esiwace

CENTRE OF EXCELLENCE IN SIMULATION OF WEATHER AND CLIMATE IN EUROPE

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

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Max-Planck-Institut für Meteorologie

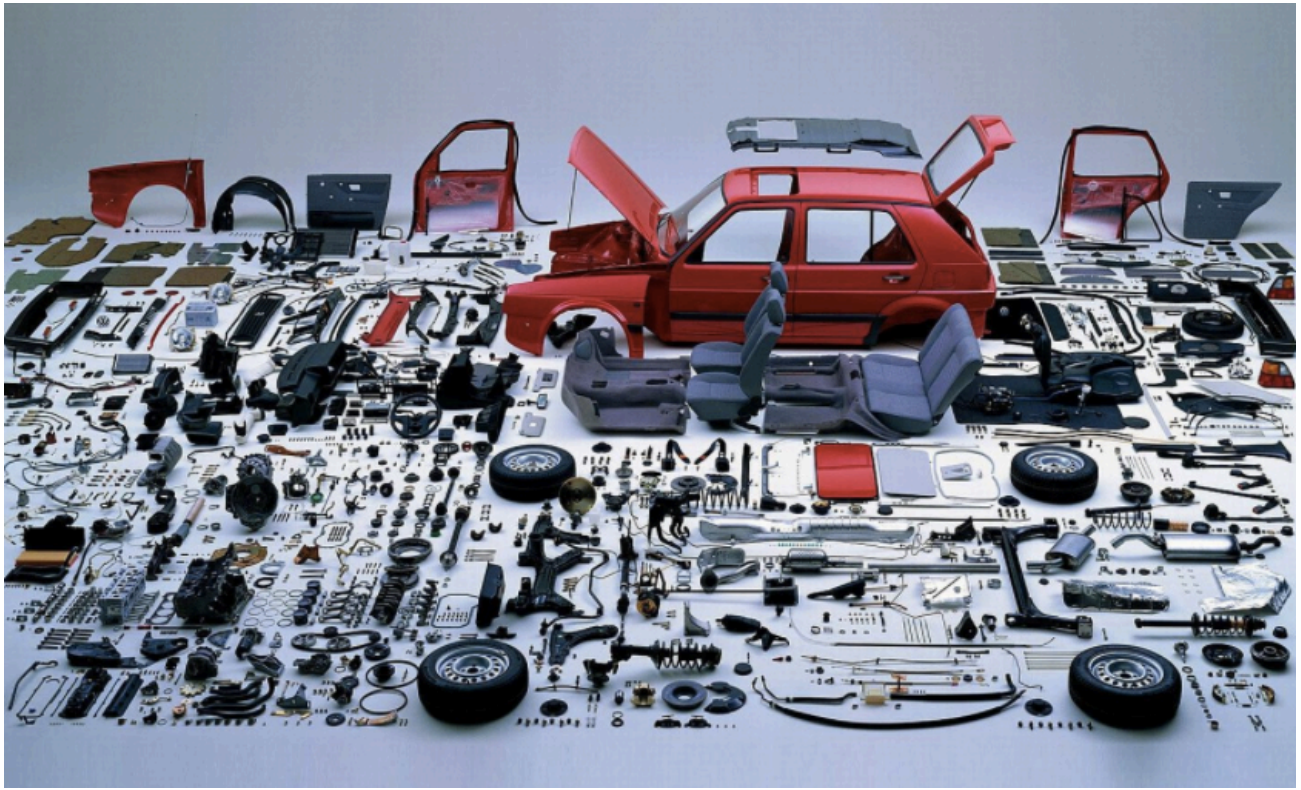
Outline

- 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)



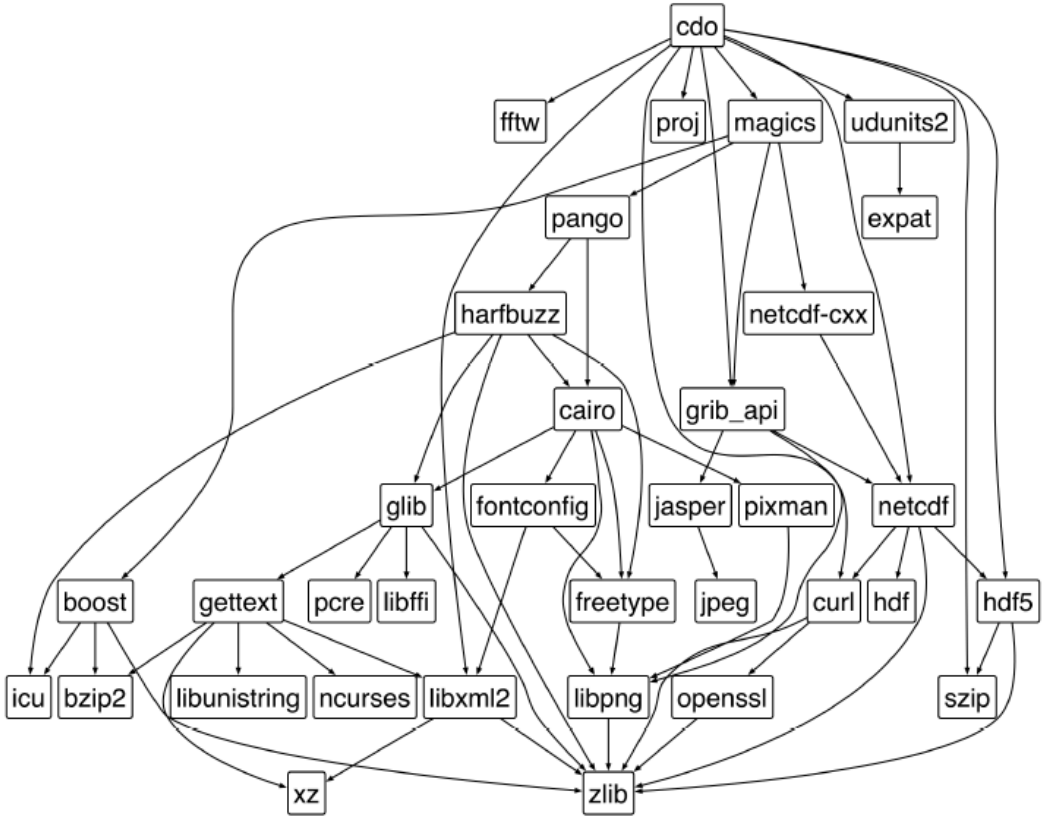
- To deploy and run an Earth System Model we need:
  - System software
  - Compilers
  - Libraries
  - Tools to pre and postprocess
- These “pieces” can change in every cluster!

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 x PREREQUISITES x VERSIONS x OPTIONS =

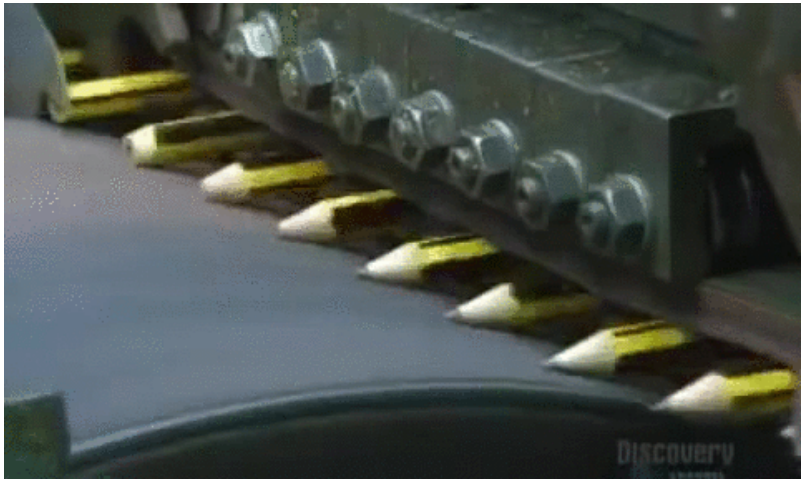
An example of a processing tool

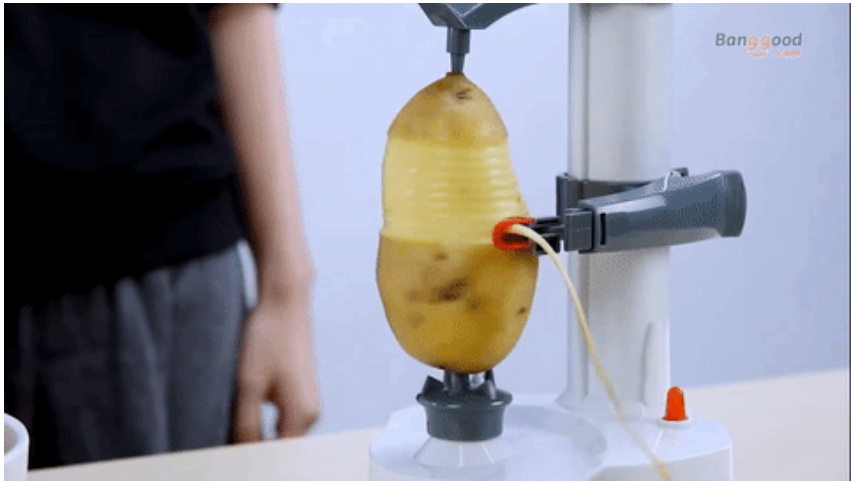


How to deal with all these issues?

- 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

Spack

Lawrence Livermore National Laboratory

- 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. Avialable at https://github.com/LLNL/spack

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

Spack is easy

- How to install Spack

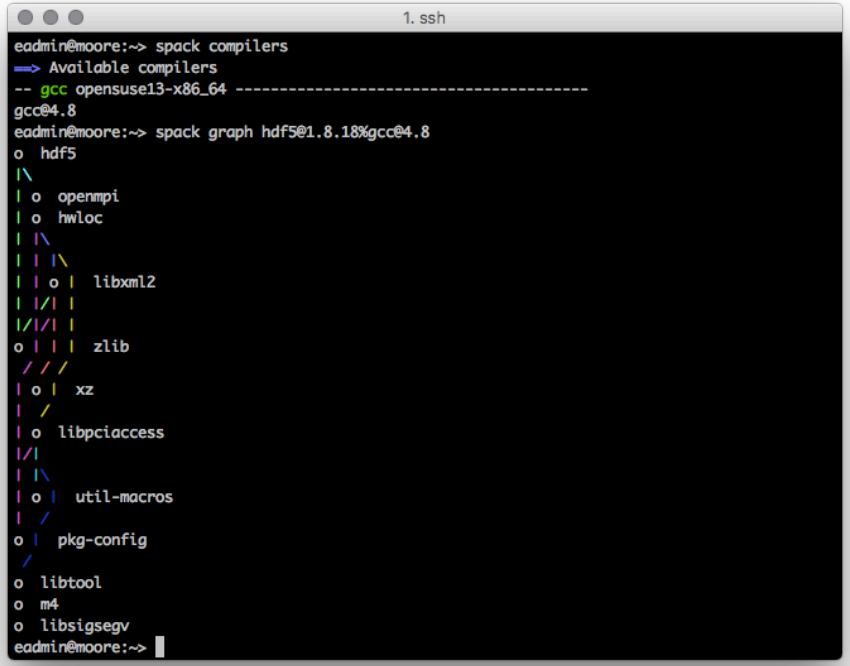
```
Get from git repository:
$ git clone https://github.com/LLNL/spack.git

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
```

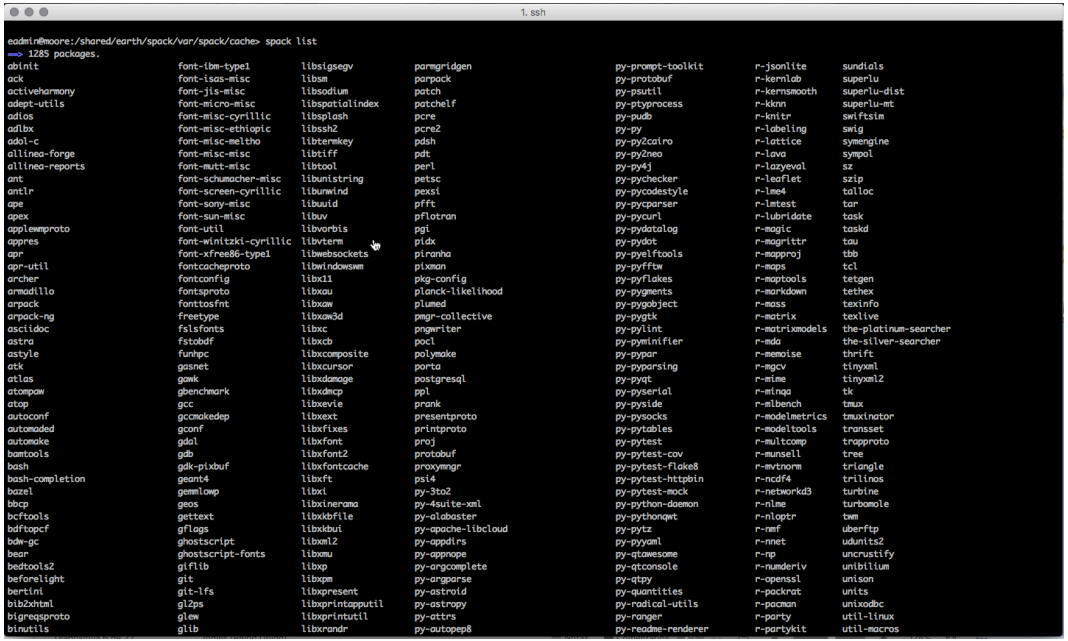
Spack is easy (II)

- 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 is easy (III)

- "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)

1. ssh

```
eadmin@moore:~> spack find
==> 155 installed packages.
-- linux-opensuse13-x86_64 / gcc@4.8 ----------------------------
antlr@2.7.7     fftw@3.3.5          icu4c@57.1          libxml2@2.9.4       pango@1.40.3          qhull@2015.2
autoconf@2.69   flex@2.6.1          inputproto@2.3.2    libxml2@2.9.4       papi@5.5.1            r@3.3.2
autoconf@2.69   flex@2.6.3          jasper@1.900.1      libxml2@2.9.4       parallel-netcdf@1.8.0 r-ncdf4@1.15
automake@1.15   font-util@1.3.1     jdk@8u92-linux-x64  libxrender@0.9.9    pcre@8.39             readline@6.3
binutils@2.27   fontconfig@2.11.1   jpeg@9b             m4@1.4.17           perl@5.24.1           renderproto@0.11.1
binutils@2.28   fontconfig@2.11.1   kbproto@1.0.7       m4@1.4.18           pixman@0.34.0         sqlite@3.8.5
bison@3.0.4     freetype@2.7        libdwarf@20160507   m4@1.4.18           pixman@0.34.0         szip@2.1
bison@3.0.4     freetype@2.7        libffi@3.2.1        mawk@1.3.4          pkg-config@0.29.1     tar@1.29
boost@1.63.0    gettext@0.19.8.1    libgcrypt@1.6.2     munge@0.5.11        proj@4.9.2            tcl@8.6.5
bzip2@1.0.6     gettext@0.19.8.1    libgpg-error@1.21   nasm@2.11.06        py-cdo@1.3.2          tcl@8.6.5
bzip2@1.0.6     gettext@0.19.8.1    libjpeg-turbo@1.5.0 nco@4.6.2           py-cycler@0.10.0      tk@8.6.5
cairo@1.14.0    glib@2.49.4         libpciaccess@0.13.4 ncurses@6.0         py-cython@0.23.5      tk@8.6.5
cairo@1.14.0    glib@2.49.4         libpng@1.6.26       ncurses@6.0         py-dateutil@2.5.2     udunits2@2.2.20
cairo@1.14.8    grib-api@1.17.0     libpng@1.6.27       netcdf@4.4.1        py-mock@1.3.0         util-macros@1.19.0
cairo@1.14.8    gsl@2.3             libpthread-stubs@0.3 netcdf@4.4.1       py-netcdf@1.2.3.1     uuid@1.6.2
cdo@1.7.2       harfbuzz@0.9.37     libsigsegv@2.10     netcdf@4.4.1        py-nose@1.3.7         xcb-proto@1.12
cmake@3.7.1     harfbuzz@0.9.37     libsigsegv@2.11     netcdf@4.4.1.1      py-numpy@1.11.2       xerces-c@3.1.4
cmake@3.7.2     harfbuzz@0.9.37     libtiff@4.0.6       netcdf-fortran@4.4.4 py-numpy@1.11.2      xextproto@7.3.0
cmor@3.2.0      hdf5@1.8.18         libtiff@4.0.6       netcdf-fortran@4.4.4 py-pbr@1.8.1         xproto@7.0.29
curl@7.50.3     hdf5@1.10.0-patch1  libtool@2.4.6       openblas@0.2.19     py-pillow@3.2.0       xtrans@1.3.5
curl@7.52.1     hdf5@1.10.0-patch1  libunwind@1.1       openmpi@2.0.1       py-pyparsing@2.0.3    xz@5.2.2
dyninst@9.3.0   hdf5@1.10.0-patch1  libx11@1.6.3        openmpi@2.0.1       py-pytz@2016.6.1      xz@5.2.3
elfutils@0.163  help2man@1.47.4     libxau@1.0.8        openmpi@2.0.2       py-scipy@0.18.1       zlib@1.2.8
elfutils@0.168  hwloc@1.11.4        libxcb@1.12         openssl@1.0.2j      py-setuptools@25.2.0  zlib@1.2.10
expat@2.2.0     hwloc@1.11.5        libxdmcp@1.1.2      openssl@1.0.2k      py-six@1.10.0         zlib@1.2.11
extrae@3.4.1    hwloc@1.11.6        libxext@1.3.3       openssl@1.1.0c      python@2.7.12
eadmin@moore:~>
```

Spack can be complex to satisfy all needs

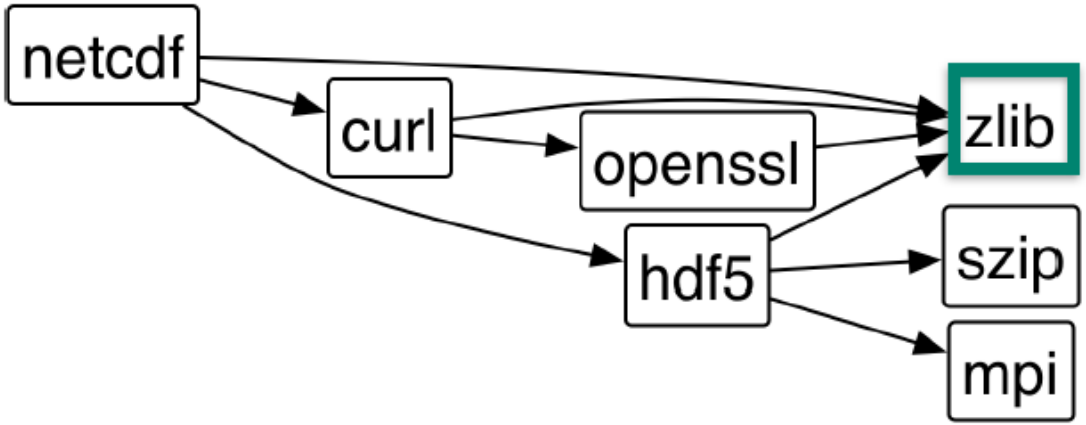
- Customizing configurations

```
$ spack install cdo                                  unconstrained
$ spack install cdo@1.7.2                            @ custom version
$ spack install cdo@1.7.2 %gcc@4.9.2                 % custom compiler
$ spack install cdo@1.7.2 %gcc@4.9.2 +grib_api       +/~ build option
$ spack install cdo@1.7.2 os=SuSE11                  os=<frontend OS>
$ spack install cdo@1.7.2 os=CNL10                   os=<backend OS>
$ spack install cdo@1.7.2 os=CNL10 target=haswell    target=<cpu target>
```

```
$ spack install ncl cflags=\'-O3 -g -fast -fpack-struct\'
```

Spack can be complex to satisfy all needs

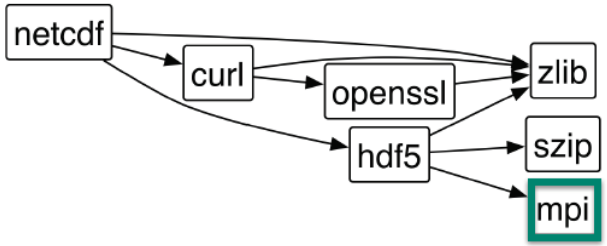
- Managing dependencies




```
$ spack install netcdf %intel@16.0.2 ^zlib@1.2.8
```

Spack can be complex to satisfy all needs

- Dealing with incompatibles packages
  - MPI is a virtual dependency
  - We have different MPI implementations




```
$ spack install netcdf ^mvapich@1.9
```

```
$ spack install netcdf ^openmpi@1.4:
```

  - Let Spack choose the MPI implementation as long it provides MPI2 interface

```
$ spack install netcdf ^mpi@2
```

Give the power to the user

- Spack can be used without interacting with HPC system team
  - The user can extend the software stack provided by default

```
compilers:
- compiler:
    environment: {}
    extra_rpaths: []
    flags: {}
    modules:
    - intel/17.0.2
    operating_system: rhel6
    paths:
      cc: icc
      cxx: icpc
      f77: ifort
      fc: ifort
    spec: intel@17.0.2
    target: x86_64
- compiler:
    environment: {}
    extra_rpaths: []
    flags: {}
    modules:
    - gcc/4.8.2
    operating_system: rhel6
    paths:
      cc: gcc
      cxx: g++
      f77: gfortran
      fc: gfortran
    spec: gcc@4.8.2
    target: x86_64
```

```
packages:
  netcdf:
    paths:
      netcdf@4.3.2~mpi%gcc@4.8: /sw/rhel6-x64/netcdf/netcdf_c-4.3.2-gcc48/
    modules:
      netcdf@4.3.2~mpi%gcc@4.8: netcdf_c/4.3.2-gcc48
    buildable: False
```

What's a package

- "spack edit package-name"
  - Description
  - Source code
  - Versions
  - Variants
  - Dependencies
  - Configuration arguments
  - Installation
  - Test (if needed)

1. ssh

```
25 from spack import *
26
27
28 class Cdo(Package):
29     """CDO is a collection of command line Operators to manipulate and analyse
30     Climate and NWP model Data. """
31
32     homepage = "https://code.zmaw.de/projects/cdo"
33
34     version('1.7.2', 'f08e4ce8739a4f2b63fc81a24db3ee31', url='https://code.zmaw.de/attachments/download/12760/cdo-1.7.2.tar.gz')
35     version('1.6.9', 'bf0997bf20e812f35e10188a930e24e2', url='https://code.zmaw.de/attachments/download/10198/cdo-1.6.9.tar.gz')
36
37     variant('szip', default=True, description='Enable szip compression for GRIB1')
38     variant('hdf5', default=False, description='Enable HDF5 support')
39     variant('netcdf', default=True, description='Enable NetCDF support')
40     variant('udunits2', default=True, description='Enable UDUNITS2 support')
41     variant('grib', default=True, description='Enable GRIB_API support')
42     variant('libxml2', default=True, description='Enable libxml2 support')
43     variant('proj', default=True, description='Enable PROJ library for cartographic projections')
44     variant('curl', default=True, description='Enable curl support')
45     variant('fftw', default=True, description='Enable support for fftw3')
46     variant('magics', default=True, description='Enable Magics library support')
47
48     depends_on('szip', when='+szip')
49     depends_on('netcdf', when='+netcdf')
50     depends_on('hdf5+threadsafe', when='+hdf5')
51     depends_on('udunits2', when='+udunits2')
52     depends_on('grib-api', when='+grib')
53     depends_on('libxml2', when='+libxml2')
54     depends_on('proj', when='+proj')
55     depends_on('curl', when='+curl')
56     depends_on('fftw', when='+fftw')
57     depends_on('magics', when='+magics')
58
59     def install(self, spec, prefix):
60         config_args = ["--prefix=" + prefix,
61                        "--enable-shared",
62                        "--enable-static"]
63
64         if '+szip' in spec:
65             config_args.append('--with-szlib=' + spec['szip'].prefix)
66         else:
67             config_args.append('--without-szlib')
68
69         if '+hdf5' in spec:
70             config_args.append('--with-hdf5=' + spec['hdf5'].prefix)
71         else:
/shared/earth/spack/var/spack/repos/builtin/packages/cdo/package.py   71,1   34%
```

Sources download

- 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/

Spack in Earth System Models

- 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

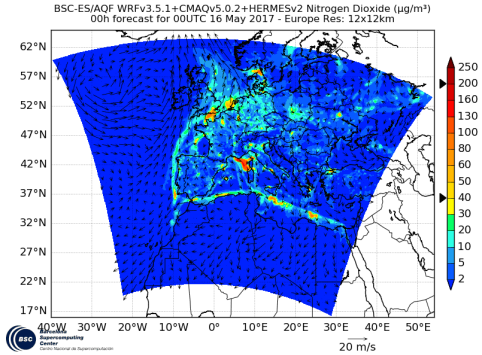
Spack in Earth System Models

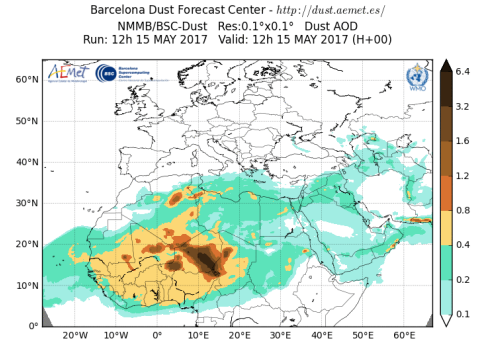
- What has been done to use Spack in ESiWACE

Demonstrator	Model	Tool/ Library	Version	Website	Package in Spack
Very high resolution atmosphere-only and ocean-only demonstrators	IFS/OpenIFS	LAPACK	3.4.2	http://www.netlib.org/lapack/	openblas
		BLAS	3.4.2	http://www.netlib.org/blas/	openblas
		GRIB-API	1.16.0	https://software.ecmwf.int/wiki/ display/GRIB/Home	grib-api
		FCM	2015.03.0	http://metomi.github.io/fcm/doc/	NA
	NEMO	XIOS	2.0	http://forge.ipsl.jussieu.fr/ioserver/	NA
		NETCDF4	4.x	http://www.unidata.ucar.edu/ software/netcdf/	netcdf netcdf-fortran
		HDF5	1.8.x	https://support.hdfgroup.org/HDF5/	hdf5
		SZIP	2.1	https://www.hdfgroup.org/ doc_resource/SZIP/	szip
		ZLIB	1.2.x	http://zlib.net	zlib
		FCM	2015.03.0	http://metomi.github.io/fcm/doc/	NA
	ICON	LAPACK	3.4.2	http://www.netlib.org/lapack/	openblas
		BLAS	3.4.2	http://www.netlib.org/blas/	openblas
		NETCDF4	4.x	http://www.unidata.ucar.edu/ software/netcdf/	netcdf-fortran
		HDF5	1.8.x	https://support.hdfgroup.org/HDF5/	hdf5
		SZIP	2.1	https://www.hdfgroup.org/ doc_resource/SZIP/	szip
		ZLIB	1.2.x	http://zlib.net	zlib
		LIBXML2	2.9.x	http://xmlsoft.org	libxml2
		GRIB-API	1.16.0	https://software.ecmwf.int/wiki/ display/GRIB/Home	grib-api

Spack in production

- 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).





But some issues are still there

- 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 demostrated to a be 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

Extra content: Spack vs. Easybuild

- 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)