11-12 December 2017
Aristotle University, Research Dissemination Center (KEDEA)

VI-SEEM NAT-GR CL: National training event in Greece

WRF
Weather Research and Forecast Model
Meteorological applications on HPC ARIS

Stergios Kartsios
PhD Candidate
Dep. of Meteorology and Climatology, AUTH
kartsios@geo.auth.gr
http://orcid.org/0000-0002-2790-3782
https://www.researchgate.net/profile/Stergios_Kartsios

Introduction

- Weather Research and Forecasting (WRF) Model:
  - A next-generation *mesoscale numerical weather* prediction system designed for both atmospheric research and operational forecasting applications
  - 2 dynamical cores (**ARW**, NMM)
  - data assimilation system
  - software architecture
  - supporting parallel computation and system extensibility
- 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/weather-research-and-forecasting-model

Introduction: WRF-ARW releases

- 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: August 4, 2005
- V2.1.1: Nov 8, 2005
- V2.1.2: Jan 27, 2006
- V2.2: Dec 21, 2006
- V2.2.1: Nov 1, 2007
- V3.0: April 4, 2008
- V3.0.1: August 5, 2008
- V3.0.1.1: August 22, 2008
- V3.1: April 9, 2009
- V3.1.1: July 31, 2009
- V3.2: March 31, 2010
- V3.2.1: August 18, 2010
- V3.3: April 6, 2011
- V3.3.1: Sept 16, 2011
- V3.4: April 6, 2012
- V3.4.1: Aug 16, 2012
- V3.5: April 18, 2013
- V3.5.1: Sept 23, 2013
- V3.6: April 18, 2014
- V3.6.1: Aug 14, 2014
- V3.7: April 20, 2015
- V3.7.1: Aug 14, 2015
- V3.8: April 8, 2016
- V3.8.1: Aug 12, 2016
- V3.9: Apr 17, 2017
- V3.9.1: Aug 17, 2017
- **V3.9.1.1: Aug 28, 2017**

NCAR
NATIONAL CENTER FOR ATMOSPHERIC RESEARCH

NATIONAL CENTERS FOR ENVIRONMENTAL PREDICTION
NCEP

NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
NOAA
U.S. DEPARTMENT OF COMMERCE

FORECAST SYSTEMS LABORATORY
BOULDER, COLORADO

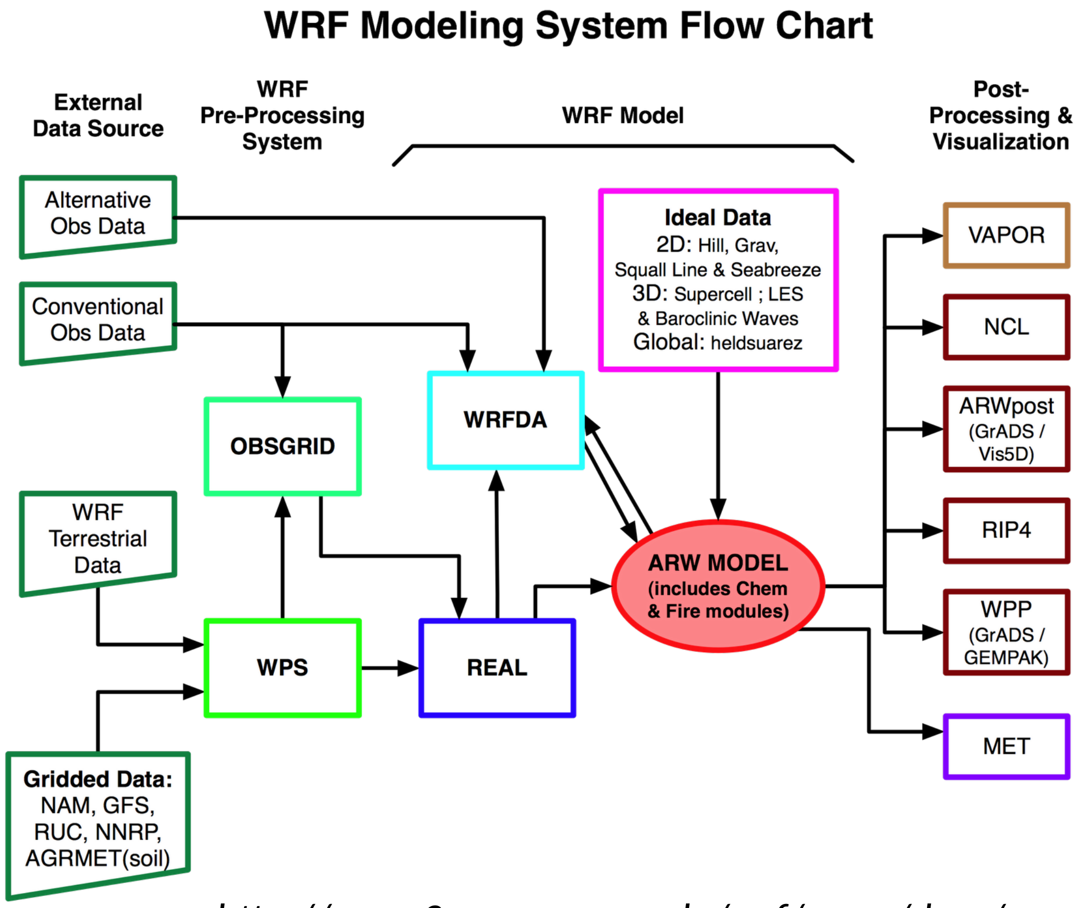
THE UNIVERSITY OF OKLAHOMA
CIVI ET REIPUBLICAE
1890

NAVAL RESEARCH LABORATORY
WASHINGTON, DC

FEDERAL AVIATION ADMINISTRATION

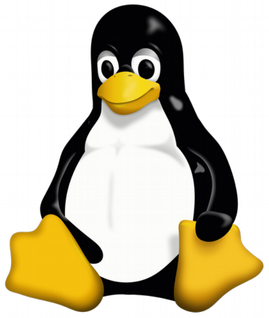
AIR FORCE WEATHER AGENCY

Introduction



- The WRF Preprocessing System (WPS)
- WRF-DA
- ARW solver
- Post-processing & Visualization tools

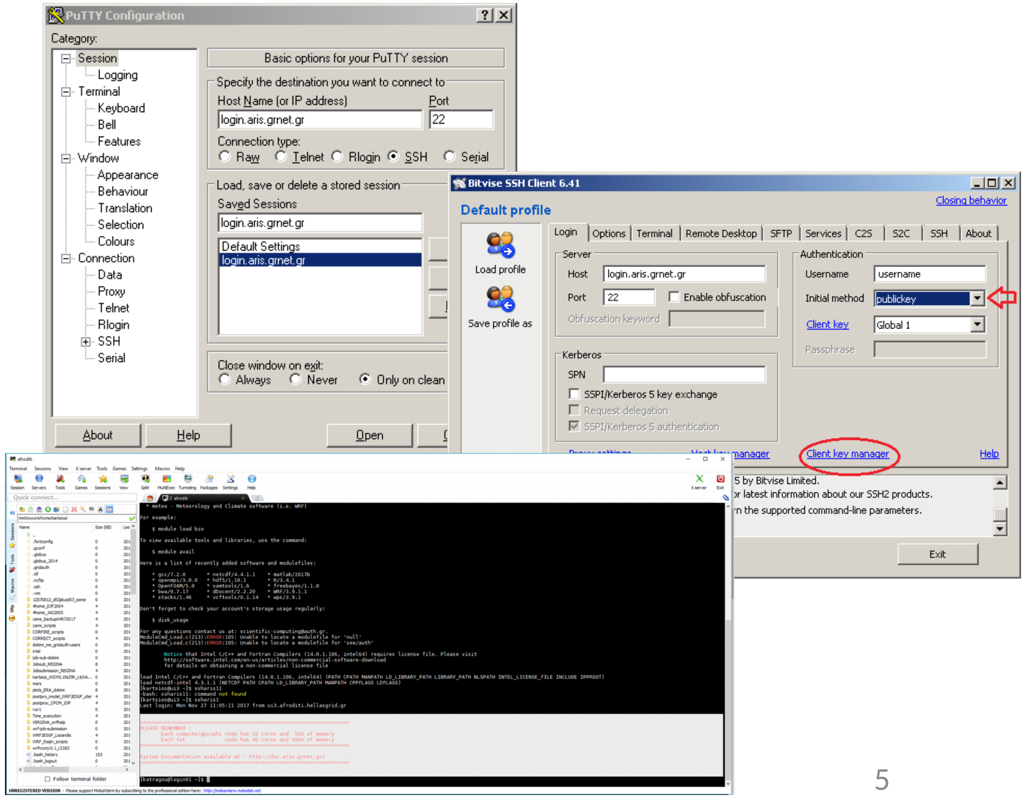
A Linux application!



http://www2.mmm.ucar.edu/wrf/users/docs/user_guide_V3.9/users_guide_chap1.htm

Basics…

- Must be familiarized with LINUX basic commands
- In order to connect to HPC ARIS infrastructure you will need an ***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/)
- On Linux/Mac just use the terminal
- ssh -YC username@login.aris.grnet.gr
  - All you need at http://doc.aris.grnet.gr/



Software Requirements

- Fortran 90 or 95 and C compiler
- 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

Configure and Compile

- Lots of on-line tutorials and examples
- Depending on the type of run you wish to make, there are various libraries that should be installed. Below are 5 libraries:
  - *mpich/intelmpi*
  - *netcdf*
  - *Jasper*
  - *libpng*
  - *zlib*
- It is important to note that these libraries must all be installed *with the same compilers* as will be used to install WRFV3 and WPS.
- On ***ARIS*** all the necessary libraries are available through <u>the environment module approach (module load)</u>

Configure and Compile: Module Usage

- **module avail:** *List all available modules*
- **module load:** *Load module into the shell environment*
- **module list:** *List loaded modules*
- **module unload:** *Remove module from the shell environment*
- **module purge:** *Unload all loaded modules*
- **module switch:** *Switch loaded module1 with module2*
- **module show:** *List all of the environment changes the module will make if loaded*
- **module whatis:** *Display what is the module information*
- **module help:** *More specific help*

http://doc.aris.grnet.gr/environment

```
File  Edit  View  Search  Terminal  Help
[kartsios@login02 ~]$ module avail

----------------------------------------------------------- /apps/modulefiles/compilers -----------------------------------------------------------
binutils/2.25          cuda/8.0.61(default)   gnu/5.3.0              intel/16.0.0           intel/18.0.1           pgi/17.4
binutils/2.26          gdb/7.11.1             gnu/5.4.0              intel/16.0.1           java/1.7.0             pgi/17.5
binutils/2.27          gdb/7.12.1(default)    gnu/5.5.0              intel/16.0.2           java/1.8.0(default)    pgi/17.7
binutils/2.28          gdb/7.9.1              gnu/6.1.0              intel/16.0.3           pgi/15.5               rcuda/16.11/8.0
binutils/2.29(default) gnu/4.1.2              gnu/6.2.0              intel/16.0.4           pgi/16.10(default)     scala/0.13.16
clang/5.0.0(default)   gnu/4.8.5              gnu/6.3.0              intel/17.0.0           pgi/16.4               sun/12.5(default)
cuda/6.5.14            gnu/4.9.2(default)     gnu/6.4.0              intel/17.0.1           pgi/16.5
cuda/7.0.28            gnu/4.9.3              gnu/7.1.0              intel/17.0.3           pgi/16.7
cuda/7.5.18            gnu/4.9.4              gnu/7.2.0              intel/17.0.4           pgi/16.9
cuda/8.0.27            gnu/5.1.0              intel/15.0.3(default)  intel/17.0.5           pgi/17.1
cuda/8.0.44            gnu/5.2.0              intel/15.0.6           intel/18.0.0           pgi/17.10

------------------------------------------------------------ /apps/modulefiles/parallel -----------------------------------------------------------
intelmpi/2017.0         intelmpi/5.1.2          openmpi/1.10.0/gnu      openmpi/1.10.5/gnu      openmpi/2.0.0/intel     openmpi/2.1.1/intel
intelmpi/2017.1         intelmpi/5.1.3          openmpi/1.10.0/intel    openmpi/1.10.5/intel    openmpi/2.0.1/gnu       openmpi/2.1.2/gnu
intelmpi/2017.2         intelmpi/5.1.3.258      openmpi/1.10.1/gnu      openmpi/1.10.7/gnu      openmpi/2.0.1/intel     openmpi/2.1.2/intel
intelmpi/2017.3         mpich/3.2/gnu           openmpi/1.10.1/intel    openmpi/1.10.7/intel    openmpi/2.0.2/gnu       openmpi/3.0.0/gnu
intelmpi/2017.4         mpich/3.2/intel         openmpi/1.10.2/gnu      openmpi/1.8.5/gnu       openmpi/2.0.2/intel     openmpi/3.0.0/intel
intelmpi/2017.5         mpich/3.2.1/gnu         openmpi/1.10.2/intel    openmpi/1.8.5/intel     openmpi/2.0.3/gnu       padb/3.3
intelmpi/2018.0         mpich/3.2.1/intel       openmpi/1.10.3/gnu      openmpi/1.8.7/gnu       openmpi/2.0.3/intel     scalasca/2.2.2
intelmpi/2018.1         mpiP/3.4.1(default)     openmpi/1.10.3/intel    openmpi/1.8.7/intel     openmpi/2.1.0/gnu       scalasca/2.3.1(default)
intelmpi/5.0.3(default) mvapich2/gnu/2.2.2a     openmpi/1.10.4/gnu      openmpi/1.8.8           openmpi/2.1.0/intel
intelmpi/5.1.1          mvapich2/intel/2.2.2a   openmpi/1.10.4/intel    openmpi/2.0.0/gnu       openmpi/2.1.1/gnu

----------------------------------------------------------- /apps/modulefiles/libraries -----------------------------------------------------------
atlas/3.10.2               fgsl/1.0.0/gnu             hdf5/1.8.15/intel          netcdf/3.6.3/gnu           openblas/0.2.18/intel
atlas/3.10.3               fgsl/1.0.0/intel           hdf5/1.8.17/gnu            netcdf/3.6.3/intel         openblas/0.2.19/gnu
atlas/3.11.34(default)     flame/5.0/gnu              hdf5/1.8.17/intel          netcdf/4.1.3/gnu           openblas/0.2.19/intel
atlas/3.11.38              flame/5.0/intel            jasper/1.900.1(default)    netcdf/4.1.3/intel         papi/5.4.1
boost/1.58.0(default)      gdal/2.2.0                 libint/1.1.5               netcdf/4.4.1/gnu           parmetis/4.0.3/gnu
boost/1.62.0               geant4/4.10.01             libjpeg-turbo/1.4.1(default) netcdf/4.4.1/intel       parmetis/4.0.3/intel
boost-py2.7/1.58.0         geant4/4.10.01.p02         libsmm/gnu                 netcdf-c/4.3.3.1/gnu       petsc/3.6.2(default)
cgnslib/3.2.1/intel        geant4/4.10.02.p03         libsmm/intel               netcdf-c/4.3.3.1/intel     petsc/3.7.2
clFFT/2.12.2               geant4/4.10.03.p01         libxc/2.2.2                netcdf-combined/4.3.3.1/intel petsc/3.7.4
clhep/2.2.0.5              geant4/4.9.5p01            libxc/3.0.0/gnu            netcdf-fortran/4.4.2/gnu   pnetcdf/1.6.1/gnu
elpa/2015.05.001/gnu       geos/3.6.1                 libxc/3.0.0/intel          netcdf-fortran/4.4.2/intel pnetcdf/1.6.1/intel
elpa/2015.05.001/intel     glpk/4.55                  libxsmm/1.8.1(default)     openblas/0.2.14/gnu/int4   proj4/4.9.3
elpa/2015.11.001/gnu       gsl/1.16/gnu               matlab/runtime/2014b       openblas/0.2.14/gnu/int8   scalapack/2.0.2/gnu
elpa/2015.11.001/intel     gsl/2.1/gnu                matlab/runtime/2015a       openblas/0.2.14/intel/int4 scalapack/2.0.2/intel
fftw/2.1.5                 gsl/2.1/intel              matlab/runtime/2016a       openblas/0.2.14/intel/int8 szip/2.1(default)
```

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:
  - *http://www2.mmm.ucar.edu/wrf/users/download/get_source.html*

```
[kartsios@login02 WRFV3]$ module load wrf
wrf/3.4.1/hybrid     wrf/3.6.1/purempi    wrf/3.7/purempi      wrf/3.8/purempi      wrf-chem             wrf-chem/3.7-hybrid
wrf/3.4.1/purempi    wrf/3.7/hybrid       wrf/3.8.1/purempi    wrf/3.9.1            wrf-chem/3.7         wrf-chem/3.8
```

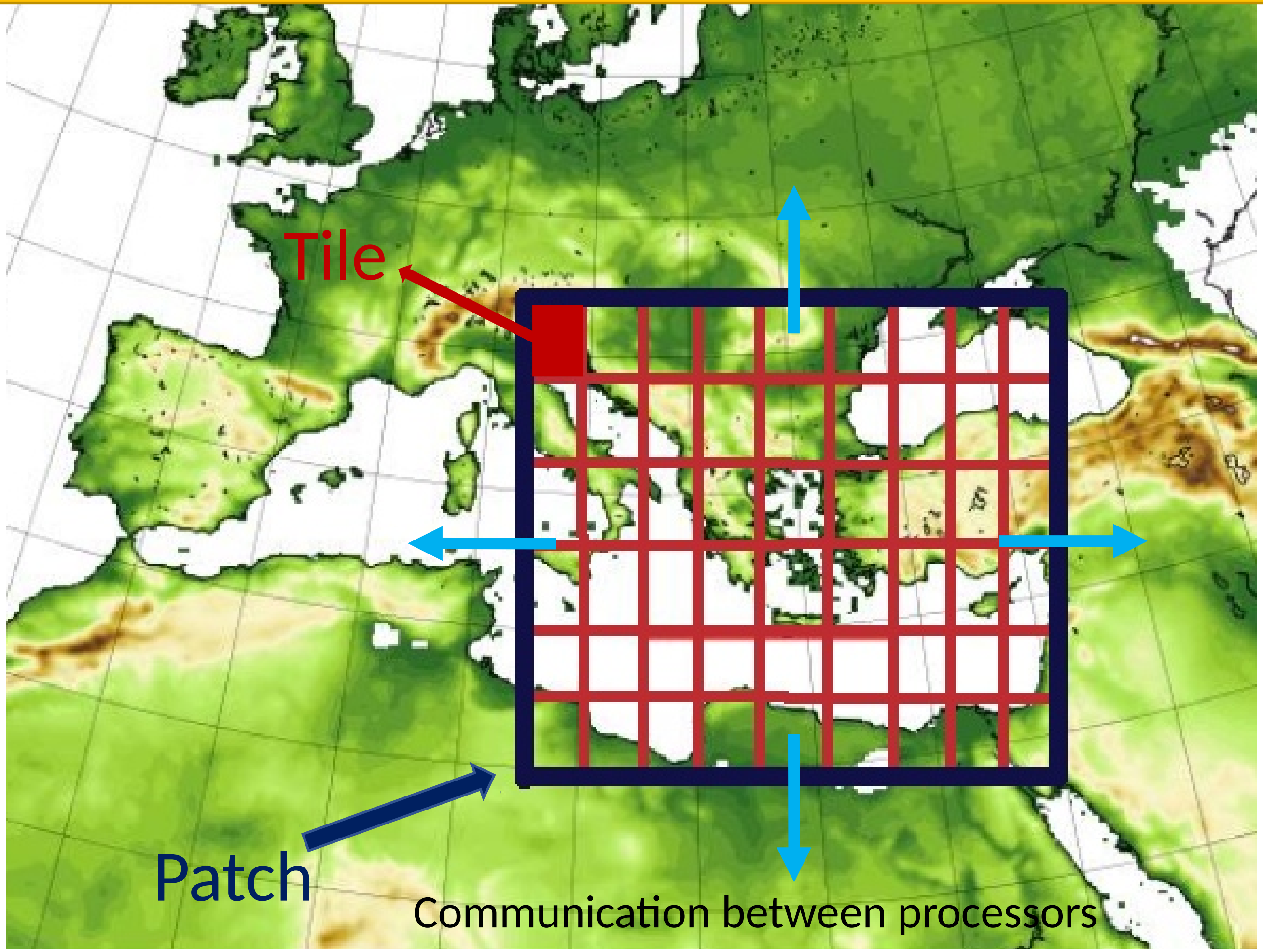
Parallelism in WRF

- 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”)*
- Experience with WRF on ARIS showed us that it is better to use DM mode (“pureMPI”)
- Although a number of configuration and compiler options are available

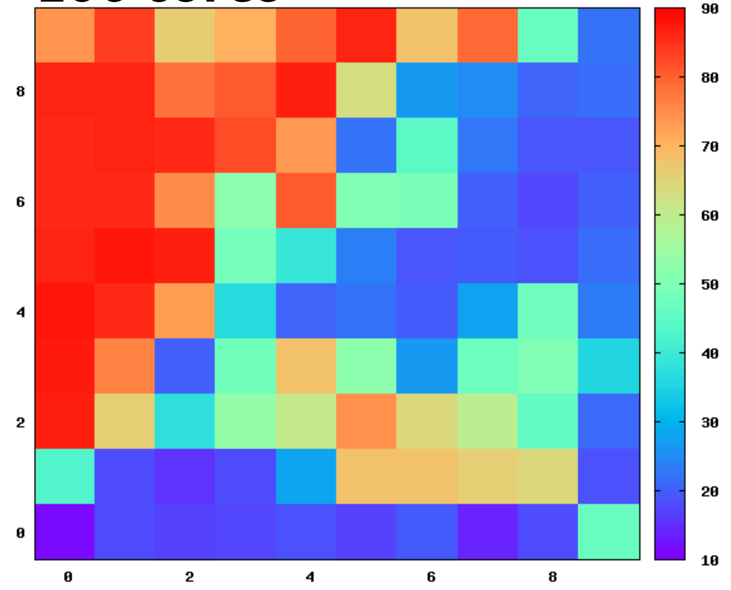
```
[kartsios@login02 WRFV3]$ ./configure
checking for perl5... no
checking for perl... found /usr/bin/perl (perl)
Will use NETCDF in dir: /apps/libraries/netcdf/4.1.3/intel
PHDF5 not set in environment. Will configure WRF for use without.
Will use 'time' to report timing information


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] :
```

Parallelism in WRF

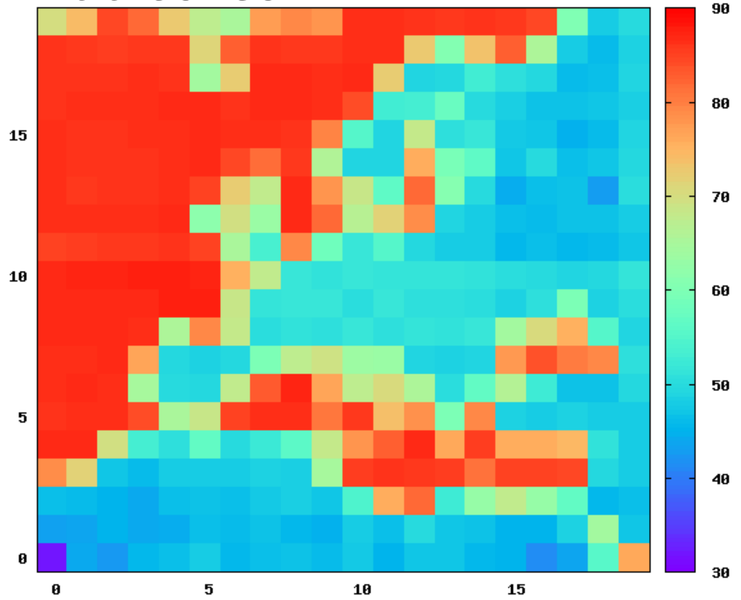
- WRF uses domain decomposition to divide total amount of work over parallel processes
- Model domains are decomposed for parallelism on two-levels
  - **Patch:** *section of model domain allocated to a distributed memory node, this is the scope of a mediation layer solver or physics driver*
  - **Tile:**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



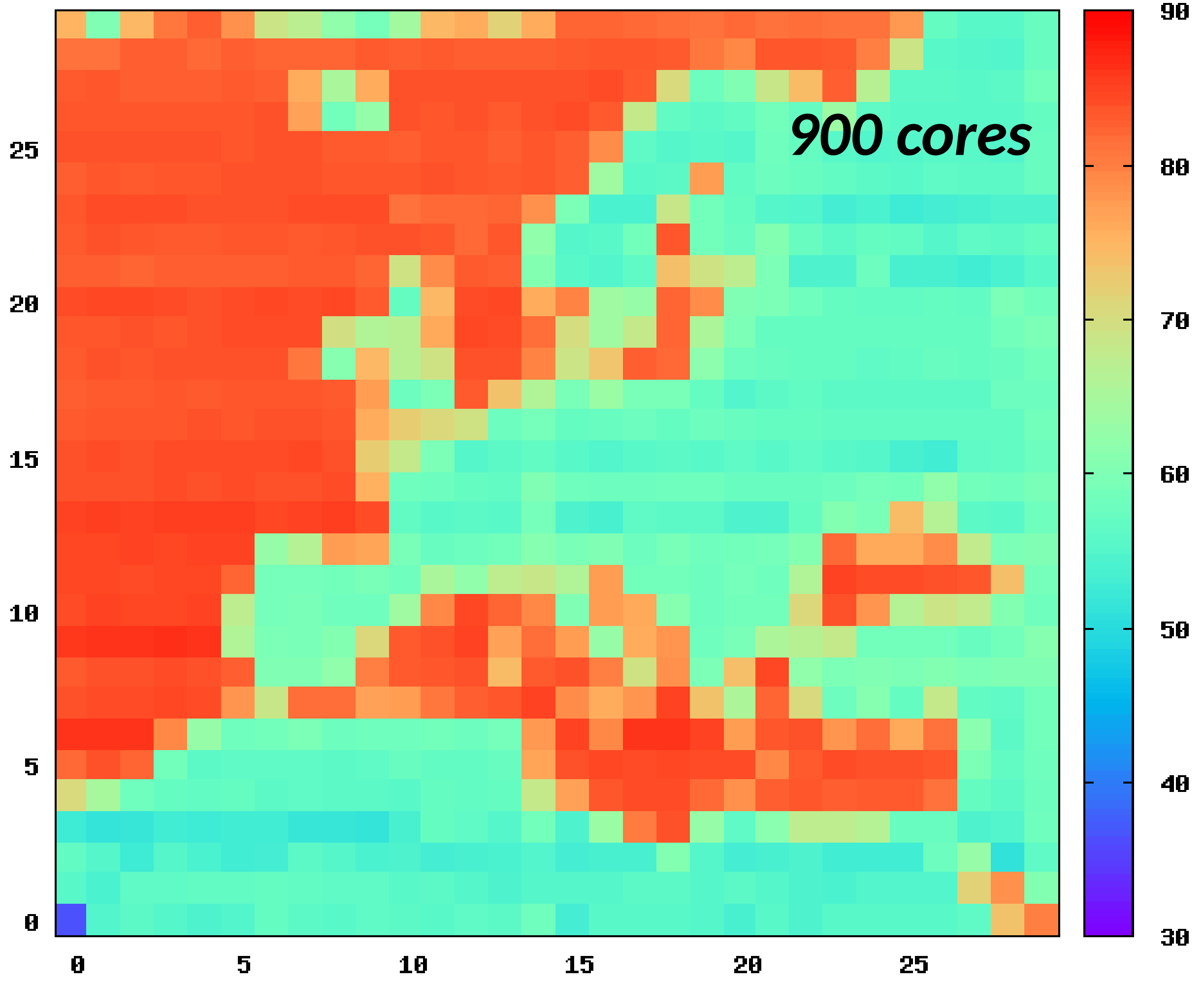
100 cores



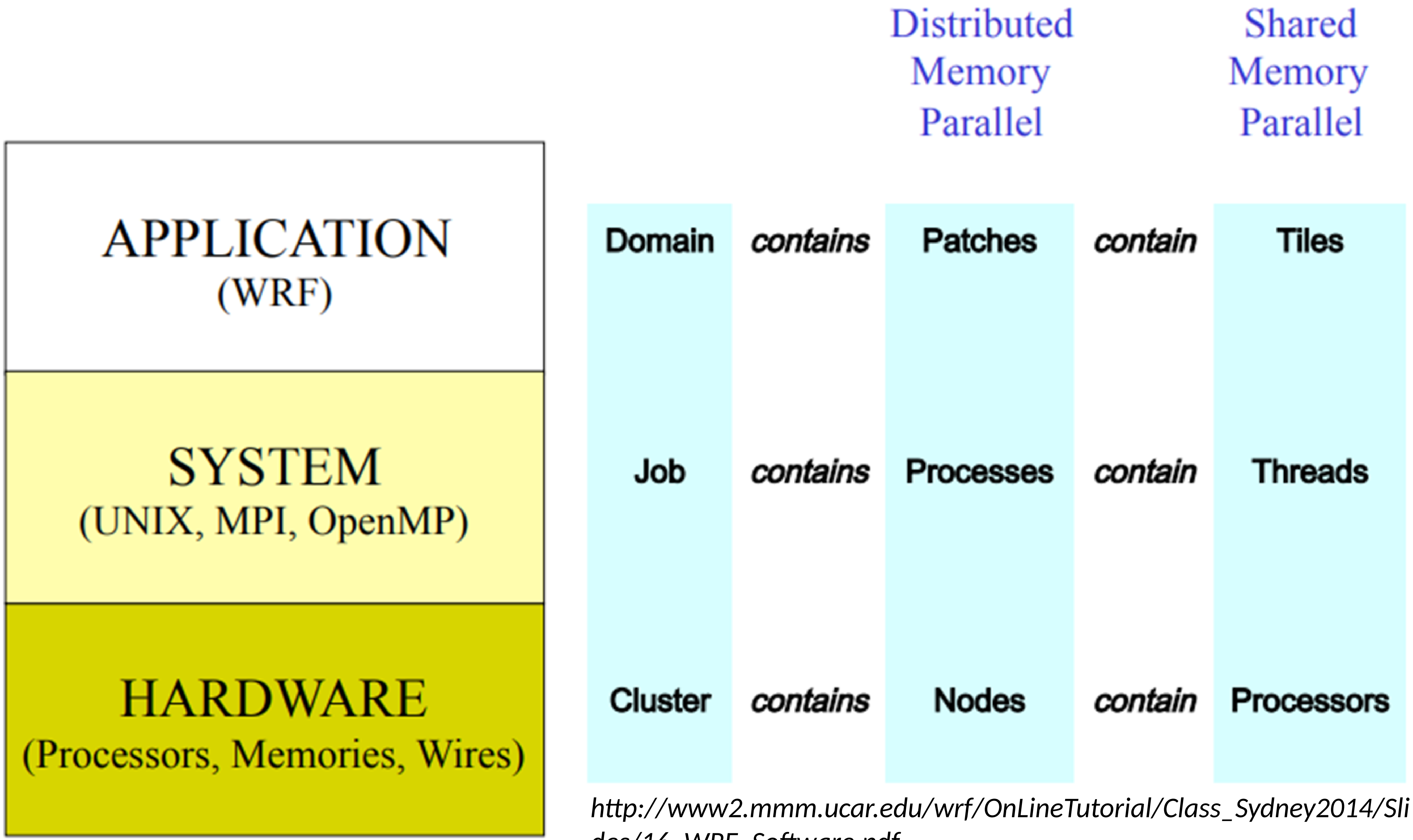
400 cores



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



Courtesy of D. Dellis



http://www2.mmm.ucar.edu/wrf/OnLineTutorial/Class_Sydney2014/Slides/16_WRF_Software.pdf

Getting started…

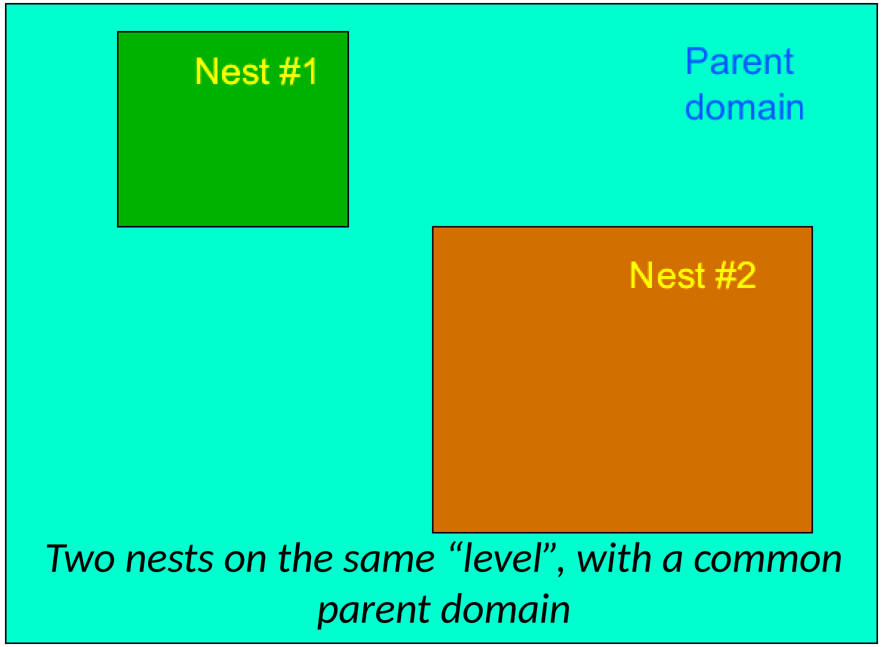
- **Think first**! What are your objectives? Why do you need WRF?
- Get to know your problem! What are the **atmospheric processes** and at **what scales** are you focusing? **Review literature**!
- How do you plan to **verify** your results? Are there any observational data available for your case?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


Domain configuration

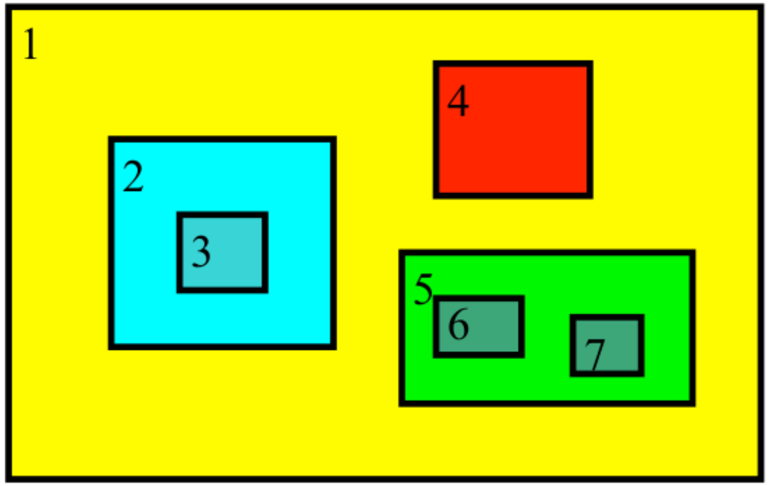
- According to you problem target your horizontal **grid resolution**
- Consider your available **initialization data** (resolution, frequency)
  - Global model, Regional model, Reanalysis?
- Most of the times a **nesting strategy** must be considered
- *namelist.wps* inside WPS folder controls domain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

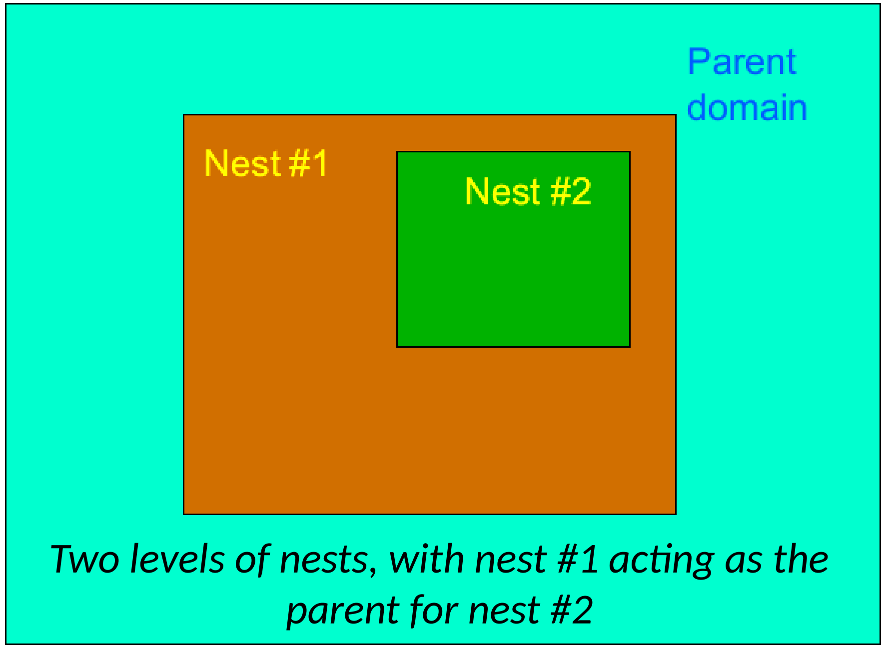
- A *nest* is a finer-resolution model run. It may be embedded simultaneously within a coarser-resolution (parent) model run, or run independently as a separate model forecast
- The nest *covers a portion* of the parent domain, and is driven along its lateral boundaries by the parent domain
- Nesting 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

```
&share
 wrf_core = 'ARW',
 max_dom = 3,
 start_date = '2016-03-24_12:00:00','2016-03-24_12:00:00','2016-03-24_12:00:00',
 end_date   = '2016-03-27_00:00:00','2016-03-27_00:00:00','2016-03-27_00:00:00',
 interval_seconds = 10800,
 io_form_geogrid = 2,
/

&geogrid
 parent_id          =   1,    1,    2,
 parent_grid_ratio =   1,    3,    3,
 i_parent_start     =   1,  150,   36,
 j_parent_start     =   1,   30,   86,
 s_we               =   1,    1,    1,
 e_we               = 240, 154,  136,
 s_sn               =   1,    1,    1,
 e_sn               = 180,  136,  109,
 geog_data_res      = '30s','30s','30s',
 dx = 18000,
 dy = 18000,
 map_proj = 'lambert',
 ref_lat   =  45.2,
 ref_lon   =  12.0,
 truelat1  =  30.0,
 truelat2  =  60.0,
 stand_lon =  16.0,
 geog_data_path = '/home/rap/Build_WRF/geog_new3.6',
/
```



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





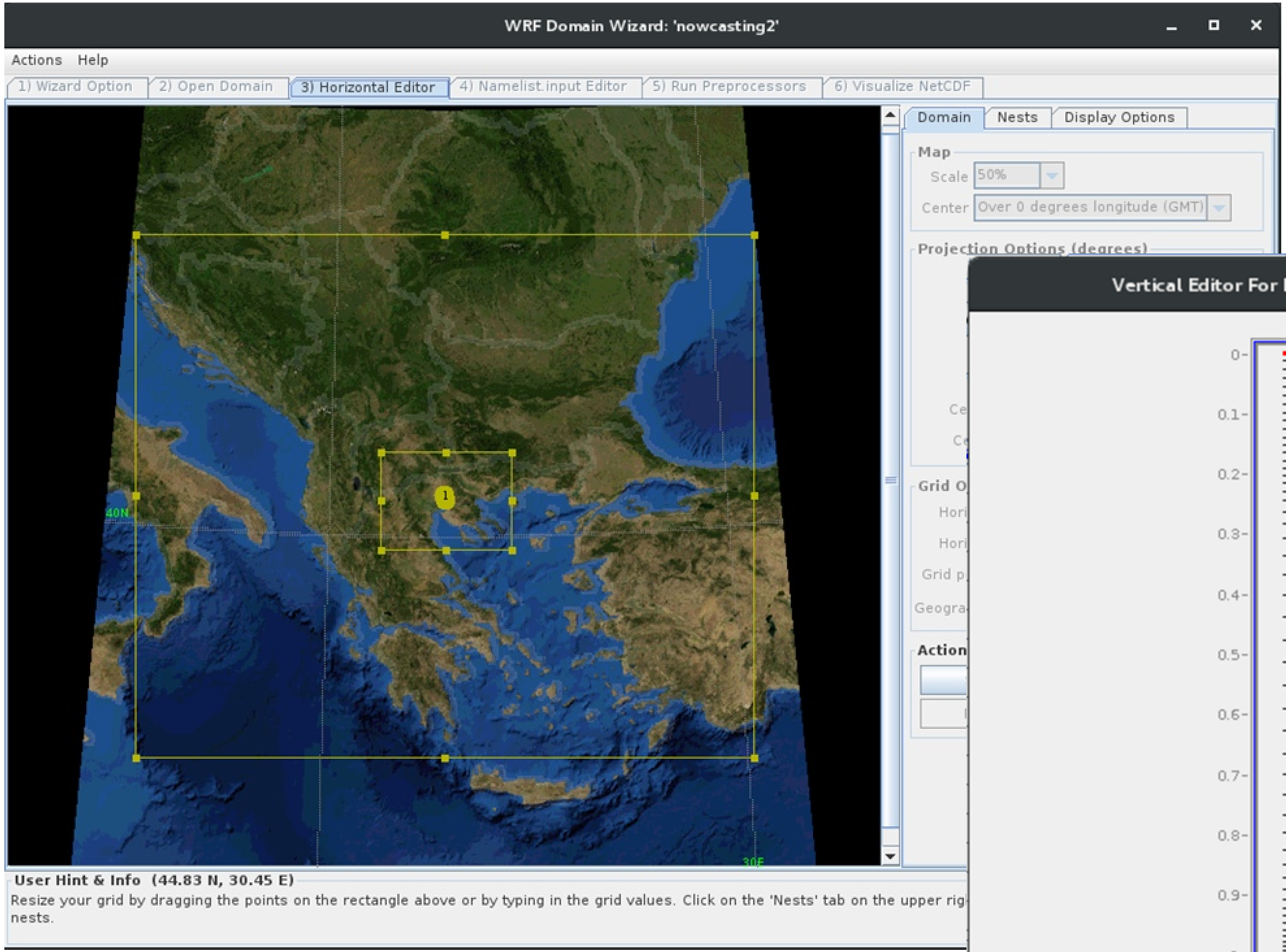
Two levels of nests, with nest #1 acting as the parent for nest #2

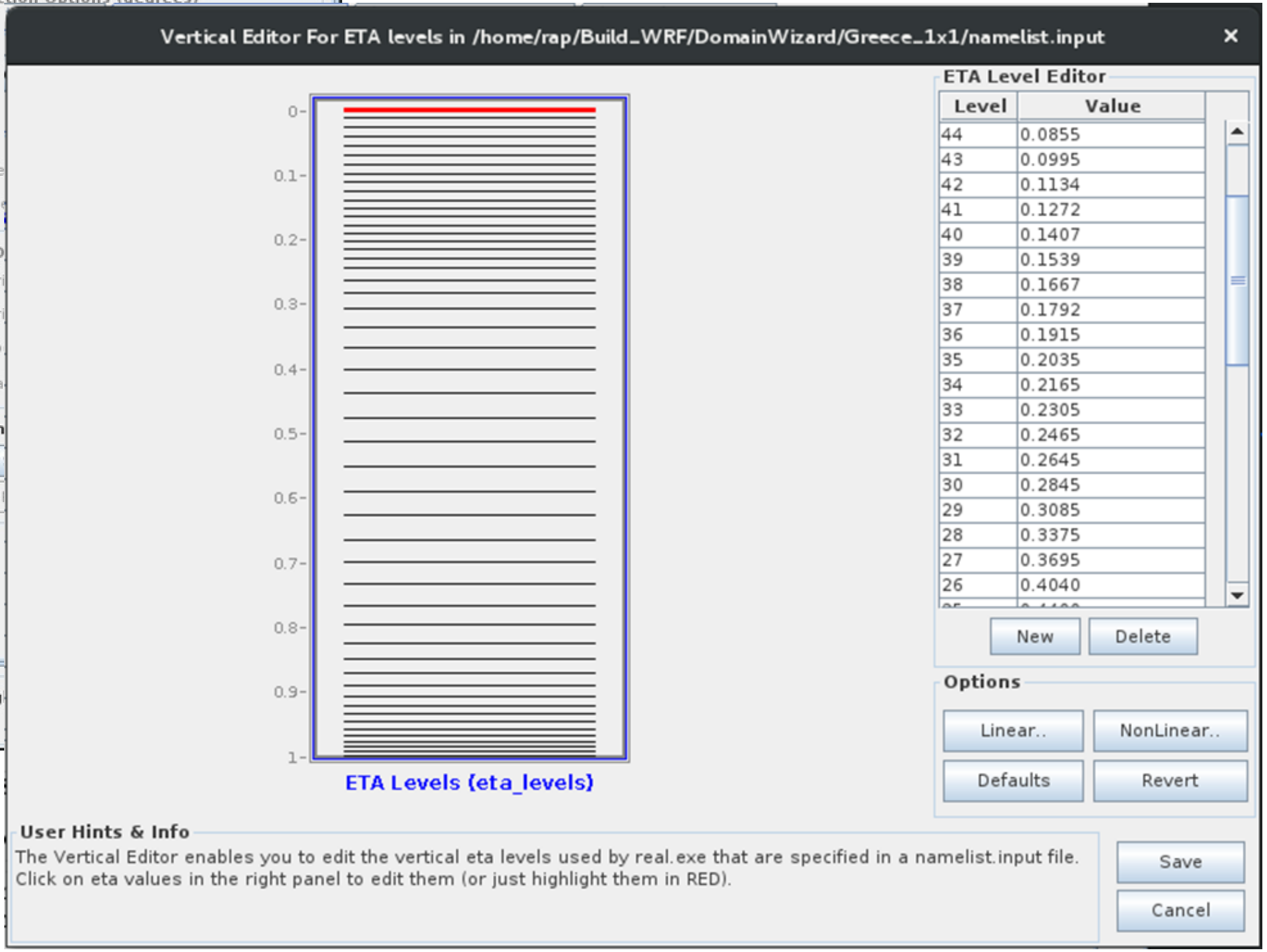
Domain configuration

- There are some NCL scripts available inside WPS folder for testing your domain properties
- A nice and easy tool for domain configuration is the *WRF Domain Wizard*, a GUI for the WRF Preprocessor System (WPS) and namelist.input
  - *https://esrl.noaa.gov/gsd/wrfportal/DomainWizard.html*

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)





- You will also need static data (topography, land use etc.)
- They are all provided in different resolutions!

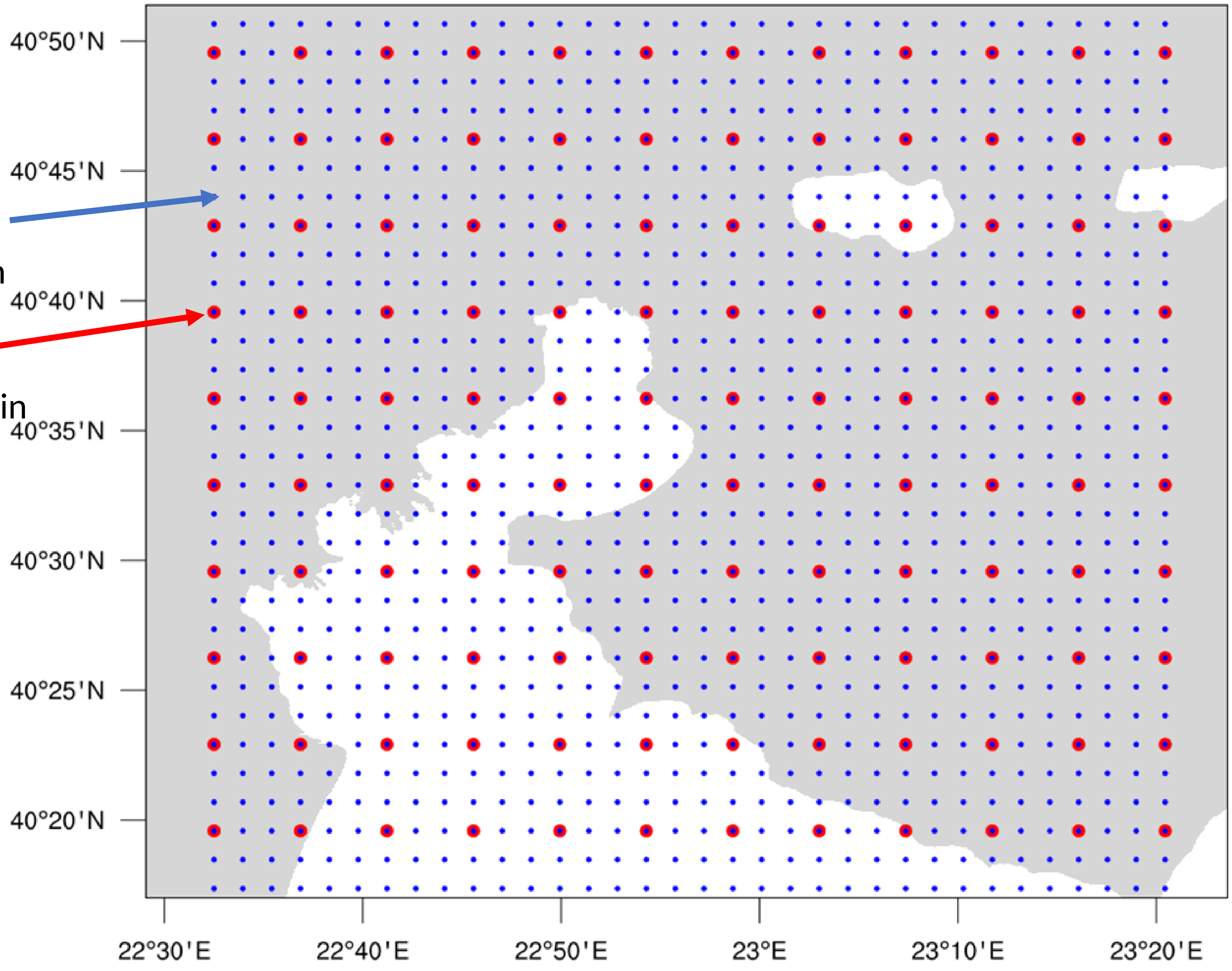
- Question yourselves: *How well my area is represented by this data?*
- May choose another dataset!

Domain configuration

- It’s all about computational resources!
- 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



Domain configuration: Nesting performance

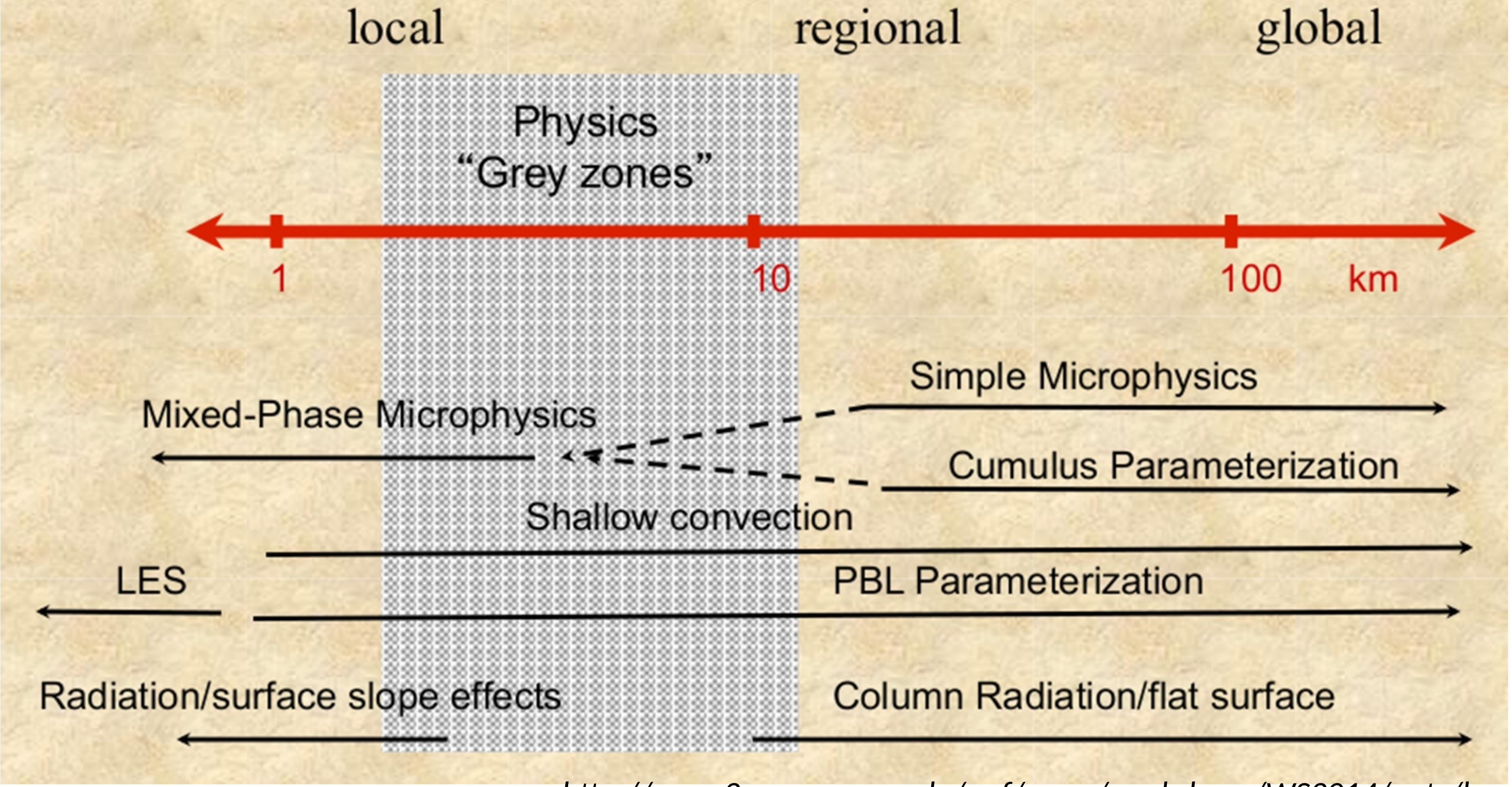
Assuming a 3:1 parent-child ratio:

- 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 compare 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.*

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 $\longrightarrow$ Different Results
- A given set of physics will perform differently depending on domain size, location, initialization and phenomenon of interest

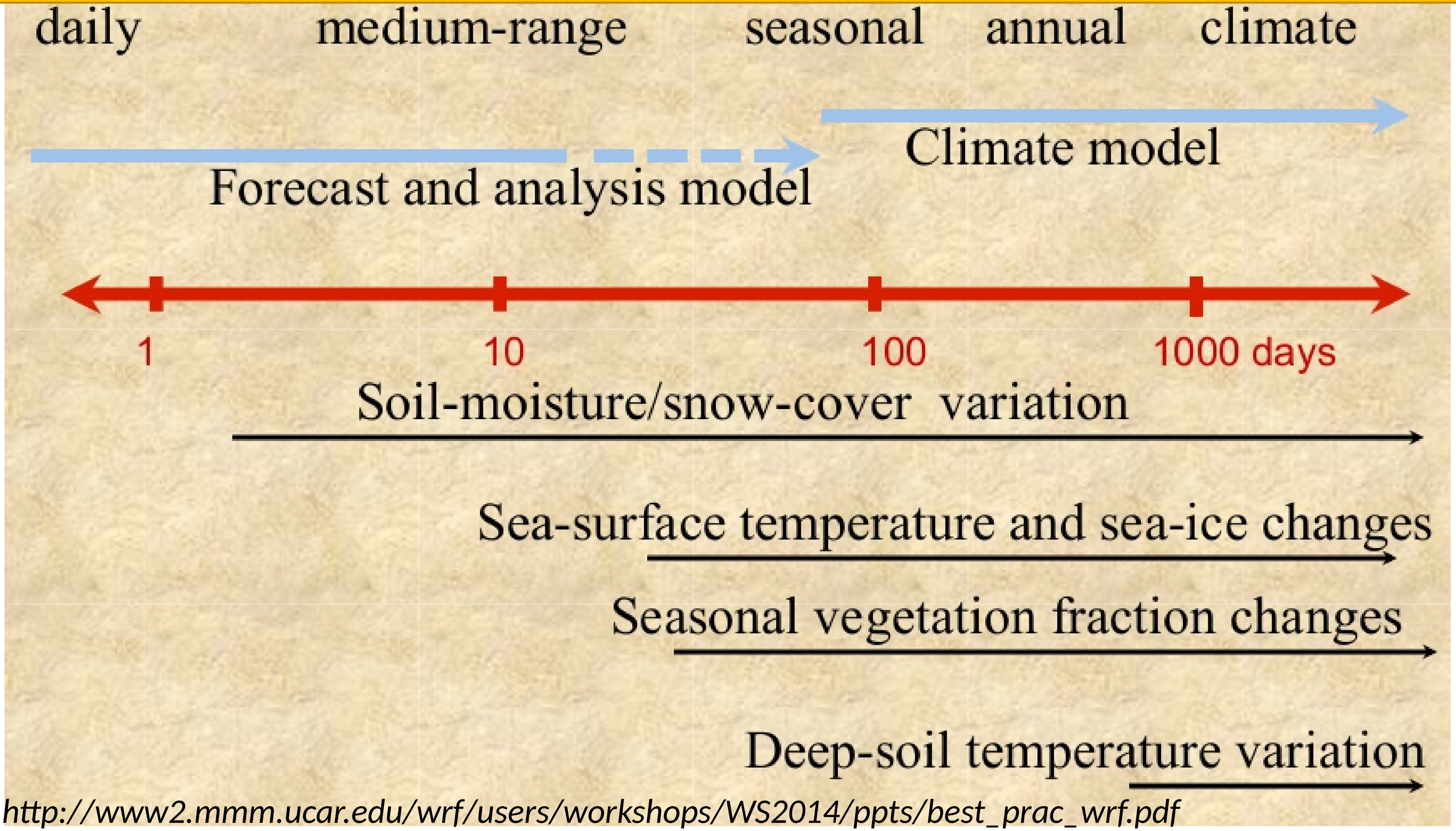
- Consider first <u>well documented</u> (tried) schemes
- Consider <u>grid size</u> when choosing sophistication of microphysics
- You *don't need* a complex scheme for a <u>10 km grid</u>
- You <u>do need</u> a microphysical scheme with <u>*graupel for convection-resolving grids*</u>
- It is better 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 < 5 km probably not
  - For grid resolutions 5-10 km, best to avoid convective cases

Model configuration: Physics

- 21 Microphysics schemes
- 16 PBL schemes
- 12 Cumulus schemes
- 9 Radiation schemes
- 7 Land Surface schemes
- 9 Shallow Convect. Schemes
- 8 Surface Layer schemes
- 3 Urban physics schemes

A large number of combinations!

Model configuration: Physics



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!
- wrfinput_d0*

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 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*

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 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*

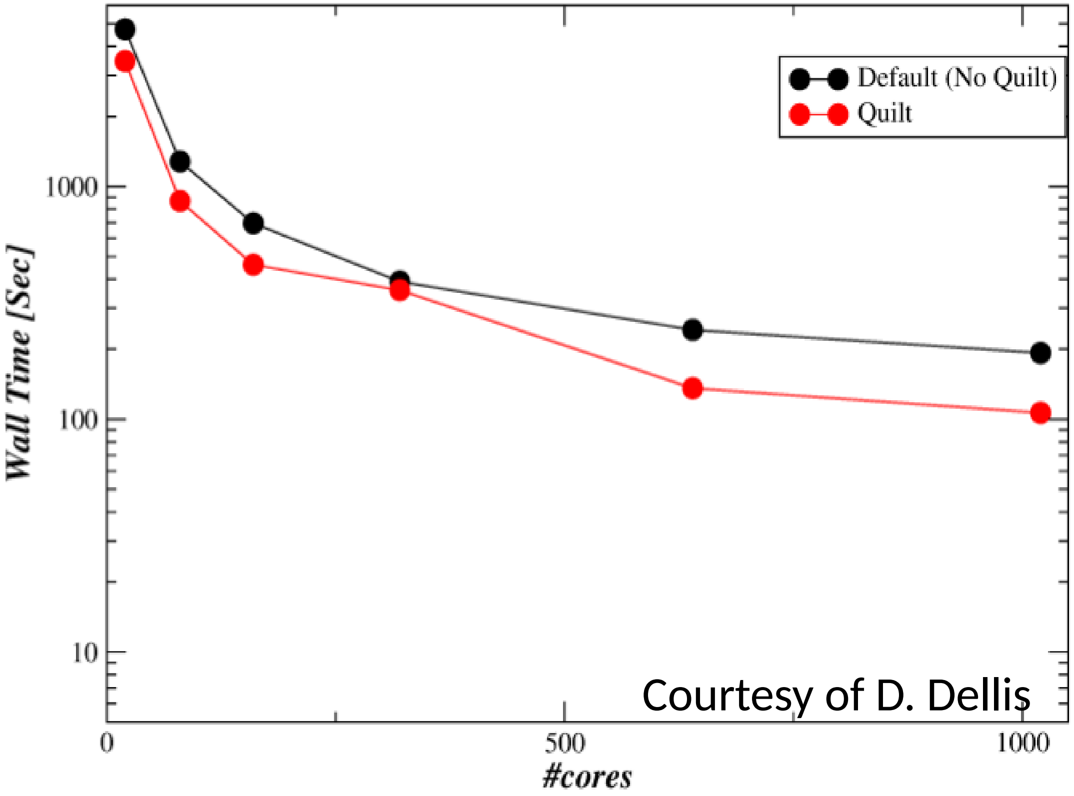
- Represents the amount of wall-clock time attributable to producing the output

Model configuration: I/O

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

Asynchronous I/O (Quilt Servers)

- 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 (not tested on ARIS)
  - 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

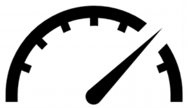
Benchmarking

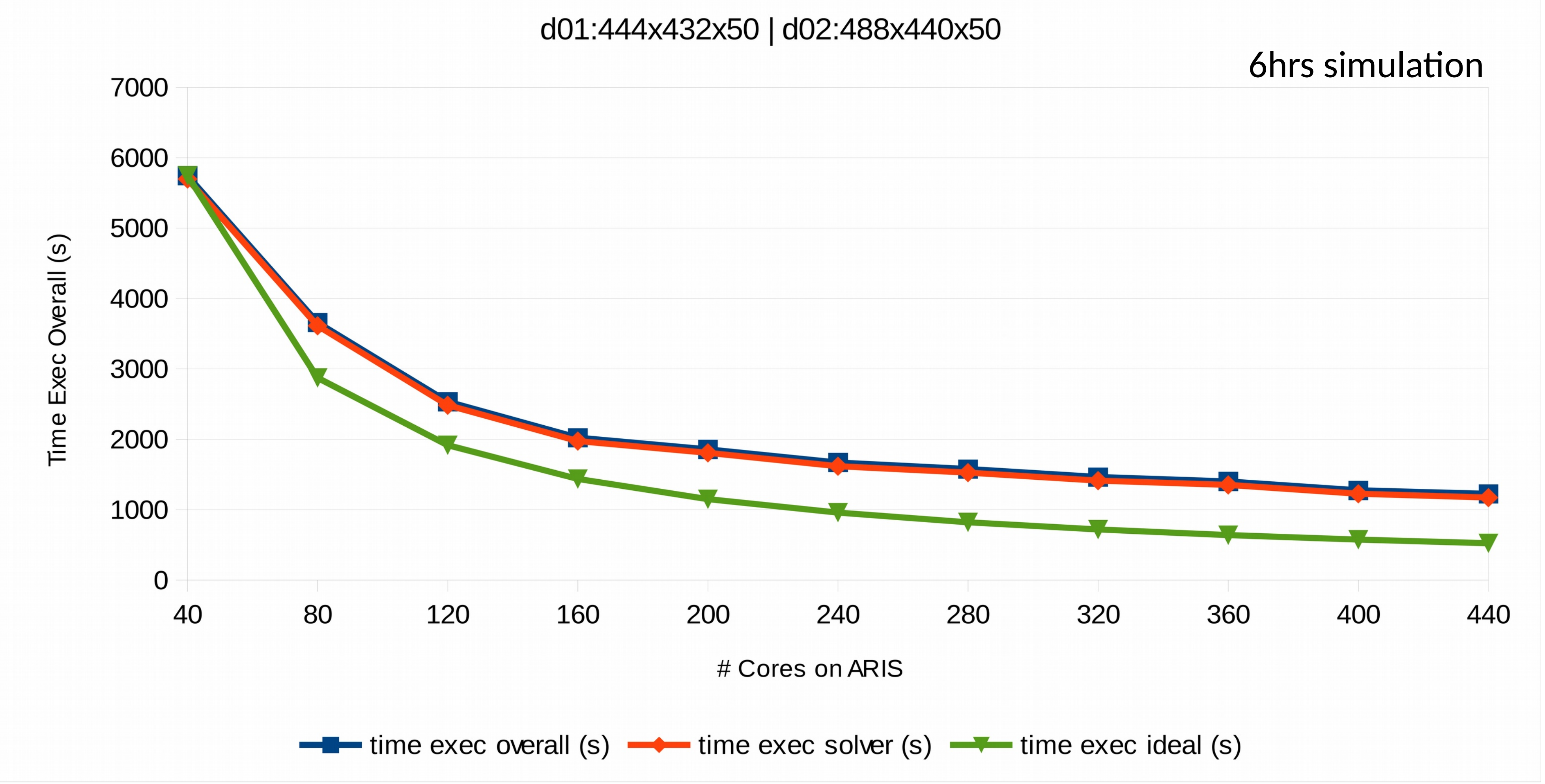
Available in namelist.input file:

- **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

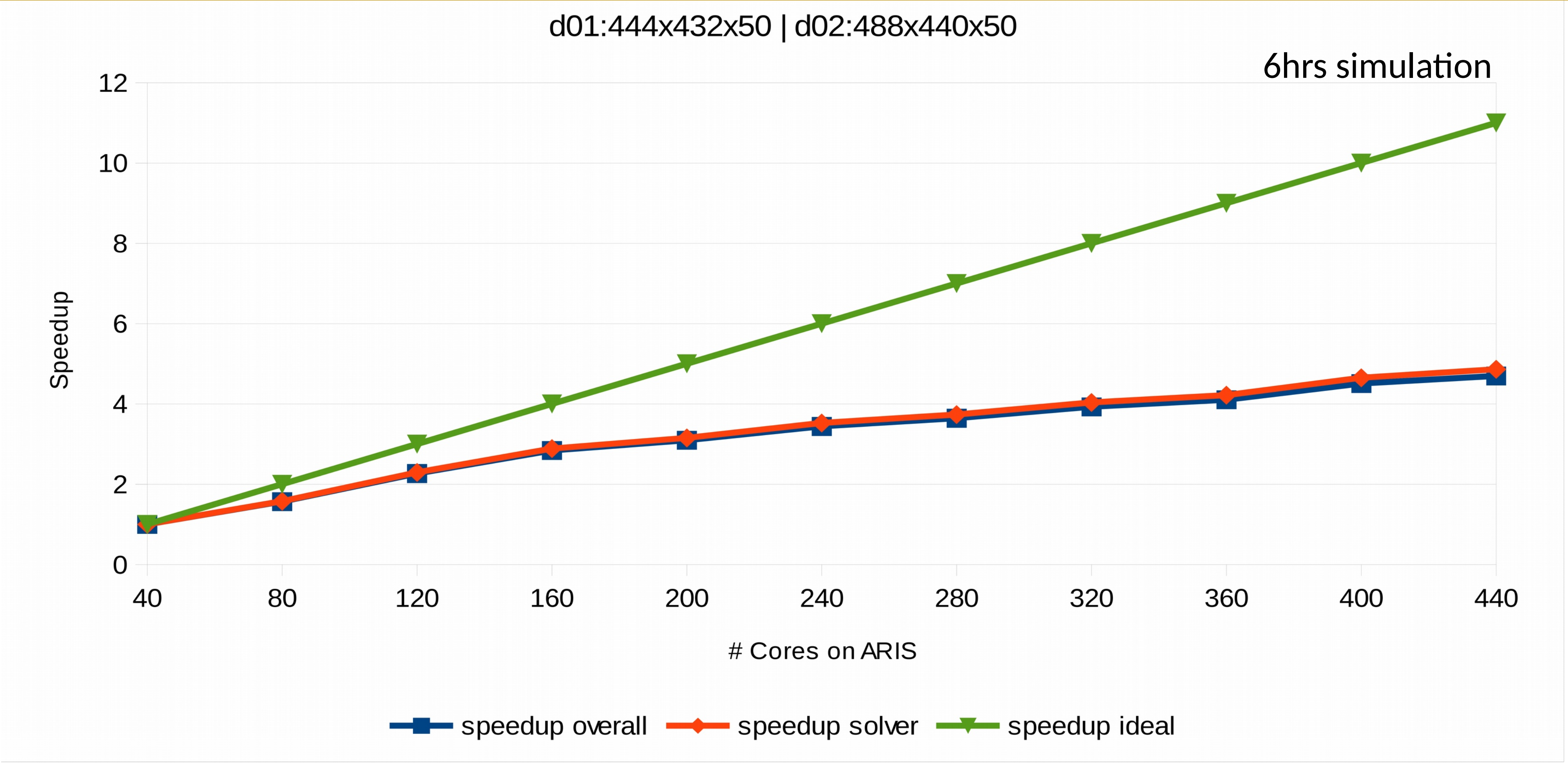




Benchmarking

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



Thank you for your attention!

Questions?

Stergios Kartsios
PhD Candidate
Dep. of Meteorology and Climatology, AUTH
kartsios@geo.auth.gr