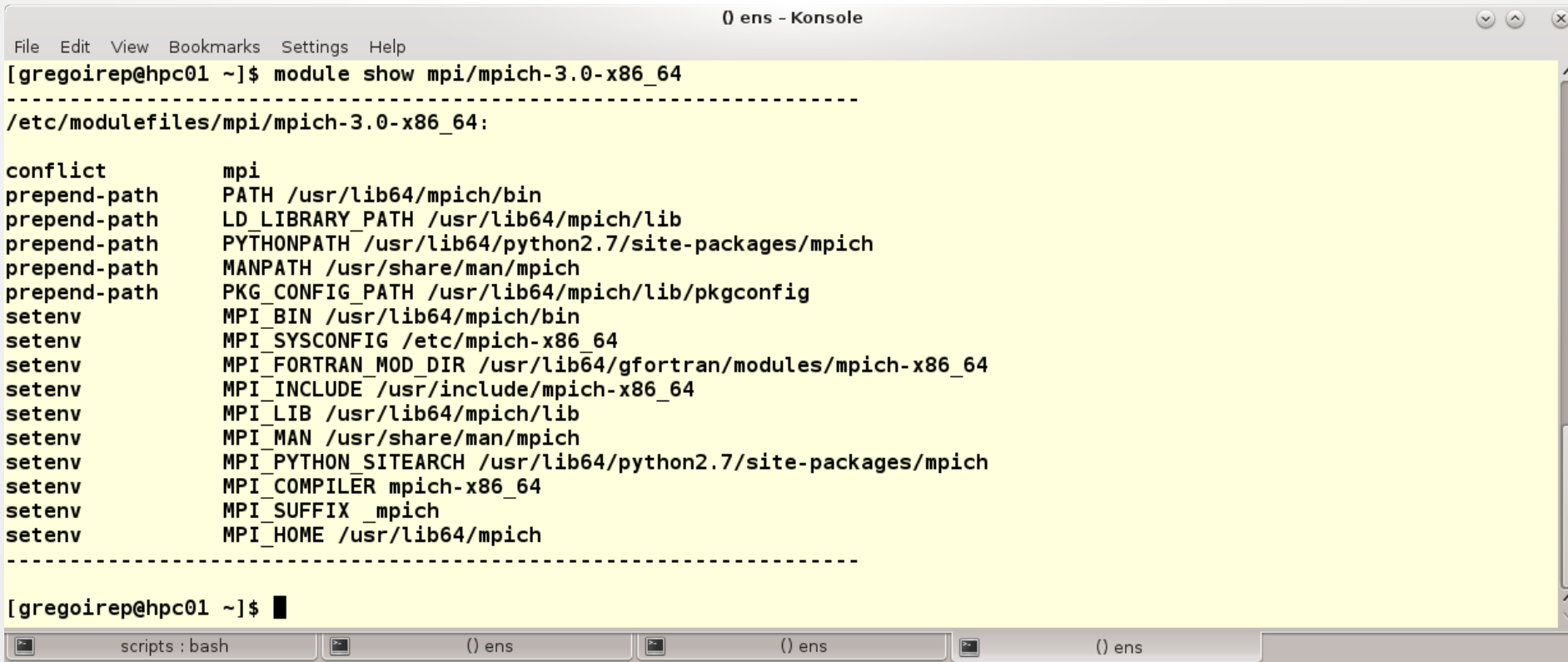
----- /usr/share/Modules/modulefiles -----
dot          module-git  module-info modules    null          use.own

----- /etc/modulefiles -----
mpi/mpich-3.0-x86_64      mpi/mpich-x86_64      mpi/mvapich2-2.2-psm2-x86_64  mpi/mvapich2-x86_64
mpi/mpich-3.2-x86_64      mpi/mvapich2-2.0-x86_64  mpi/mvapich2-2.2-x86_64      mpi/openmpi-x86_64
[gregoirep@hpc01 ~]$ module load mpi/openmpi-x86_64
[gregoirep@hpc01 ~]$ echo $PATH
/usr/lib64/openmpi/bin:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/gregoirep/bin
[gregoirep@hpc01 ~]$ module switch mpi/openmpi-x86_64 mpi/mvapich2-2.0-x86_64
[gregoirep@hpc01 ~]$ echo $PATH
/usr/lib64/mvapich2/bin:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/gregoirep/bin
[gregoirep@hpc01 ~]$ █

scripts : bash      () ens      () ens      () ens
```

Module : s'informer sur un module

- module show | display
- module whatis
- module apropos



```
0 ens - Konsole
File Edit View Bookmarks Settings Help
[gregoirep@hpc01 ~]$ module show mpi/mpich-3.0-x86_64
-----
/etc/modulefiles/mpi/mpich-3.0-x86_64:

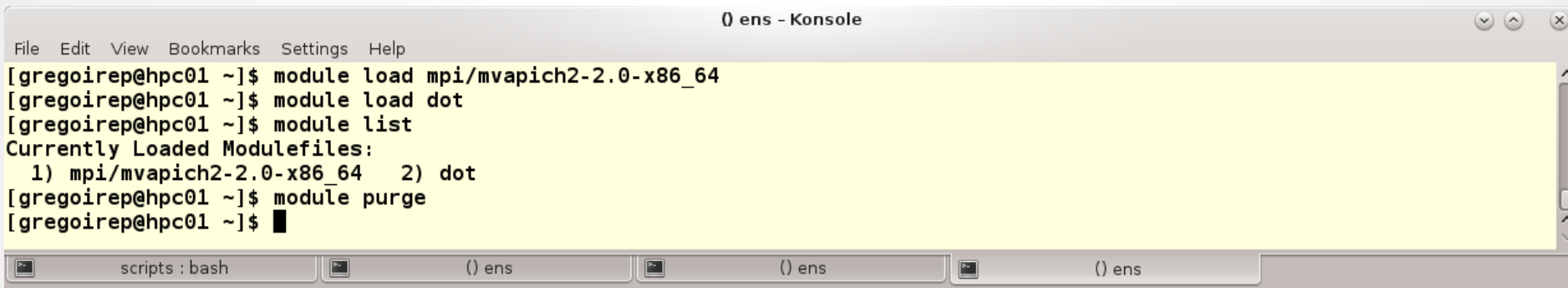
conflict          mpi
prepend-path      PATH /usr/lib64/mpich/bin
prepend-path      LD_LIBRARY_PATH /usr/lib64/mpich/lib
prepend-path      PYTHONPATH /usr/lib64/python2.7/site-packages/mpich
prepend-path      MANPATH /usr/share/man/mpich
prepend-path      PKG_CONFIG_PATH /usr/lib64/mpich/lib/pkgconfig
setenv            MPI_BIN /usr/lib64/mpich/bin
setenv            MPI_SYSCONFIG /etc/mpich-x86_64
setenv            MPI_FORTRAN_MOD_DIR /usr/lib64/gfortran/modules/mpich-x86_64
setenv            MPI_INCLUDE /usr/include/mpich-x86_64
setenv            MPI_LIB /usr/lib64/mpich/lib
setenv            MPI_MAN /usr/share/man/mpich
setenv            MPI_PYTHON_SITEARCH /usr/lib64/python2.7/site-packages/mpich
setenv            MPI_COMPILER mpich-x86_64
setenv            MPI_SUFFIX _mpich
setenv            MPI_HOME /usr/lib64/mpich
-----

[gregoirep@hpc01 ~]$
```

The screenshot shows a terminal window titled "0 ens - Konsole". The terminal displays the command `module show mpi/mpich-3.0-x86_64` and its output. The output is a list of environment variables and their values for the `mpi/mpich-3.0-x86_64` module, separated by a dashed line. The variables include `conflict`, `prepend-path` for `PATH`, `LD_LIBRARY_PATH`, `PYTHONPATH`, `MANPATH`, `PKG_CONFIG_PATH`, `setenv` for `MPI_BIN`, `MPI_SYSCONFIG`, `MPI_FORTRAN_MOD_DIR`, `MPI_INCLUDE`, `MPI_LIB`, `MPI_MAN`, `MPI_PYTHON_SITEARCH`, `MPI_COMPILER`, `MPI_SUFFIX`, and `MPI_HOME`. The terminal window has a menu bar with "File", "Edit", "View", "Bookmarks", "Settings", and "Help". The bottom of the window shows a tab bar with "scripts : bash" and three empty tabs labeled "() ens".

Module : nettoyer son environnement

- module purge



```
File Edit View Bookmarks Settings Help
[gregoirep@hpc01 ~]$ module load mpi/mvapich2-2.0-x86_64
[gregoirep@hpc01 ~]$ module load dot
[gregoirep@hpc01 ~]$ module list
Currently Loaded Modulefiles:
  1) mpi/mvapich2-2.0-x86_64  2) dot
[gregoirep@hpc01 ~]$ module purge
[gregoirep@hpc01 ~]$
```

The screenshot shows a terminal window with a menu bar (File, Edit, View, Bookmarks, Settings, Help) and a title bar ('() ens - Konsole'). The terminal output shows the user loading two modules, listing them, and then purging them. The prompt is '[gregoirep@hpc01 ~]\$' and the cursor is at the end of the last line.

Module : dans un job ?

- Module est initialisé pour les sessions interactives
- Pour un job batch : charger le fichier `/etc/profile/modules.sh`
- Slurm peut le faire dans ses prologues

Module : toolchain

- Toolchain : ensemble d'outils qui forment un ensemble cohérent de développement.
- Sont utilisés pour compiler les autres packages.
- Les packages compilés avec le même toolchain (toolchain+ version) peuvent travailler ensemble.
 - Pas de conflit d'environnement
 - Pas d'incompatibilité de librairies

Module : toolchain Intel

- Intel toolchain :
 - Compilateurs intel c, c++ et fortran : icc, ifort
 - Librairie MPI intelMPI : mpicc, mpifort
 - Librairie mathématique Intel : MKL

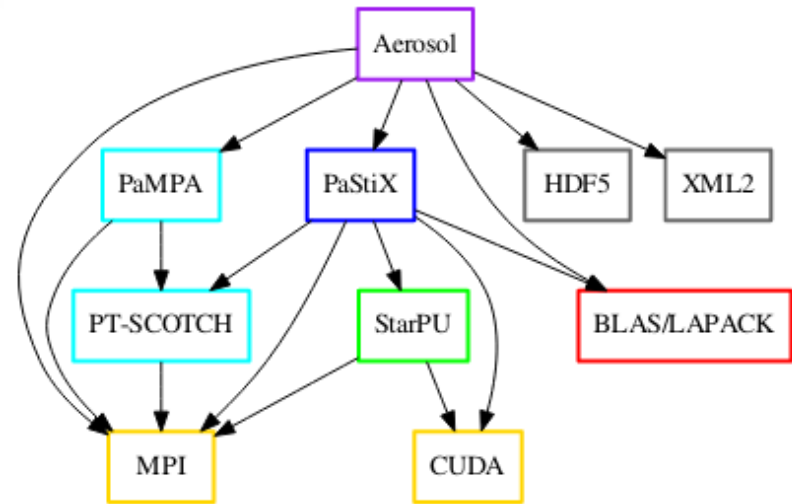
Module : toolchain Foss

- Foss toolchain :
 - Compilateurs GNU : gcc, gfort
 - Librairie MPI OpenMPI : mpicc, mpifort
 - Bibliothèques mathématiques : OpenBlas, Lapack, ScaLapack, FFTW

Comment installer efficacement
de nombreux logiciels ?

Problématique

- L'OpenSource facilite l'utilisation de composants logiciels
- En R&D, les prototypes nécessitent de nombreux essais
- Les logiciels HPC doivent s'exécuter dans des environnements différents
- Importance du choix des composants, de leurs options
- Importance de la reproductibilité
- Importance de la tracabilité



Problématique

- Interdépendances des logiciels OpenSource
 - Composants haut niveau
 - Bibliothèques numériques
 - Version de langage (python, C++)
 - Chaînes de développement
 - Outils divers
- La demande est toujours plus forte
 - Les ressources restent les mêmes
 - Activé chronophage

Installation multi-version par rpm
Une solution ?

Installation de plusieurs rpm

- Rpm ne permet pas l'installation de plusieurs versions (sauf kernel)
- La version la plus récente remplace la précédente
- Solution
 - Changer le nom du paquet pour inclure la version :
 - Newname = pkgname+pkgvversion
 - Changer les répertoires d'installation
- Exemple : python
 - Version 2 : Rpm python-2.X.Y install standard
 - Version 3 : Rpm python3-3.Z.T

Installation de plusieurs rpm

- Configurer yum :
 - Installonlypkgs : donne la liste des paquets à ne jamais mettre à jour
 - installonly_limit : donne le nombre maximum de versions à installer (pour un paquet)
 - Attention quand la limite est atteinte !
 - Man 8 yum.conf

Méthode RPM

- Trop limitée
- Nécessite une modification des specfiles
- Nécessite les droits root
- Impossible de déléguer l'activité

Framework de génération de logiciels

EasyBuild

- Automatisation de génération/installation de logiciels scientifiques
- Flexible et configurable (build recipes)
- Résout automatiquement les dépendances
- Génération automatique des modules associés
- Bonne documentation,
- Large communauté d'utilisateurs
- Parfois un peu compliqué à customiser
- Concurrent : SPACK

EasyBuild : Terminologie

- EasyBuild framework
 - Moteur d'EasyBuild
 - Enchaîne les étapes nécessaires pour l'installation d'un logiciel
- EasyBlock :
 - Module python, greffon (plugin) dans le framework EasyBuild
 - Définit comment générer et installer un logiciel
- Easyconfig :
 - Fichiers de configuration (*.eb) pour l'installation d'une version donnée d'un logiciel
- Toolchain
 - Ensemble cohérent d'outils de compilation et librairies

EasyBuild : commandes de base

- `eb -list-easyblocks` – lists available easyblocks
 - | -- ConfigureMake
 - | | -- CMakeMake
 - | | -- EB_GROMACS
- `eb -list-toolchains` – lists available toolchains
 - goolf: BLACS, FFTW, GCC, OpenBLAS, OpenMPI, ScaLAPACK
- `eb -S pkgname` – search for package easyconfig
 - `eb -S GROMACS`
 - * \$CFG\$1/GROMACS-4.6.5-goolf-1.4.10-hybrid.eb
- `eb pkgname -r` –install package with dependencies (- r)

Un easyblock pour construire Flex

```
name = 'flex'
version = '2.6.4'

homepage = 'http://flex.sourceforge.net/'

description = """
Flex (Fast Lexical Analyzer) is a tool for generating scanners. A scanner,
sometimes called a tokenizer, is a program which recognizes lexical patterns
in text.
"""

toolchain = {'name': 'dummy', 'version': ''}
toolchainopts = {'pic': True}

source_urls = ['https://github.com/westes/flex/releases/download/v%(version)s/']
sources = [SOURCELOWER_TAR_GZ]
checksums = ['e87aae032bf07c26f85ac0ed3250998c37621d95f8bd748b31f15b33c45ee995']

builddependencies = [
    ('Bison', '3.0.4'),
    ('help2man', '1.47.4'),
]

dependencies = [
    ('M4', '1.4.18'),
]

moduleclass = 'lang'
```

Encore des easyblocks !

```
[gregoirep@hpc01 easyconfigs]$ ls -l w/WRF
total 192
-rw-r--r--. 1 gregoirep teachers 1049 Apr  4 21:15 WRF-3.3.1-goolf-1.4.10-dmpar.eb
-rw-r--r--. 1 gregoirep teachers  970 Apr  4 21:15 WRF-3.3.1-ictce-5.3.0-dmpar.eb
-rw-r--r--. 1 gregoirep teachers  445 Apr  4 21:15 WRF-3.3.1_GCC-build_fix.patch
-rw-r--r--. 1 gregoirep teachers  634 Apr  4 21:15 WRF-3.3.1_known_problems.patch
-rw-r--r--. 1 gregoirep teachers 1091 Apr  4 21:15 WRF-3.4-goolf-1.4.10-dmpar.eb
-rw-r--r--. 1 gregoirep teachers 1047 Apr  4 21:15 WRF-3.4-ictce-5.3.0-dmpar.eb
-rw-r--r--. 1 gregoirep teachers 1046 Apr  4 21:15 WRF-3.4.1-ictce-5.3.0-dmpar.eb
-rw-r--r--. 1 gregoirep teachers 9122 Apr  4 21:15 WRF-3.4.1_netCDF-Fortran_separate_path.patch
-rw-r--r--. 1 gregoirep teachers 7035 Apr  4 21:15 WRF-3.4_known_problems.patch
-rw-r--r--. 1 gregoirep teachers 1059 Apr  4 21:15 WRF-3.5-goolf-1.4.10-dmpar.eb
-rw-r--r--. 1 gregoirep teachers 1058 Apr  4 21:15 WRF-3.5-ictce-5.3.0-dmpar.eb
-rw-r--r--. 1 gregoirep teachers 1009 Apr  4 21:15 WRF-3.5.1-goolf-1.4.10-dmpar.eb
-rw-r--r--. 1 gregoirep teachers 1009 Apr  4 21:15 WRF-3.5.1-goolf-1.5.14-dmpar.eb
-rw-r--r--. 1 gregoirep teachers 1008 Apr  4 21:15 WRF-3.5.1-ictce-5.3.0-dmpar.eb
-rw-r--r--. 1 gregoirep teachers 1776 Apr  4 21:15 WRF-3.5_known_problems.patch
-rw-r--r--. 1 gregoirep teachers 9442 Apr  4 21:15 WRF-3.5_netCDF-Fortran_separate_path.patch
-rw-r--r--. 1 gregoirep teachers 1055 Apr  4 21:15 WRF-3.6.1-CrayGNU-2015.06-dmpar.eb
-rw-r--r--. 1 gregoirep teachers 1055 Apr  4 21:15 WRF-3.6.1-CrayGNU-2015.11-dmpar.eb
-rw-r--r--. 1 gregoirep teachers 1057 Apr  4 21:15 WRF-3.6.1-CrayIntel-2015.11-dmpar.eb
-rw-r--r--. 1 gregoirep teachers 1047 Apr  4 21:15 WRF-3.6.1-foss-2015a-dmpar.eb
-rw-r--r--. 1 gregoirep teachers 1048 Apr  4 21:15 WRF-3.6.1-intel-2014b-dmpar.eb
-rw-r--r--. 1 gregoirep teachers 1048 Apr  4 21:15 WRF-3.6.1-intel-2015a-dmpar.eb
-rw-r--r--. 1 gregoirep teachers 2437 Apr  4 21:15 WRF-3.6.1_known_problems.patch
-rw-r--r--. 1 gregoirep teachers 9790 Apr  4 21:15 WRF-3.6.1_netCDF-Fortran_separate_path.patch
-rw-r--r--. 1 gregoirep teachers 1072 Apr  4 21:15 WRF-3.8.0-intel-2016a-dmpar.eb
-rw-r--r--. 1 gregoirep teachers 1070 Apr  4 21:15 WRF-3.8.0-intel-2016b-dmpar.eb
-rw-r--r--. 1 gregoirep teachers 4443 Apr  4 21:15 WRF-3.8.0_known_problems.patch
-rw-r--r--. 1 gregoirep teachers 10926 Apr  4 21:15 WRF-3.8.0_netCDF-Fortran_separate_path.patch
-rw-r--r--. 1 gregoirep teachers 5712 Apr  4 21:15 WRF_FC-output-spec_fix.patch
-rw-r--r--. 1 gregoirep teachers 8953 Apr  4 21:15 WRF_netCDF-Fortran_separate_path.patch
-rw-r--r--. 1 gregoirep teachers 1201 Apr  4 21:15 WRF_no-GCC-graphite-loop-opts.patch
-rw-r--r--. 1 gregoirep teachers  684 Apr  4 21:15 WRF_parallel_build_fix.patch
-rw-r--r--. 1 gregoirep teachers 9944 Apr  4 21:15 WRF_tests_limit-runtimes.patch
[gregoirep@hpc01 easyconfigs]$ █
```

EasyBuild : Pro & Cons

- Pros

- Conçu pour l'écosystème HPC
- Supporte les logiciels propriétaires : Intel, Nvidia, ...
- Large communauté : Beaucoup easyblock disponibles
- Ré-utilisabilité
- Gestion des dépendances automatiques

- Cons

- Accès Internet (bootstrap)
- Gère mal la combinatoire
- 1 fichier par combinaison software / toolchain

Evolution de l'architecture

- EasyBuild facilite la génération/installation des logiciels
 - Mutualisation des procédures,
 - Mutualisation des produits générés.
- La génération des environnements utilisateurs est chronophage :
 - Demandes d'installation de nouvelles versions
 - Gestion de la combinatoire avec les toochains
 - Gestion des conflits lors de l'utilisation
 - Suivi des logiciels installés

Evolution de l'architecture

- Les utilisateurs travaillent dans de nombreux environnements
 - Dans leurs entreprises
 - Dans plusieurs centres de calcul
- Un environnement applicatif =
 - Une distribution Linux
 - Un ensemble de logiciels
 - L'application

Evolution de l'architecture

- Solution : Les utilisateurs fournissent leur application avec l'environnement adéquat.
 - Containers (Docker)
 - Machines virtuelles
- A eux de construire leurs piles logicielles :-)

