Numerical weather prediction in high-performance computing (HPC) environments

Implementation of the WRF model on ARIS

Theodore M. Giannaros

Post-doc Researcher

National Observatory of Athens, IERSD

Email: thgian@noa.gr

Web: http://theodoregiannaros.eu

Vi-SEEM

Virtual Research Environment Portal

Introduction: Overview of parallelism

Parallelism in WRF

- Distributed memory (DM) - “MPI”
- Shared-memory (SM) - “OpenMP”
- Clusters of SM processors (“hybrid MPI+OpenMP”)

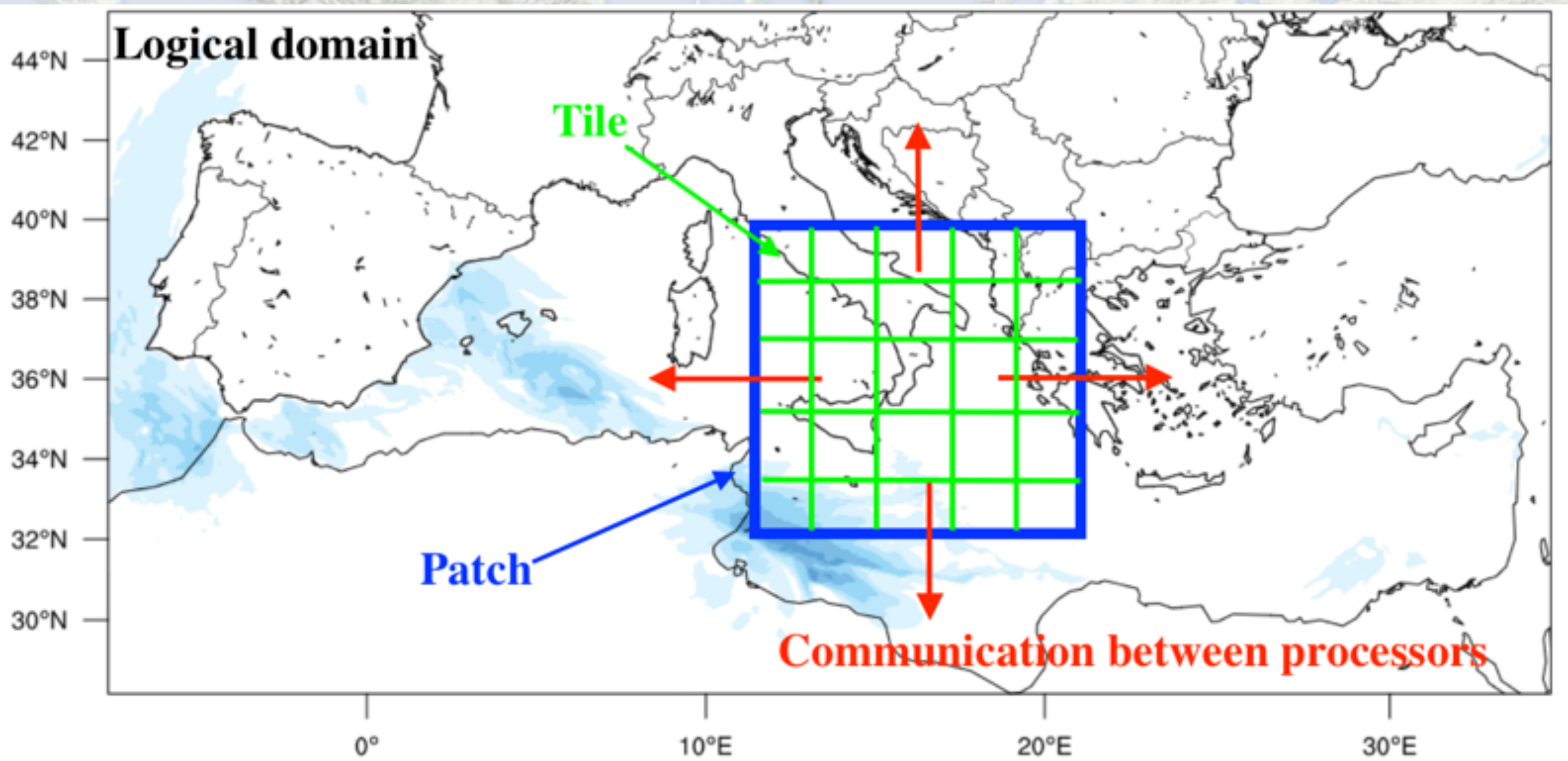
50+ compilation options: Serial, DM, SM, Hybrid (DM+SM), numerous compilers and architectures

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

Introduction: DM versus SM (1)

Domain decomposition

DM works in "**patches**": MPI processes
SM works in "**tiles**": Threads in each MPI process



Introduction: DM versus SM (2)

Example

2 nodes on ARIS, each with 20 CPUs; 40 CPUs in total

So, what are my options?

40 MPI processes, 1 thread per each (pure DM)
OMP_NUM_THREADS=1; mpirun -np 40 ./wrf.exe

OR

20 MPI processes, 2 threads per each (hybrid DM+SM)
OMP_NUM_THREADS=2; mpirun -np 20 ./wrf.exe

OR

10 MPI processes, 4 threads per each (hybrid DM+SM)
OMP_NUM_THREADS=4; mpirun -np 10 ./wrf.exe

OR

and so on …

Experience with WRF has shown that hybrid DM+SM does not always have a positive effect on computational performance
Better to use “pure MPI”

Define your objectives

What are your **scientific** and/or **practical objectives**? **Why** do you need to run WRF? **How** will you know that your simulations are successful?

Get to know your problem

Review literature! What are the **atmospheric processes** involved? Which are the most **important** (clouds, radiation, convection, etc.)? **What** is known? Is anything **missing**? **Judge** the **efficacy** of your "simulations-to-do".

Determine available observational datasets

What **observations** are available? Again, become familiar with the **processes** that you want to study. How will the observations be used for **verifying** and/or **complementing** your simulations? **Judge** the **adequacy** of your "simulations-to-do".

Prepare your strategy

Are you going to focus on a **case study**? If yes, **which** one and **why**? Are there adequate **observations** for verifying your "simulations-to-do"? Will you set up an **operational** weather forecasting service? What are the practical **requirements**?

Model configuration: Domains (1)

Consider first

- **Target** horizontal grid spacing
- Resolution of **initialization** data

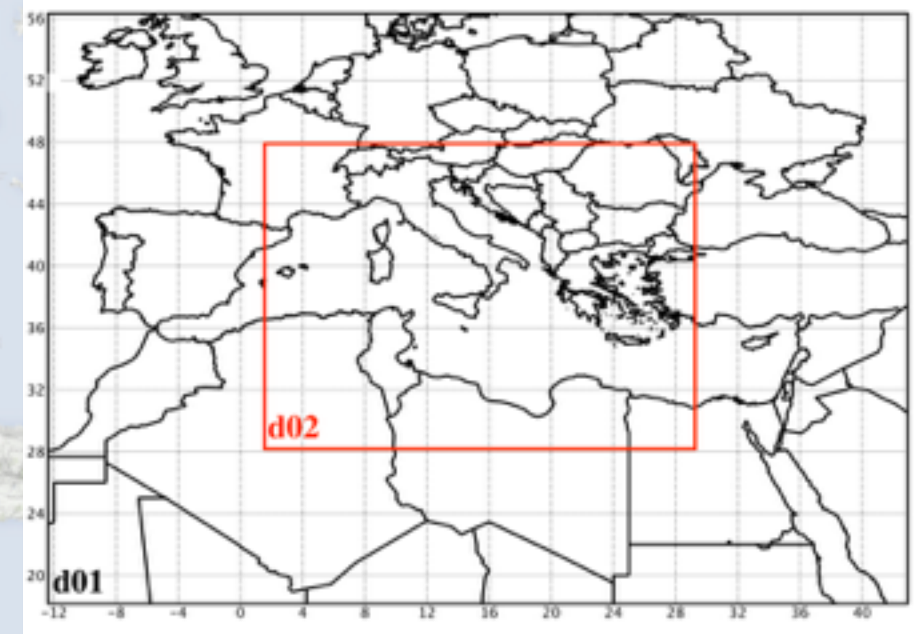
Most often, you will need to adopt a **nesting** strategy.

Hints

- Place domain **boundaries away** from each other, and away from **steep** topography
- Odd parent-child ratios are preferred (e.g. **3:1, 5:1**)
- Higher **horizontal** resolution will also require higher **vertical** resolution
- Use at least **30-35** vertical levels; larger density closer to the ground and to the model top
- Lambert: mid-latitudes, Mercator: low-latitudes, Lat-Lon: global, Rotated lat-lon: regional
- Start **inside-out** (first the nest, move up)

Do remember!

Avoid the “grey zone” (4-10 km)



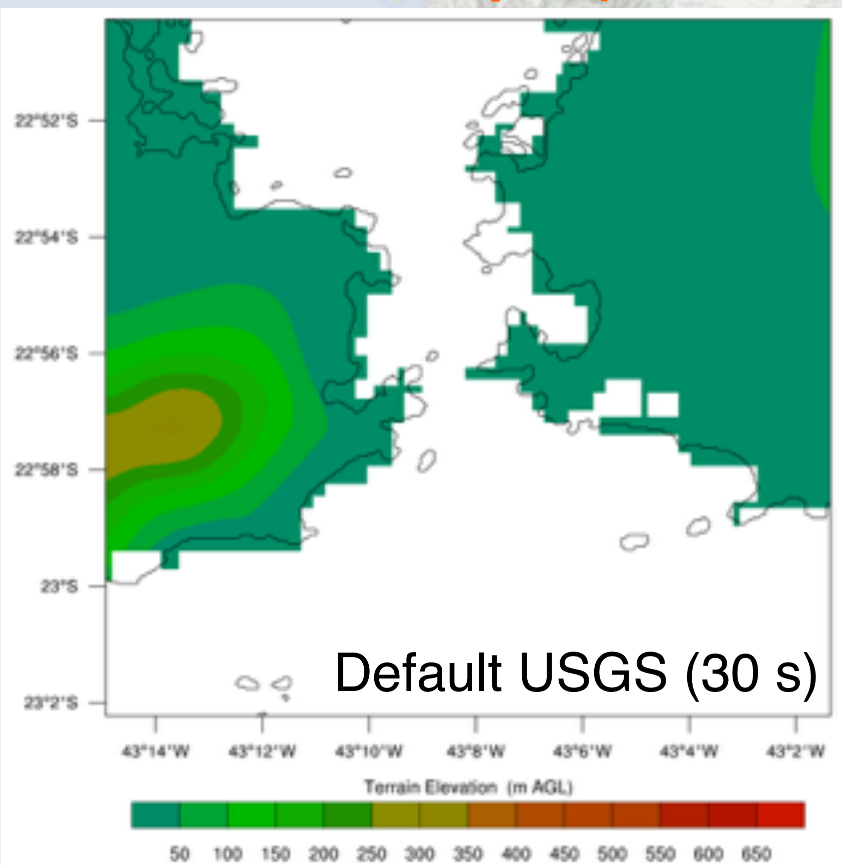
Model configuration: Domains (2)

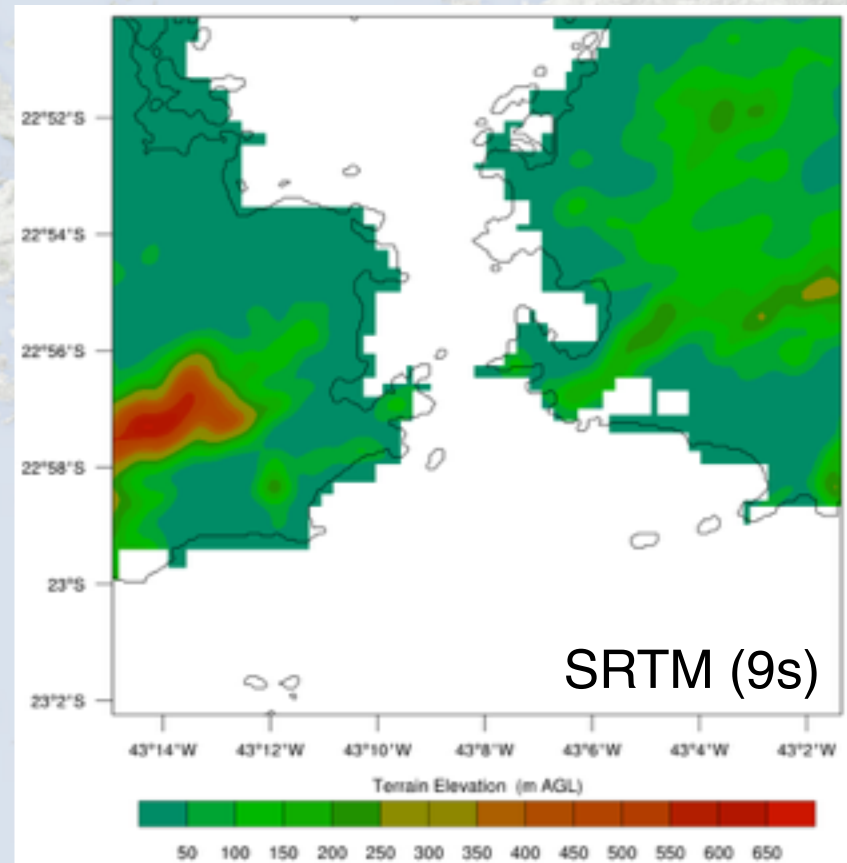
Static (input) data

Does land data represent your area adequately well? If not, consider using alternative datasets (land use, topography)

May have profound impact on your results!

Real-world example (200 m domain for Rio de Janeiro)





Model configuration: Domains (3)

Dynamic (input) data

Ask yourself: how good are the data used for initializing WRF?

Real-world example

Wind forecasts for the Guanabara bay in Rio de Janeiro, Brazil, verified against observed data

GFS: Forecasts driven by 0.5deg, 6h NCEP/GFS

ECMWF: Forecasts driven by 0.5deg, 6h ECMWF/IFS

	B (º)		RMSE (º)		WBE (º)		% WBE<20º	
Location	GFS	ECMWF	GFS	ECMWF	GFS	ECMWF	GFS	ECMWF
SBRJ	-6.4	25.5	91.8	90.9	65.8	62.3	43	54
RJ1	-25.8	-8.4	82.3	70.5	66.4	56.2	16	21
RJ2	-3.3	-9.0	94.9	84.9	74.0	62.9	21	28
RJ3	-7.2	-8.0	83.1	74.8	72.8	60.9	2	20

Table 5. Same as Table 4 but for wind direction verification statistics.

Model configuration: Domains (4)

From a computational point of view

- Assuming a **3:1** parent-child ratio, the nest will require **3x** as many time steps to keep pace with the parent grid.
- Rule of thumb: a nested WRF simulation costs **~4x** the cost of a single parent domain simulation.
- Coarse domains are not a “headache”: **doubling** their grid points will result to **~25%** increase in nested domain simulation time.

Estimating (roughly) the cost (3:1 ratio example)

1. If the fine and the coarse grid have the **same** dimensions (**# of grid points**), then the required **CPU** for integrating a single time step will be **about the same** for both domains.
2. Given that the fine grid time step is **1/3** of the coarse grid time step, it is deduced that the nest will require **3x** the **CPU** to catch up with the coarse domain.

Model configuration: Physics

Too many options! Where to start from?

- Back to basics: Which processes are important? **Review literature**. What others did?
- Consider first well documented (**tried**) schemes

Hints

- Convective schemes are generally not required at **dx<4 km**
- Sophisticated microphysics schemes (double-moment, detailed species) may not be necessary at **dx>>10 km**
- Try to have **consistent physics** between the domains or use 1-way nesting
- If your simulation spans more than **5 days**, you could start thinking to adopt the **SST update** option

		Rad	MP	CP	PBL	Sfc
Atmospheric State or Tendencies	Momentum			i	io	
	Pot. Temp.	io	io	io	io	
	Water Vapor	i	io	io	io	
	Cloud	i	io	o	io	
	Precip	i	io	o		
Surface Fluxes	Longwave Up	i				o
	Longwave Down	o				i
	Shortwave Up	i				o
	Shortwave Down	o				i
	Sfc Convective Rain			o		i
	Sfc Resolved Rain		o			i
	Heat Flux				i	o
	Moisture Flux				i	o
	Surface Stress				i	o

Model configuration: Initialization & Spin-up

Garbage in, garbage out

NWP is a problem of initial conditions! Common “**problematic**” variables:

- **Soil** moisture and temperature
- **Sea**-surface temperature
- Bad representation of **land/sea** mask

Double-check your initial conditions (**wrfinput_d0***)!

Let the model warm-up

- Allow for a **reasonable spin-up period** to avoid “noise” in certain fields (e.g. pressure).
- Spin-up is of great importance for **convection**, particularly deep convection.
- No rules of thumb; **Trial and error** process to identify the “ideal” spin-up period
- Computationally costly, but desired!

Model configuration: Integration

"Stability" versus "efficiency"

Recommended (maximum) integration time step (s) equals **6*dx**(km)
Most often, this needs to be **downscaled** to avoid numerical instability (CFL violation)

Example

1-way nested, **15 km** coarse grid (**CG**) and **5 km** fine grid (**FG**)

- Ideally: CG dt=6*15=90s, FG dt=90/3=30s (parent dt divided by 3:1 ratio)

Result: Model "**blows up**" quickly after the beginning of the simulation

- Reduce time step: CG dt=60s, FG=60/3=20s

Result: Model becomes numerically **steady**; but also 90/60=**1.5x more expensive**

- Reduce time step only for CG: CG dt=60s, FG=60/2 (parent dt divided by 2:1 **time step ratio**)

Result: Model becomes numerically **steady**; **save** computational time

Remember

You can reduce the CG time step without reducing model performance, as long as you are able to tweak the FG time step (adjust parent-child time step ratio; trial and error)

Model configuration: Tackling CFL errors

Model "blows up" with CFL errors

Troubleshooting:

Check "**where**" the model becomes unstable: (a) which **vertical** level, (b) which **i,j** in model domain

A. If CFL violation occurs at the **first few vertical levels**, then it's probably due to steep **orography**: (i) 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**)

B. If CFV violation occurs at **upper vertical levels**, then the available options you have are: (i) use the damping option for vertical velocities (**w_damping=1**), (ii) use a different damping option (**damp_opt=1,2,3**), (iii) **reduce** your integration time step, (iv) consider restructuring your **eta_levels** (if you defined them explicitly)

I/O optimization

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

Good to remember

Output data **quick**ly
Output **small** data
Output **less** data

Hints

- Use runtime i/o to reduce output variables (**iofields_filename**=“varsout.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** 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.

Benchmarking WRF (1)

Definitions

Performance: Model speed ignoring I/O and initialization costs, measured directly as the *average cost per model time step* over a representative integration period. Can be expressed as either *simulation speed* or floating-point rate.

Simulation speed: Measure of the *actual time-to-solution*. Expresses the ratio of model time simulated to the actual time, and is computed as the *ratio of model time step to the average time per time step,* over a representative integration period.

Scaling: The *ratio of increase in simulation speed to the increase in parallel processes*.

ARIS benchmarking tests

Case A: Single domain, 235x175x40, 24 km (Europe)

Case B: Case A & 685x235x40, 6 km (Mediterranean)

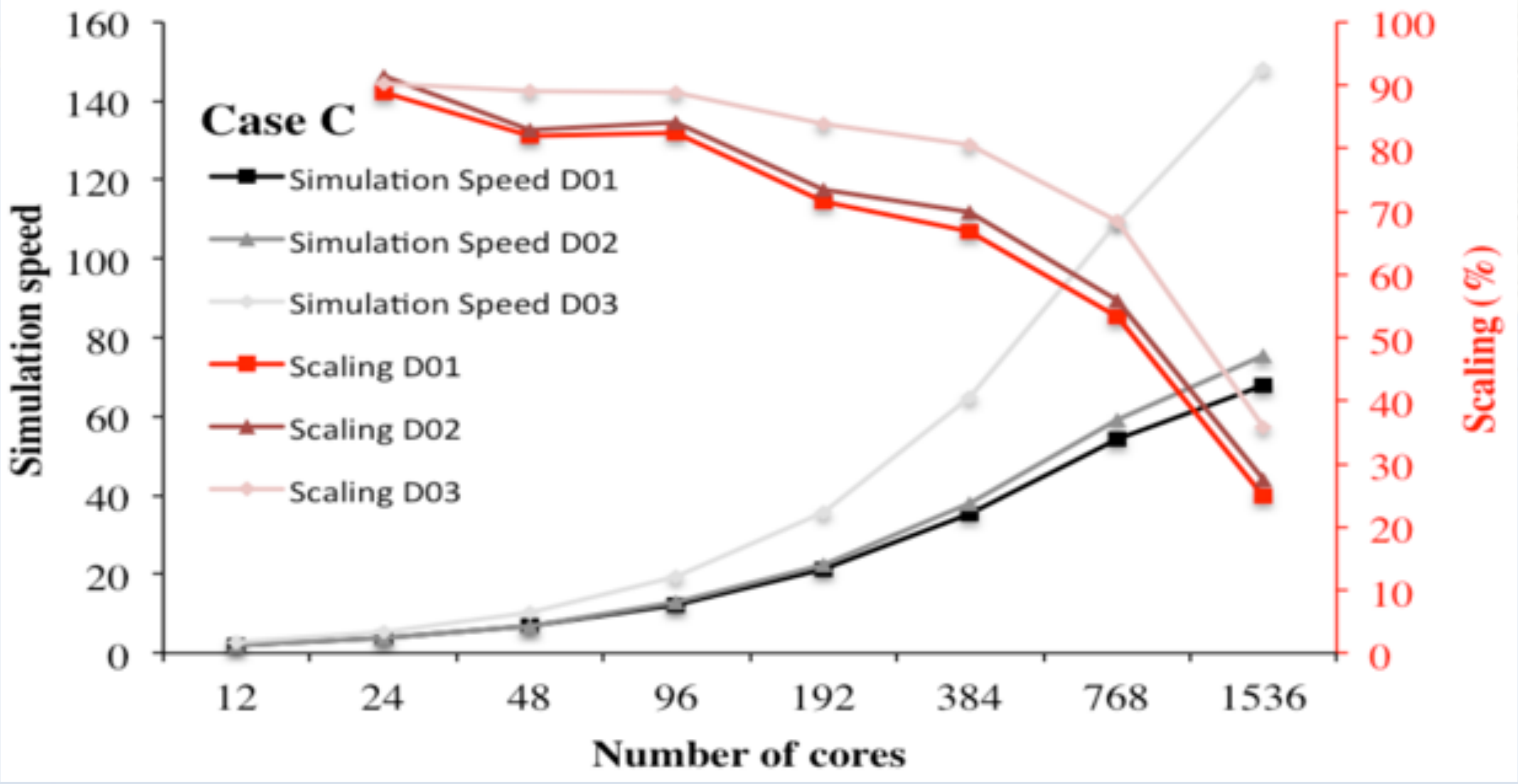
Case C: Case B & 538x499, 2 km (Greece)

- 60 h numerical simulations
- Benchmark period: T0+13 - T0+60 (48 hours)
- Same physics for all cases and domains

Benchmarking WRF (2)

Case C

Number of cores	Time-to-solution (hh:mm:ss)
12	26:34:53
24	15:59:33
48	08:47:03
96	04:50:50
192	02:49:45
384	01:42:29
768	01:08:09
1536	00:54:57



Definitions

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.

Hint

WRF responds better to a more rectangular decomposition, i.e. **nproc_x<<nproc_y**:

- Longer inner loops for better vector and registry reuse
- Better cache blocking
- More efficient halo exchange communication pattern

Best combination defined by **trial and error**!

Take-away for MPI

- As the number o MPI tasks increases, the amount of work inside each MPI task decreases
- More MPI tasks, more contention for due to communications is likely
- As the computation time gets smaller compared to the communications time, parallel efficiency suffers

Thank you for your attention!
Questions? Comments?

Theodore M. Giannaros

Post-doc Researcher

National Observatory of Athens, IERSD

Email: thgian@noa.gr

Web: http://theodoregiannaros.eu

Vi-SEEM
Virtual Research Environment Portal