

Installing Software on WestGrid Systems

Belaid Moa

Compute Canada/WestGrid/University Systems

University of Victoria

Topics

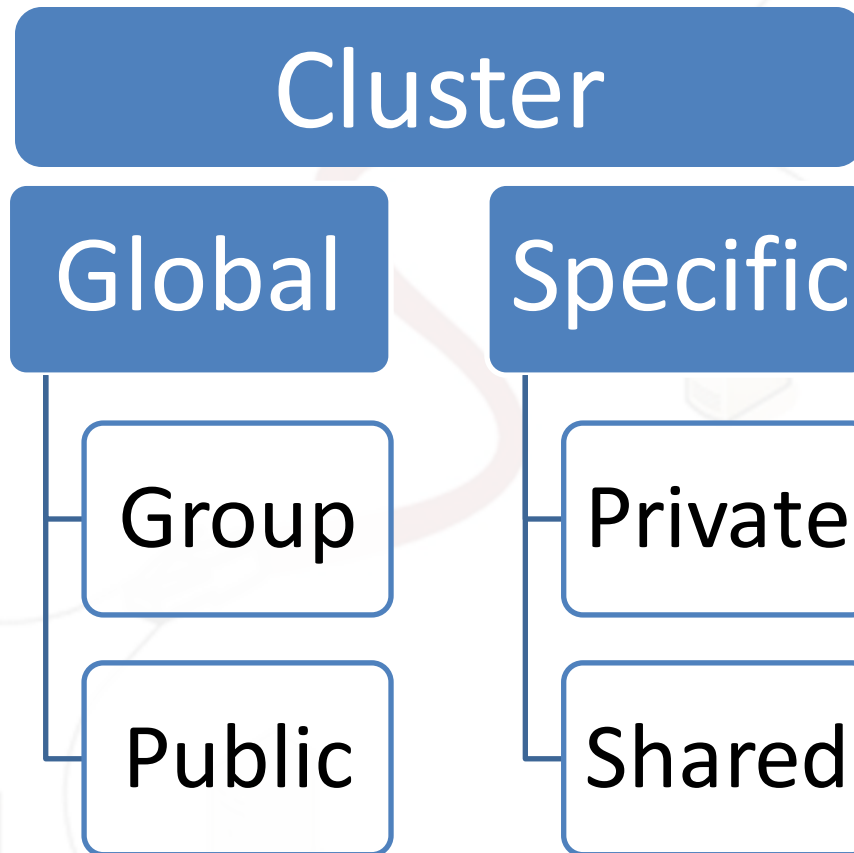
- Global software installation on a WestGrid system
- User-based software installation:
 - Who, When, Where, How
- Specific software package customization
- Software installation challenges faced by WestGrid users
- Effective guidelines to address these challenges

Software on WestGrid

- Each cluster has its own independent software stack
- The list of software available on WestGrid systems is available at:

<http://www.westgrid.ca/support/software>

Software on WestGrid



Global Software on WestGrid

Group

- Terms/conditions
- Join the group
- [Load modules]

Public

- Available to all
- [Load modules]

Location

- /global/software
- /usr/local
- ?



Global Software Installation on WestGrid

- Who:
 - a site admin and/or
 - a support staff
- The challenges faced:
 - The list of software to install is usually long
 - Expect a long waiting time, especially when the software is used by a single user

**Better to install the software needed yourself in
your own home directory**

User-Specific Software on WestGrid

Private

- Specific to a single user
- Under his home directory
- Set environment variables, especially PATH

Shared

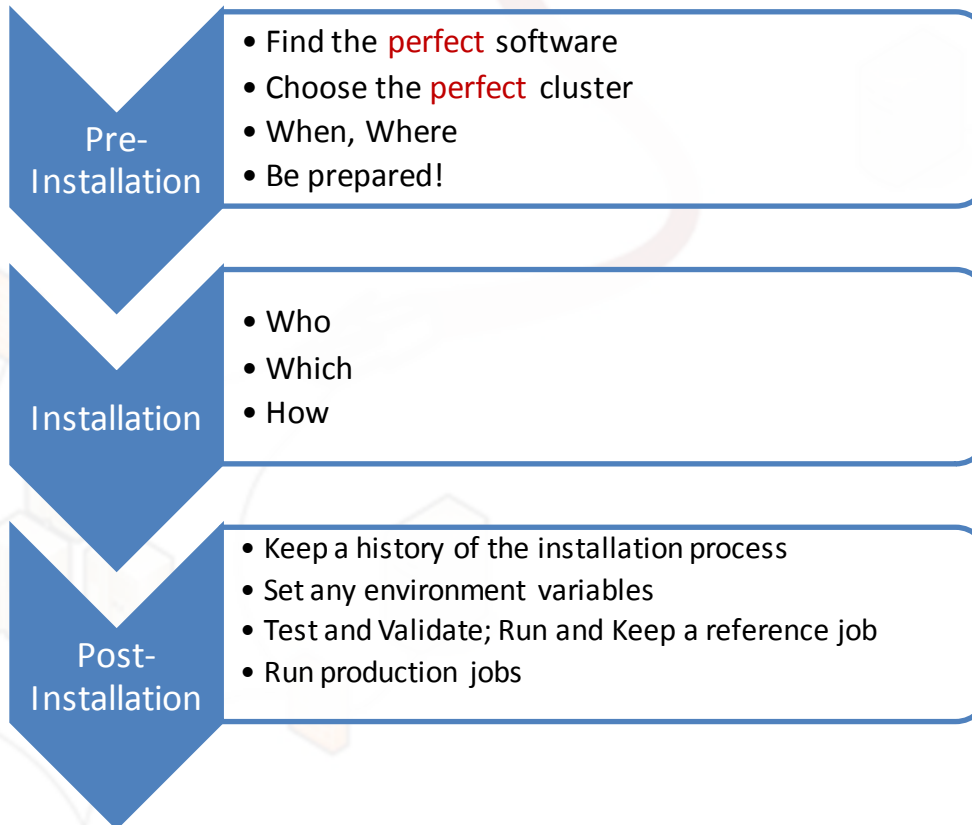
- Ask for a UNIX group
- Ask for users to be added
- Change the permissions on the software folder for the group

Location

- Source: /global/scratch/username/source
- Install: /home/username/software

User-Specific Software Installation on WestGrid

- WestGrid allows users to install software under their home directories
- The installation process:



User-Specific Software Installation on WestGrid

Who

- You
- Support staff are ready to help
 - just ask support@westgrid.ca

When

- Not available globally
 - check the software page
- No other globally available software is good enough
- Very specific to you/closed source
- Cannot wait for a global installation

Where

- Choose the right cluster
 - Serial, openMP/threads, MPI
 - Memory
 - Dependencies
- Put the source in /global/scratch and the binaries in /home

Specific Software Installation on WestGrid



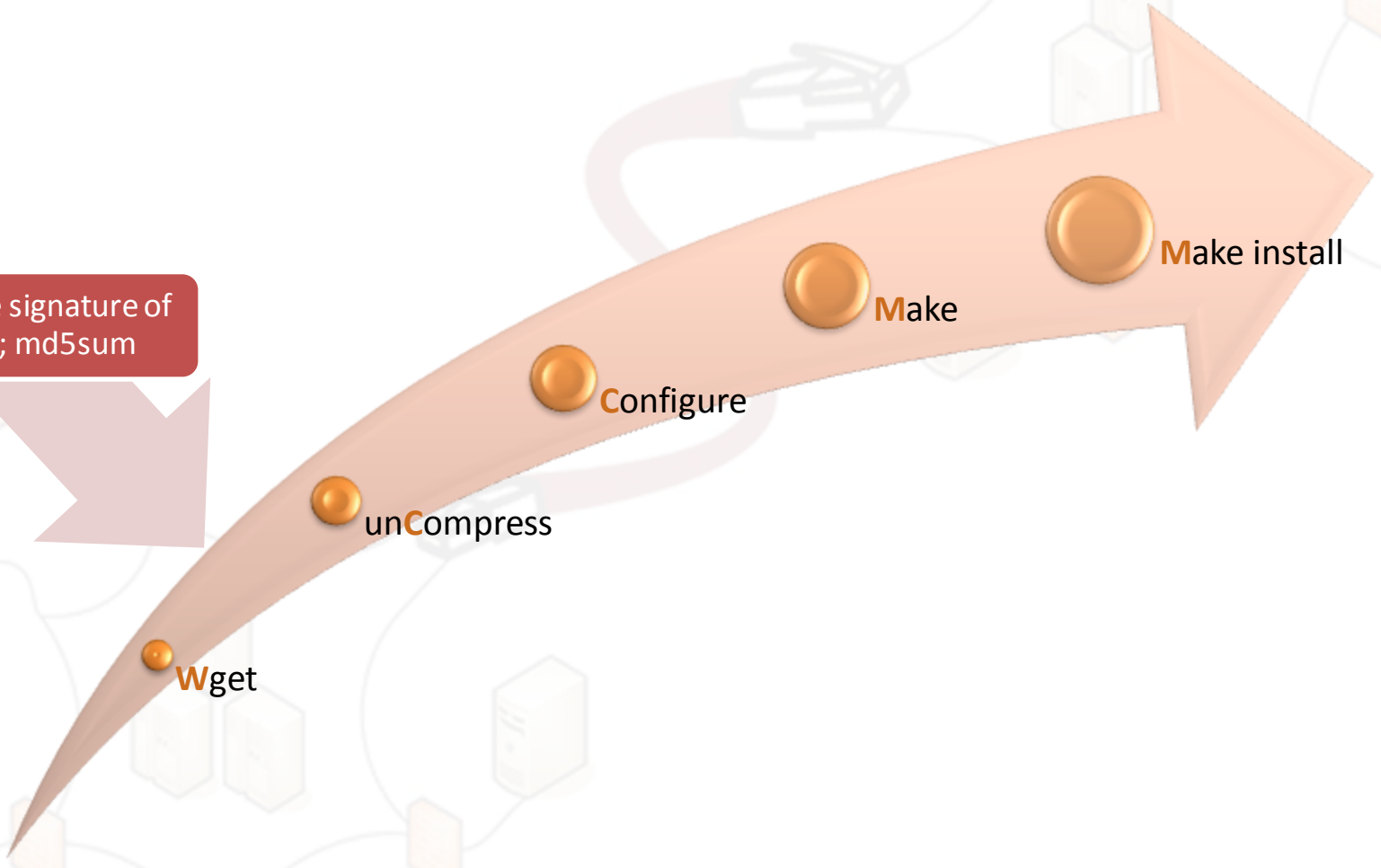
Which: Binary vs. Source



How: WCCMM

Specific Software Installation: How

Verify the signature of the file; md5sum



Specific Software Installation: cufflinks

- Set PATH
- Run cufflinks

wget

unCompress

Configure

Make

Make install

```
tar xzvf cufflinks-1.2.0.Linux_x86_64.tar.gz
```

```
wget http://cufflinks.cbcb.umd.edu/downloads/cufflinks-1.2.0.Linux_x86_64.tar.gz
```

Specific Software Installation: ALMOST

- Set PATH
- Run almost

wget

unCompress

Configure

Make

Make install

make

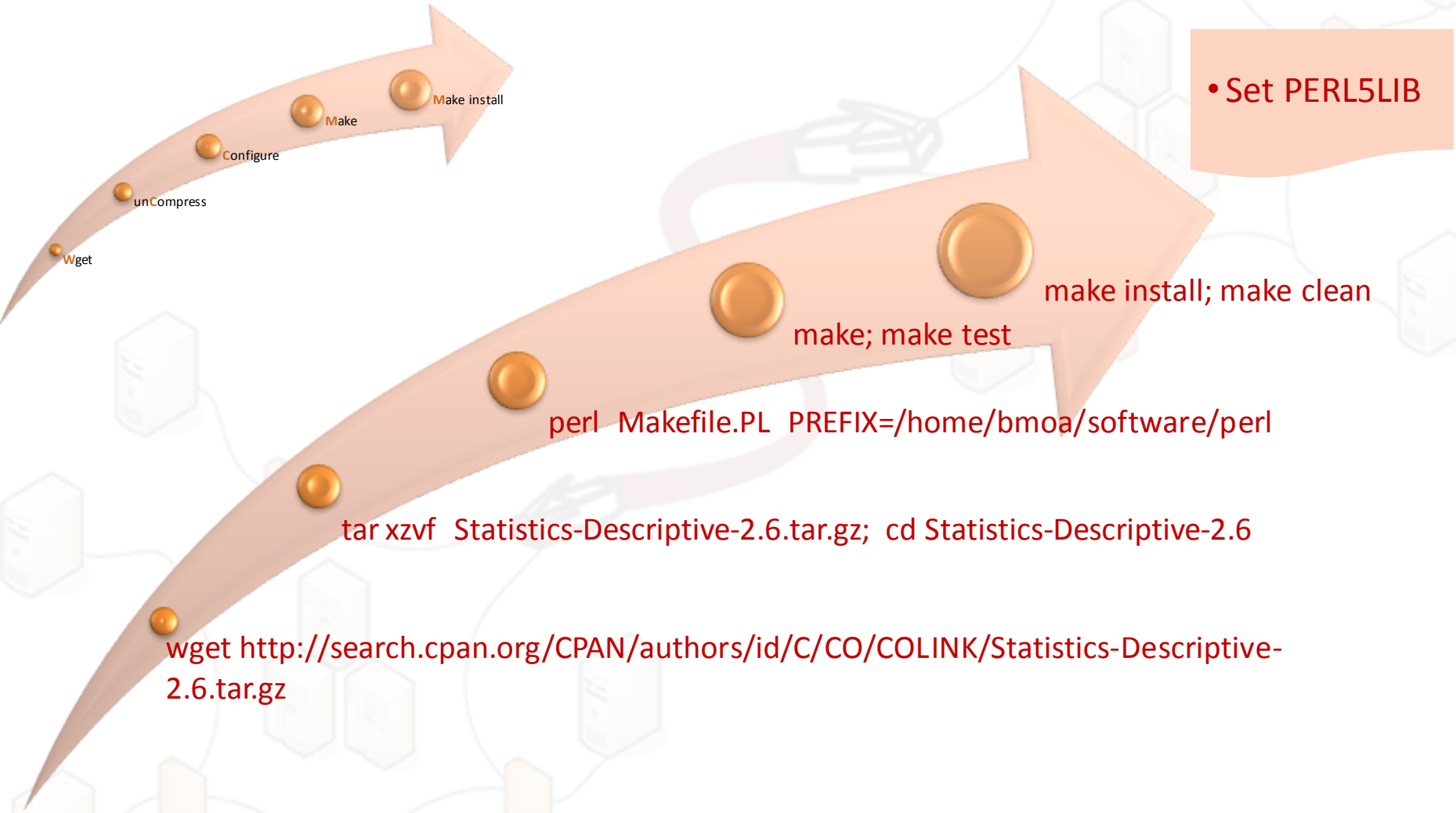
make install

```
./configure --prefix=/home/bmoa/software/almost-1.0.4
```

```
tar xzvf almost-1.0.4.tar.gz; cd almost-1.0.4
```

```
wget http://www-almost.ch.cam.ac.uk/site/downloads/almost-1.0.4.tar.gz
```

Specific Software Installation: A Perl Package

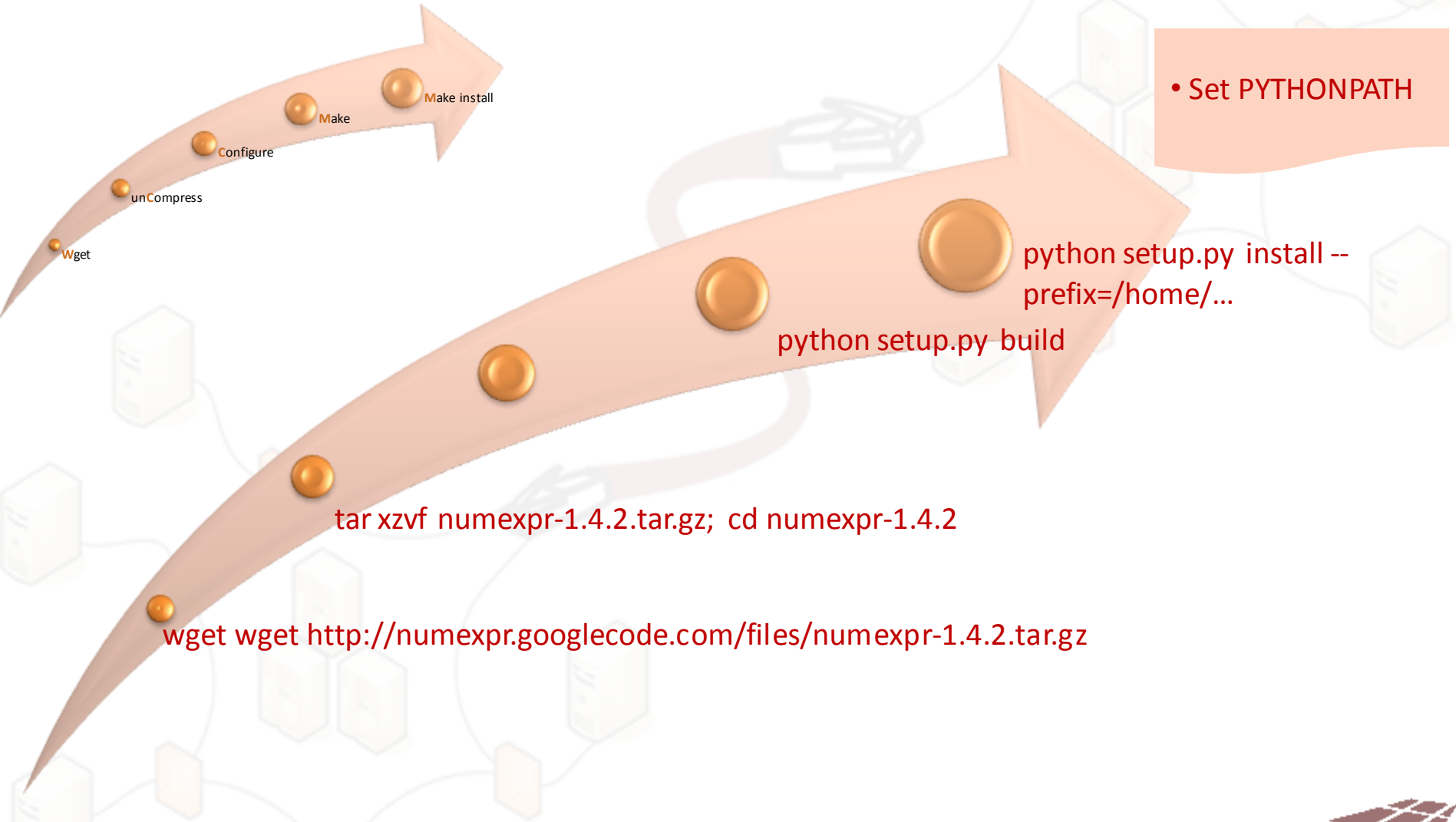


More ...

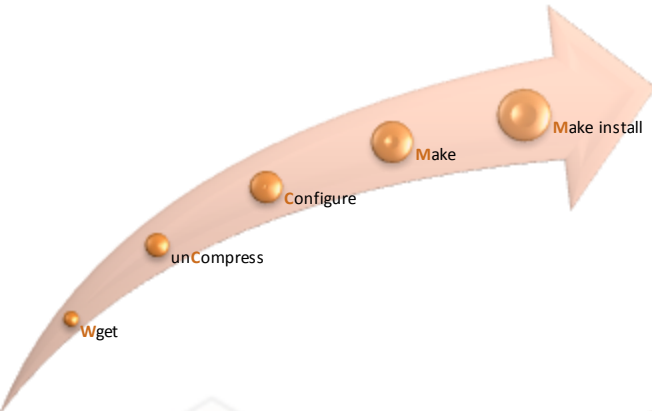
```
litai05:/home/bmoa$ GTK_CFLAGS=-I/home/bmoa/software/include/gtk-2.0 GTK_LIBS=-L/home/bmoa/software/lib CFLAGS="-I/home/bmoa/software/include -I/home/bmoa/software/include/gtk-2.0" FFTW3_CFLAGS=-I/global/software/fftw-3.2.2/include FFTW3_LIBS=-L/global/software/fftw-3.2.2/lib LDFLAGS="-L/home/bmoa/software/lib -L/global/software/netcdf-4.1.2/lib -L/home/bmoa/software/readline-6.2/shlib" LIBS=-lcurses ./configure --prefix=/home/bmoa/software/Galacticus/Tools --with-boost=/home/bmoa/software/boost_1_35_0 --with-mpi=/global/software/openmpi-1.4.2
```

```
litai05:/home/bmoa$ make CXXFLAGS="-I/global/software/netcdf-4.1.2/include -I/home/bmoa/software/include -I/usr/include -O3" CFLAGS="-O3"
```

Specific Software Installation: A Python Module



Specific Software Installation: An R Package



- Set R_LIBS_USER

```
Rscript -e "install.packages('evd',  
lib='~/R/library',  
contriburl=contrib.url('http://cran.r-  
project.org/'))"
```

```
Rscript -e  
"source('http://bioconductor.org/biocLite.R');  
biocLite(lib='~/R/library')"
```

Specific Software Installation: How

- Get the software
 - Usually a tar ball (tar.gz, tar.bz2, .tgz)
 - Do not forget to check and get any patch
- Uncompress the tar ball
 - `tar xvf[z,j] tar_ball`
- [Apply any patch]
- [Flags] `./configure [options]`
 - Read the README and/or INSTALL files; look in the doc/ folder
 - `./configure --help`
- Make [options]
- [Make test] [make check]
- Make install

Configure

- Gather information about the machine
- Check for dependencies/things needed
- Yes/No messages

Make

- Requires a Makefile
- Targets, rules and dependencies

Make test/check [no always available]

- Test whether the package is built and runs properly

Make Install

- Copy the executables, scripts, includes and libraries to their the prefix location

Configure

- Detect information about the system
- Do requirements testing
- Report the available/missing requirements/dependencies
- Helps in customizing the build process
- [flags] ./configure [options]
 - ./configure –help
- **Very important option: --prefix**
 - Allows you to install the software under a specific folder
 - ./configure --prefix=WHERE_TO_INSTALL_IT

The main outcome: makefiles needed by make

Make

- A utility to simplify the building and the maintenance of programs, especially large ones
 - A programming utility
- Make reads and processes instructions from a text file
 - By default the file used is makefile/Makefile
- The instructions consists of targets, dependencies and rules, and structured as shown below

```
cxx=icpc -Wall
hello: hello.o
    $(cxx) hello.o -o $@
hello.o: hello.cpp hello.h
    $(cxx) -c hello.cpp
```

```
target: dependencies
      commands
```

```
make hello
```

Make

- The main purpose of make is to
 - Compile the code
 - Generate the executables and the libraries
- **The main outcome: executables and libraries**
- **make test/check:** runs the target test/check
 - Run few tests on the executable
- **make install:** runs the target install

The main outcome: Copy headers, scripts, libraries, executables into the Prefix directory

Challenges

- Which software to use?
 - MPI > OpenMP/Threads > serial
 - Efficient; uses modest amount of resources
- Which cluster to use?
 - Has the software
 - Matches the requirements of your jobs
 - procs, mem, walltime, storage, etc.

Pre-installation



Challenges

- Headers/libraries not found
 - Make sure they are installed
 - Point to the right location using [...]FLAGS/LIBS
 - Check whether the existing libraries are compatible with the software
- Error not clear
 - Read config.log
 - Find which test code failed

Installation:
Configure



Challenges

- Warnings
 - Usually Ignore, especially with Intel compilers
- Not found error - compilation
 - Header files missing
 - Include folders using [...]FLAGS with -I
- Undefined reference error – linking
 - Libraries are missing: install them
 - Libraries not found: help the system to find them - LDFLAGS/--with-libraries

Installation:
make



Challenges

- Nothing copied
 - Run make first
- Permission problem
 - Prefix not specified
- Quota exceeded
 - Number of files
 - User storage exceeded
 - Take advantage of large space on /global/scratch

Installation:
make install



Challenges

- Command not found
 - Missed make install?
 - Add the path of the command to your PATH
- Library couldn't be loaded
 - Use ldd to check the libraries used; adjust LD_LIBRARY_PATH
 - Consider recompiling with rpath
 - Static linking ?!
- Argument missing
 - Read the manul/help of the software
 - Start by running an installation check test

Post-Installation



Challenges

- Segmentation faults
 - Check how the software should be used
 - Did you apply patches? Did you set the stack size as required?
 - Google!, Software User Forum – Join the mailing list...
 - Debug the code: gdb, valgrind, etc ...
- Results are not what you expect
 - Run a test for which others reported the results and compare
 - Find what's different
 - The compilers
 - The compilation flags
 - The libraries
- ... Ask support@westgrid.ca

Post-Installation





Thank You!