Software stack deployment for Earth System Modelling using Spack

Kim Serradell Maronda (BSC)
Sergey Kosukhin (MPI-M)

The ESiWACE project has received funding from the European Union’s Horizon 2020 research and innovation program under grant agreement No 675191

This material reflects only the author's view and the Commission is not responsible for any use that may be made of the information it contains.
• Motivation
• Tool description
• Developments for CoE ESiWACE
• Conclusions
• Work done in the framework of CoE ESiWACE (Excellence in Simulation in Weather and Climate in Europe)

• Included in “Usability” work package
  – Goal: Build a system software stack

• Done in collaboration between BSC – MPI-M
Earth System Models (an analogy)
Earth System Models (an analogy)
• To deploy and run an Earth System Model we need:
  – System software
  – Compilers
  – Libraries
  – Tools to pre and postprocess
• These “pieces” can change in every cluster!
Environments are very different

- Single machine or large-scale HPC site?
- Build everything from scratch or use provided system software?
- Which compiler? Which prerequisite packages and their versions?

COMPILERS $\times$ PREREQUISITES $\times$ VERSIONS $\times$ OPTIONS $=$

PRACEdays17 Barcelona, 17/05/2017
An example of a processing tool
How to deal with all these issues?

• Manually
• Binary package managers
  – Designed to manage a single, stable and well tested stack.
  – Install one version of each package in a single prefix (/usr).
• Port systems
  – Macports, Homebrew, Gentoo, etc.
  – Minimal support for builds parameterized by compilers, dependency versions.
• Virtual Machines and Linux Containers (Docker)
  – Containers allow users to build environments for different applications.
  – Does not solve the build problem (someone has to build the image)
  – Performance, security, and upgrade issues prevent widespread HPC deployment.
Do it automatic
Spack is a package management tool designed to support multiple versions and configurations of software on a wide variety of platforms and environments.

It was designed for **large supercomputing** centers, where **many users** and application teams share **common installations** of software on **clusters with exotic architectures**, using **libraries** that do not have a standard ABI.

- Released under Lesser GPL. Available at [https://github.com/LLNL/spack](https://github.com/LLNL/spack)

- More than 140 contributors and currently >1400 packages (libraries, tools, python modules, R packages...)

Spack
Spack is easy

• How to install Spack

Get from git repository:
$ git clone https://github.com/LLNL/spack.git

Or download the archive and unzip it:
$ wget https://github.com/LLNL/spack/archive/develop.zip
$ unzip develop.zip

Setup environmental variables:
$ ./spack/share/spack/setup-env.sh

• How to install a package

$ spack install hdf5
Spack will detect compilers installed
Will build the list of dependencies
And install the package
Spack is easy (III)

- "spack list" (packages available to install)
• “spack load package-name”
  – Spack also generates module files
    • Modules interaction is being improved
  – Spack manages all environnement variables
    • $PATH, $LD_LIBRARY_PATH, ...
Spack is easy (III)

- “spack find” (installed so far)
Spack can be complex to satisfy all needs

- Customizing configurations

```bash
$ spack install cdo
$ spack install cdo@1.7.2
$ spack install cdo@1.7.2 %gcc@4.9.2
$ spack install cdo@1.7.2 %gcc@4.9.2 +grib_api
$ spack install cdo@1.7.2 os=SuSE11
$ spack install cdo@1.7.2 os=CNL10
$ spack install cdo@1.7.2 os=CNL10 target=haswell
```

```bash
$ spack install ncl cflags=\`-O3 -g -fast -fpack-struct\'
```
Spack can be complex to satisfy all needs

- Managing dependencies

```
$ spack install netcdf %intel@16.0.2 ^zlib@1.2.8
```
• Dealing with incompatibles packages
  – MPI is a virtual dependency
  – We have different MPI implementations

Let Spack choose the MPI implementation as long it provides MPI2 interface
• Spack can be used without interacting with HPC system team
  – The user can extend the software stack provided by default
• “spack edit package-name”
  – Description
  – Source code
  – Versions
  – Variants
  – Dependencies
  – Configuration arguments
  – Installation
  – Test (if needed)
• Spack will try to download sources (using curl)
• Some HPC (for security reasons) can not download from login and compute nodes

• **Solution:**
  – Download to a machine with Internet access using Spack:
    `spack fetch -D {package-name}`
  – Copy via ssh to your server:
    `scp -r ./var/spack/cache {server-name}:{/spack-dir}/var/spack/`
• Integration of the ESM applications:
  – CDO, Magics, libemos, grib-api, NCL, cmor

• Improvements for system software:
  – harfbuzz, pango, qt, libtiff, python, uuid, ...

• Improvements for core functionality
### What has been done to use Spack in ESiWACE

<table>
<thead>
<tr>
<th>Demonstrator</th>
<th>Model</th>
<th>Tool/ Library</th>
<th>Version</th>
<th>Website</th>
<th>Package in Spack</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very high resolution atmosphere-only and ocean-only demonstrators</td>
<td>IFS/OpenIFS</td>
<td>LAPACK</td>
<td>3.4.2</td>
<td><a href="http://www.netlib.org/lapack/">http://www.netlib.org/lapack/</a></td>
<td>openblas</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BLAS</td>
<td>3.4.2</td>
<td><a href="http://www.netlib.org/blas/">http://www.netlib.org/blas/</a></td>
<td>openblas</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GRIB-API</td>
<td>1.16.0</td>
<td><a href="https://software.ecmwf.int/wiki/display/GRIB/Home">https://software.ecmwf.int/wiki/display/GRIB/Home</a></td>
<td>grib-api</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XIOS</td>
<td>2015.03.0</td>
<td><a href="http://forge.ipsl.jussieu.fr/ioserver/">http://forge.ipsl.jussieu.fr/ioserver/</a></td>
<td>NA</td>
</tr>
<tr>
<td></td>
<td>NEMO</td>
<td>NETCDF4</td>
<td>4.x</td>
<td><a href="http://www.unidata.ucar.edu/software/netcdf/">http://www.unidata.ucar.edu/software/netcdf/</a></td>
<td>netcdf, netcdf-fortran</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HDF5</td>
<td>1.8.x</td>
<td><a href="https://support.hdfgroup.org/HDF5/">https://support.hdfgroup.org/HDF5/</a></td>
<td>hdf5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SZIP</td>
<td>2.1</td>
<td><a href="https://www.hdfgroup.org/doc_resource/SZIP/">https://www.hdfgroup.org/doc_resource/SZIP/</a></td>
<td>szip</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ZLIB</td>
<td>1.2.x</td>
<td><a href="http://zlib.net">http://zlib.net</a></td>
<td>zlib</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FCM</td>
<td>2015.03.0</td>
<td><a href="http://metomi.github.io/fcm/doc/">http://metomi.github.io/fcm/doc/</a></td>
<td>NA</td>
</tr>
<tr>
<td></td>
<td>ICON</td>
<td>LAPACK</td>
<td>3.4.2</td>
<td><a href="http://www.netlib.org/lapack/">http://www.netlib.org/lapack/</a></td>
<td>openblas</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BLAS</td>
<td>3.4.2</td>
<td><a href="http://www.netlib.org/blas/">http://www.netlib.org/blas/</a></td>
<td>openblas</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NETCDF4</td>
<td>4.x</td>
<td><a href="http://www.unidata.ucar.edu/software/netcdf/">http://www.unidata.ucar.edu/software/netcdf/</a></td>
<td>netcdf-fortran</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HDF5</td>
<td>1.8.x</td>
<td><a href="https://support.hdfgroup.org/HDF5/">https://support.hdfgroup.org/HDF5/</a></td>
<td>hdf5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SZIP</td>
<td>2.1</td>
<td><a href="https://www.hdfgroup.org/doc_resource/SZIP/">https://www.hdfgroup.org/doc_resource/SZIP/</a></td>
<td>szip</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ZLIB</td>
<td>1.2.x</td>
<td><a href="http://zlib.net">http://zlib.net</a></td>
<td>zlib</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LIBXML2</td>
<td>2.9.x</td>
<td><a href="http://xmlsoft.org">http://xmlsoft.org</a></td>
<td>libxml2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GRIB-API</td>
<td>1.16.0</td>
<td><a href="https://software.ecmwf.int/wiki/display/GRIB/Home">https://software.ecmwf.int/wiki/display/GRIB/Home</a></td>
<td>grib-api</td>
</tr>
</tbody>
</table>
• **Examples of Spack production use:**
  - Due to Mare Nostrum update, BSC Earth air quality operational products where deployed in other Spanish HPC clusters
    - Altamira in Universidad de Cantabria
    - Nimbus in Spanish Meteorological Agency
  - **CALIOPE system** (combination of 3 ESM) running in less than two days (usually 1-2 weeks).
• Non-standard installation systems can be handled but not that easily (i.e. ESMF library)

• Some packages from your package’s dependency tree are not at the production level

• There are many implicit dependencies (e.g. icc -> gcc)
Conclusions

• Spack has demonstrated to be a useful tool for Earth System Models

• Spack is easy to test and deploy. Reasonable learning curve

• Next step is gathering all packages in a one step process

• Some issues are still there
QUESTIONS
• ESIWACE started with Easybuild
• So, why moving?
• Each tools has pros and cons
• For those interested
  – In Spain (15th and 16th June 2017), HPCKP (talks from both developers)