EURO Greece

The Weather Research and Forecasting Model on HPC

ReACT

Remote sensing of Aerosols, Clouds and Trace gases

IAASARS

Dr. Stergios Kartsios

Introduction

- Weather Research and Forecasting (WRF) Model:
  - A next-generation *mesoscale numerical weather* prediction system (*and much more*) designed for both atmospheric research and operational forecasting applications
  - 1 dynamical core (**ARW – non-hydrostatic)**
    - A legacy hydrostatic core (NMM, last version 3.9)
  - data assimilation system (**WRFDA**)
  - **supporting parallel computation and system extensibility**
  - **Coupling framework (Chem, Hydro, Urban, Fire, Solar, Hurricanes)**
- Serves a wide range of meteorological applications across scales from tens of meters to thousands of kilometers

http://www2.mmm.ucar.edu/wrf/users

https://www.mmm.ucar.edu/models/wrf

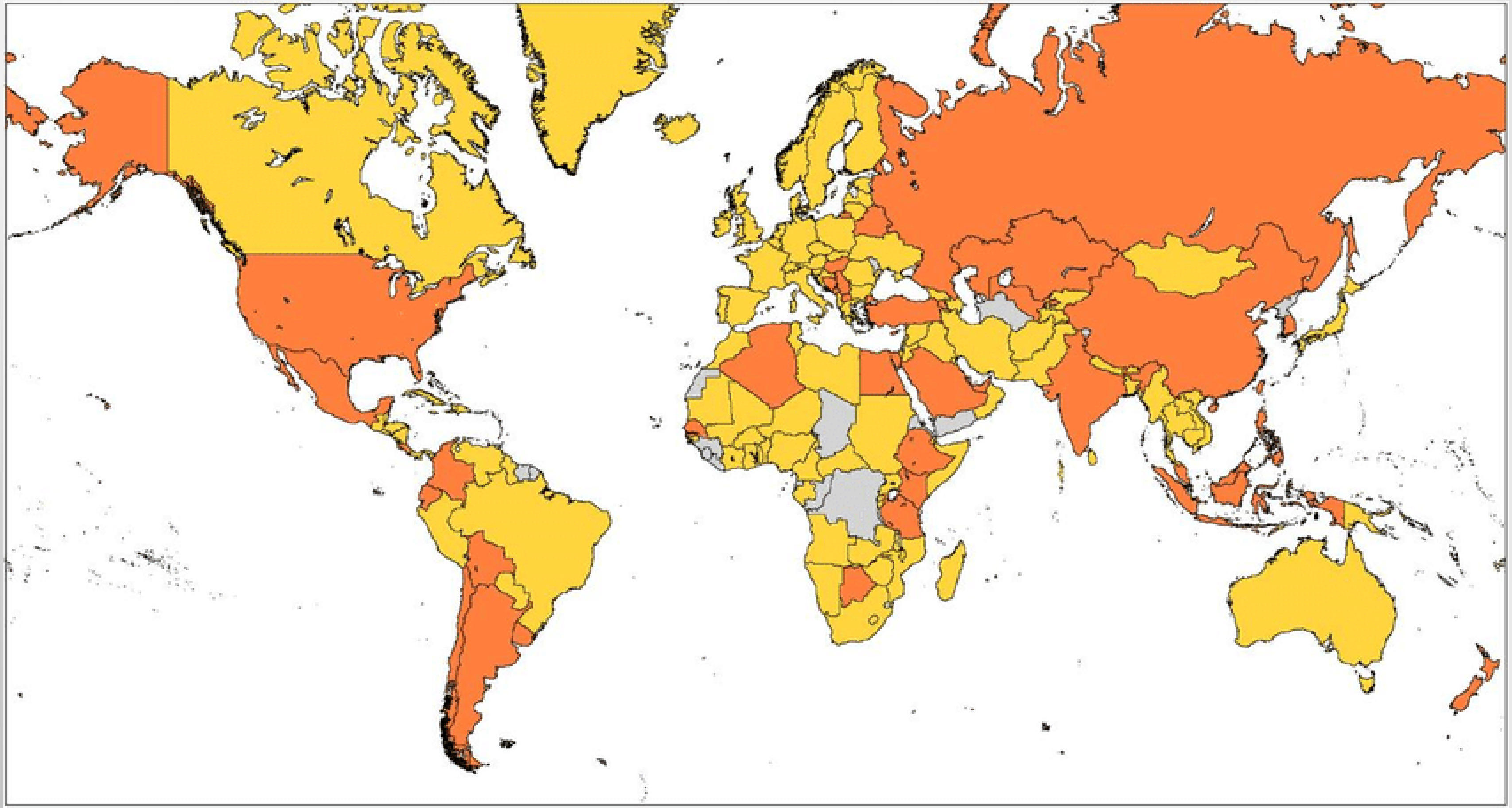
Introduction

- V2.0.1: May 21, 2004
- V2.0.2: June 3, 2004
- V2.0.3: Nov 12, 2004
- V2.0.3.1: Dec 3, 2004
- V2.1.0: August 4, 2005
- V2.1.1: Nov 8, 2005
- V2.1.2: Jan 27, 2006
- V2.2.0: Dec 21, 2006
- V2.2.1: Nov 1, 2007
- V3.0.0: April 4, 2008
- V3.0.1: August 5, 2008
- V3.0.1.1: August 22, 2008
- V3.1.0: April 9, 2009
- V3.1.1: July 31, 2009
- V3.2.0: March 31, 2010
- V3.2.1: August 18, 2010
- V3.3.0: April 6, 2011
- V3.3.1: Sept 16, 2011
- V3.4.0: April 6, 2012
- V3.4.1: Aug 16, 2012
- V3.5.0: April 18, 2013
- V3.5.1: Sept 23, 2013
- V3.6.0: April 18, 2014
- V3.6.1: Aug 14, 2014
- V3.7.0: April 20, 2015
- V3.7.1: Aug 14, 2015
- V3.8.0: April 8, 2016
- V3.8.1: Aug 12, 2016
- V3.9.0: Apr 17, 2017
- V3.9.1: Aug 17, 2017
- **V3.9.1.1: Aug 28, 2017**
- **V4.0.0: Jun 9, 2018**
- V4.0.1: Oct 3, 2018
- V4.0.2: Nov 10, 2018
- V4.0.3: Dec 18, 2018
- V4.1.0: Apr 12, 2019
- V4.1.1: Jun 4, 2019
- V4.1.2: Jul 12, 2019
- V4.1.3: Nov 25, 2019
- V4.1.4: Feb 12, 2020
- V4.1.5: Mar 10, 2020
- V4.2.0: Apr 23, 2020
- V4.2.1: Jul 22, 2020
- V4.2.2: Jan 15, 2021
- V4.3.0: May 11, 2021
- V4.3.1: Oct 28, 2021
- V4.3.2: Dec 15, 2021
- V4.3.3: Jan 11, 2022
- V4.4.0: Apr 26, 2022
- V4.4.1: Aug 26, 2022
- V4.4.2: Dec 19, 2022
- V4.5.0: Apr 21, 2023
- V4.5.1: Jul 26, 2023
- V4.5.2: Dec 22, 2023
- V4.6.0: May 9, 2024
- V4.6.1: Oct 17, 2024
- **V4.7.0: April 25, 2025**

New hybrid option available

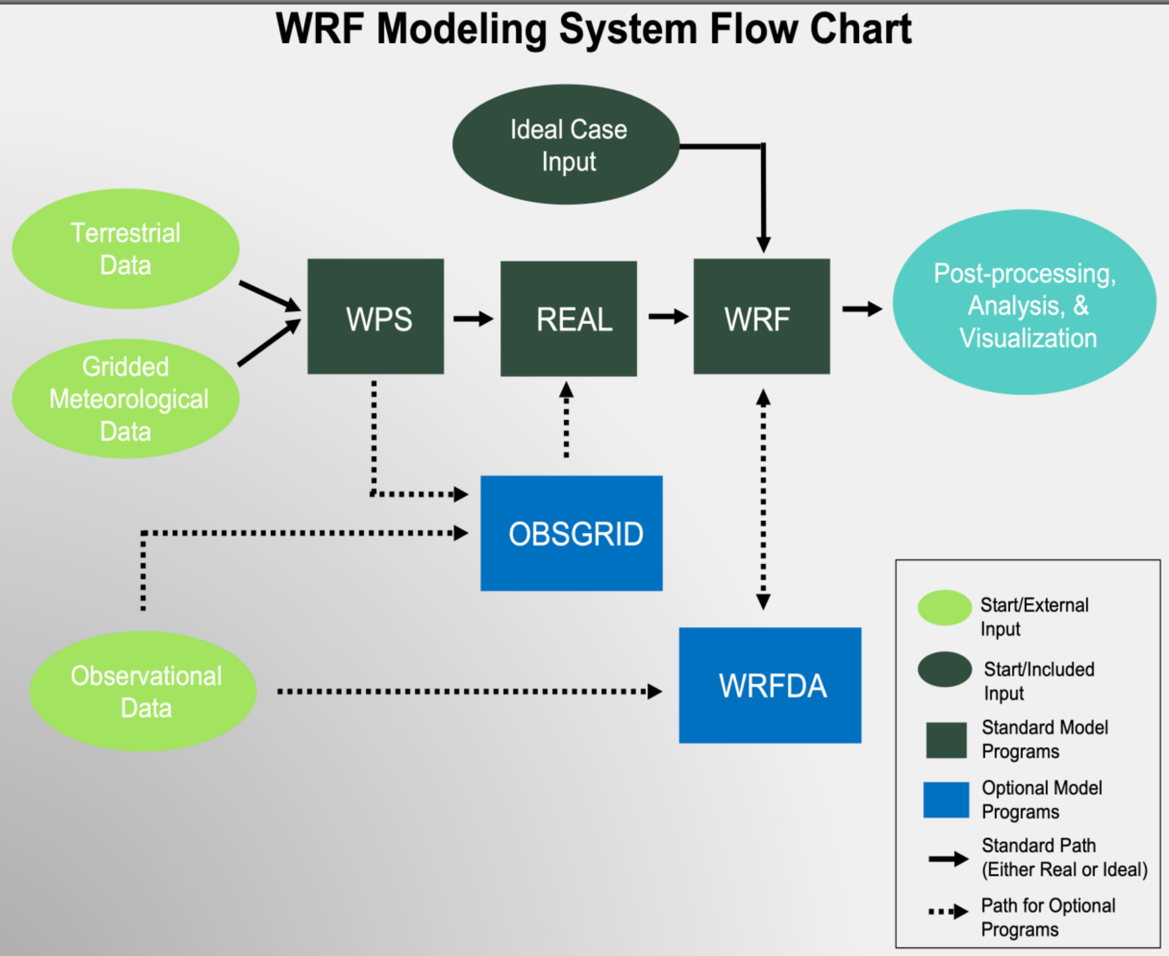
https://github.com/wrf-model/WRF/releases?page=1

Introduction



Powers et al. (2017)

Introduction


WRF Modeling System Flow Chart

- Main components:
  - The WRF Preprocessing System (WPS)
  - Initialization (REAL, IDEAL)
  - ARW solver
  - WRF-DA
  - Post-processing & Visualization tools

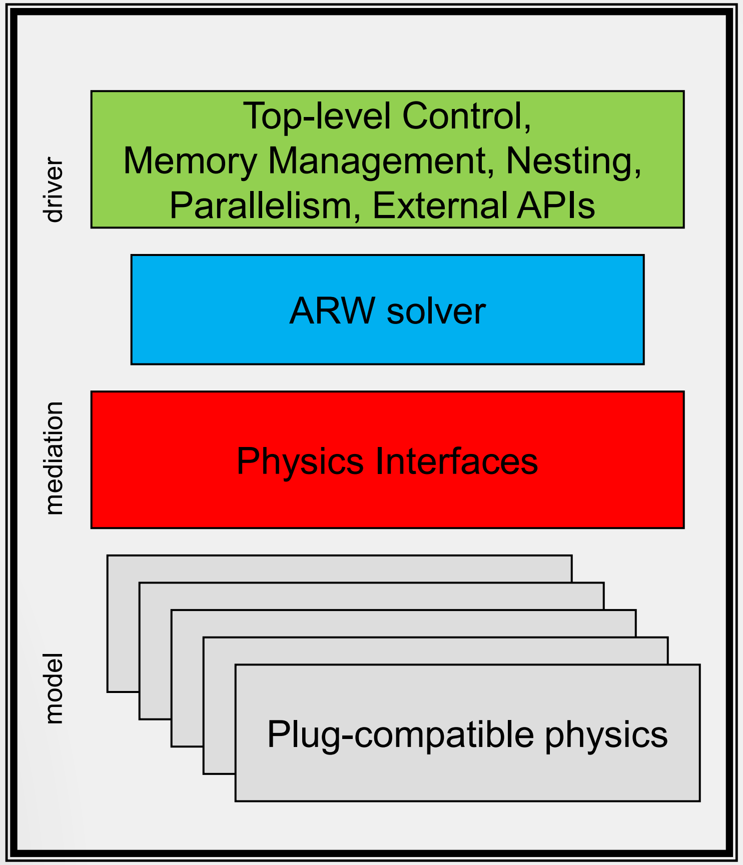
A Linux application!

Can be installed also on

https://www2.mmm.ucar.edu/wrf/users/wrf_users_guide/build/html/overview.html

Introduction

WRF Software Architecture



Software Requirements

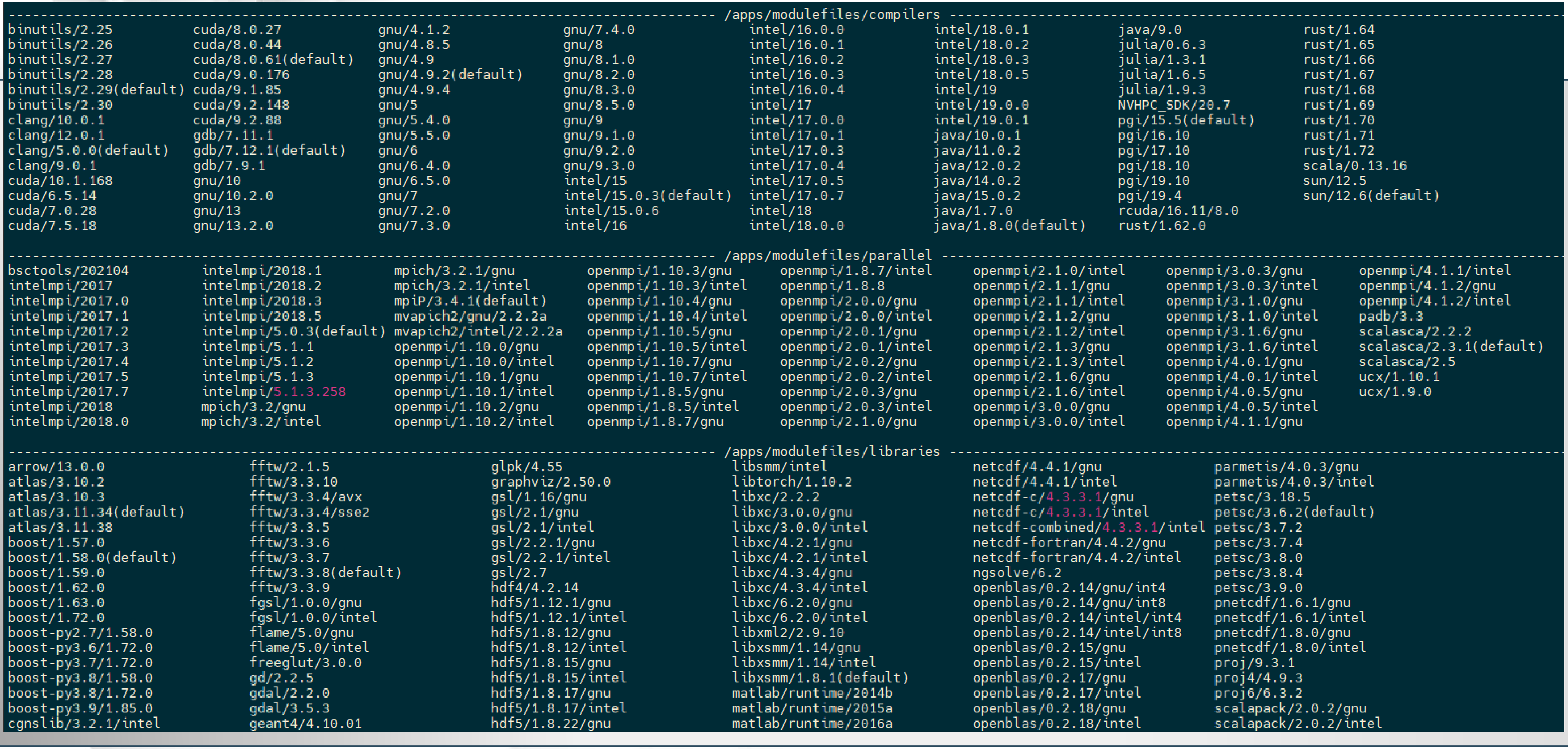
- Fortran 90 or 95 and C compiler (e.g. gnu, intel)
- **perl** 5.04 or later
- If **MPI** and **OpenMP** compilation is desired, MPI or OpenMP libraries are required
- WRF I/O API supports **netCDF**, **pnetCDF**, **HDF5**, **GriB 1** and **GriB 2** formats; hence one of these libraries needs to be available on the computer on which you compile and run WRF
- UNIX utilities: **csh** and **Bourne** shell, **make**, **M4**, **sed**, **awk**, and the **uname** command

Library requirements

- Lots of on-line tutorials and examples (e.g. https://www2.mmm.ucar.edu/wrf/users/tutorial/tutorial.html)
- Depending on the type of run you wish to make, there are various libraries that should be installed. Below are 5 standard libraries:
  - *mpich/intelmpi*
  - *netcdf**
  - *jasper*
  - *libpng*
  - *zlib*
- It is important to note that these libraries must all be installed *with the same compiler* as will be used to install WRF and WPS.
- Export some variables to the environment (e.g. WRF_EM_CORE=1)
- On ***ARIS*** all the necessary libraries are available through the environment module approach (module load)

*If compression of output files is needed, then you should have also HDF5 where netcdf supports hdf5 compression

./module avail



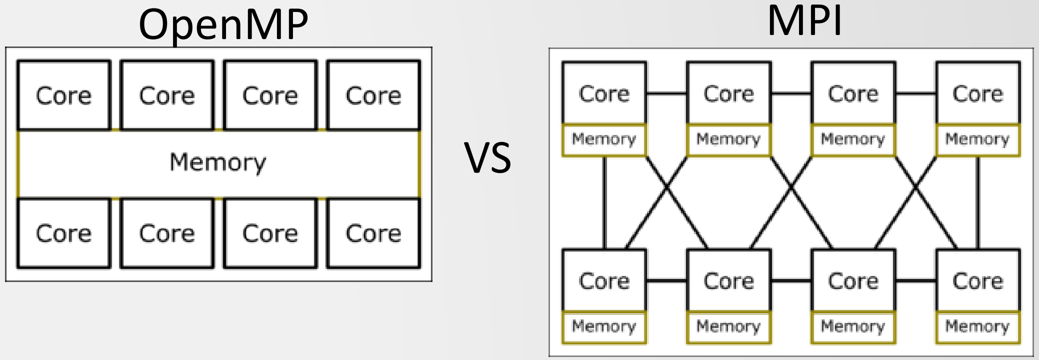
```
------------------------------------------------------------------------- /apps/modulefiles/compilers -------------------------------------------------------------------------
binutils/2.25          cuda/8.0.27            gnu/4.1.2              gnu/7.4.0              intel/16.0.0           intel/18.0.1           java/9.0               rust/1.64
binutils/2.26          cuda/8.0.44            gnu/4.8.5              gnu/8                  intel/16.0.1           intel/18.0.2           julia/0.6.3            rust/1.65
binutils/2.27          cuda/8.0.61(default)   gnu/4.9                gnu/8.1.0              intel/16.0.2           intel/18.0.3           julia/1.3.1            rust/1.66
binutils/2.28          cuda/9.0.176           gnu/4.9.2(default)     gnu/8.2.0              intel/16.0.3           intel/18.0.5           julia/1.6.5            rust/1.67
binutils/2.29(default) cuda/9.1.85            gnu/4.9.4              gnu/8.3.0              intel/16.0.4           intel/19               julia/1.9.3            rust/1.68
binutils/2.30          cuda/9.2.148           gnu/5                  gnu/8.5.0              intel/17               intel/19.0.0           NVHPC_SDK/20.7         rust/1.69
clang/10.0.1           cuda/9.2.88            gnu/5.4.0              gnu/9                  intel/17.0.0           intel/19.0.1           pgi/15.5(default)      rust/1.70
clang/12.0.1           gdb/7.11.1             gnu/5.5.0              gnu/9.1.0              intel/17.0.1           java/10.0.1            pgi/16.10              rust/1.71
clang/5.0.0(default)   gdb/7.12.1(default)    gnu/6                  gnu/9.2.0              intel/17.0.3           java/11.0.2            pgi/17.10              rust/1.72
clang/9.0.1            gdb/7.9.1              gnu/6.4.0              gnu/9.3.0              intel/17.0.4           java/12.0.2            pgi/18.10              scala/0.13.16
cuda/10.1.168          gnu/10                 gnu/6.5.0              intel/15               intel/17.0.5           java/14.0.2            pgi/19.10              sun/12.5
cuda/6.5.14            gnu/10.2.0             gnu/7                  intel/15.0.3(default)  intel/17.0.7           java/15.0.2            pgi/19.4               sun/12.6(default)
cuda/7.0.28            gnu/13                 gnu/7.2.0              intel/15.0.6           intel/18               java/1.7.0             rcuda/16.11/8.0
cuda/7.5.18            gnu/13.2.0             gnu/7.3.0              intel/16               intel/18.0.0           java/1.8.0(default)    rust/1.62.0

------------------------------------------------------------------------- /apps/modulefiles/parallel --------------------------------------------------------------------------
bsctools/202104         intelmpi/2018.1         mpich/3.2.1/gnu         openmpi/1.10.3/gnu      openmpi/1.8.7/intel     openmpi/2.1.0/intel     openmpi/3.0.3/gnu       openmpi/4.1.1/intel
intelmpi/2017           intelmpi/2018.2         mpich/3.2.1/intel       openmpi/1.10.3/intel    openmpi/1.8.8           openmpi/2.1.1/gnu       openmpi/3.0.3/intel     openmpi/4.1.2/gnu
intelmpi/2017.0         intelmpi/2018.3         mpiP/3.4.1(default)     openmpi/1.10.4/gnu      openmpi/2.0.0/gnu       openmpi/2.1.1/intel     openmpi/3.1.0/gnu       openmpi/4.1.2/intel
intelmpi/2017.1         intelmpi/2018.5         mvapich2/gnu/2.2.2a     openmpi/1.10.4/intel    openmpi/2.0.0/intel     openmpi/2.1.2/gnu       openmpi/3.1.0/intel     padb/3.3
intelmpi/2017.2         intelmpi/5.0.3(default) mvapich2/intel/2.2.2a   openmpi/1.10.5/gnu      openmpi/2.0.1/gnu       openmpi/2.1.2/intel     openmpi/3.1.6/gnu       scalasca/2.2.2
intelmpi/2017.3         intelmpi/5.1.1          openmpi/1.10.0/gnu      openmpi/1.10.5/intel    openmpi/2.0.1/intel     openmpi/2.1.3/gnu       openmpi/3.1.6/intel     scalasca/2.3.1(default)
intelmpi/2017.4         intelmpi/5.1.2          openmpi/1.10.0/intel    openmpi/1.10.7/gnu      openmpi/2.0.2/gnu       openmpi/2.1.3/intel     openmpi/4.0.1/gnu       scalasca/2.5
intelmpi/2017.5         intelmpi/5.1.3          openmpi/1.10.1/gnu      openmpi/1.10.7/intel    openmpi/2.0.2/intel     openmpi/2.1.6/gnu       openmpi/4.0.1/intel     ucx/1.10.1
intelmpi/2017.7         intelmpi/5.1.3.258      openmpi/1.10.1/intel    openmpi/1.8.5/gnu       openmpi/2.0.3/gnu       openmpi/2.1.6/intel     openmpi/4.0.5/gnu       ucx/1.9.0
intelmpi/2018           mpich/3.2/gnu           openmpi/1.10.2/gnu      openmpi/1.8.5/intel     openmpi/2.0.3/intel     openmpi/3.0.0/gnu       openmpi/4.0.5/intel
intelmpi/2018.0         mpich/3.2/intel         openmpi/1.10.2/intel    openmpi/1.8.7/gnu       openmpi/2.1.0/gnu       openmpi/3.0.0/intel     openmpi/4.1.1/gnu

------------------------------------------------------------------------- /apps/modulefiles/libraries -------------------------------------------------------------------------
arrow/13.0.0                  fftw/2.1.5                    glpk/4.55                     libsmm/intel                  netcdf/4.4.1/gnu              parmetis/4.0.3/gnu
atlas/3.10.2                  fftw/3.3.10                   graphviz/2.50.0               libtorch/1.10.2               netcdf/4.4.1/intel            parmetis/4.0.3/intel
atlas/3.10.3                  fftw/3.3.4/avx                gsl/1.16/gnu                  libxc/2.2.2                   netcdf-c/4.3.3.1/gnu          petsc/3.18.5
atlas/3.11.34(default)        fftw/3.3.4/sse2               gsl/2.1/gnu                   libxc/3.0.0/gnu               netcdf-c/4.3.3.1/intel        petsc/3.6.2(default)
atlas/3.11.38                 fftw/3.3.5                    gsl/2.1/intel                 libxc/3.0.0/intel             netcdf-combined/4.3.3.1/intel petsc/3.7.2
boost/1.57.0                  fftw/3.3.6                    gsl/2.2.1/gnu                 libxc/4.2.1/gnu               netcdf-fortran/4.4.2/gnu      petsc/3.7.4
boost/1.58.0(default)         fftw/3.3.7                    gsl/2.2.1/intel               libxc/4.2.1/intel             netcdf-fortran/4.4.2/intel    petsc/3.8.0
boost/1.59.0                  fftw/3.3.8(default)           gsl/2.7                       libxc/4.3.4/gnu               ngsolve/6.2                   petsc/3.8.4
boost/1.62.0                  fftw/3.3.9                    hdf4/4.2.14                   libxc/4.3.4/intel             openblas/0.2.14/gnu/int4      petsc/3.9.0
boost/1.63.0                  fgsl/1.0.0/gnu                hdf5/1.12.1/gnu               libxc/6.2.0/gnu               openblas/0.2.14/gnu/int8      pnetcdf/1.6.1/gnu
boost/1.72.0                  fgsl/1.0.0/intel              hdf5/1.12.1/intel             libxc/6.2.0/intel             openblas/0.2.14/intel/int4    pnetcdf/1.6.1/intel
boost-py2.7/1.58.0            flame/5.0/gnu                 hdf5/1.8.12/gnu               libxml2/2.9.10                openblas/0.2.14/intel/int8    pnetcdf/1.8.0/gnu
boost-py3.6/1.72.0            flame/5.0/intel               hdf5/1.8.12/intel             libxsmm/1.14/gnu              openblas/0.2.15/gnu           pnetcdf/1.8.0/intel
boost-py3.7/1.72.0            freeglut/3.0.0                hdf5/1.8.15/gnu               libxsmm/1.14/intel            openblas/0.2.15/intel         proj/9.3.1
boost-py3.8/1.58.0            gd/2.2.5                      hdf5/1.8.15/intel             libxsmm/1.8.1(default)        openblas/0.2.17/gnu           proj4/4.9.3
boost-py3.8/1.72.0            gdal/2.2.0                    hdf5/1.8.17/gnu               matlab/runtime/2014b          openblas/0.2.17/intel         proj6/6.3.2
boost-py3.9/1.85.0            gdal/3.5.3                    hdf5/1.8.17/intel             matlab/runtime/2015a          openblas/0.2.18/gnu           scalapack/2.0.2/gnu
cgnslib/3.2.1/intel           geant4/4.10.01                hdf5/1.8.22/gnu               matlab/runtime/2016a          openblas/0.2.18/intel         scalapack/2.0.2/intel
```

Parallelism in WRF

- WRF can be configured to run either in:
  - *serial*
  - *distributed memory (DM, “MPI”)*
  - *shared memory (SM, “OpenMP”)*
  - *or clusters of SM processors (hybrid, “MPI+OpenMP”)*
- Although several configuration and compiler options are available,
  - experience with WRF on ARIS showed us that it is better to use **DM mode** (MPI)
  - with Intel compiler due to hardware architecture

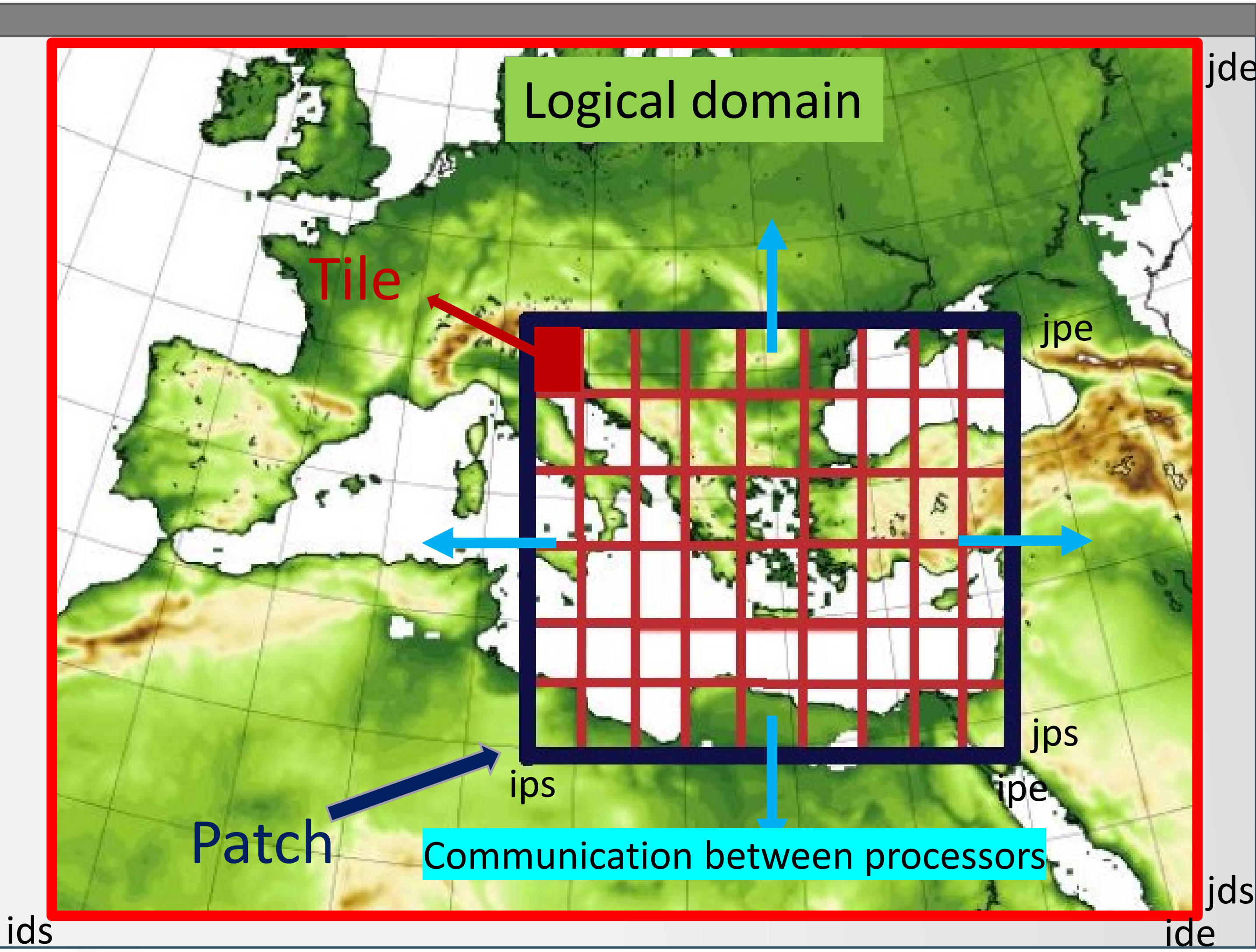

OpenMP

VS

MPI

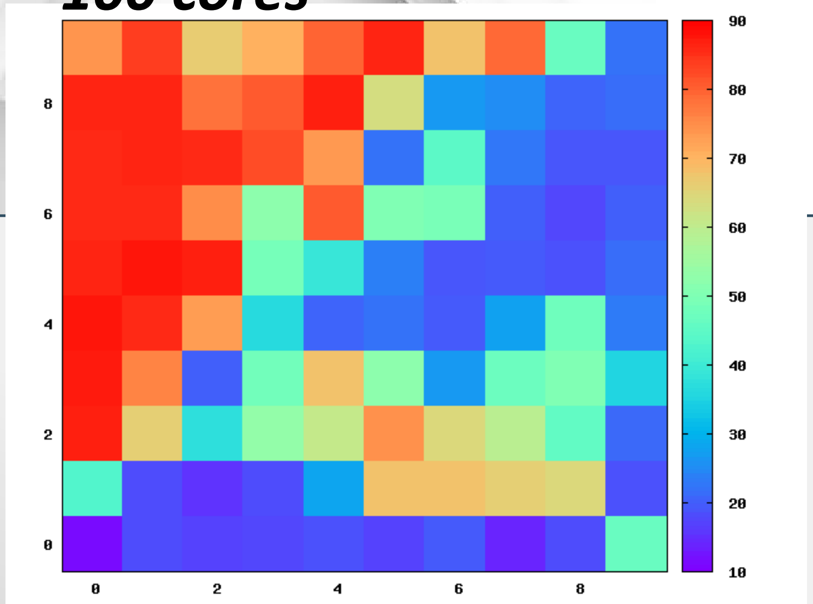
Parallelism in WRF

- WRF uses domain decomposition to divide total amount of work over parallel processes
- Model domains are decomposed for parallelism on two-levels
  - **Patch:** *section of model domain allocated to a distributed memory node, this is the scope of a mediation layer solver or physics driver*
  - **Tile:** *section of a patch allocated to a shared-memory processor within a node; this is also the scope of a model layer subroutine*
- Distributed memory parallelism is over patches
- Shared memory parallelism is over tiles within patches

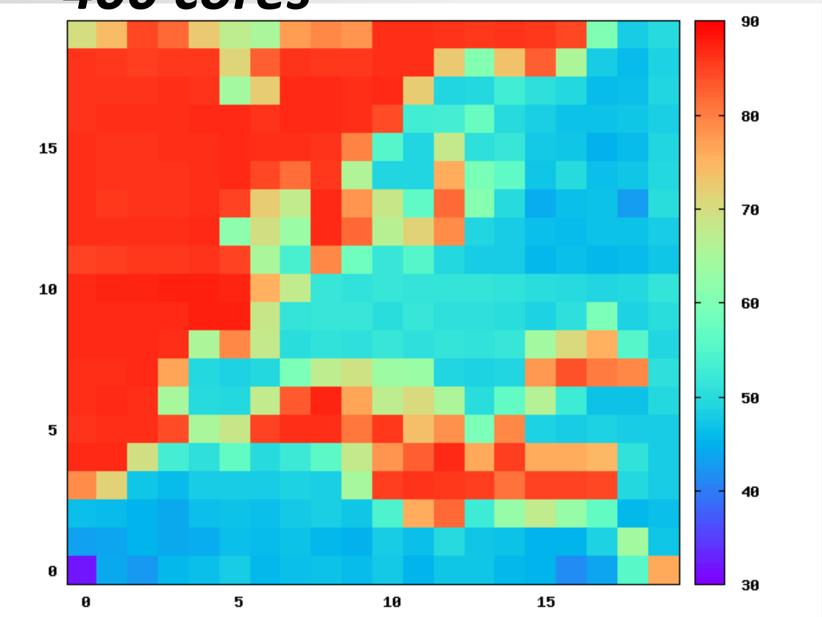


Waiting time (%) in each task (core) over a single domain

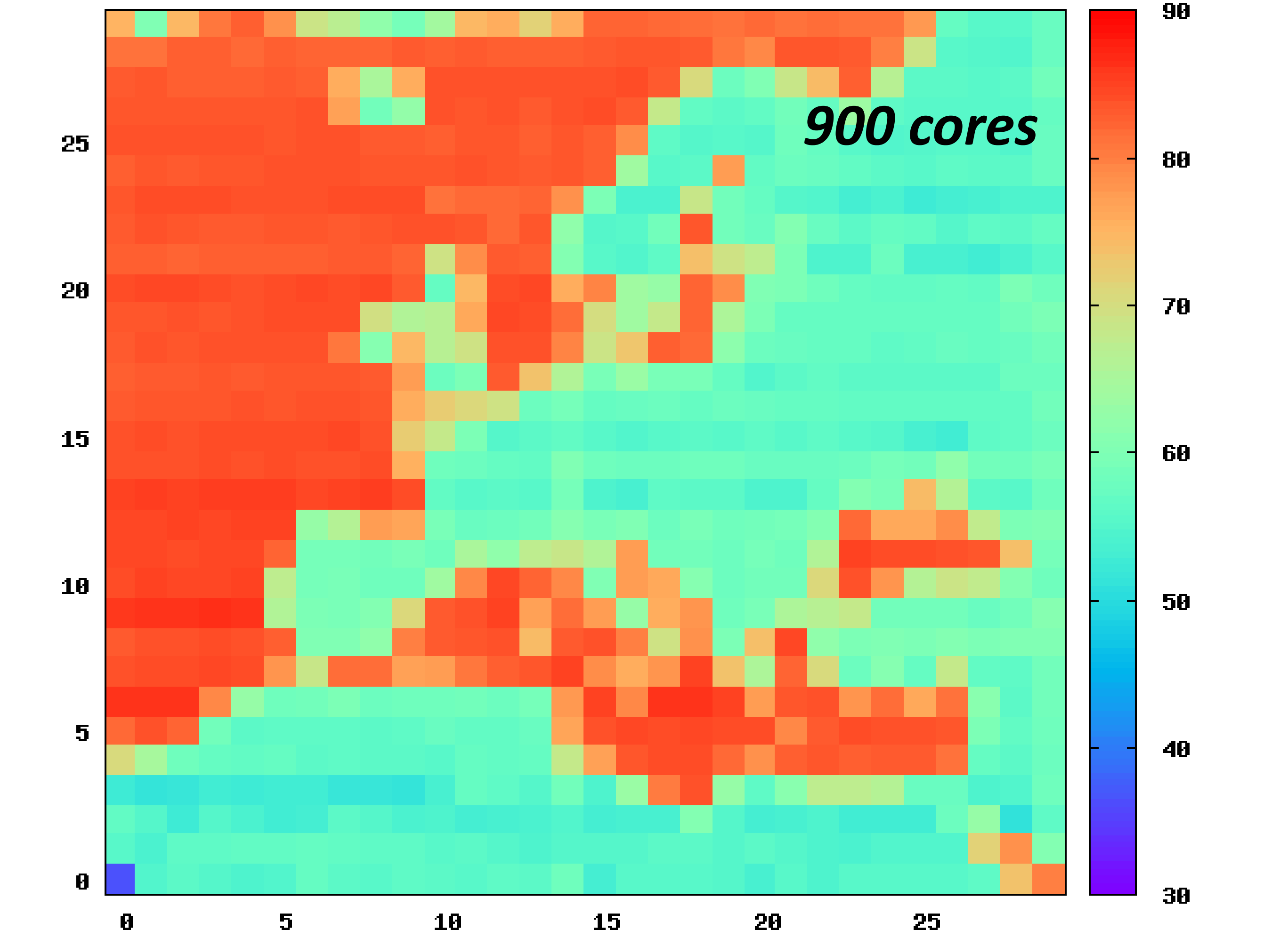
100 cores



400 cores



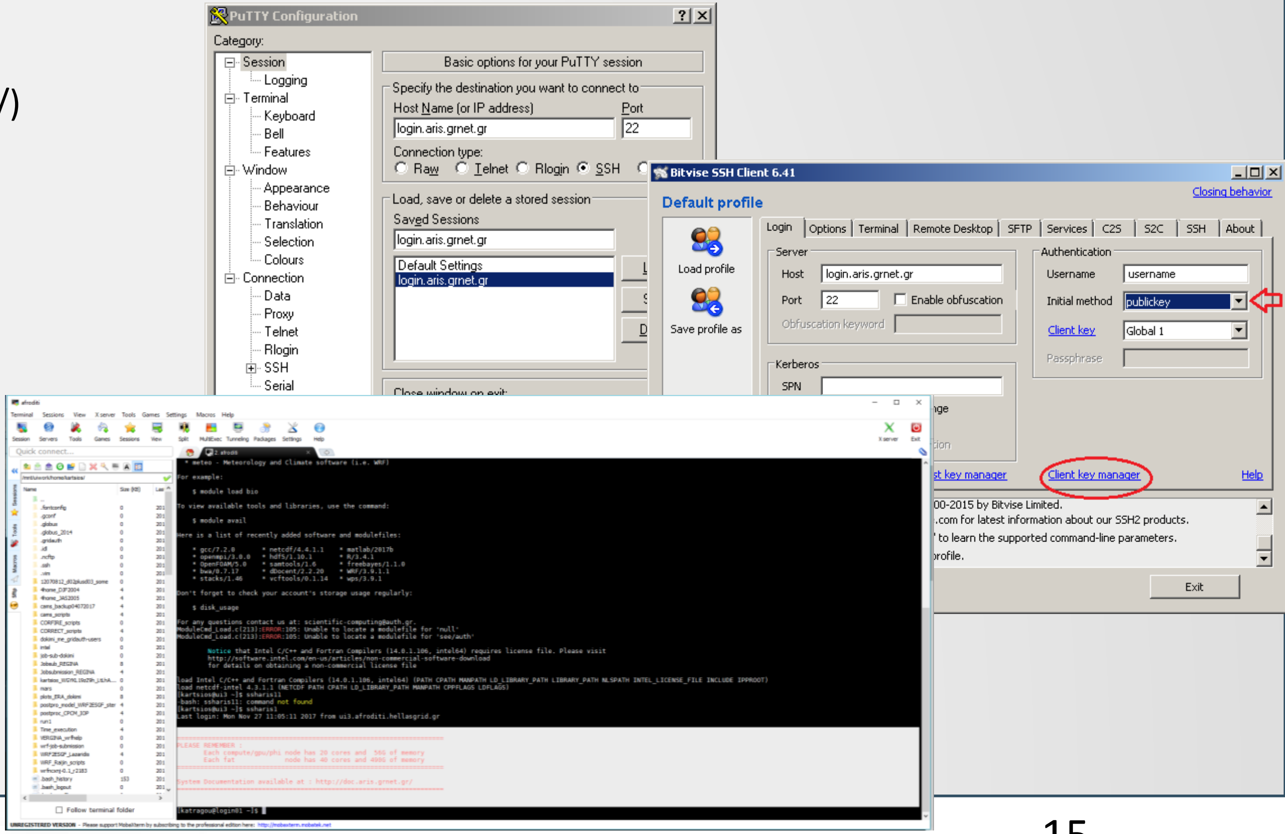
900 cores



Courtesy of Dr. Dimitris Dellis

Some basics…

- Must be familiarized with LINUX basic commands
- In order to connect to HPC ARIS infrastructure you will need a ***ssh client*** if operating from a WINDOWS PC
  - PUTTY (http://www.chiark.greenend.org.uk/~sgtatham/putty/)
  - BitVise (https://www.bitvise.com/ssh-client-download)
  - MobaXterm (https://mobaxterm.mobatek.net/)
  - PowerShell
- On Linux/Mac just use the terminal
- ssh -YC username@login.aris.grnet.gr
  - All you need at http://doc.aris.grnet.gr/



Host Name (or IP address): login.aris.grnet.gr
Port: 22
Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH
Saved Sessions: login.aris.grnet.gr
Default Settings
login.aris.grnet.gr

Host: login.aris.grnet.gr
Port: 22 □ Enable obfuscation
Username: username
Initial method: publickey
Client key: Global 1

00-2015 by Bitvise Limited.
.com for latest information about our SSH2 products.
' to learn the supported command-line parameters.
rofile.

```
  * meteo - Meteorology and Climate software (i.e. WRF)

For example:

    $ module load bio

To view available tools and libraries, use the command:

    $ module avail

Here is a list of recently added software and modulefiles:

    * gcc/7.2.0       * netcdf/4.4.1.1    * matlab/2017b
    * openmpi/3.0.0   * hdf5/1.10.1       * R/3.4.1
    * OpenFOAM/5.0    * samtools/1.6      * freebayes/1.1.0
    * bwa/0.7.17      * dDocent/2.2.20    * WRF/3.9.1.1
    * stacks/1.46     * vcftools/0.1.14   * wps/3.9.1

Don't forget to check your account's storage usage regularly:

    $ disk_usage

For any questions contact us at: scientific-computing@auth.gr.
ModuleCmd_Load.c(213):ERROR:105: Unable to locate a modulefile for 'null'
ModuleCmd_Load.c(213):ERROR:105: Unable to locate a modulefile for 'see/auth'

        Notice that Intel C/C++ and Fortran Compilers (14.0.1.106, intel64) requires license file. Please visit
        http://software.intel.com/en-us/articles/non-commercial-software-download
        for details on obtaining a non-commercial license file

load Intel C/C++ and Fortran Compilers (14.0.1.106, intel64) (PATH CPATH MANPATH LD_LIBRARY_PATH LIBRARY_PATH NLSPATH INTEL_LICENSE_FILE INCLUDE IPPROOT)
load netcdf-intel 4.3.1.1 (NETCDF PATH CPATH LD_LIBRARY_PATH MANPATH CPPFLAGS LDFLAGS)
[kartsios@ui3 ~]$ ssharis11
-bash: ssharis11: command not found
[kartsios@ui3 ~]$ ssharis1
Last login: Mon Nov 27 11:05:11 2017 from ui3.afroditi.hellasgrid.gr

=====================================================================
PLEASE REMEMBER :
      Each compute/gpu/phi node has 20 cores and  56G of memory
      Each fat             node has 40 cores and 490G of memory
=====================================================================

System Documentation available at : http://doc.aris.grnet.gr/
=====================================================================

[katragou@login01 ~]$
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: http://mobaxterm.mobatek.net

WRF folders

```
wget https://github.com/wrf-model/WRF/archive/refs/tags/v4.5.1.tar.gz
```

- cd WRF /

Some source code directories:

- **dyn_em/** Directory for ARW dynamics and numerics
- **dyn_exp/** Directory for a 'toy' dynamic core
- **external/** Directory containing external packages, such as those for IO, time keeping, and MPI
- **frame/** Directory containing modules for the WRF framework
- **inc/** Directory containing 'include' files
- **main/** Directory for main routines, such as wrf.F, and all executables after compilation
- **phys/** Directory for all physics modules
- **share/** Directory containing mostly modules for WRF mediation layer and WRF I/O
- **tools/** Directory containing tools for developers

Scripts:

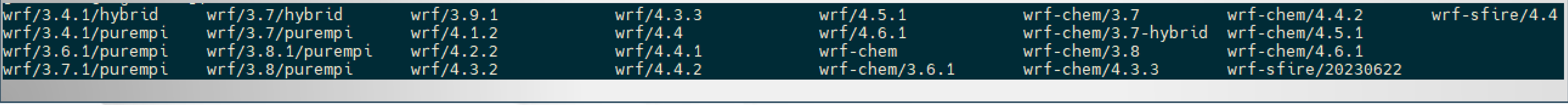
- clean Script to clean created files and executables
- compile Script for compiling the WRF code
- configure Script to configure the *configure.wrf* file for compilation

Makefile Top-level makefile

- **Registry/** Directory for WRF Registry files
- **arch/** Directory where compile options are gathered
- **run/** Directory where one may run WRF
- **test/** Directory that contains several test case directories, may be used to run WRF

Configure and Compile

- What is your scientific or practical objectives?
- If you are only planning on running **Idealized Cases**, you would need:
  - *WRF ARW Model + post-processing tools*
- If you are planning on running **Real Cases**, you would need:
  - *WPS + WRF ARW Model + post-processing tools*
- If you are planning on running **Real Cases with Variational Analysis**, you would need:
  - *WPS + WRF-Var + WRF ARW Model + post-processing tools*
- Download the code from (git clone):
  - *https://github.com/wrf-model/WRF/releases?page=1*
- Or just...

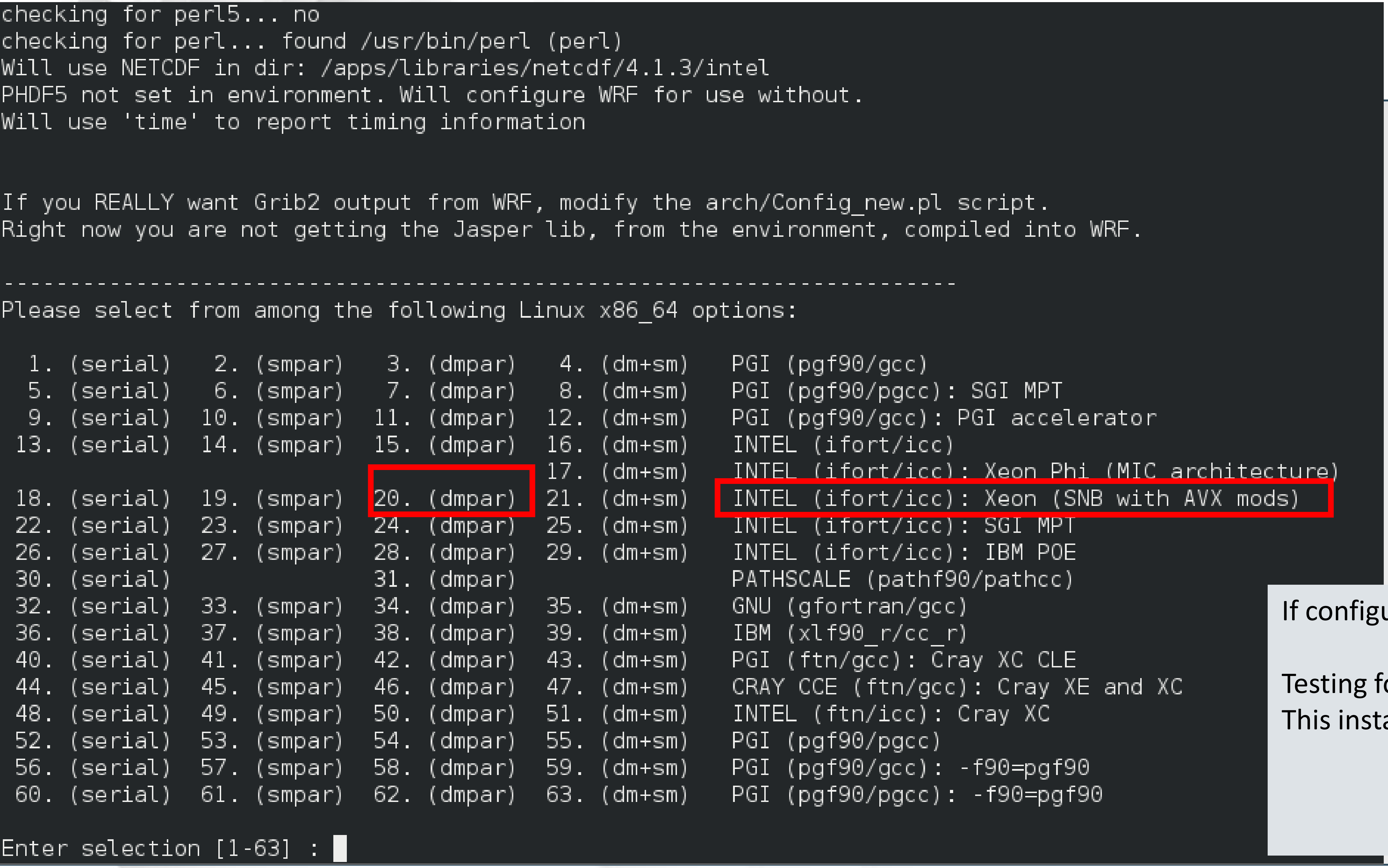


```
wrf/3.4.1/hybrid     wrf/3.7/hybrid       wrf/3.9.1            wrf/4.3.3            wrf/4.5.1            wrf-chem/3.7         wrf-chem/4.4.2       wrf-sfire/4.4
wrf/3.4.1/purempi    wrf/3.7/purempi      wrf/4.1.2            wrf/4.4              wrf/4.6.1            wrf-chem/3.7-hybrid  wrf-chem/4.5.1
wrf/3.6.1/purempi    wrf/3.8.1/purempi    wrf/4.2.2            wrf/4.4.1            wrf-chem             wrf-chem/3.8         wrf-chem/4.6.1
wrf/3.7.1/purempi    wrf/3.8/purempi      wrf/4.3.2            wrf/4.4.2            wrf-chem/3.6.1       wrf-chem/4.3.3       wrf-sfire/20230622
```

Configure and Compile

- Set the environment correctly
  - export WRF_EM_CORE=1
  - export WRFIO_NCD_LARGE_FILE_SUPPORT=1
- In order to configure the code a number of modules must be loaded
- module load
  - gnu/9.3.0
  - intel/18.0.5
  - intelmpi/2018.5
  - netcdf/4.4.1/intel
  - udunits2/2.2.19
  - jasper/1.900.1
  - hdf5/1.8.17/intel
  - szip/2.1
  - gsl/2.2.1
- libpng and zlib are already loaded to the system

./configure



```
checking for perl5... no
checking for perl... found /usr/bin/perl (perl)
Will use NETCDF in dir: /apps/libraries/netcdf/4.1.3/intel
PHDF5 not set in environment. Will configure WRF for use without.
Will use 'time' to report timing information


If you REALLY want Grib2 output from WRF, modify the arch/Config_new.pl script.
Right now you are not getting the Jasper lib, from the environment, compiled into WRF.

----------------------------------------------------------------------
Please select from among the following Linux x86_64 options:

  1. (serial)   2. (smpar)   3. (dmpar)   4. (dm+sm)   PGI (pgf90/gcc)
  5. (serial)   6. (smpar)   7. (dmpar)   8. (dm+sm)   PGI (pgf90/pgcc): SGI MPT
  9. (serial)  10. (smpar)  11. (dmpar)  12. (dm+sm)   PGI (pgf90/gcc): PGI accelerator
 13. (serial)  14. (smpar)  15. (dmpar)  16. (dm+sm)   INTEL (ifort/icc)
                                         17. (dm+sm)   INTEL (ifort/icc): Xeon Phi (MIC architecture)
 18. (serial)  19. (smpar)  20. (dmpar)  21. (dm+sm)   INTEL (ifort/icc): Xeon (SNB with AVX mods)
 22. (serial)  23. (smpar)  24. (dmpar)  25. (dm+sm)   INTEL (ifort/icc): SGI MPT
 26. (serial)  27. (smpar)  28. (dmpar)  29. (dm+sm)   INTEL (ifort/icc): IBM POE
 30. (serial)               31. (dmpar)                PATHSCALE (pathf90/pathcc)
 32. (serial)  33. (smpar)  34. (dmpar)  35. (dm+sm)   GNU (gfortran/gcc)
 36. (serial)  37. (smpar)  38. (dmpar)  39. (dm+sm)   IBM (xlf90_r/cc_r)
 40. (serial)  41. (smpar)  42. (dmpar)  43. (dm+sm)   PGI (ftn/gcc): Cray XC CLE
 44. (serial)  45. (smpar)  46. (dmpar)  47. (dm+sm)   CRAY CCE (ftn/gcc): Cray XE and XC
 48. (serial)  49. (smpar)  50. (dmpar)  51. (dm+sm)   INTEL (ftn/icc): Cray XC
 52. (serial)  53. (smpar)  54. (dmpar)  55. (dm+sm)   PGI (pgf90/pgcc)
 56. (serial)  57. (smpar)  58. (dmpar)  59. (dm+sm)   PGI (pgf90/gcc): -f90=pgf90
 60. (serial)  61. (smpar)  62. (dmpar)  63. (dm+sm)   PGI (pgf90/pgcc): -f90=pgf90

Enter selection [1-63] :
```

If configuration is successful:

Testing for NetCDF, C and Fortran compiler
This installation of NetCDF is 64-bit
C compiler is 64-bit
Fortran compiler is 64-bit
It will build in 64-bit

Configure and Compile

- Edit configure.wrf file
- Change the following lines:
  - DM_FC = mpif90 -f90=$(SFC)
  - DM_CC = mpicc -cc=$(SCC)
  - LIB_EXTERNAL = \
  - -L$(WRF_SRC_ROOT_DIR)/external/io_netcdf -lwrfio_nf -L/apps/libraries/netcdf/4.4.1/intel/lib -lnetcdff –lnetcdf
- to
  - DM_FC = mpiifort
  - DM_CC = mpiicc
  - LIB_EXTERNAL = \
  - -L$(WRF_SRC_ROOT_DIR)/external/io_netcdf -lwrfio_nf -L/apps/libraries/netcdf/4.4.1/intel/lib -lnetcdff -lnetcdf -L/apps/libraries/hdf5/1.8.17/intel/lib -lhdf5hl_fortran -lhdf5_hl -lhdf5_fortran -lhdf5 -lm -lz
- Compile: ./compile em_real >& log.compile &
- To monitor compilation process: tail –f log.compile

Configure and Compile

- If successful then at the end of the log.compile file:

```
=================================================
build started:   Mon May 27 17:50:42 EEST 2024
build completed: Mon May 27 19:34:39 EEST 2024

--->                  Executables successfully built                  <---

-rwxr-xr-x 1 user user 62429320 May 27 19:34 main/ndown.exe
-rwxr-xr-x 1 user user 62475256 May 27 19:34 main/real.exe
-rwxr-xr-x 1 user user 61442920 May 27 19:34 main/tc.exe
-rwxr-xr-x 1 user user 70757584 May 27 19:32 main/wrf.exe

=================================================
```

- Or just…
    - module load wrf/4.5.1

Build and Compile WPS

https://github.com/wrf-model/WPS/releases

```
wget https://github.com/wrf-model/WPS/archive/refs/tags/v4.5.tar.gz
```

- WPS program can be built either in serial or parallel mode
  - Usually, we need parallel mode if we are going for climate runs
- cd WPS
  - ./configure
  - ./compile >& log.compile &
- If your compilation is successful, you should see these executables created
  - geogrid.exe -> geogrid/src/geogrid.exe   Generates static data
  - metgrid.exe -> metgrid/src/metgrid.exe   Generates input data for WRF
  - ungrib.exe -> ungrib/src/ungrid.exe   Unpacks GRIB data*

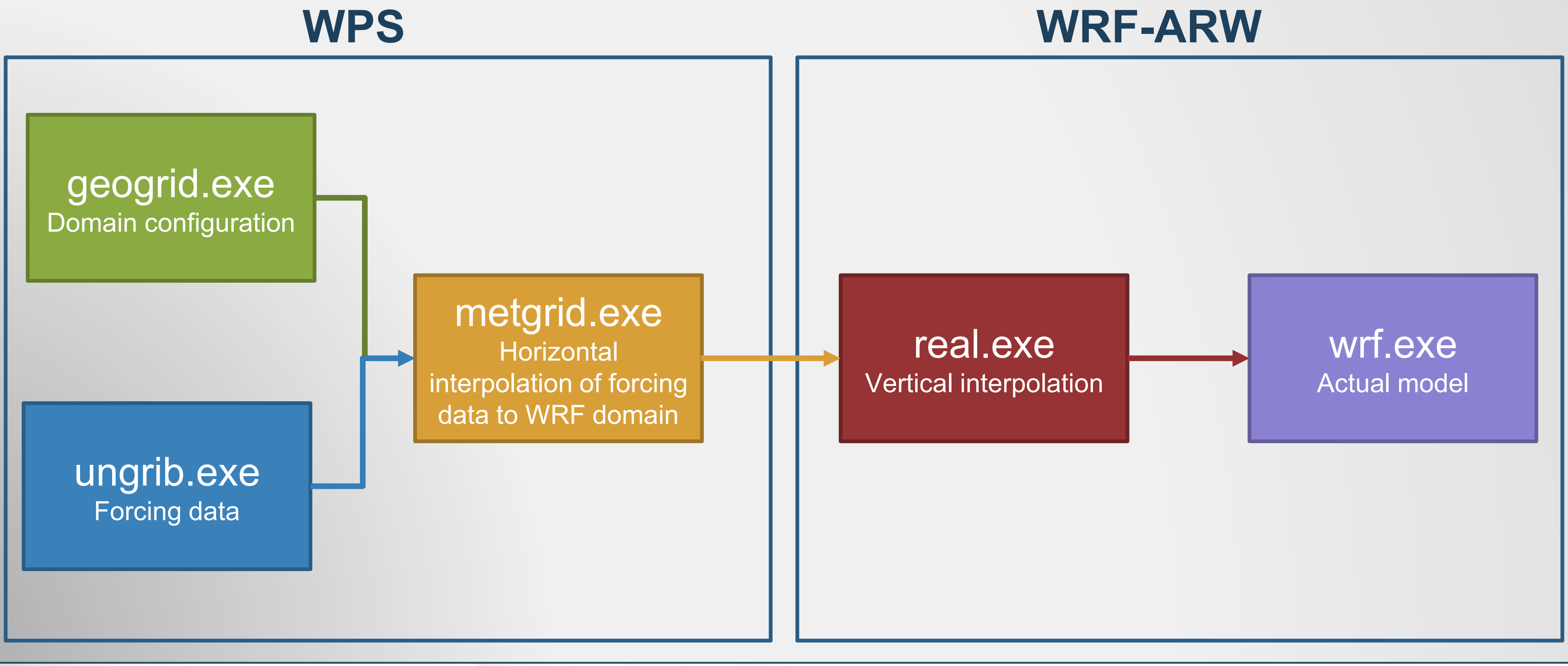
*You will need the jasper library for unpacking GRIB2 data

- Visit http://www2.mmm.ucar.edu/wrf/users/download/get_sources_wps_geog.html
- 2 variations of the WPS geographical input data download sets (high and low resolution)
- Choose carefully what you need
- *Usually, WPS program is running locally and all necessary met_em.d0* files can be transferred to ARIS (scp)*

Getting started…

- **Think first**! What are your objectives? Why do you need WRF?
- Get to know your problem! What are the **atmospheric processes** and at **what scales** are you focusing? **Review literature**!
- How do you plan to **verify** your results? Are there any observational data available for your case? Are you familiar with any post-processing tools?
- Always have a **strategy** plan for your simulations!

Setting up your simulation…



WPS

geogrid.exe
Domain configuration

ungrib.exe
Forcing data

metgrid.exe
Horizontal interpolation of forcing data to WRF domain

WRF-ARW

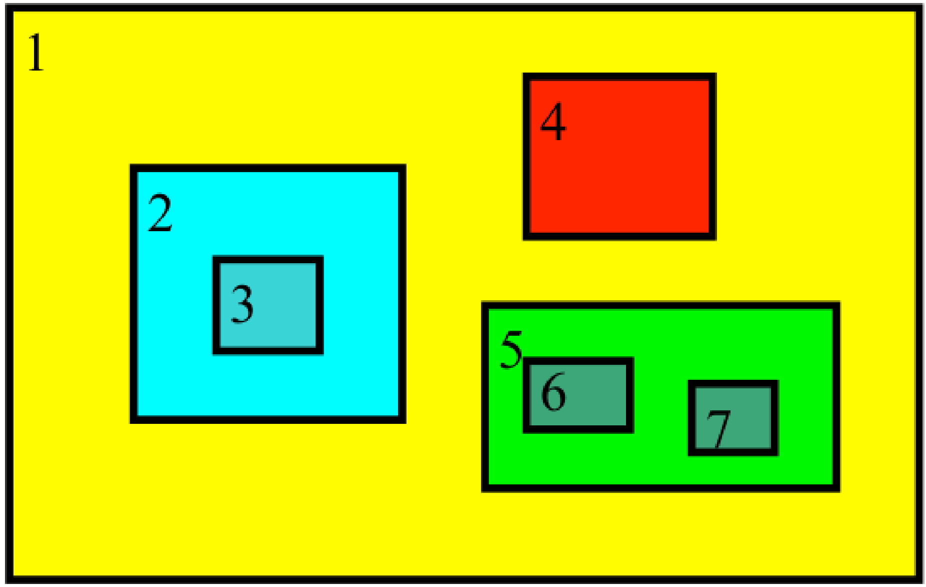
real.exe
Vertical interpolation

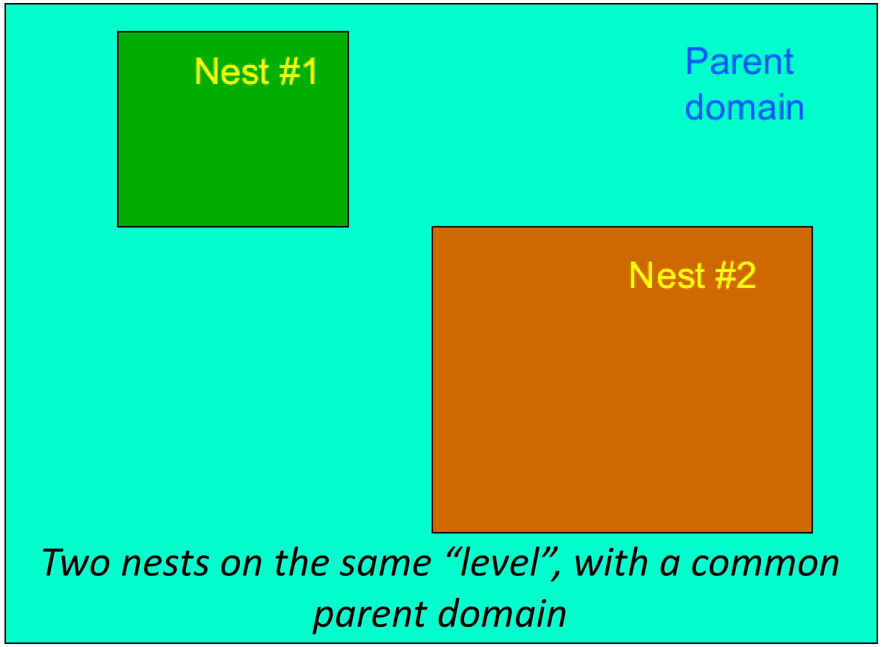
wrf.exe
Actual model

Domain configuration

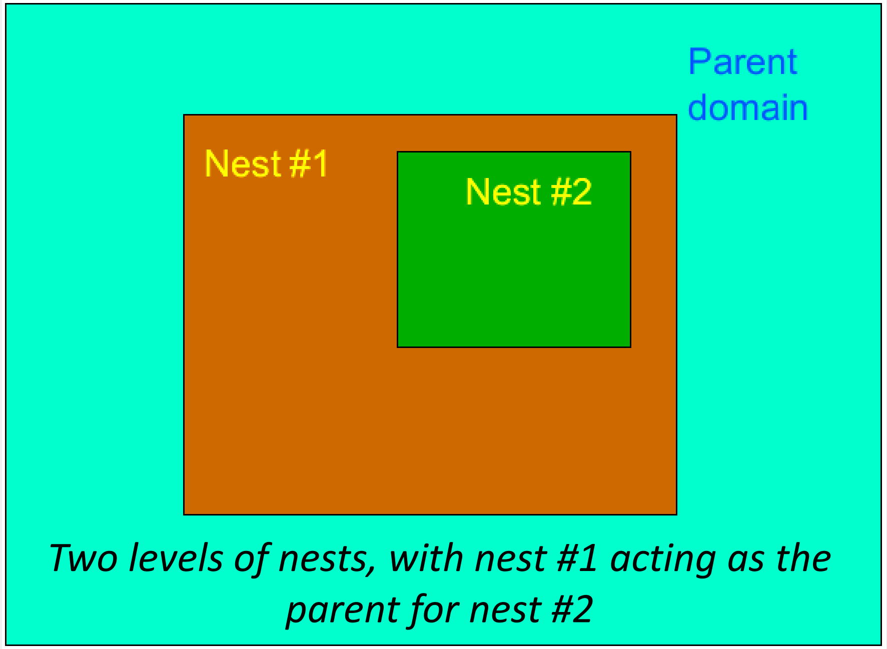
- According to your problem target your horizontal **grid resolution**
- Consider your available **initialization data** (resolution, frequency)
  - Global model, Regional model, Reanalysis?
- Most of the times a **nesting strategy** must be considered
- *namelist.wps* inside WPS folder controls domain configuration (*more on that later*)

- A *nest* is a finer-resolution model run. It may be embedded simultaneously within a coarser-resolution (parent) model run, or run independently as a separate model forecast
- The nest *covers a portion* of the parent domain, and is driven along its lateral boundaries by the parent domain
- Nesting enables running at finer resolution without the following problems:
  - Uniformly high resolution over a large domain – prohibitively expensive
  - High resolution for a very small domain with mismatched time and spatial lateral boundary conditions





Two nests on the same "level", with a common parent domain

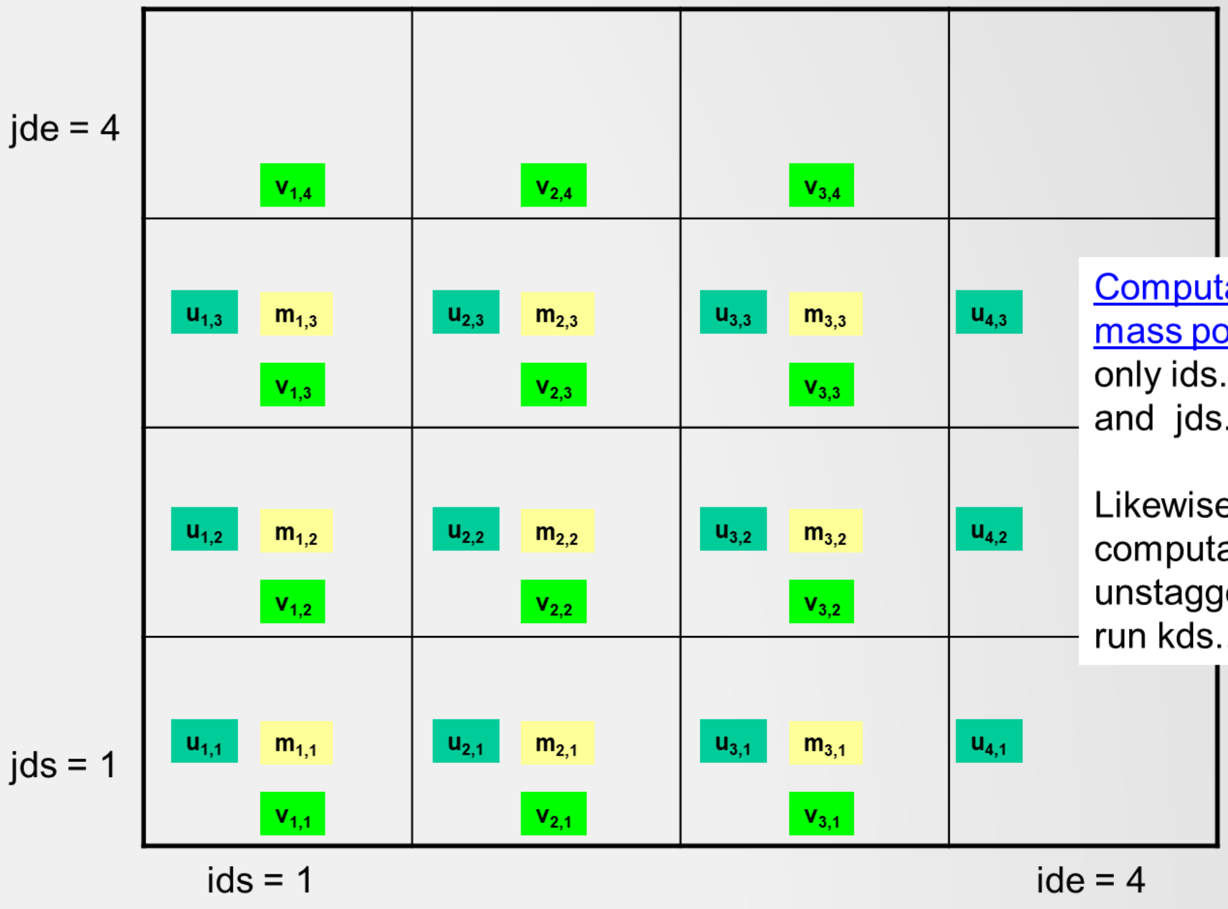


Two levels of nests, with nest #1 acting as the parent for nest #2

Grid representation in arrays

- Increasing indices in WRF arrays run
  - West to East (X, or I-dimension)
  - South to North (Y, or J-dimension)
  - Bottom to Top (Z, or K-dimension)
- Storage order in WRF is IKJ but this is a WRF Model convention, not a restriction of the WRF Software Framework
- The extent of the logical or *domain* dimensions is always the "staggered" grid dimension. That is, from the point of view of a non-staggered dimension, there is always an extra cell on the end of the domain dimension.

Grid Indices Mapped onto Array Indices (C-grid example)



jde = 4
jds = 1
ids = 1
ide = 4
$v_{1,4}$ $v_{2,4}$ $v_{3,4}$
$u_{1,3}$ $m_{1,3}$ $u_{2,3}$ $m_{2,3}$ $u_{3,3}$ $m_{3,3}$ $u_{4,3}$
$v_{1,3}$ $v_{2,3}$ $v_{3,3}$
$u_{1,2}$ $m_{1,2}$ $u_{2,2}$ $m_{2,2}$ $u_{3,2}$ $m_{3,2}$ $u_{4,2}$
$v_{1,2}$ $v_{2,2}$ $v_{3,2}$
$u_{1,1}$ $m_{1,1}$ $u_{2,1}$ $m_{2,1}$ $u_{3,1}$ $m_{3,1}$ $u_{4,1}$
$v_{1,1}$ $v_{2,1}$ $v_{3,1}$

Computation over mass points runs only ids..ide-1 and jds..jde-1

Likewise, vertical computation over unstaggered fields run kds..kde-1

namelist.wps

```
&share
 wrf_core = 'ARW’,
 max_dom = 3,
 start_date = '2006-08-16_12:00:00','2006-08-16_12:00:00’,
'2006-08-16_12:00:00'
 end_date   = '2006-08-16_18:00:00','2006-08-16_18:00:00’,
'2006-08-16_18:00:00'
 interval_seconds = 21600
 io_form_geogrid = 2,
&END

&ungrib
 out_format = 'WPS',
 prefix = 'FILE',
&END

&metgrid
 !constants_name = ‘TAVGSFC',
 fg_name = 'FILE',
 io_form_metgrid = 2,
&END
```

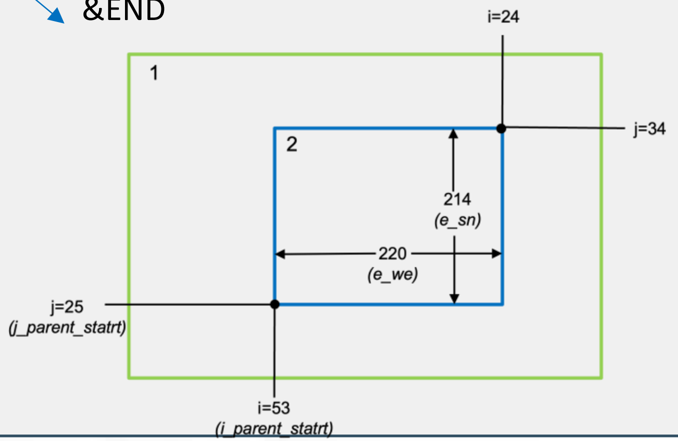
```
&geogrid
 parent_id         = 1,1,2,
 parent_grid_ratio = 1,3,3,
 i_parent_start    = 1,155,100,
 j_parent_start    = 1,50,110,
 e_we          = 368,436,640,
 e_sn          = 263,394,622,
 geog_data_res = '30s','30s','30s’,
 dx = 15000,
 dy = 15000,
 map_proj =  'lambert’,
 ref_lat   = 39.694,
 ref_lon   = 18.202,
 truelat1  = 39.694,
 truelat2  = 39.694,
 stand_lon = 18.202,
 geog_data_path = ‘/work/pr001/user/geog’,
 ref_x = 190.0,
 ref_y = 124.0,
&END
```

e_we, e_sn: Domains size

ref_lat, ref_lon: Center of the domain (parent)

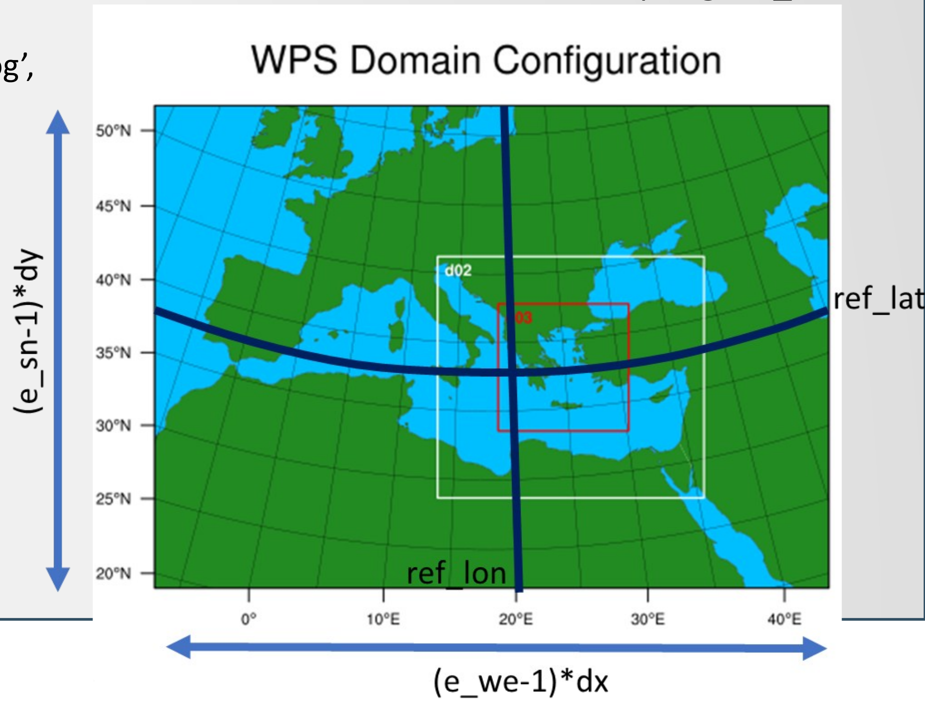
truelat1, truelat2, stand_lon: Parameters of the projection

```
./geogrid.exe (serial or parallel)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Successful completion of geogrid. !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
geo_em.d01.nc
geo_em.d02.nc (nest)
geo_em.d03.nc (nest)
```



https://www2.mmm.ucar.edu/wrf/users/wrf_users_guide/build/html/_images/wps_ij_parent_start.png

~ncl util/plotgrids_new.ncl



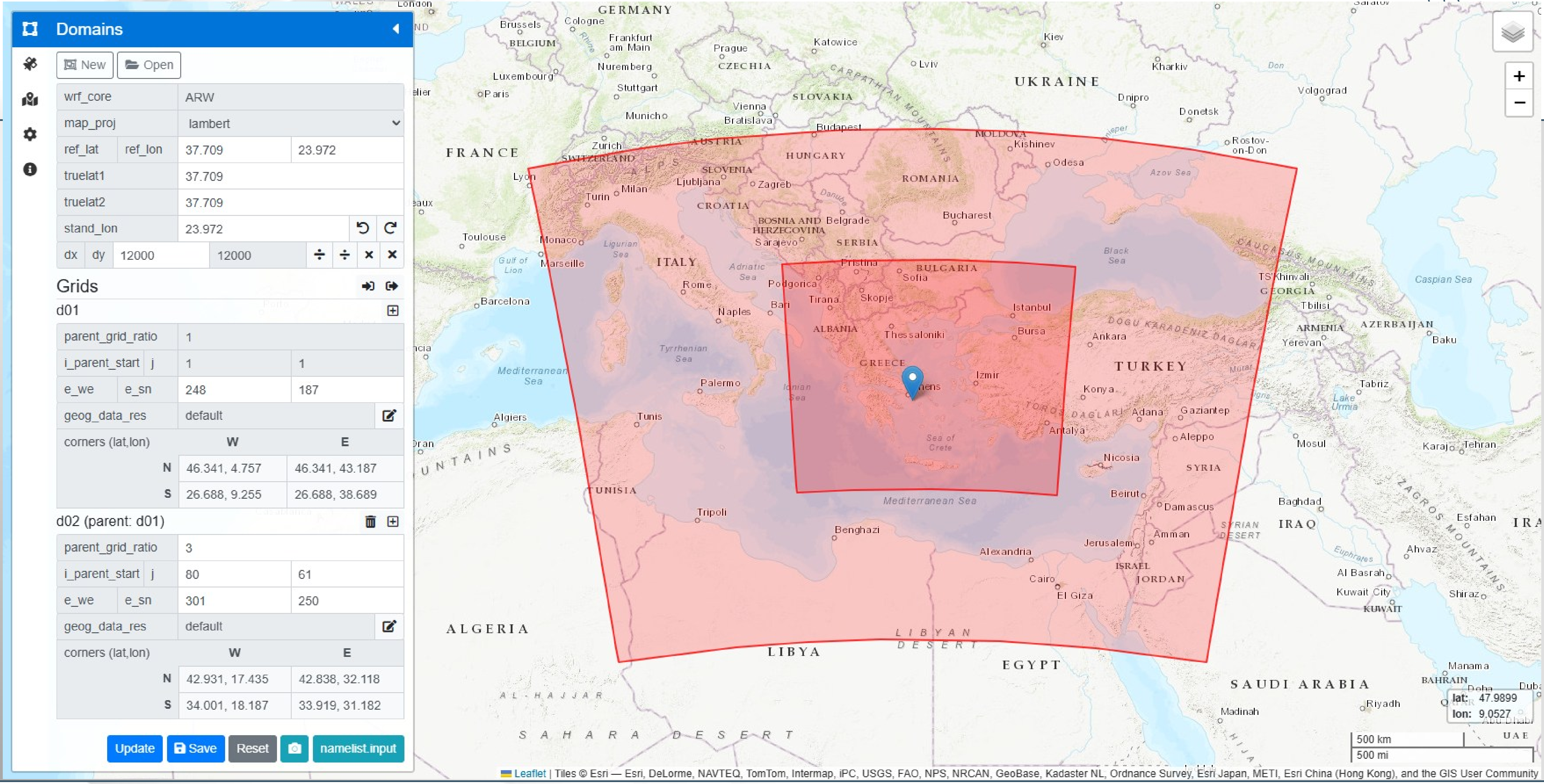
Domain configuration

- There are some NCL scripts available inside WPS folder (util) for testing your domain properties
- A nice and easy tool for domain configuration is the *WRF Domain Wizard*, an online client-side SPA for the WRF Preprocessor System (WPS)
    - *https://github.com/JiriRichter/WRFDomainWizard*
    - *https://jiririchter.github.io/WRFDomainWizard/*
    - *https://wrfdomainwizard.net/*

Hints

- An **odd** grid ratio (e.g. 3:1, 5:1) introduces parent/nest points being coincident, and a 3:1 ratio is preferred as it has been extensively tested
- *Minimum distance* between the nest boundary and the parent boundary is *4 grid cells*. You should have a much larger buffer zone
- Higher horizontal resolution will also require higher vertical resolution, typically 30-35 vertical levels; by default, larger density closer to the ground and to the model top
- Map projection: *Lambert*: mid-latitudes, *Mercator*: low-latitudes, *Lat-Lon*: global, *Rotated Lat-Lon:* regional
- Start inside-out (first the nest, move up)

https://jiririchter.github.io/WRFDomainWizard/



Domains

New | Open

wrf_core	ARW	
map_proj	lambert	
ref_lat / ref_lon	37.709	23.972
truelat1	37.709	
truelat2	37.709	
stand_lon	23.972	
dx / dy	12000	12000

Grids

d01

parent_grid_ratio	1	
i_parent_start / j	1	1
e_we / e_sn	248	187
geog_data_res	default	

corners (lat,lon)	W	E
N	46.341, 4.757	46.341, 43.187
S	26.688, 9.255	26.688, 38.689

d02 (parent: d01)

parent_grid_ratio	3	
i_parent_start / j	80	61
e_we / e_sn	301	250
geog_data_res	default	

corners (lat,lon)	W	E
N	42.931, 17.435	42.838, 32.118
S	34.001, 18.187	33.919, 31.182

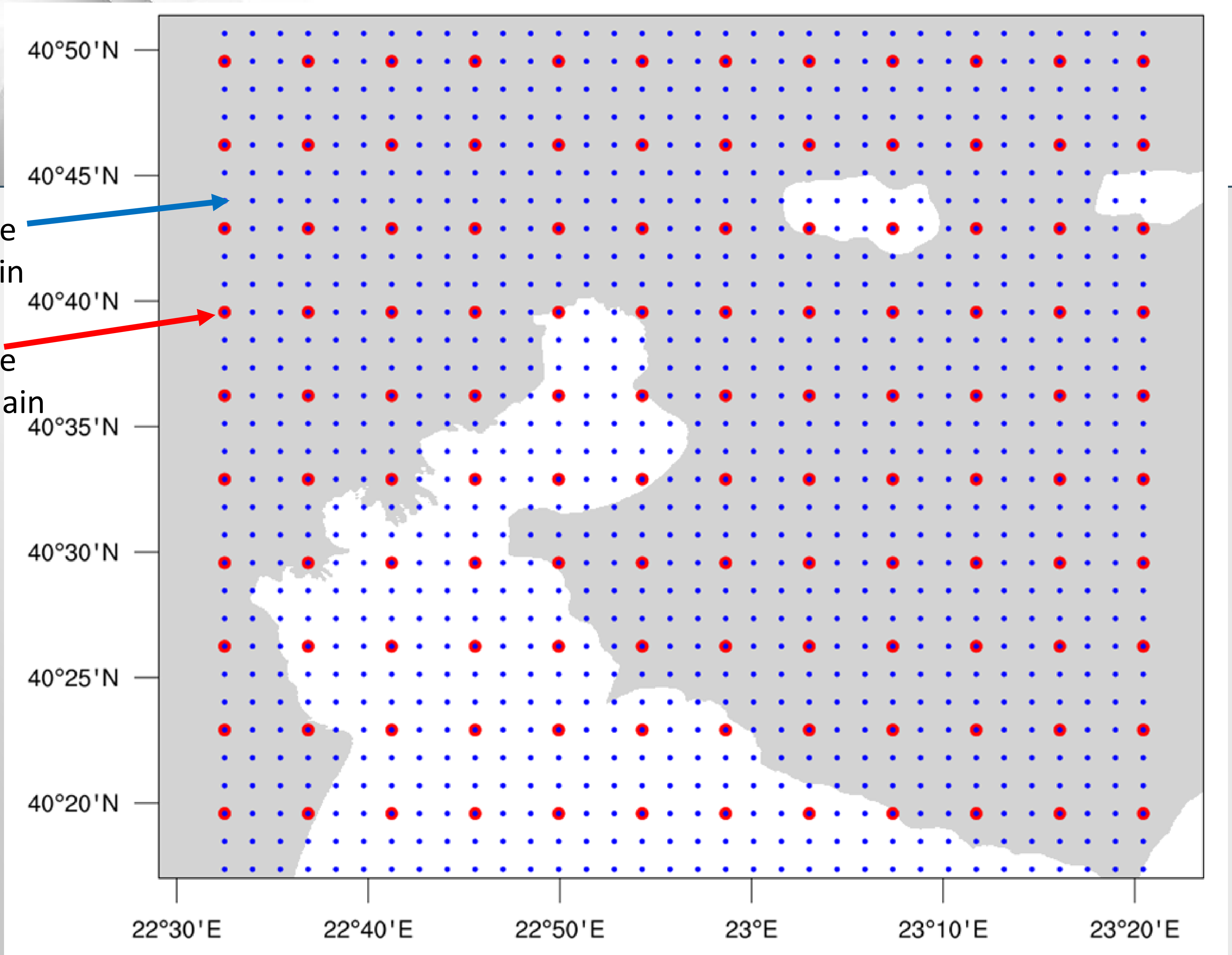
Update | Save | Reset | namelist.input

lat: 47.9899
lon: 9.0527

Leaflet | Tiles © Esri — Esri, DeLorme, NAVTEQ, TomTom, Intermap, iPC, USGS, FAO, NPS, NRCAN, GeoBase, Kadaster NL, Ordnance Survey, Esri Japan, METI, Esri China (Hong Kong), and the GIS User Community

Domain configuration

- It’s all about computational resources!
- **Computational time** = (*operation per equation*) x (*number of equations per grid box*) x (*number of grid boxes*) x (*number of time steps per simulations*)
- **Increasing spatial resolution** by a factor of **2**, the number of grid cells increase by a factor of **2^3 = 8** and with the **doubled number of time steps**, the computational time of the run increases by a factor of **2^4 = 16**
- Keep in mind that the *size of the nested domain* may need to be chosen along with *computing performance*
- If a 3:1 ratio is assumed, with the same number of grid points between the parent and the nest domain, then the fine grid will require **3x** as many time steps to keep pace with the coarse domain
- A simple nested domain forecast is approximately **4x** the cost of just the coarse domain
- **Remember!** Doubling the coarse grid points results in only a 25% nested forecast time increase



Grid points of the inner (FG) domain

Grid points of the coarse (CG) domain

EURO Greece

Domain configuration: Nesting performance

Assuming a 3:1 parent-child ratio:

- If the nest has the same number of grid points, then the amount of CPU to do a single time step for a coarse grid (CG) and a fine grid step (FG) is *approximately the same*
- Since the FG has 1/3 the grid distance, it requires 1/3 the model time step. Therefore, the FG requires **3x the CPU** to catch up with the CG domain
- If you try to cover the same area with a FG domain as a CG domain, you need ***ratio^2*** grid points
- With the associated FG time step ratio, you require ***ratio^3*** computational resources in compared to CG domain
- Thus, with a **3:1** nest ratio, a FG domain covering the same area as the CG domain requires ***27x computational resources*** (CPU)
- Assuming a **5:1** nest ratio, the FG domain for the same area as the CG would be ***125x more expensive***

Domain configuration: Nesting performance

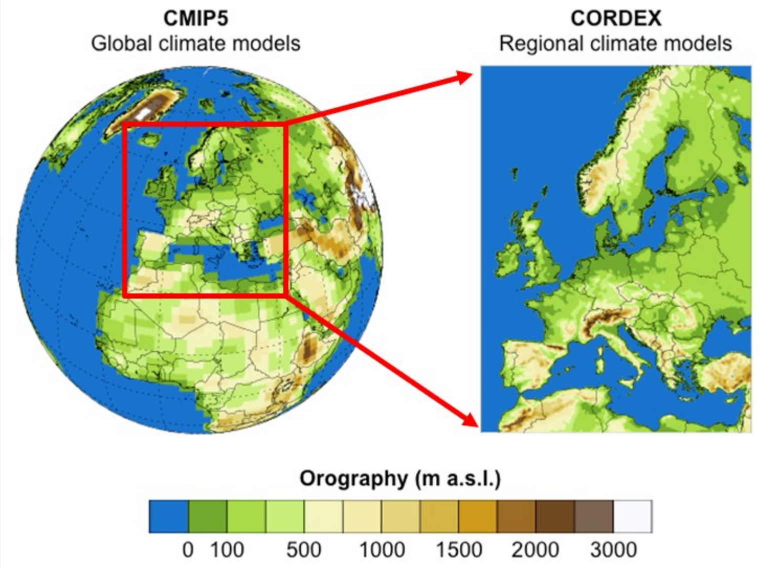
- Start with the inner-most domain. For a traditional forecast, you want everything important for that forecast to be entirely contained inside the domain.
- Then start adding parent domains at a 3:1 or 5:1 ratio. A parent should not have a smaller size (in grid points).
- Keep adding domains until the most coarse grid has a no more than a 3:1 to 5:1 ratio to the initialization (first guess) data.
- Larger domains tend to be better than smaller domains (although not in all cases).
- *Consider a 2 km resolution grid with 100x100 grid points. An upper level parcel moves at 200 km/h, meaning that within a couple of hours, most of the upper-level initial data will be swept out of the domain.*

Preparing the forcing data

- **What does the model need as input?**
  - 3D input data (*Temperature, Relative/Specific Humidity, Geopotential height, Wind U & V*)
  - 2D input data (*Surface pressure, Sea level pressure, Skin temperature, 2m Temperature, 2m Relative / Absolute / Specific humidity, 10m wind U & V*)

- Recommended but not necessary:
  - Sea surface temperature changes (*SSTs, for regional climate modeling this is a mandatory field*)
  - Surface height
  - Equivalent snow depth
  - Sea ice changes (*in case of missing, results will present a systematic bias over regions at high latitudes*)

- **Provided either from a GCM (CCSM4, EC-EARTH, GISS, HadGEM2, MPI etc.) or reanalysis data (FNL, ERA-Interim, ERA5 etc.)**



Preparing the forcing data

- Supposing our forcing data (ERA5) is available…
  - *`cd /work/pr001/user/WRFV4.5.1/WPS`*
  - Link the correct Vtable to the file name “Vtable” in the run directory
  - Some Vtables are provided with WPS in the *./WPS/ungrib/Variable_Tables* directory
  - Ungrib.exe always expects to find a file named Vtable in the run directory

  - *`ln -s ungrib/Variable_Tables/Vtable.ERA-interim.{pl,ml} Vtable`*

- Link GRIB files to the correct file names in the run directory
  - *`./link_grib.csh /work/pr001/user/ERA5/ERA5_*.grb`*
  - *`ls GRIBFILE.A*`*

- `./ungrib.exe` (only serial)

```
&share
  wrf_core = 'ARW’,
  max_dom = 2,
  start_date = '2006-08-27_12:00:00','2006-08-27_12:00:00’,
  end_date   = '2006-08-29_00:00:00','2006-08-29_00:00:00’,
  interval_seconds = 21600
  io_form_geogrid = 2,
&END

&ungrib
  out_format = 'WPS’,
  prefix = 'FILE',
&END
```

If the lakes are resolved in the WRF domain but they are not resolved in the input data, then it is recommended to use the alternative initialization lake SST option (avg_tsfc.exe)

Preparing the forcing data

```
• *** Starting program ungrib.exe ***
• Start_date =  2006-08-27_12:00:00 ,      End_date = 2006-08-29_00:00:00
• output format is WPS
• Path to intermediate files is ./
• ###################################################################################
• Inventory for date = 2006-08-27 12:00:00
• PRES   GEOPT    HGT      TT      UU      VV      RH      DEWPT    LANDSEA  SOILGEO  SOILHGT  PSFC     PMSL     SKINTEMP SEAICE   SST
  SNOW_DEN SNOW_EC  SNOW     SNOWH    ST000007 ST007028 ST028100 ST100289 SM000007 SM007028 SM028100 SM100289
• ------------------------------------------------------------------------------
• 1000.0  X           X       X       X       X
•  975.0  X           X       X       X       X
•  950.0  X           X       X       X       X
•  925.0  X           X       X       X       X
•  900.0  X           X       X       X       X
•  875.0  X           X       X       X       X
```

X: input data exist

O: missing

metgrid.exe

- &share and &metgrid namelist’s sections need to be edited
- `./metgrid.exe (or srun metgrid.exe)`

```
Processing domain 1 of 2
 Processing 2006-08-27_12
    FILE
 Processing 2006-08-27_18
    FILE
 Processing 2006-08-28_00
    FILE
 Processing 2006-08-28_06
    FILE
 Processing 2006-08-28_12
    FILE
 Processing 2006-08-28_18
    FILE
 Processing 2006-08-29_00
    FILE
Processing domain 2 of 2
 Processing 2006-08-27_12
    FILE
 Processing 2006-08-27_18
    FILE
 Processing 2006-08-28_00
…..

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!  Successful completion of metgrid.  !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
&share
  wrf_core = 'ARW’,
  max_dom = 2,
  start_date = '2006-08-27_12:00:00','2006-08-27_12:00:00’,
  end_date   = '2006-08-29_00:00:00','2006-08-29_00:00:00’,
  interval_seconds = 21600
  io_form_geogrid = 2,
&END

&metgrid
 fg_name = 'FILE',
 io_form_metgrid = 2,
&END
```

```
ls -1 ./met_em.d0*
     met_em.d01.2006-08-27_12:00:00.nc
     met_em.d01.2006-08-27_18:00:00.nc
     met_em.d01.2006-08-28_00:00:00.nc
     met_em.d01.2006-08-28_06:00:00.nc
     met_em.d01.2006-08-28_12:00:00.nc
     met_em.d01.2006-08-28_18:00:00.nc
     met_em.d01.2006-08-29_00:00:00.nc
     met_em.d02.2006-08-27_12:00:00.nc
     met_em.d02.2006-08-27_18:00:00.nc
     met_em.d02.2006-08-28_00:00:00.nc
     met_em.d02.2006-08-28_06:00:00.nc
     met_em.d02.2006-08-28_12:00:00.nc
     met_em.d02.2006-08-28_18:00:00.nc
     met_em.d02.2006-08-29_00:00:00.nc
```

Running the model…

- `cd WRF/test/em_real`
- `ln -s ../../../WPS/met_em.d0* .`
- Edit *namelist.input** according to your application
- Run real.exe (serial or parallel)
  - real.exe executable is responsible for the vertical interpolation of the input data
  - Creates **wrfbdy_d01** & **wrfinput_d0*** (in some cases also **wrflowinp_d0***)
- Run wrf.exe (parallel)
  - wrf.exe executable is THE MODEL itself
  - Creates **wrfout_d0*, wrfxtrm_d0*, wrfpress_d0*** files (and some other…)

```
#!/bin/bash -l
###############################
#SBATCH --job-name=WRF
#SBATCH --ntasks=240
#SBATCH --nodes=12
#SBATCH --ntasks-per-node=20
#SBATCH --cpus-per-task=1
#SBATCH --time=48:00:00
#SBATCH --partition=compute
#SBATCH --mem=56G
#SBATCH --account=pr003005
#SBATCH --output=mpijob.%j.out
#SBATCH --error=mpijob.%j.err
###############################
#LOAD MODULES
#
export WRF_EM_CORE=1
export WRFIO_NCD_LARGE_FILE_SUPPORT=1
export I_MPI_FABRICS=shm:dapl
```

```
# RUN THE PROGRAM
 t0=$(date +%s)
 cd $RUNDIR
#
 ulimit unlimited
 ulimit -c unlimited
 ulimit -s unlimited
 srun real.exe
 wait
 srun wrf.exe
 t1=$(date +%s)
 echo "WRF execution: $(( $t1 - $t0 )) sec"
```

*https://www2.mmm.ucar.edu/wrf/users/wrf_users_guide/build/html/namelist_variables.html

Running the model…

- For example, in **climate simulations** (not only) several options in *namelist.input* must be considered:

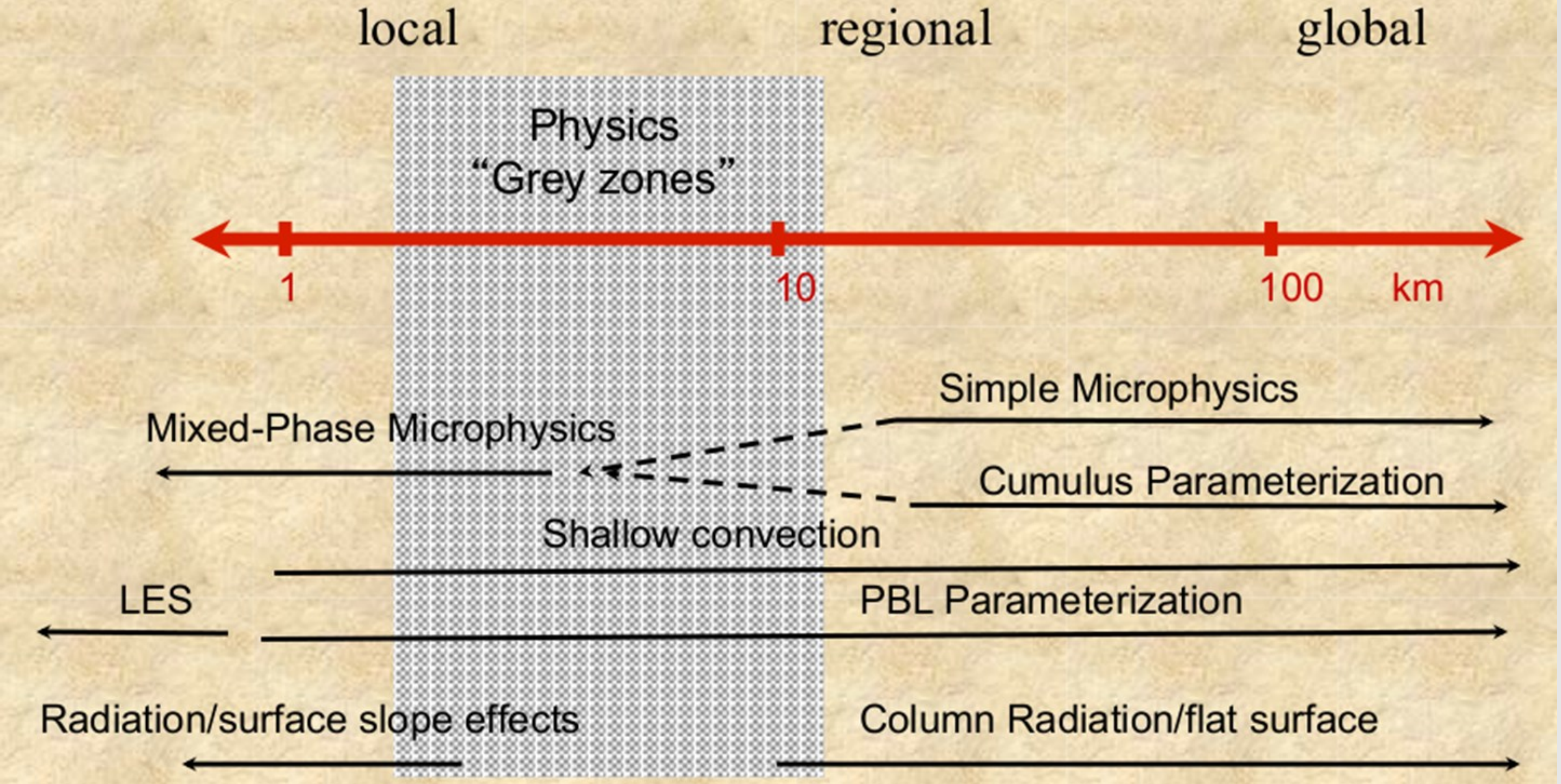
&time_control section

restart	= .true.,	if it is a restart run or not (.false.)
restart_interval	= 105120,	in minutes (how often a restart file is produced, equal or less than simulation length)
debug_level	= 0,	print additional information during integration
auxhist3_outname	= 'wrfxtrm_<domain>_<date>,	name of the file containing some statistics
auxhist3_interval	= 1440,	in minutes (how often statistics are computed)
frames_per_auxhist3	= 7,	number of timestamps per wrfxtrm file
auxinput4_inname	= 'wrflowinp_d<domain>',	created by real.exe, information for update SST, sea-ice etc.
auxinput4_interval	= 360,	in minutes (also in &physics must set sst_update = 1)
override_restart_timers	= .true.,	uses all output intervals given by the wrfrst files
write_hist_at_0h_rst	= .true.,	If history output is desired at the time of restart

- Also consider at &physics section in namelist.input:

sst_skin	= 1,	mandatory for climate simulations (calculate skin SST)
sst_update	= 1,	employ time-varying sea-surface temperature
tmn_update	= 1,	update deep soil temperature
lagday	= 150,	days over which deep soil temp is computed using skin temperature
usemonalb	=.true.,	use monthly albedo fields from geogrid, instead of table values
bucket_mm	= 1000.0,	bucket reset values for water accumulation

Model configuration: Physics



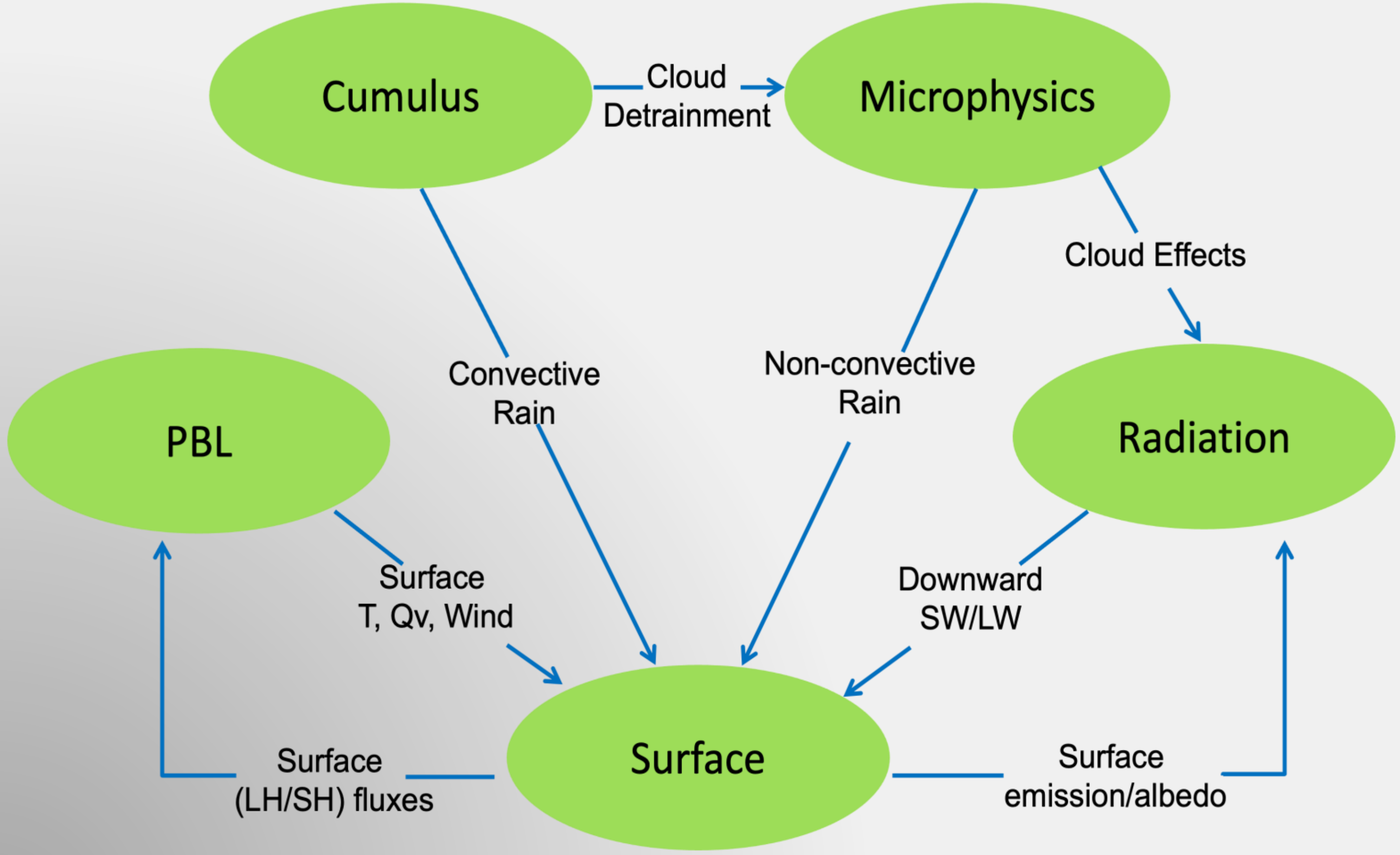
http://www2.mmm.ucar.edu/wrf/users/workshops/WS2014/ppts/best_prac_wrf.pdf

Model configuration: Physics

- **A large number of schemes available**
- Which processes are important? *Review literature*. What others did?
- Different Schemes ⟶ Different Results
- A given set of physics will perform differently depending on domain size, location, initialization and phenomenon of interest

- Consider first well documented (tried) schemes
- Consider grid size when choosing sophistication of microphysics
- You *don't need* a complex scheme for a 10 km grid
- You do need a microphysical scheme with *graupel for convection-resolving grids*
- It is better if you have consistent physics between the domains (must have if 2-way nesting)
- Cumulus parameterization:
  - For grid resolutions > 10 km you must activate it
  - For grid resolutions < 3 km probably not
  - For grid resolutions 3-10 km, best to avoid convective cases (grey zone)

Model configuration: Physics



https://www2.mmm.ucar.edu/wrf/users/wrf_users_guide/build/html/_images/phys_scheme_interaction.png

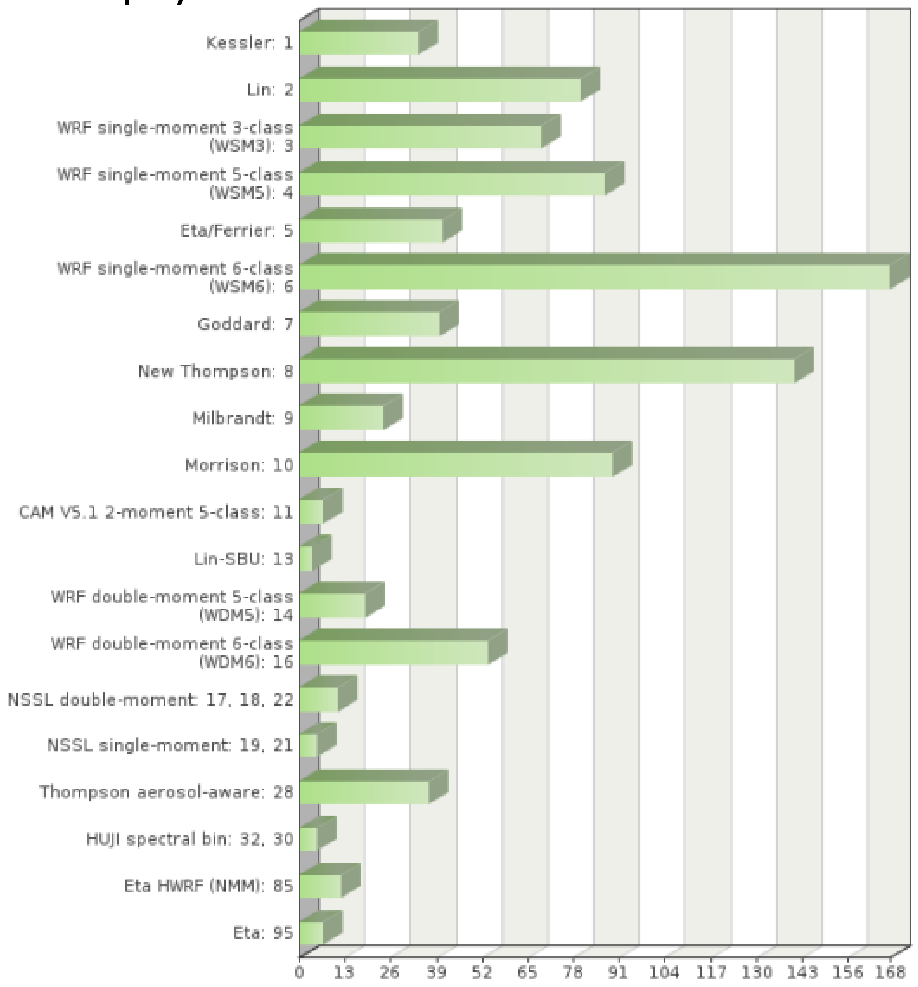
- 27 Microphysics schemes
- 14 PBL schemes
- 14 Cumulus schemes
- 8 Radiation schemes
- 7 Land Surface schemes
- 9 Shallow Convect. Schemes
- 7 Surface Layer schemes
- 3 Urban physics schemes
- 2 Ocean physics schemes

A large number of combinations!

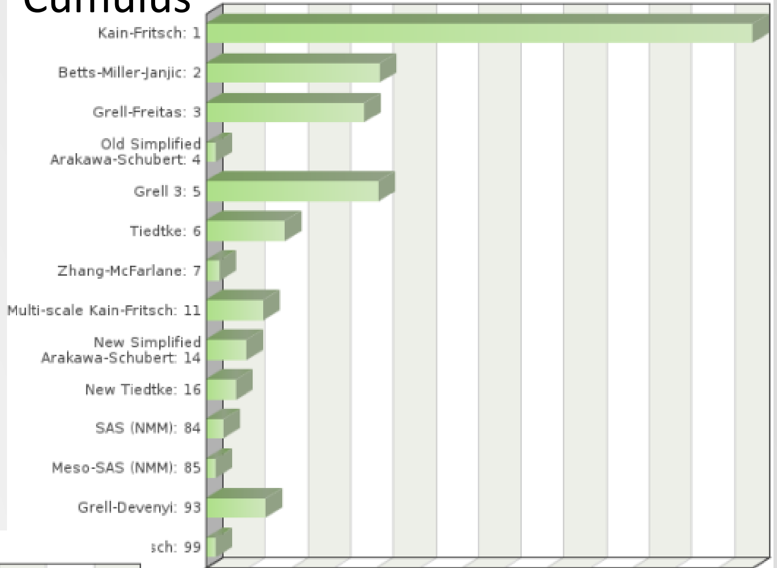
WRF physics survey

https://www2.mmm.ucar.edu/wrf/users/physics/wrf_physics_survey.pdf

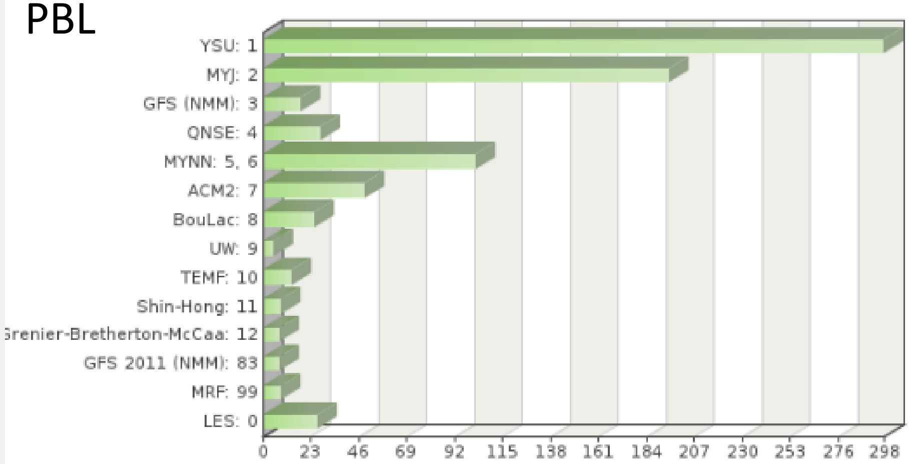
Microphysics



Cumulus

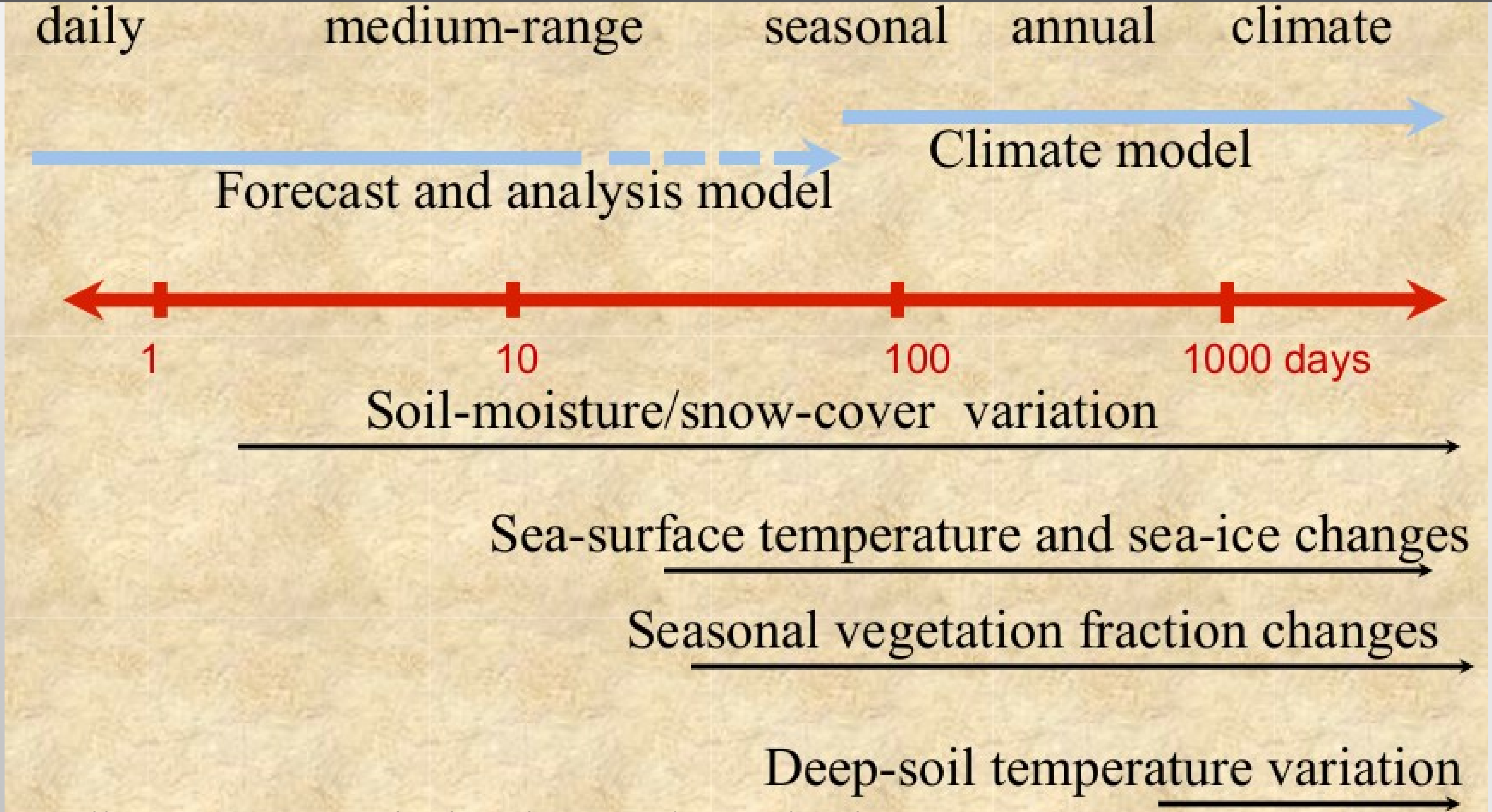


PBL



An old survey (2015)…

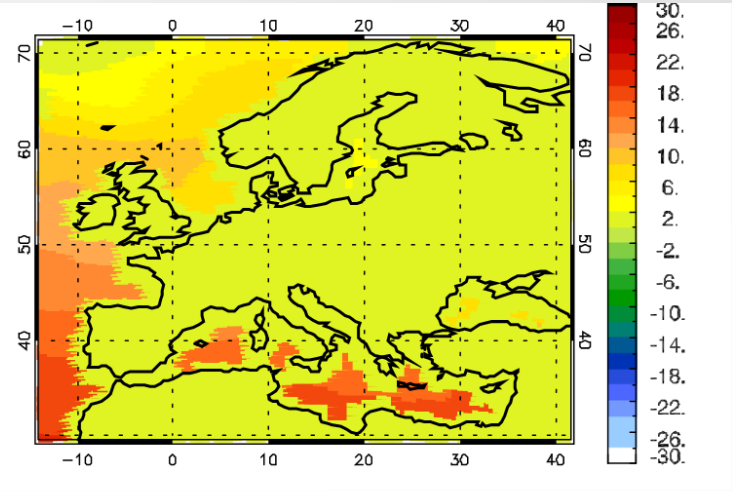
Model configuration: Physics


daily
medium-range
seasonal
annual
climate
Climate model
Forecast and analysis model
1
10
100
1000 days
Soil-moisture/snow-cover variation
Sea-surface temperature and sea-ice changes
Seasonal vegetation fraction changes
Deep-soil temperature variation

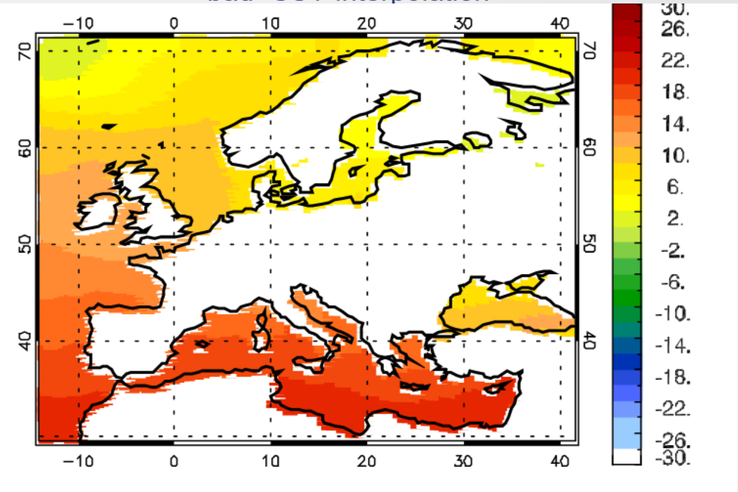
http://www2.mmm.ucar.edu/wrf/users/workshops/WS2014/ppts/best_prac_wrf.pdf

Model configuration: Initialization and Spin up

- Usually, model problems occur due to initialization (poor initial conditions)
  - Poor soil temperature and moisture representation
  - Missing or inappropriate sea surface temperatures (SSTs) masking at coastlines
  - Wrong representation of land/sea mask
- Check your inputs carefully!
  - met_em.d0*, wrfinput_d0*



“bad” SST interpolation



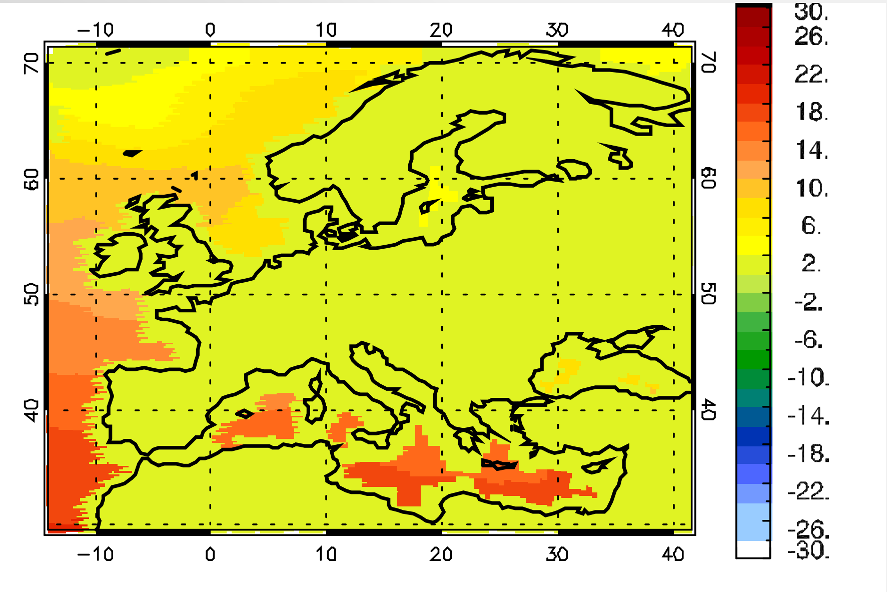
Corrected

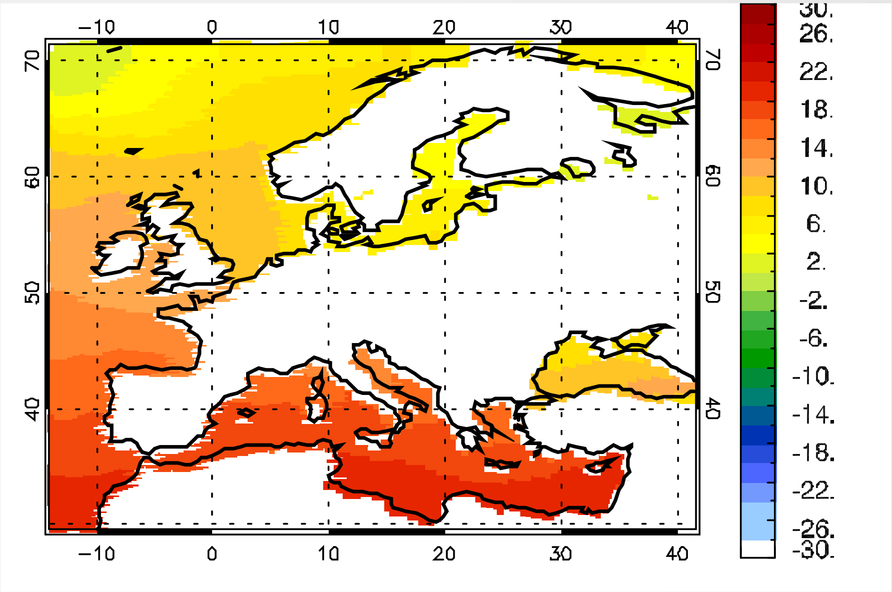
Model configuration: Initialization and Spin up

METGRIB.TBL

```
name=SST
        interp_option=sixteen_pt+four_pt
        fill_missing=0.
        missing_value=-1.E30
        flag_in_output=FLAG_SST
```

```
name=SST
        interp_option=sixteen_pt+four_pt+(wt_)average_4pt
        fill_missing=0.
        missing_value=-1.E30
        flag_in_output=FLAG_SST
```





Model configuration: Initialization and Spin up

- **Noise** in pressure fields in the *first hours* of the simulation
- Sound waves adjusting winds to terrain and this disappears in about time-scales for sound waves to leave the domain
- For large domains, this time-scale is longer, e.g. ~1 hour per 1000km
- **Allow** a reasonable *spin-up period*
- Very important is also the *convection spin-up*, where model will take some time to develop deep convection
- This delay may also be followed by high bias when convection finally spins up
- *For a daily 96hrs forecast usually the first 6-9 hours are considered as spin up period*

Model configuration: Integration

- Model time step is always proportional to the time step of the most coarse grid
- Recommended (maximum) integration time step (s) equals **6*dx** (km)
- Most often, this needs to be downscaled to avoid *numerical instability* (CFL violation)
- Reducing the coarse grid time step does not significantly reduce model performance if you can *tweak the time step ratio*

$$\frac{\partial \varphi}{\partial t} + c\frac{\partial \varphi}{\partial x} = 0$$

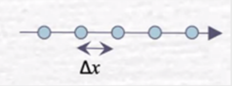
Forward Time – Centered Space (FTCS)

$$\frac{\varphi_j^{n+1}-\varphi_j^{n}}{\Delta t} + c\frac{\varphi_{j+1}^{n}-\varphi_{j-1}^{n}}{2\Delta x} = 0 \quad \Longrightarrow \quad \varphi_j^{n+1} = \varphi_j^{n} - \frac{c\Delta t}{2\Delta x}\left(\varphi_{j+1}^{n}-\varphi_{j-1}^{n}\right)$$

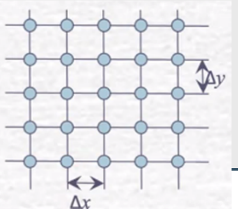
Centered Time – Centered Space (CTCS)

$$\frac{\varphi_j^{n+1}-\varphi_j^{n-1}}{2\Delta t} + c\frac{\varphi_{j+1}^{n}-\varphi_{j-1}^{n}}{2\Delta x} = 0 \quad \Longrightarrow \quad \varphi_j^{n+1} = \varphi_j^{n-1} - \frac{c\Delta t}{\Delta x}\left(\varphi_{j+1}^{n}-\varphi_{j-1}^{n}\right)$$

$$\left|\frac{c\cdot\Delta t}{\Delta x}\right| \le 1$$



N διαστάσεις: $\frac{c\cdot\Delta t\cdot\sqrt{N}}{\Delta x} \le 1$



$$\frac{c\Delta t}{\Delta x} \le \frac{1}{\sqrt{2}}$$

Model configuration: Integration

For example,

- If we have a 15 km coarse grid (CG) and a 5 km fine grid (FG) (1-way nested) then:
  - CG dt=6*15=90s, FG dt=90/3=30s (parent dt divided by 3:1 ratio)
  - time_step = **90**
  - dx = 15000, 5000,
  - grid_id = 1, 2,
  - parent_id = 0, 1,
  - parent_grid_ratio = 1, 3,
  - parent_time_step_ratio = 1, **3**,
- For some reason model “blows up” quickly after the beginning of the simulation

Model configuration: Integration

- We can reduce the time step: CG dt=60s, FG=60/3=20s
  - time_step = **60**
  - dx = 15000, 5000,
  - grid_id = 1, 2,
  - parent_id = 0, 1,
  - parent_grid_ratio = 1, 3,
  - parent_time_step_ratio = 1, **3**,
- Model becomes numerically steady
- But also 90/60 = 1.5x more expensive

Model configuration: Integration

- Reduce time step only for CG: CG dt=60s, FG=60/2=30s (parent time step divided by 2:1 time step ratio)
    - time_step = **60**
    - dx = 15000, 5000,
    - grid_id = 1, 2,
    - parent_id = 0, 1,
    - parent_grid_ratio = 1, 3,
    - parent_time_step_ratio = 1, **2**,
- Model becomes numerically steady
- **Save computational time!**

Model configuration: I/O

- During model's integration, a large number of files are produced
  - History files (*wrfout**)
  - Restart files (*wrfrst**)
  - Other auxiliary files (*wrfxtrm*, wrfpress**)
  - Standart output files rsl.out.0000 and rsl.error.0000 (along with rsl.out.* and rsl.error.* according to cores number)

For example:

  - rsl.out.0000: *Timing for Writing wrfout_d01_2017-08-09_12_00_00 for domain 1: 5.54356 elapsed seconds*

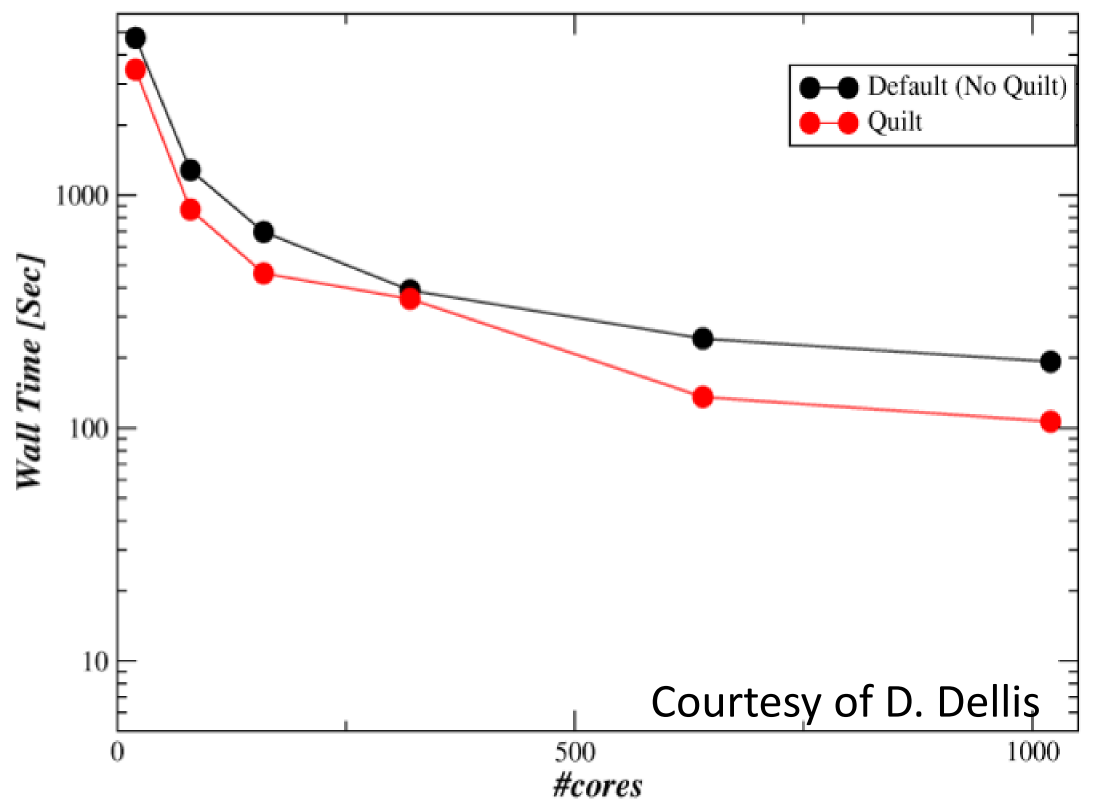
- Represents the amount of wall-clock time attributable to producing the output

Model configuration: I/O

- I/O optimization can be a “*bottleneck*” for improving WRF performance
- On some occasions, <u>I/O takes more time compared to integration</u>!

<u>**Asynchronous I/O (Quilt Servers)**</u>

- WRF provides such I/O server functionality, enabling the user to select at runtime via the input namelist_quilt, *the number of groups of I/O servers to allocate* (nio_groups) *and the number of I/O ranks per group* (nio_tasks_per_group)
- Trial and Error!



Courtesy of D. Dellis

Model configuration: I/O

- If no quilting is desirable these may help also:
  - Output less data
  - Use runtime i/o to reduce output variables via *namelist.input* (iofields_filename=“my_variables.txt”).
    - This will even allow you to cut your file sizes down to half!
  - Consider your experiment. Do you need to output data every 1 h or less?
  - Use parallel netCDF (p-netCDF) during compilation
  - Use option to output *1 file per MPI* process (io_form_history=102). Reported to save a lot time, but you need to manually join files at the end. Officially unsupported.

Model configuration: CFL errors

- WRF develops numerical instability, *CFL* errors, that cause high-resolution runs (not always necessary) to fail occasionally
- *Courant–Friedrichs–Lewy* (CFL) condition is a necessary condition for convergence while solving certain partial differential equations numerically by the method of finite differences
- If the model “blew” up due CFL error then in rsl.error.0000 (for example):

```
3  points exceeded cfl=2 in domain d02 at time 2014-04-28_12:00:16 hours
MAX AT i,j,k:  40 80 4  vert_cfl,w,d(eta)= 2.263442993 -80.54151917 0.2999961376E-02
3  points exceeded cfl=2 in domain d03 at time 2014-04-28_12:00:16 hours
MAX AT i,j,k:  40 80 4  vert_cfl,w,d(eta)= 2.485260963 13.09560013 0.2999961376E-02
```

Model configuration: CFL errors

How to overcome the CFL error

- Check “*where*” the model becomes unstable (vertical level, or which i,j) in model domain by examining the rsl.error.0000 file
- If CFL violation occurs at the first few vertical levels, then it’s probably due to steep orography:
  - Check i,j to verify (even approximately) whether the instability is over complex terrain;
  - If that is the case, consider smoothing orography (GEOGRID.TBL; smooth option: 1-2-1)
- If CFL violation occurs at upper vertical levels, then the available options are:
  - Use the damping option for vertical velocities (w_damping=1)
  - Use a different damping option (damp_opt=1,2,3)
  - Reduce your integration time step or use *adaptive time step* option
  - Consider restructuring your eta_levels (if you defined them explicitly)
- Try to avoid putting domain boundaries near steep orography. If you can't avoid, use more smoothing passes in geogrid table before you create domain

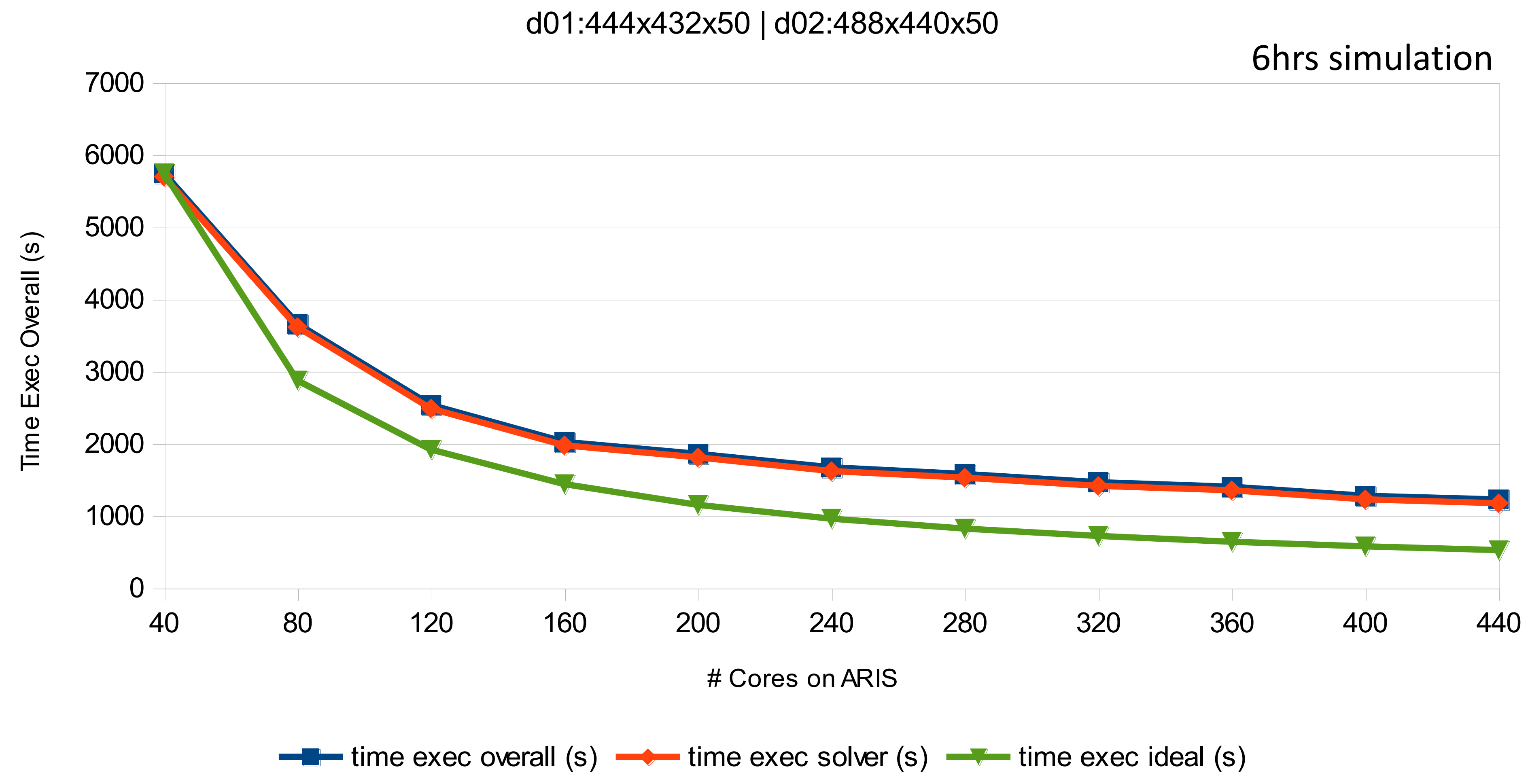
Benchmarking

Available in namelist.input file:

- **nproc_x:** number of processors to use for decomposition in x-direction
- **nproc_y:** number of processors to use for decomposition in y-direction

- By default, WRF will use the square root of processors for deriving values for nproc_x and nproc_y
- If this is not possible, some close values will be used
- WRF responds better to a more rectangular decomposition,
  - i.e. nproc_x << nproc_y
- This leads to longer inner loops for better vector and register reuse, better cache blocking, and more efficient halo exchange communication pattern

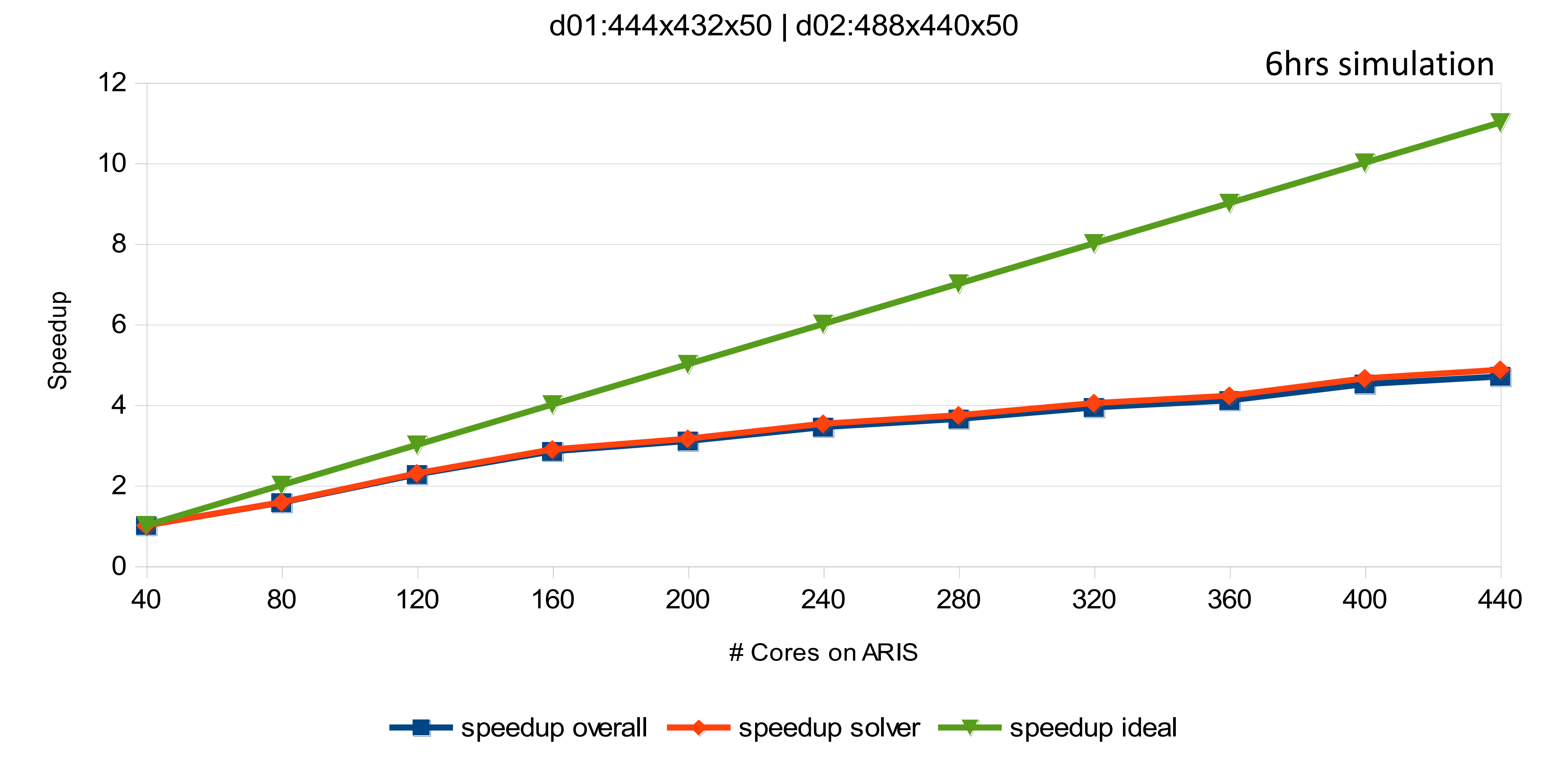
Benchmarking



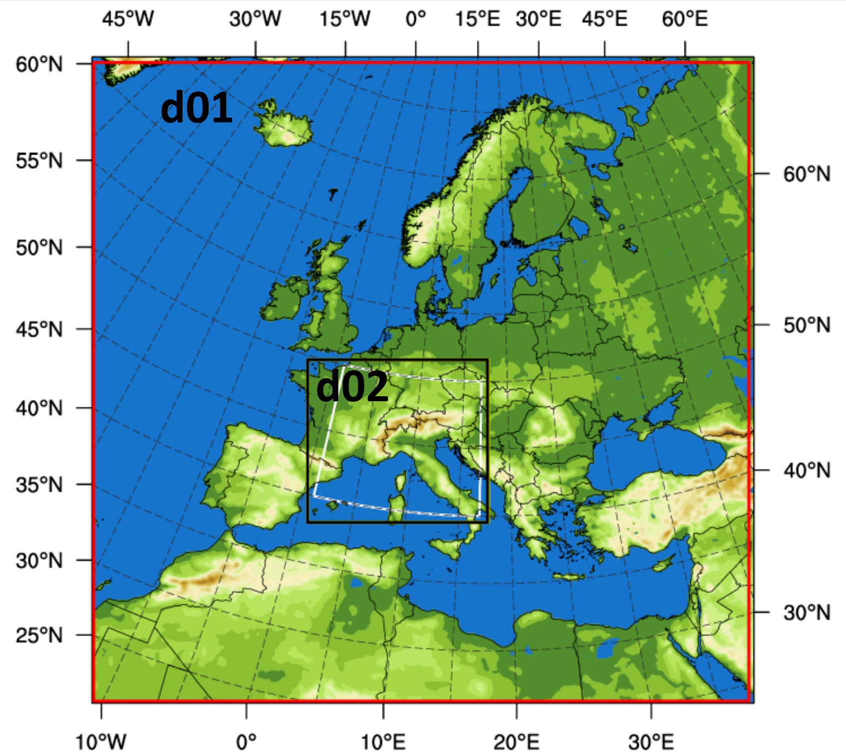
Benchmarking

No nodes	No. cores	Decomposition	time exec solver (s)	time exec overall (s)	speedup solver	speedup overall
2	40	5x8	5697.57	5744.11	1	1
4	80	8x10	3613.01	3658.09	1.58	1.57
6	120	8x15	2482.81	2532.64	2.29	2.27
8	160	10x16	1973.49	2021.57	2.89	2.84
10	200	10x20	1806.81	1855.83	3.15	3.1
12	240	12x20	1616.72	1669.01	3.52	3.44
14	280	14x20	1525.48	1576.29	3.73	3.64
16	320	16x20	1412.04	1462.55	4.04	3.93
18	360	18x20	1351.15	1400.33	4.22	4.1
20	400	16x25	1225.19	1274.43	4.65	4.51
22	440	20x22	1171.56	1222.92	4.86	4.7

Benchmarking



CORDEX-FPSCONV



RCMs: WRF-ARW, RegCM4, AROME, UM, COSMO

CMIP5 GCMs: EC-EARTH, HadGEM2-ES, HadGEM3-GC3.1-N512, MPI-ESM-LR, NorESM1-ME, IPSL-CM5A-MR, IPSL-CM5-MR

Scientific questions:

- How do convective events and associated damaging phenomena (heavy precipitation, windstorms, flash-floods) respond to changing climate conditions in different climatic regions of Europe
- Does an improved representation of convective processes and precipitation at convection permitting scales lead to upscaled added value?
- Is it possible to augment costly convection-permitting experiments with physically defensible statistical downscaling approaches such as “convection emulators” that mimic CP-RCMs and are fed by output of conventional-scale RCMs?

Coppola et al. (2020)
https://doi.org/10.1007/s00382-018-4521-8

- Spatial resolution: 15 Km (Europe) – 3 Km (Alps)
- Period: hindcast (2000-2015), projection (2090-2099)
- Restart (wrfrst) every month
- Wall-clock time: ~42hrs per month (240 cores)
- For 29 years (including spin-up): ~7.500hrs, 610 days !!!!
- Total core hours: ~3.600.000
- 1.4 TB per month raw output → **!!! 488 TB !!!**
- **3 production projects on HPC-ARIS (2018-2021)**

Molina et al. (2024), Sangelantoni et al. (2024), Belusic Vozila et al. (2024), Ha et al. (2024), Soares et al. (2024), Muller et al. (2023), Ban et al. (2021)

CORDEX-FPSLUCAS

The overall objective of FPS LUCAS is to identify robust biophysical impacts of land use changes on climate across regional to local spatial scales and at various time scales from extreme events to multi-decadal trends.

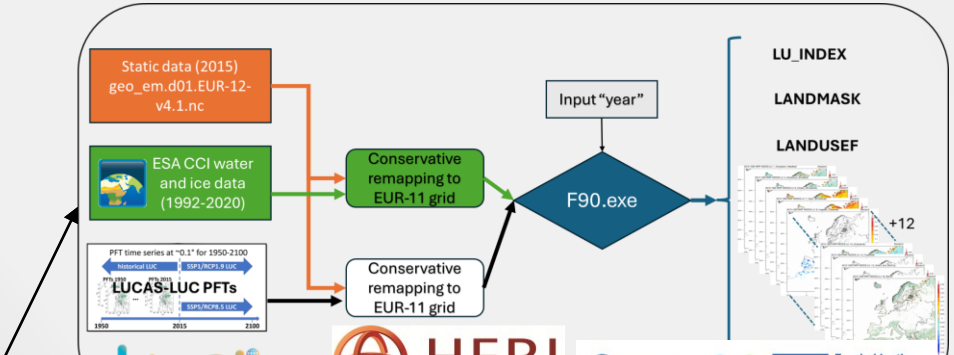
https://ms.hereon.de/cordex_fps_lucas/index.php.en

Sofiadis et al. 2022; Davin et al. 2019; 2020; Mooney et al. 2022; Daloz et al. 2022

✓ **1st phase:**
Idealized simulations

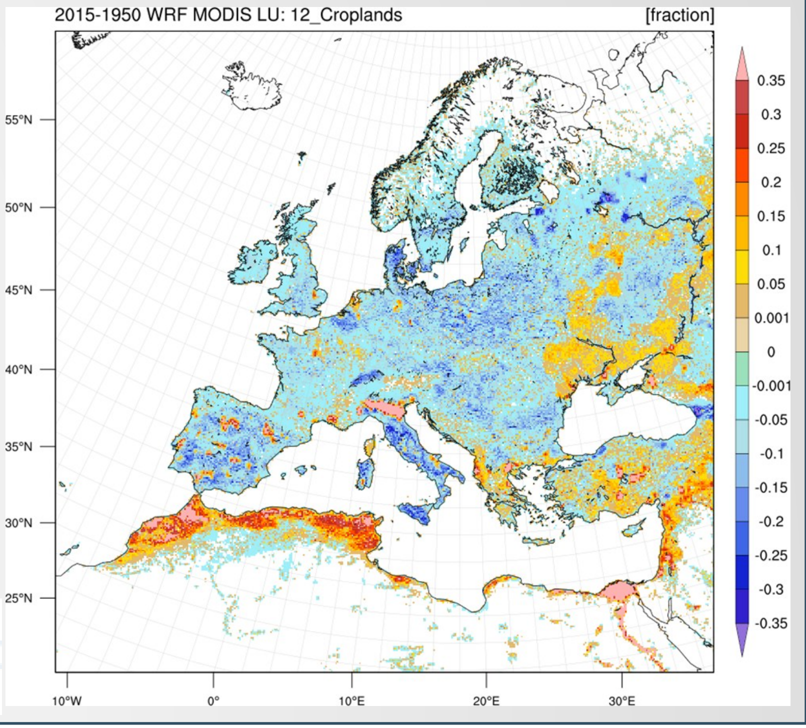
2024... **2nd phase:**
EUR11 Evaluation, historical & projection

...2026 **3rd phase:**
High resolution simulations (~5km)


Static data (2015) geo_em.d01.EUR-12-v4.1.nc
ESA CCI water and ice data (1992-2020)
PFT time series at ~0.1° for 1950-2100
historical LUC
SSP1/RCP1.9 LUC
LUCAS-LUC PFTs
SSP5/RCP8.5 LUC
1950
2015
2100
Conservative remapping to EUR-11 grid
Conservative remapping to EUR-11 grid
Input "year"
F90.exe
LU_INDEX
LANDMASK
LANDUSEF
+12

The research project is implemented in the framework of H.F.R.I call "Basic research Financing (Horizontal support of all Sciences)" under the National Recovery and Resilience Plan "Greece 2.0" funded by the European Union –NextGenerationEU (H.F.R.I. Project Number: 14696)."

- **RCM:** WRF-ARW v4.5.1 (CORDEX WRF community fork, bug fixes) (*WRFv3.8.1 in Phase 1*)
- **ICs & BCs:** MPI-ESM1-2-HR
- **Time periods:** 1950-2014 (**64 yrs**, historical), 2015-2100 (**85 yrs**, ssp126), 1980-2020 (**40 yrs**, evaluation)
- EUR-11 (~13 hours per month)
- ~3.200 core hours / ~240 GB per month
- Estimated core hours: **~7.000.000**



Thank you for attention!

Co-funded by
the European Union

EuroHPC
Joint Undertaking

Funded by the European Union. This work has received funding from the European High Performance Computing Joint Undertaking (JU) and Germany, Bulgaria, Austria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, Greece, Hungary, Ireland, Italy, Lithuania, Latvia, Poland, Portugal, Romania, Slovenia, Spain, Sweden, France, Netherlands, Belgium, Luxembourg, Slovakia, Norway, Türkiye, Republic of North Macedonia, Iceland, Montenegro, Serbia under grant agreement No 101101903.