Introduction to WRF: The Single-Domain Test Case and The 3-Domain Test Case

Dave Gill
National Center for Atmospheric Research

Gerardo Cisneros-Stoianowski
HPC-AI Advisory Council

Ophir Maor
HPC-AI Advisory Council

This material is based upon work supported by the National Center for Atmospheric Research, which is a major facility sponsored by the National Science Foundation under Cooperative Agreement No. 1852977.

WRF

- What?
- Who?
- Why?
- Basic Concepts
- Nesting
- WRF Single-Domain Case
- WRF 3-domain Case
- Quick Guide to WRF Parallelism

Overview on the ISC21 SCC WRF task

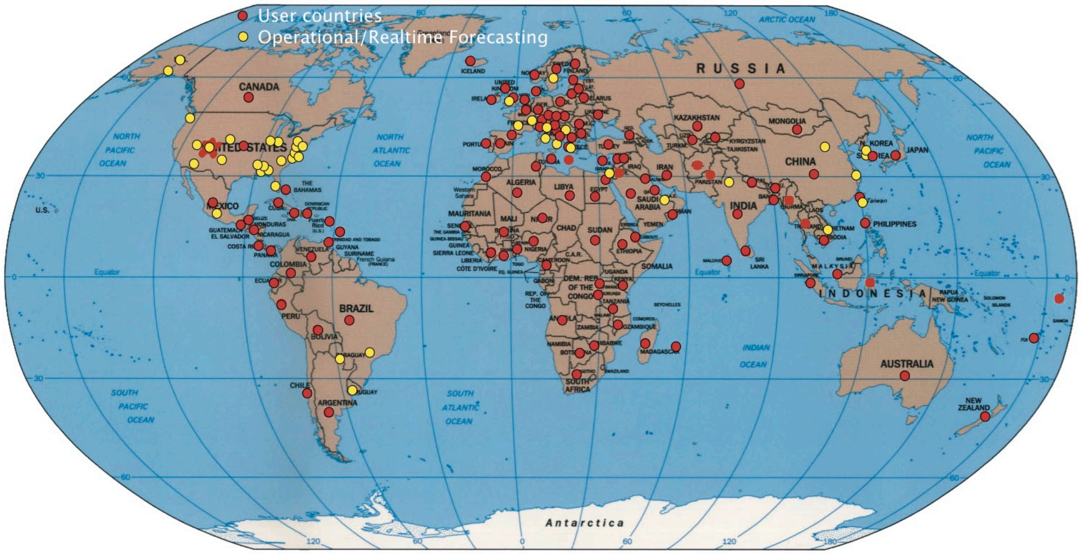
- Minimum requirements for participation
- Optimization: what to tune and *not*

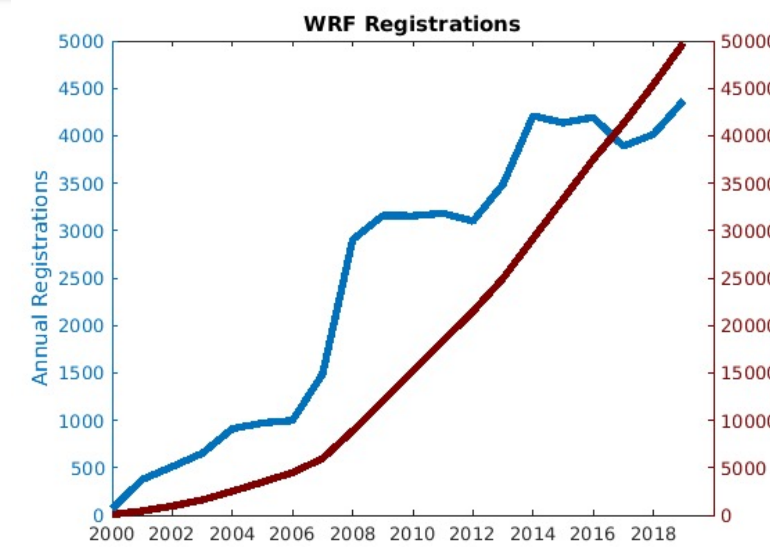
This material is based upon work supported by the National Center for Atmospheric Research, which is a major facility sponsored by the National Science Foundation under Cooperative Agreement No. 1852977.

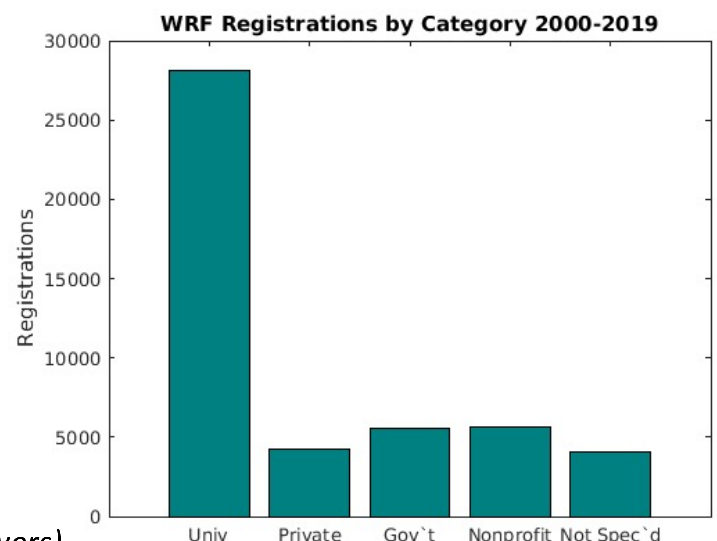
What is WRF?

- WRF: Weather Research and Forecasting Model
- It is a supported "community model", i.e. a free and shared resource with distributed development and support
- Its development is led by NCAR, with partnerships and collaborations with universities and other government agencies in the US and overseas

Who Uses WRF







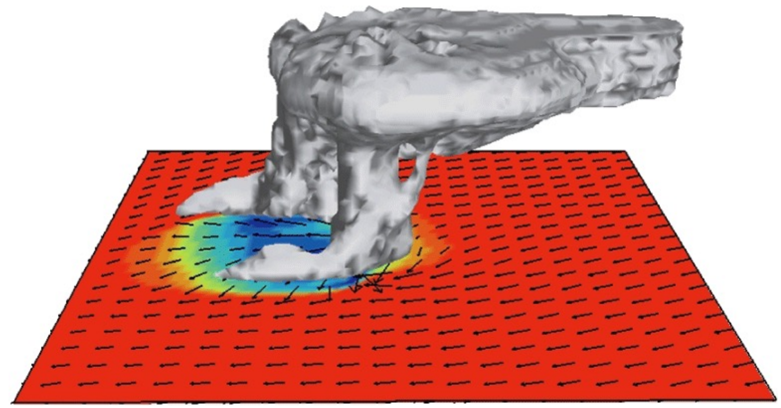
As of Dec 2020:
No. of countries: 184
No. of users: 52570
US: 13970 (~ 26%)
Foreign: 38600

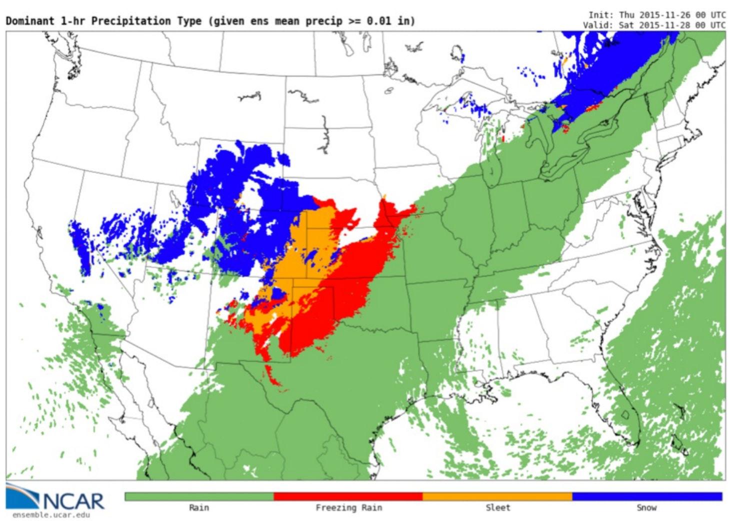
(From Powers)

Why Use WRF

- A research tool:

Idealized simulations →



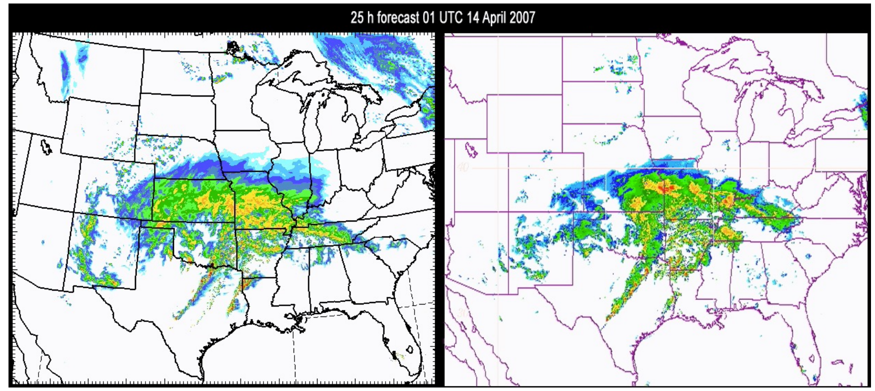


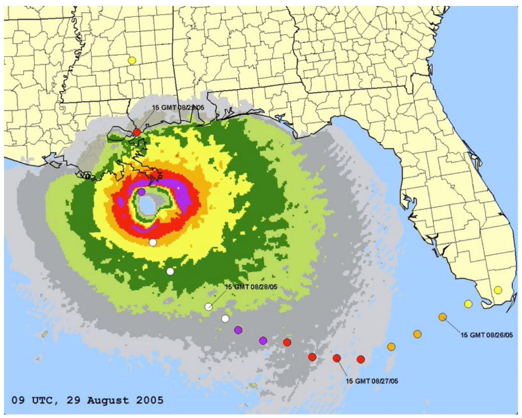
← Experimental real-time forecast

Why Use WRF

- A research tool:

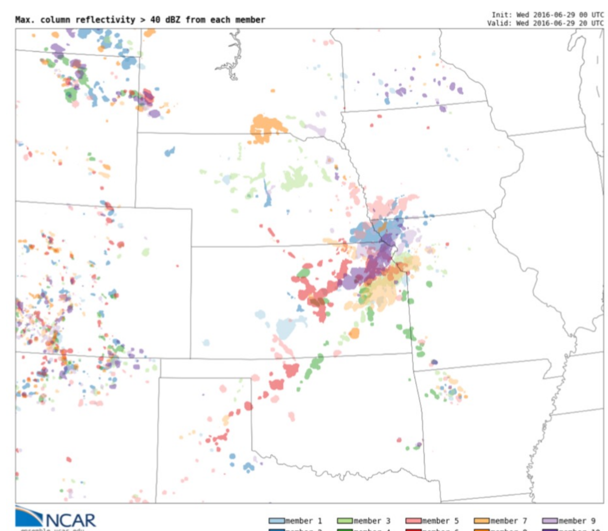
Convection forecast →





← High-resolution hurricane simulations

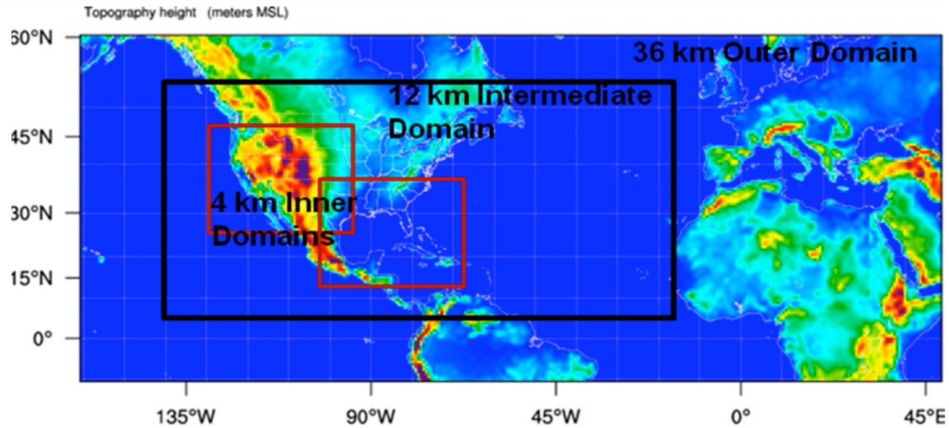
Development of ensemble forecasting technology →

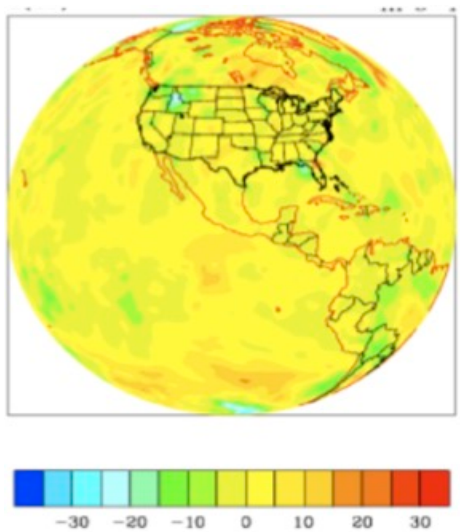


Why Use WRF

- A research tool:

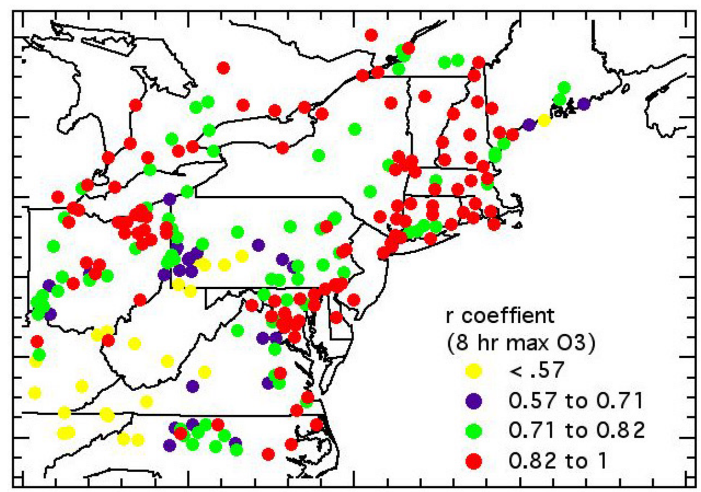
Regional Climate Modeling →





← Data assimilation (*analysis increments*)

WRF-Chemistry →
(*O3 forecast*)



Why Use WRF

- A tool for research
  - Develop and test physical parameterizations
  - Case-study research for specific weather events
  - Regional climate studies
  - Coupled-chemistry, fire, and hydrological applications
  - Data assimilation research
  - Teaching modeling and NWP

- A tool for numerical weather prediction
  - Hind-casting
  - Real-time (operational) forecasting
  - Forecasting for wind, solar and air quality (online and offline)

Basic Concepts

- How does a model work and what does time integration mean?

$$\frac{\Delta A}{\Delta t} = F(A)$$

ΔA = change in a forecast variable at a particular point in space

$F(A)$ = represents the dynamical and physical processes that can change the value of A

Δt = change in time

So a forecast is

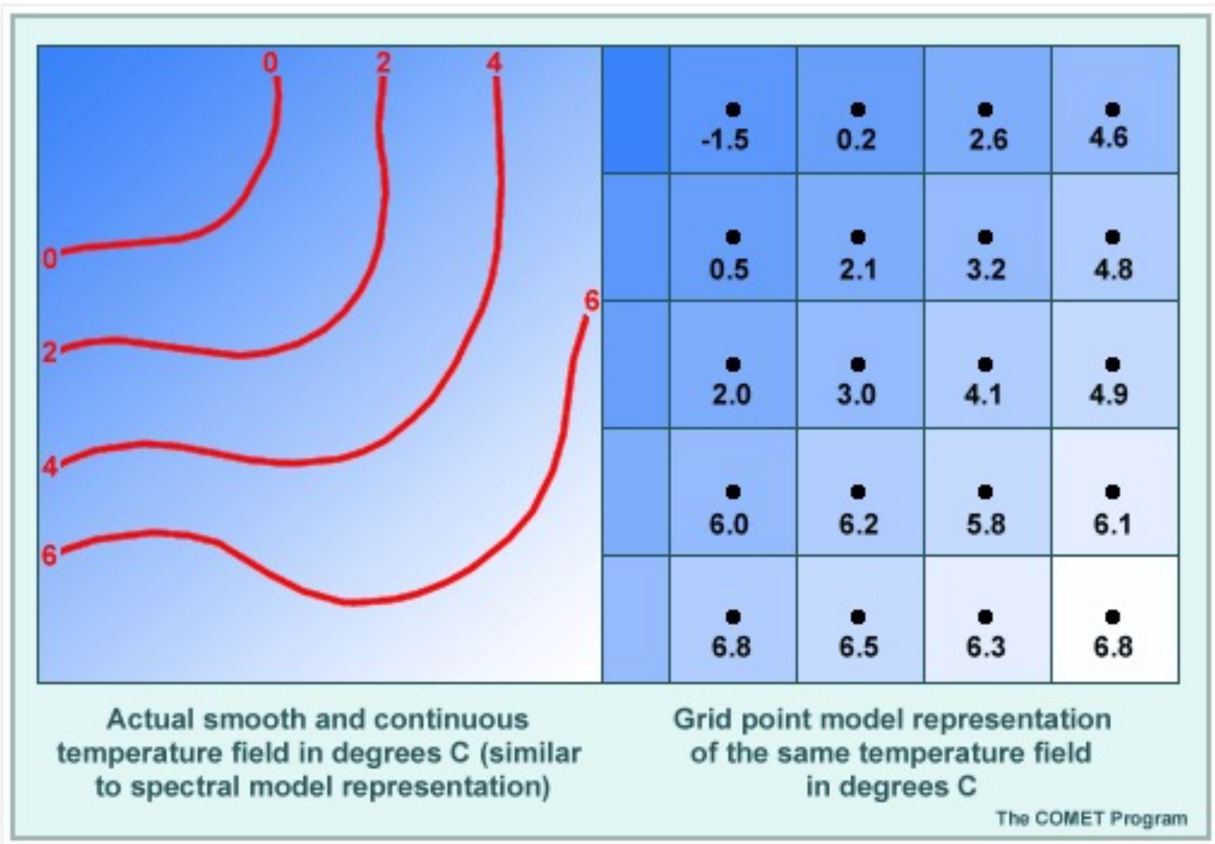
$$A^{forecast} = A^{initial} + F(A)\,\Delta t$$

$$A^{N} = A^{initial} + F_1(A)\,\Delta t + F_2(A)\,\Delta t \ldots + F_N(A)\,\Delta t$$

(adapted from COMET)

Basic Concepts

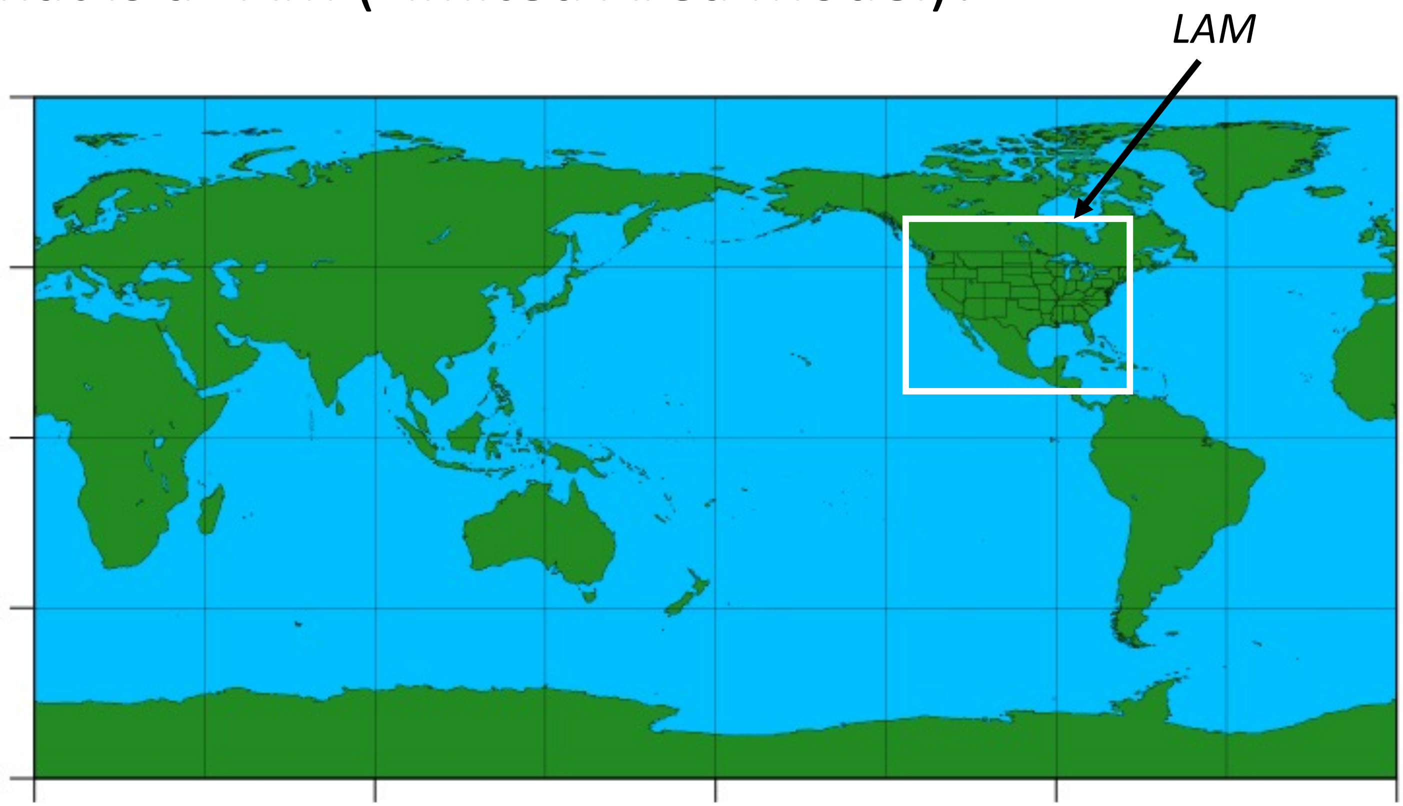
- How are data represented, and equations solved on a model grid?



(from COMET)

Basic Concepts

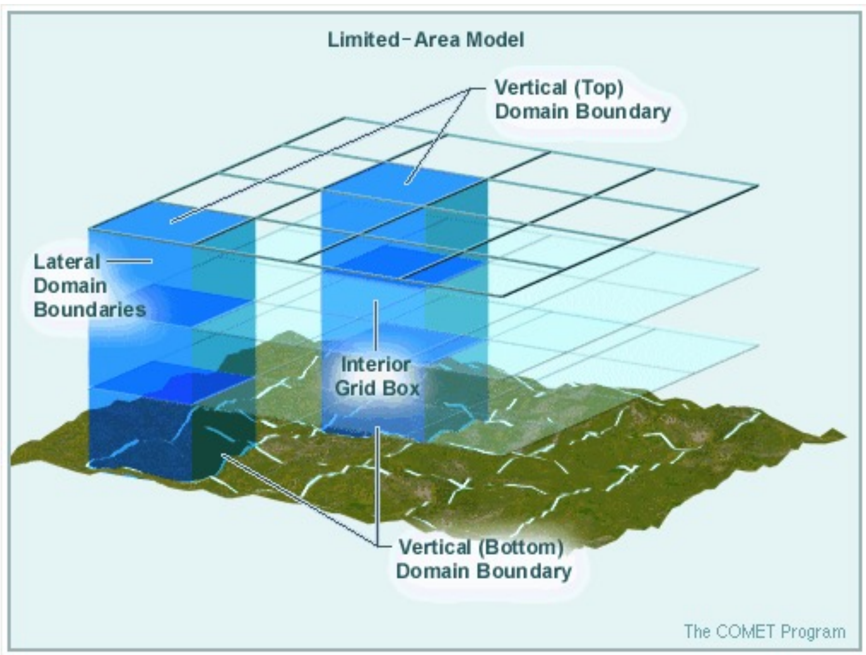
- What is a LAM (Limited Area Model)?

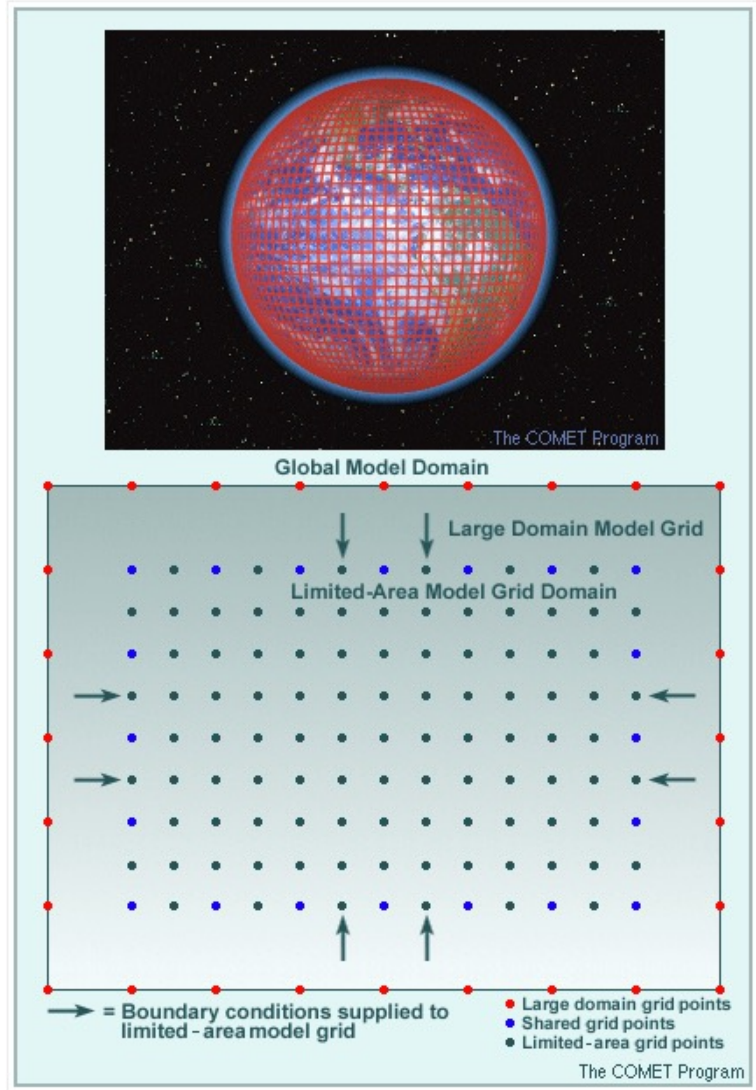


Global Model

Basic Concepts

- What is a LBC (lateral boundary condition)?

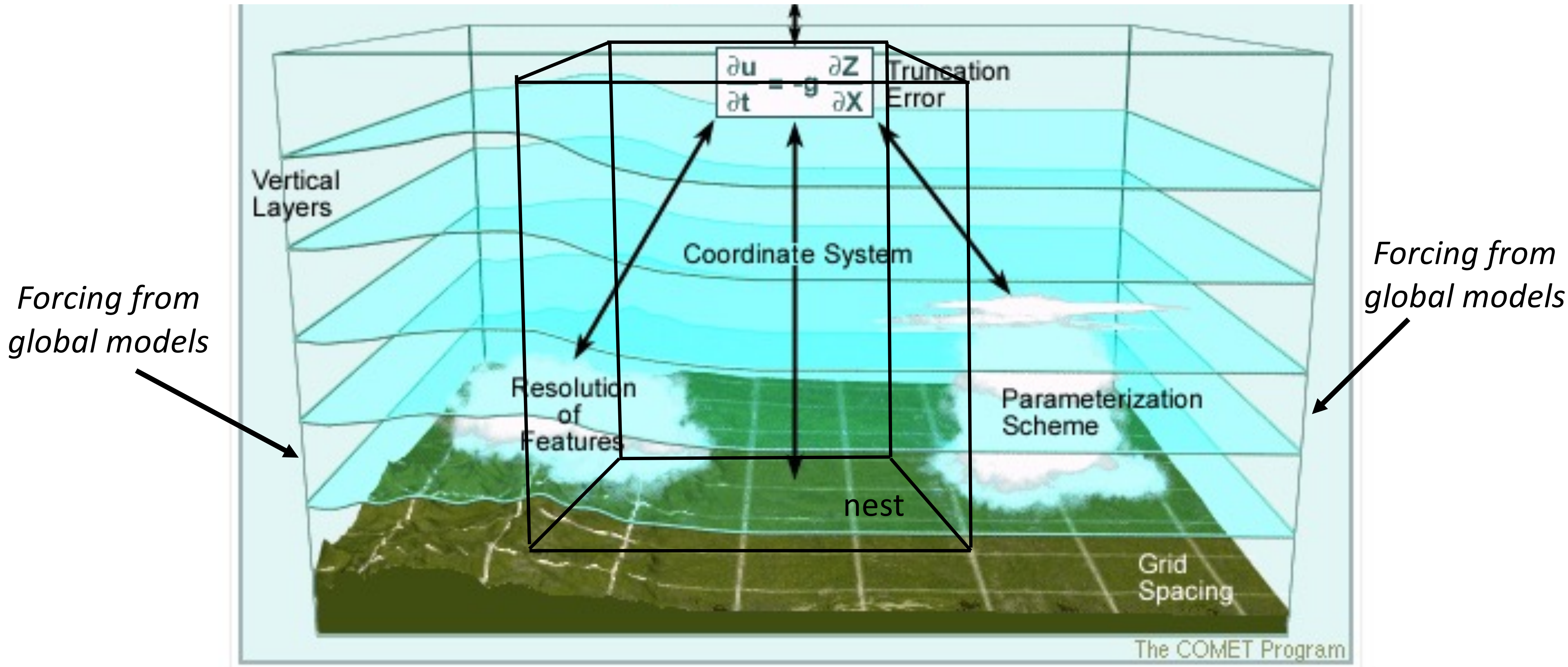




(from COMET)

Basic Concepts

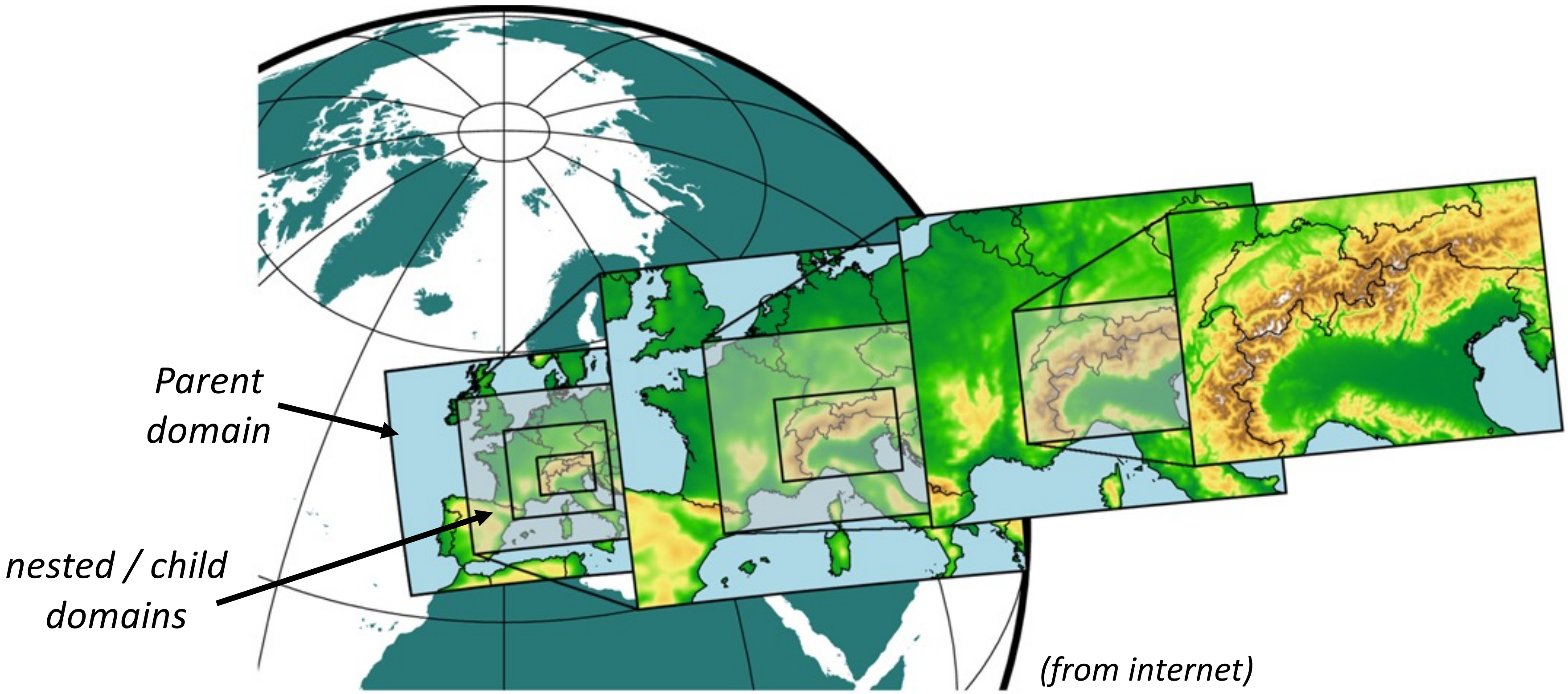
- A 3D view of LAM



(partially from COMET)

Nesting

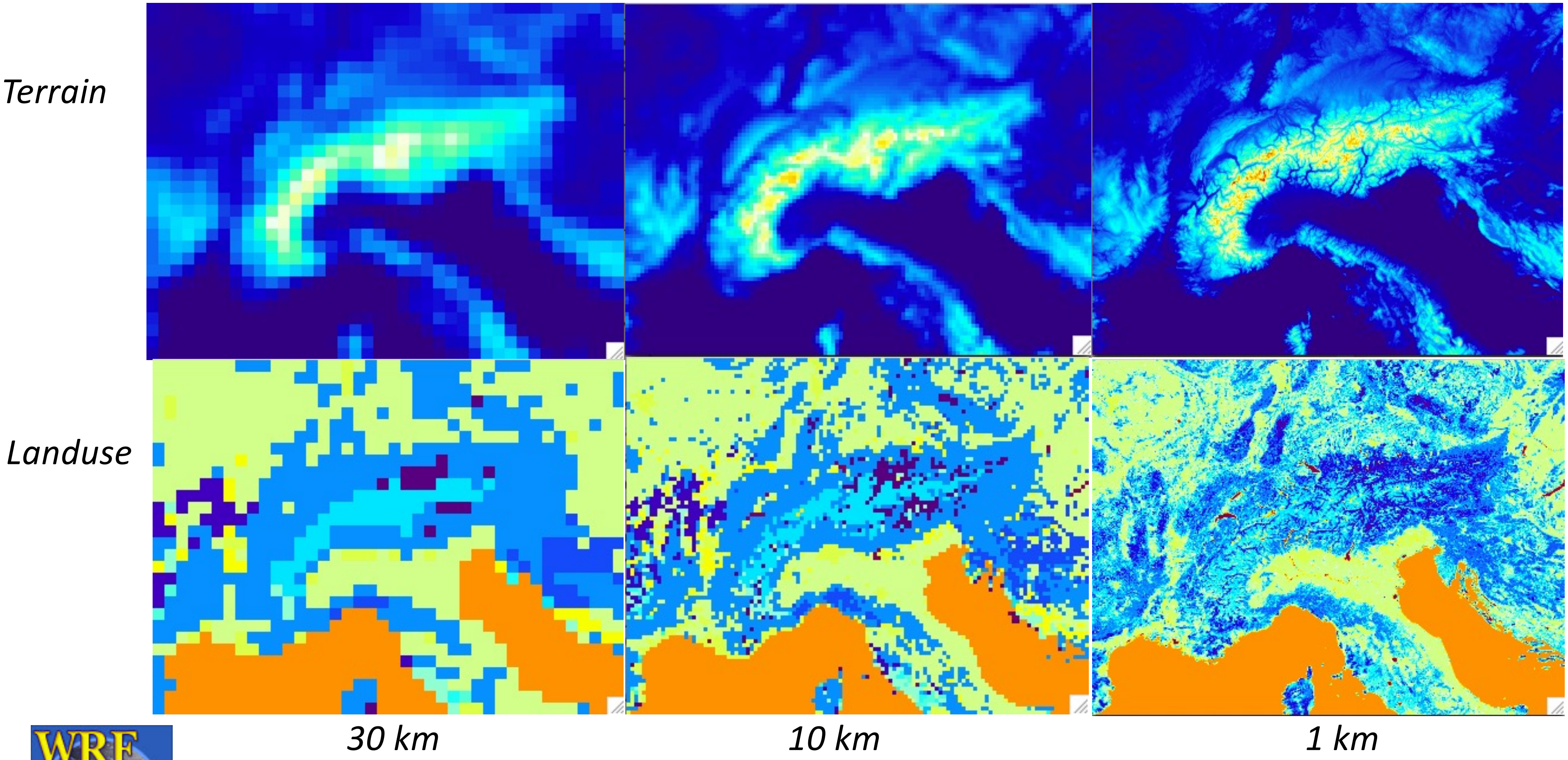
- Nesting in limited area model



(from internet)

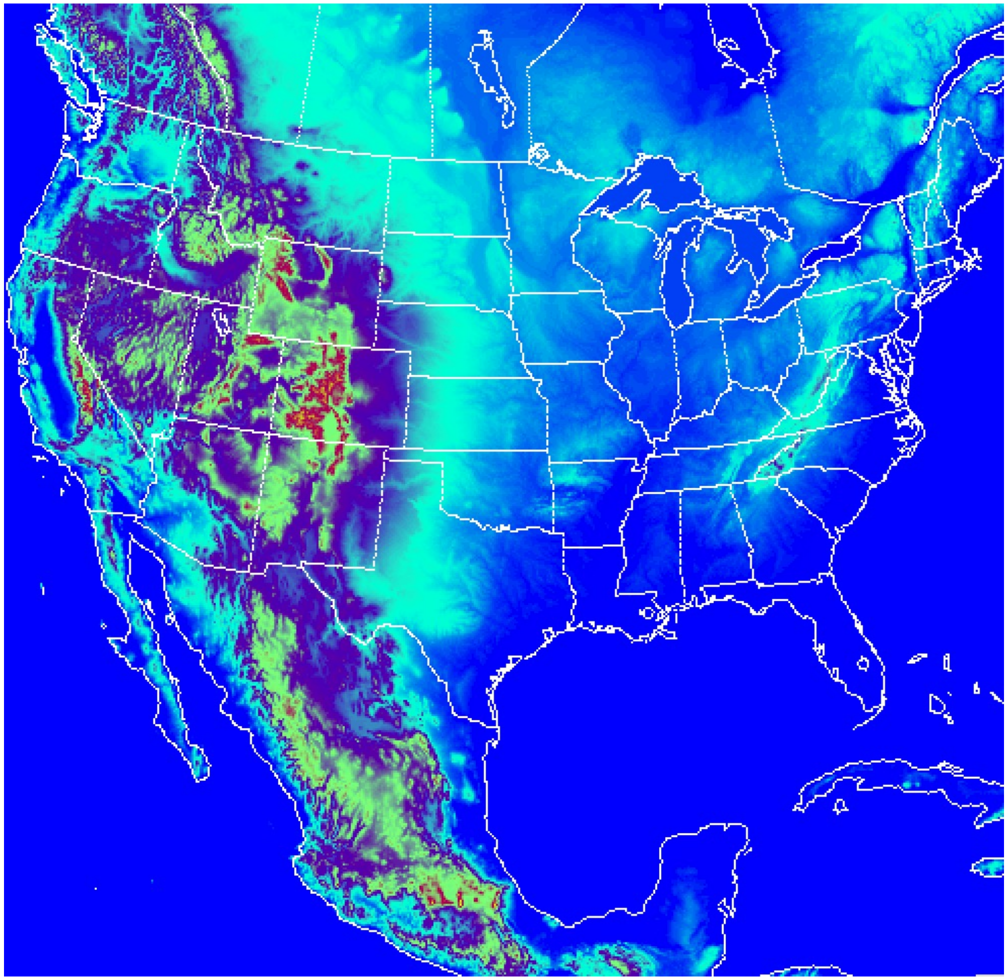
Nesting

- Why nesting? An efficient way to obtain high resolution model solutions.



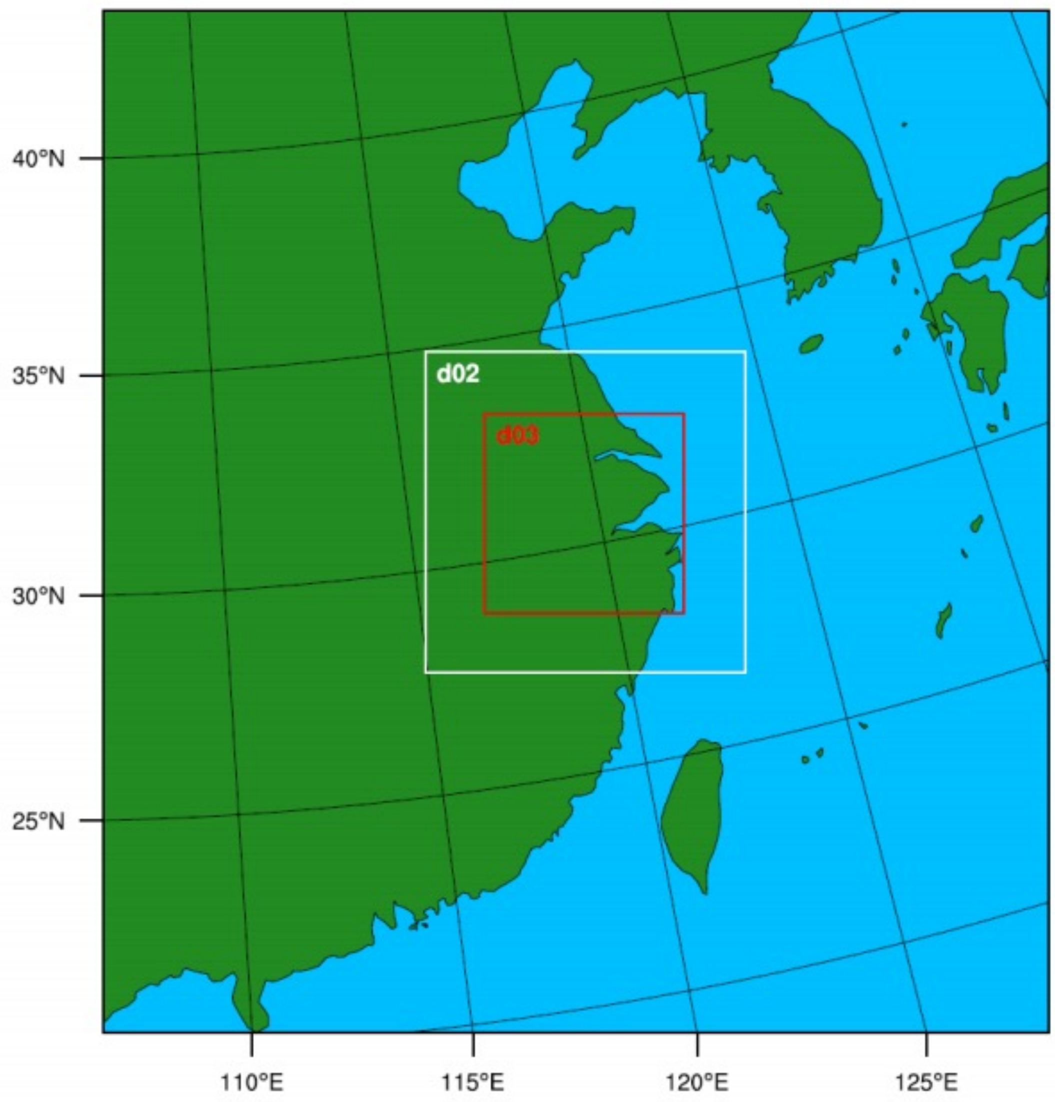
WRF Single-Domain Case

- *50 vertical levels*
- *No cumulus*
- *Hybrid vertical activated*
- *Moist theta*
- *18 s dt*
- *6 minute simulation*
- *10 hour spin-up, then restart*
- *No I/O included in timing*
- *Single radiation time step*
- *18 non-radiation time steps*



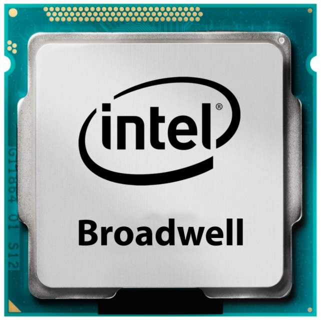
WRF 3-Domain Case

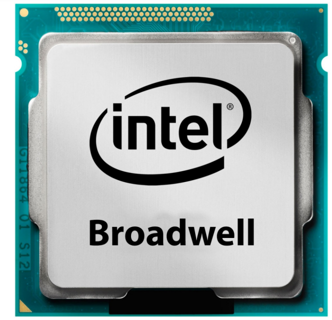
- *Domain 1:*
  - *3.0 km*
  - *793x853*
- *Domain 2:*
  - *1.0 km*
  - *805x805*
- *Domain 3:*
  - *0.5 km*
  - *1001x1001*

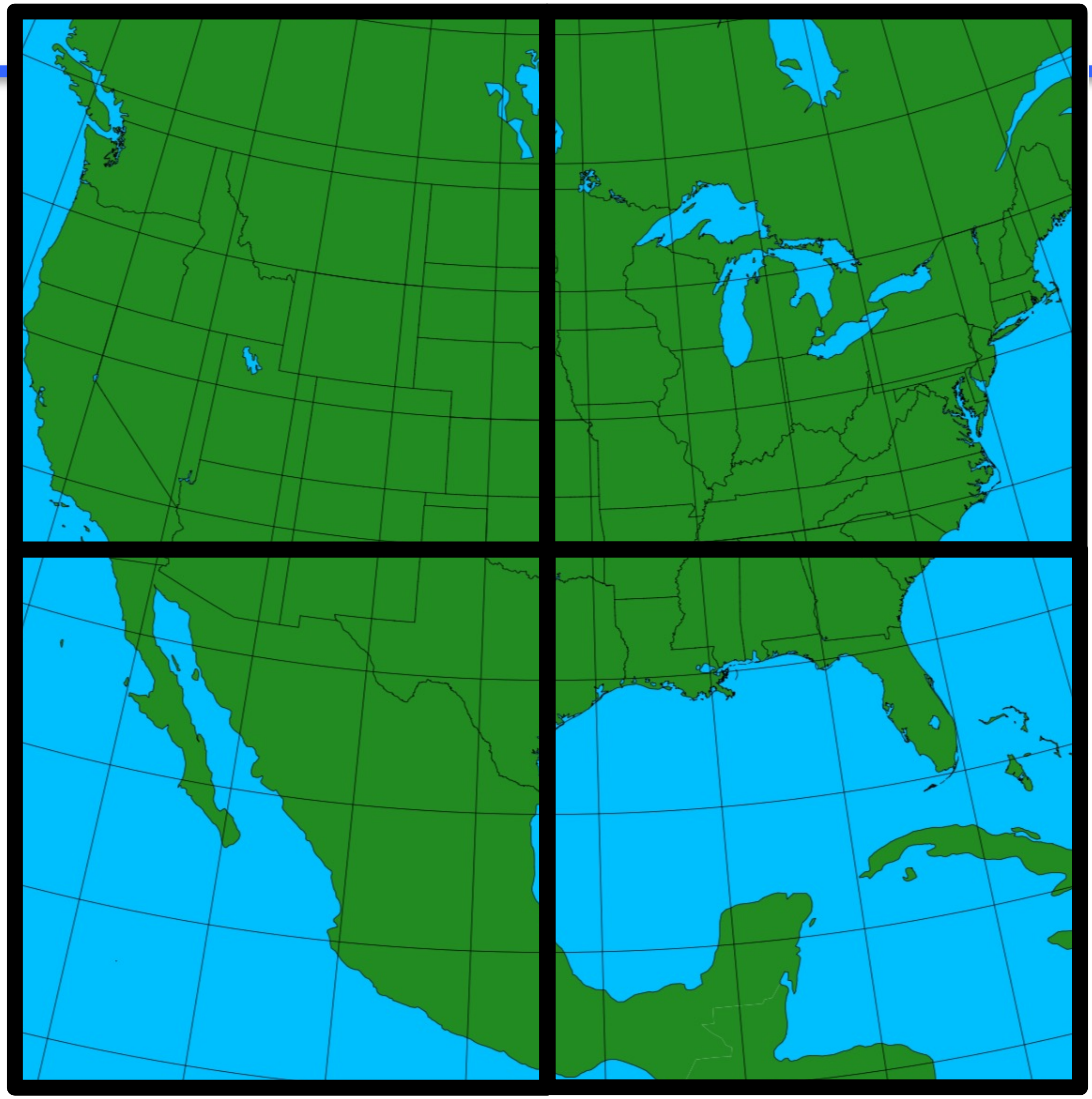


Quick Guide to WRF Parallelism

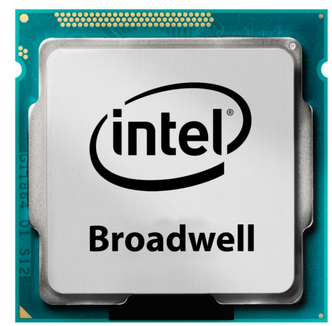
- OpenMP is ALWAYS within a single share-memory processing unit
- With MPI patches need to sometimes send and receive information from each other, referred to as messages and message passing
- WRF uses HALO regions to assist with message passing

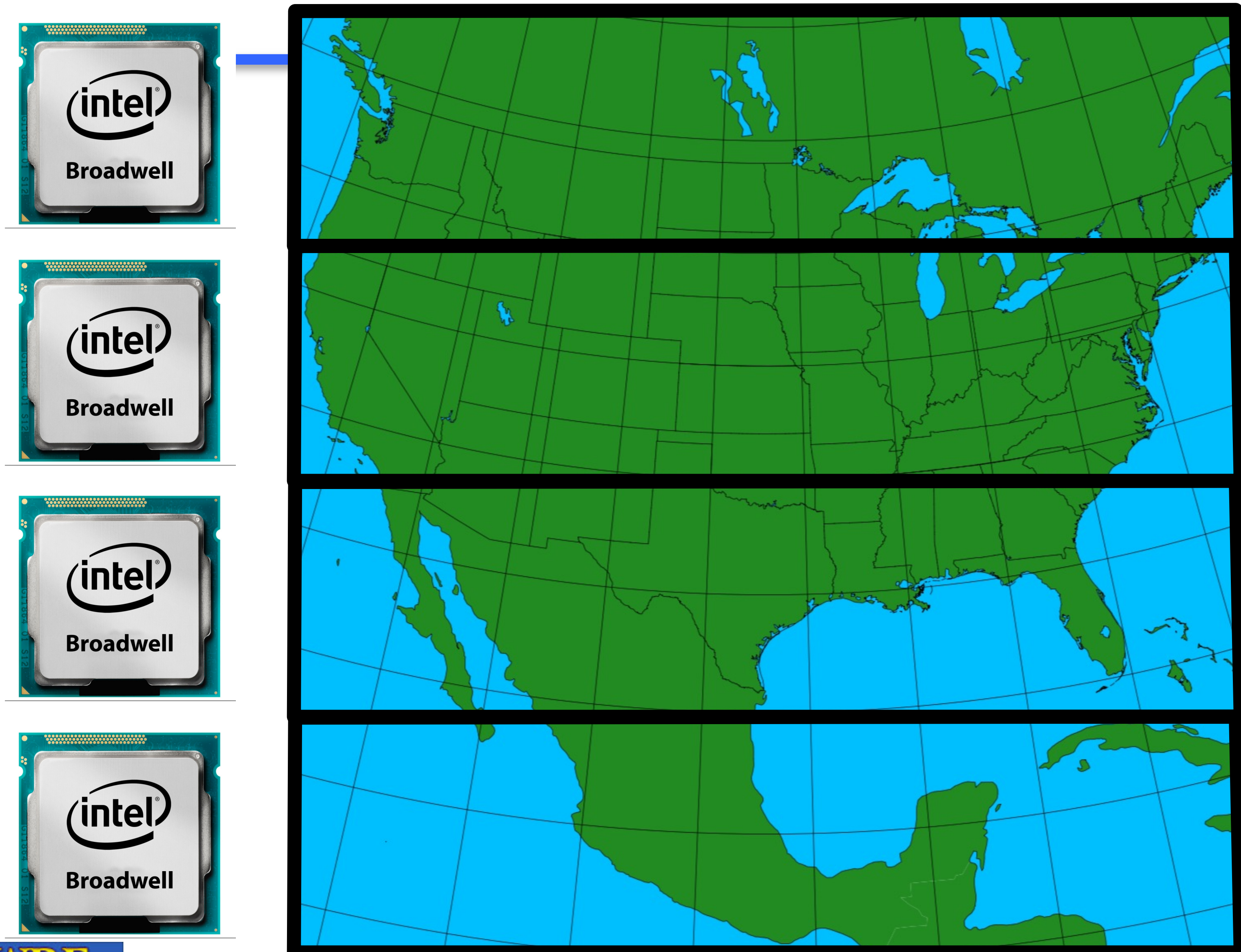

intel
Broadwell


intel
Broadwell




intel
Broadwell

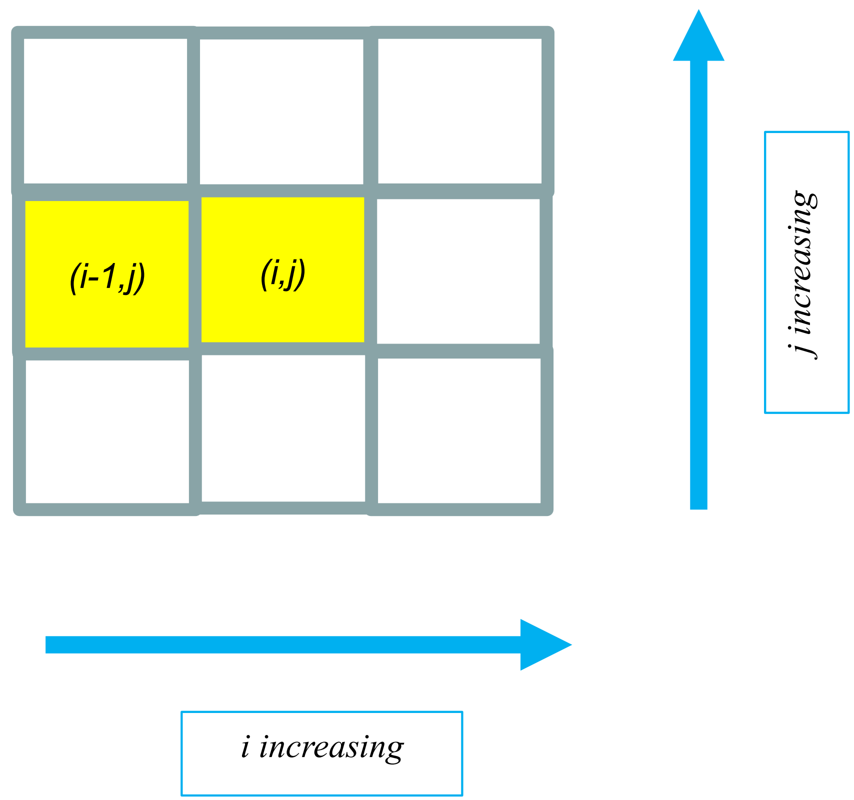

intel
Broadwell


intel
Broadwell
intel
Broadwell
intel
Broadwell
intel
Broadwell

Quick Guide to WRF Parallelism

```
porig(i,k,j) = ( po(i,k,j) + po(i-1,k,j) ) * 0.5
```

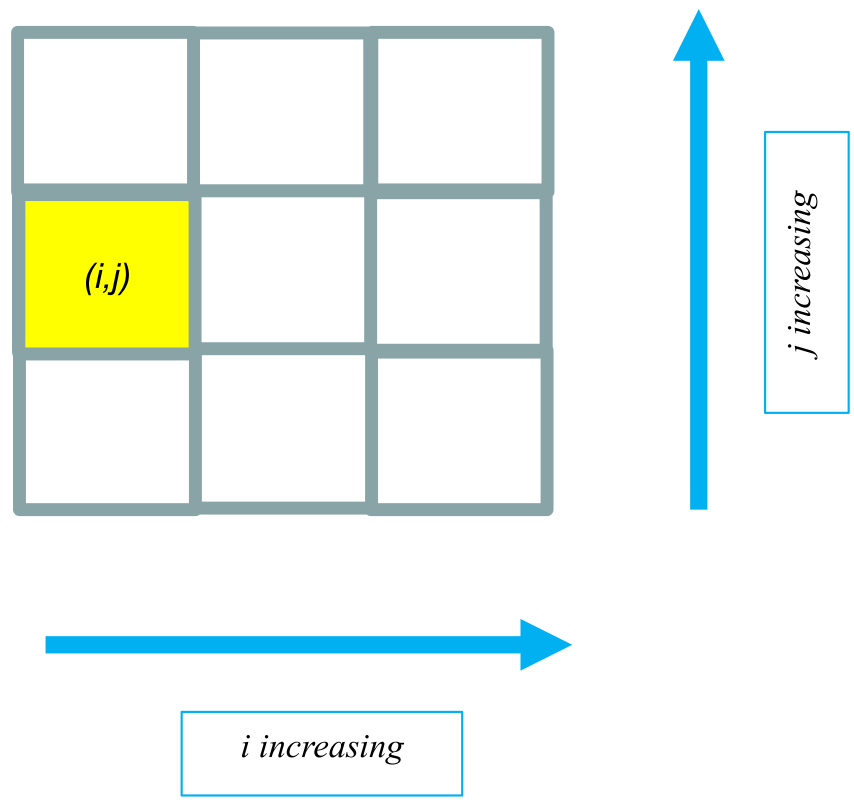
- Solve a simple 2-point stencil, used for averaging a mass-point pressure to a momentum cell face location
- The assumption is that for each (i,j), the (i-1,j) location is a neighboring point



Quick Guide to WRF Parallelism

```
porig(i,k,j) = ( po(i,k,j) + po(i-1,k,j) ) * 0.5
```

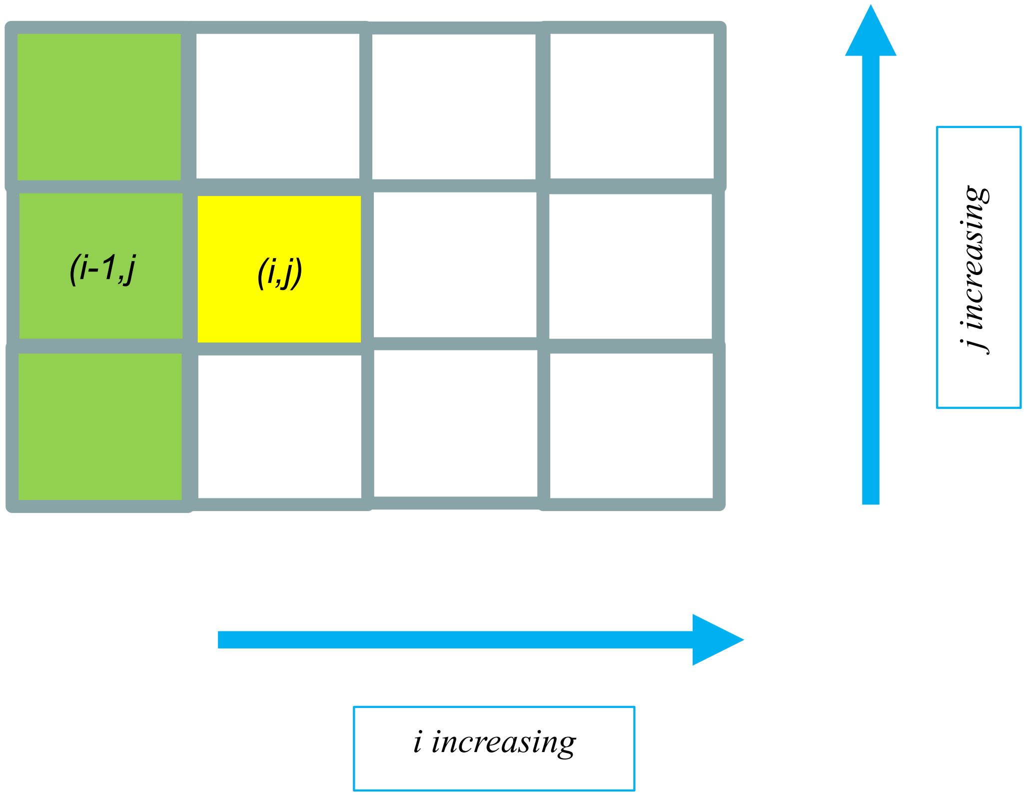
- For this stencil, if our grid cell lies on a western boundary, there is no neighboring point
- To get the information, we could communicate with the next patch and request the data
- However, communication is much slower than local memory access

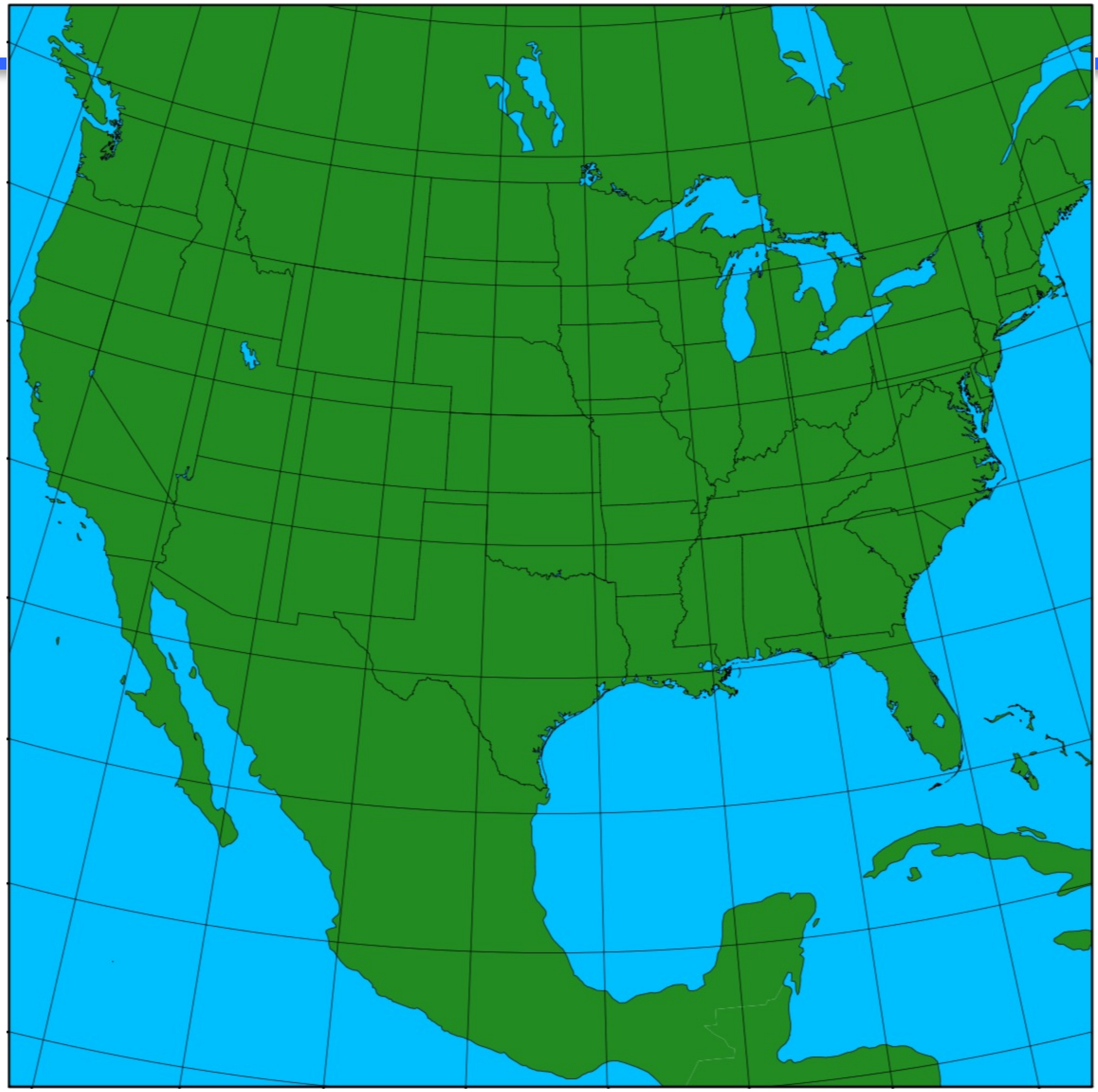


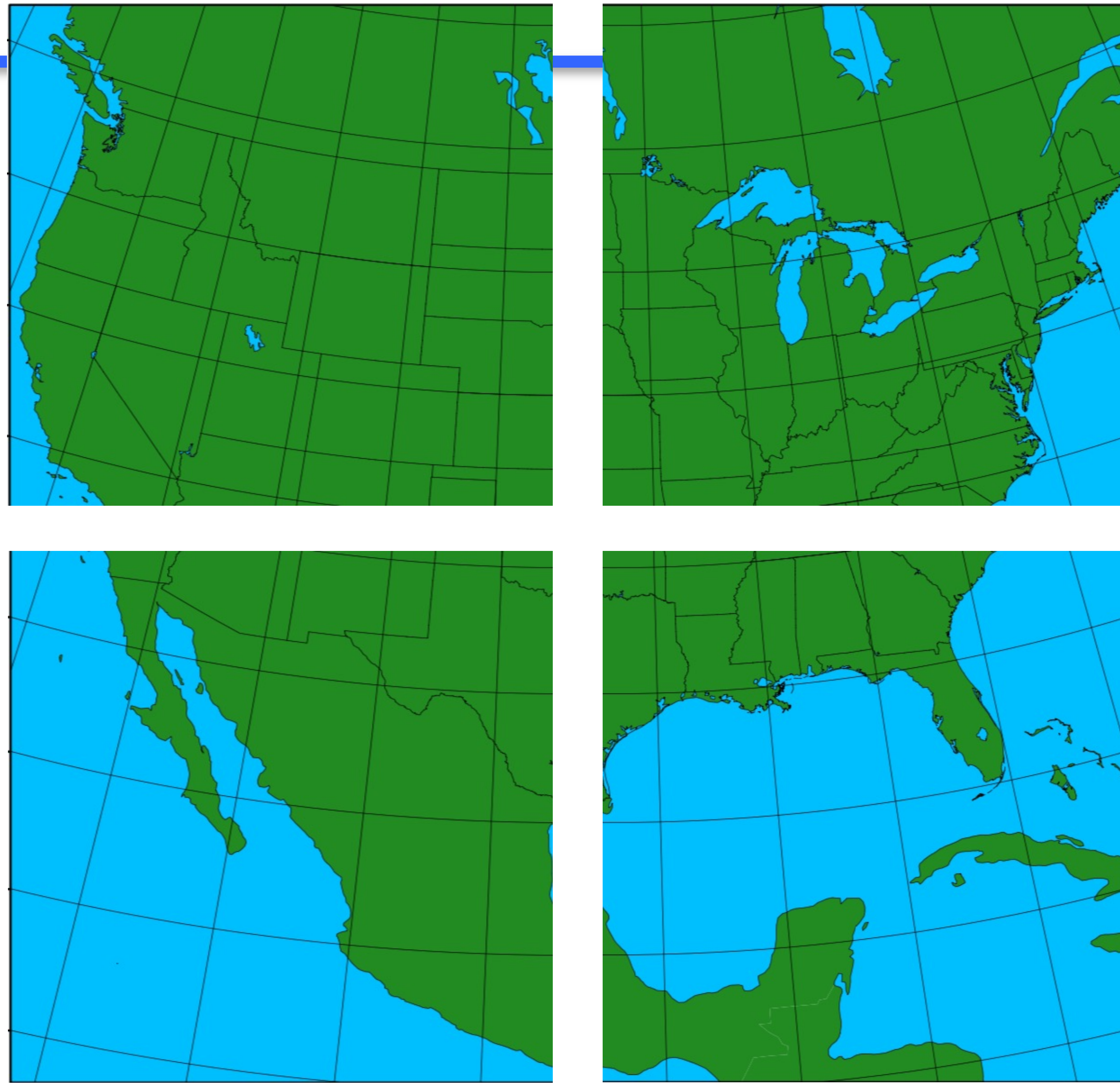
Quick Guide to WRF Parallelism

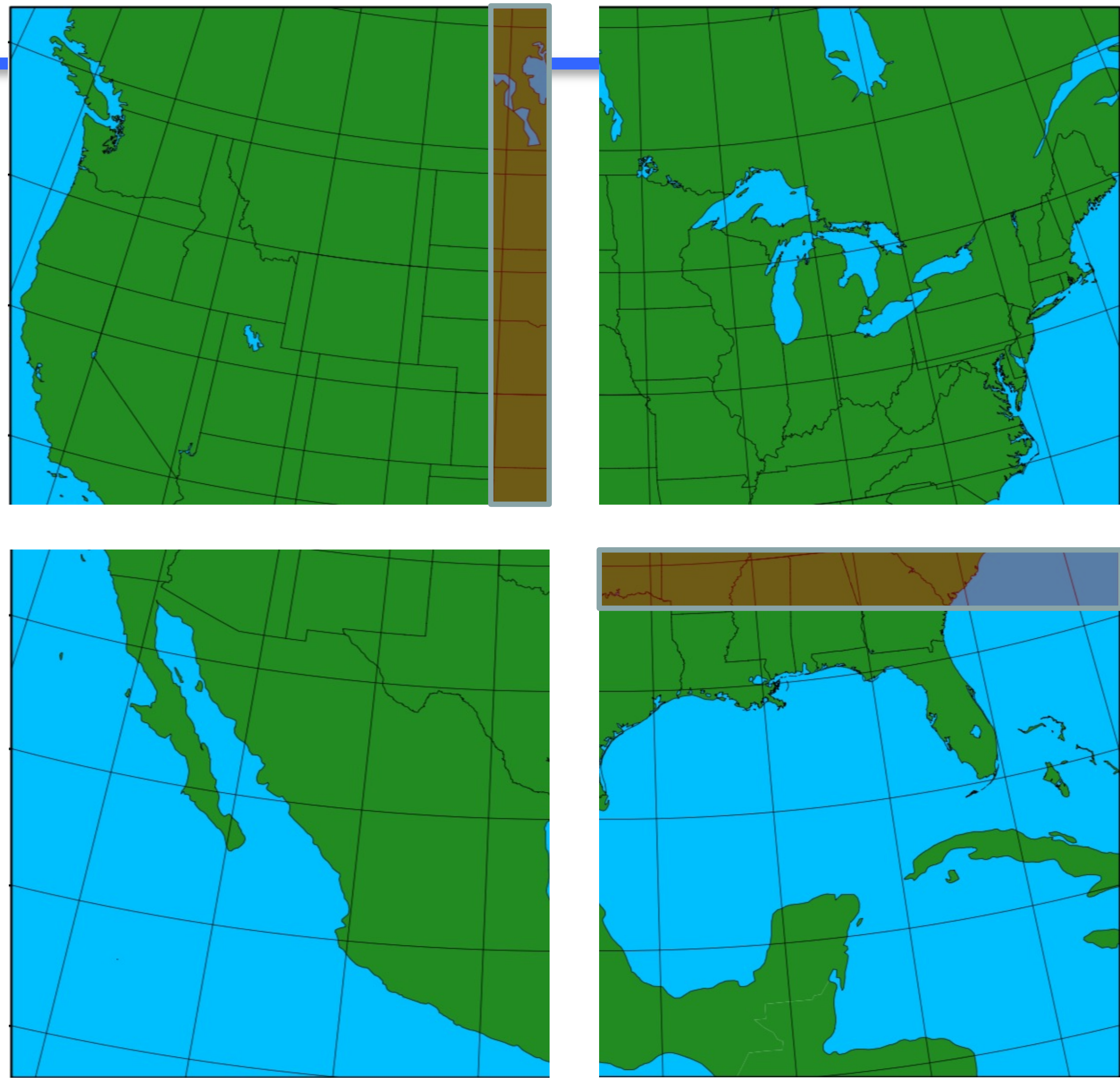
```
porig(i,k,j) = ( po(i,k,j) + po(i-1,k,j) ) * 0.5
```

- Instead of communicating with a distributed memory processor for each computation, a surrounding group of cells along the boundary holds read-only information
- This halo region is kept updated periodically from the neighboring distributed memory processor



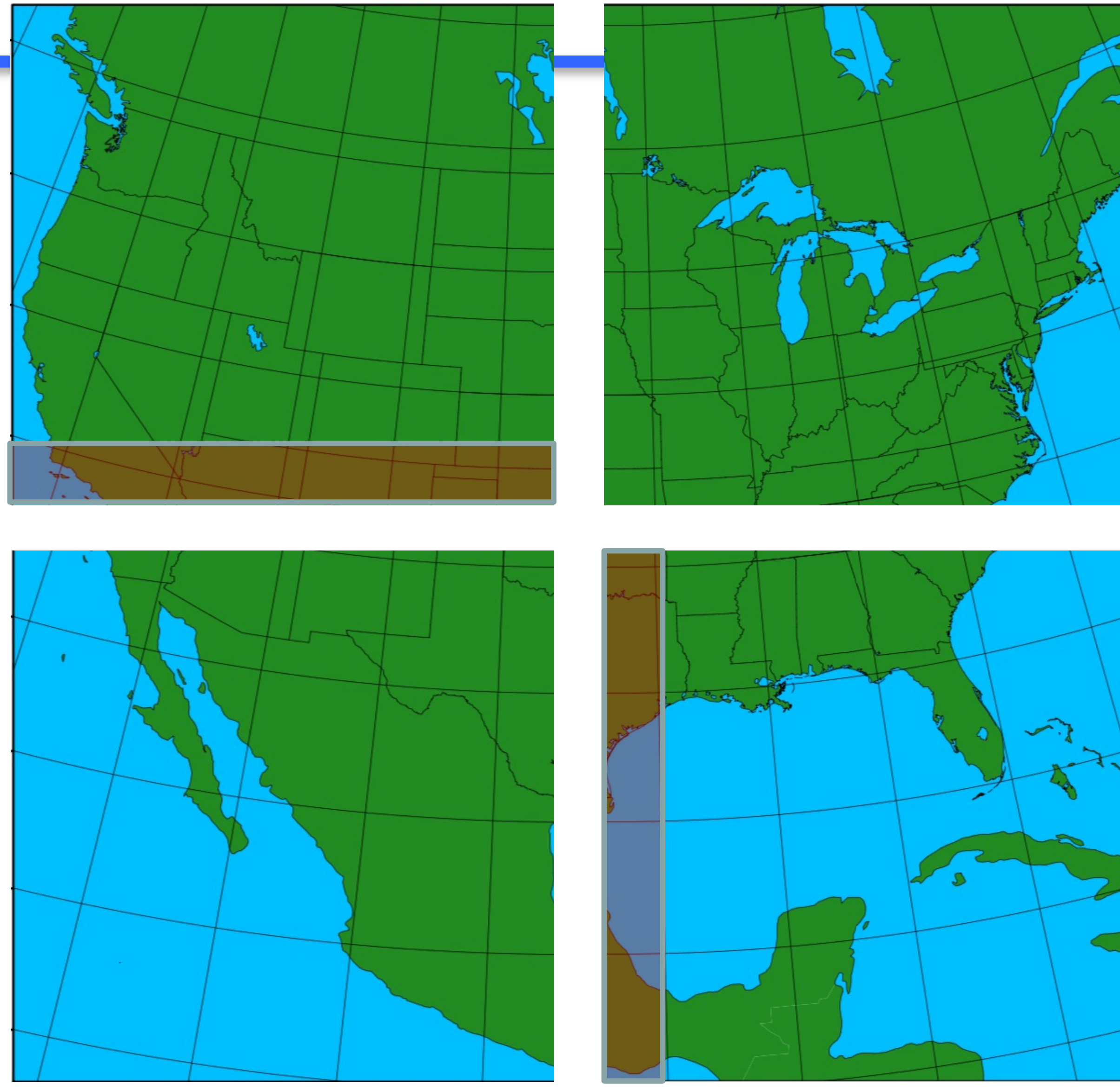


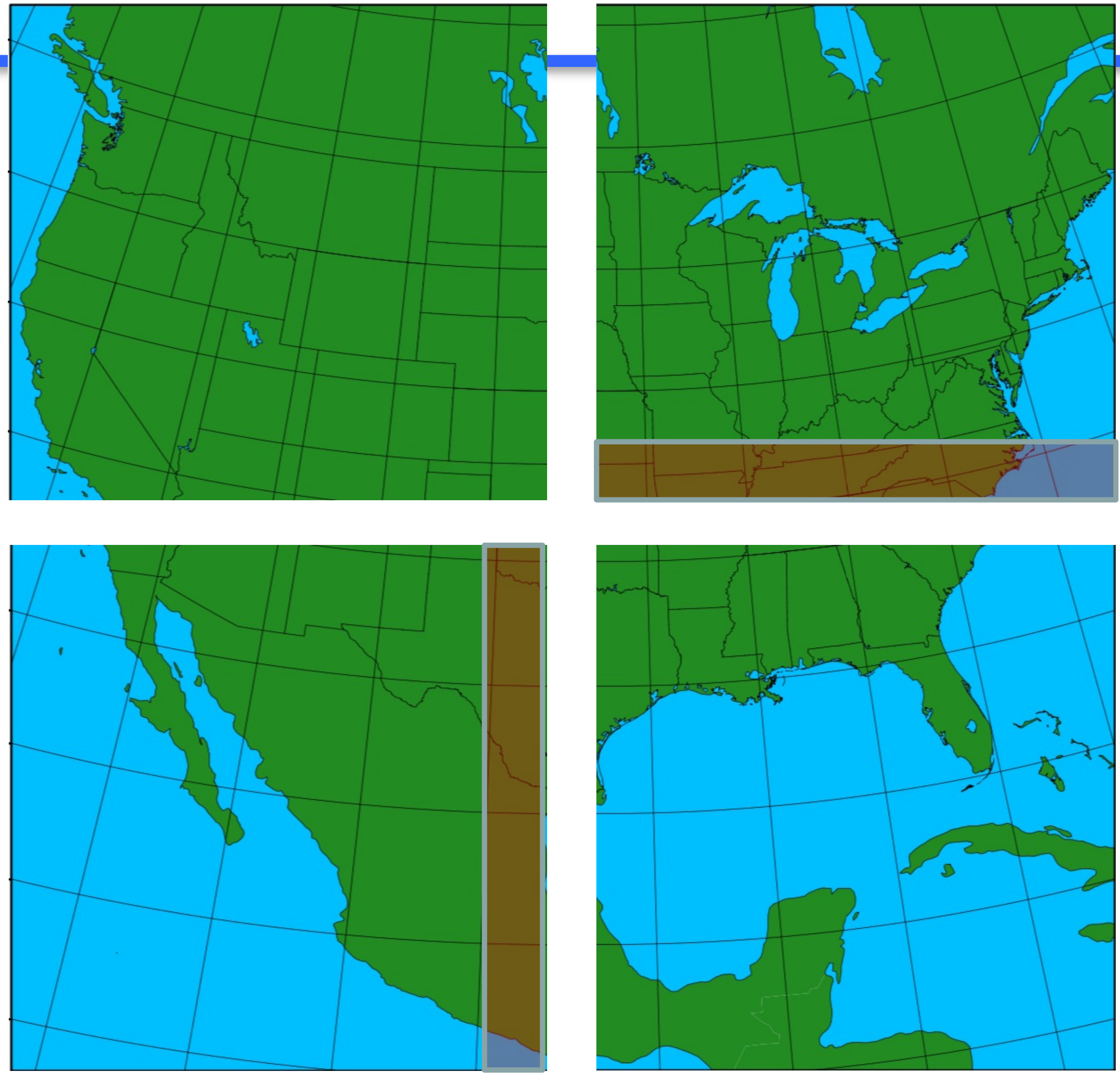




Halos to the left and below

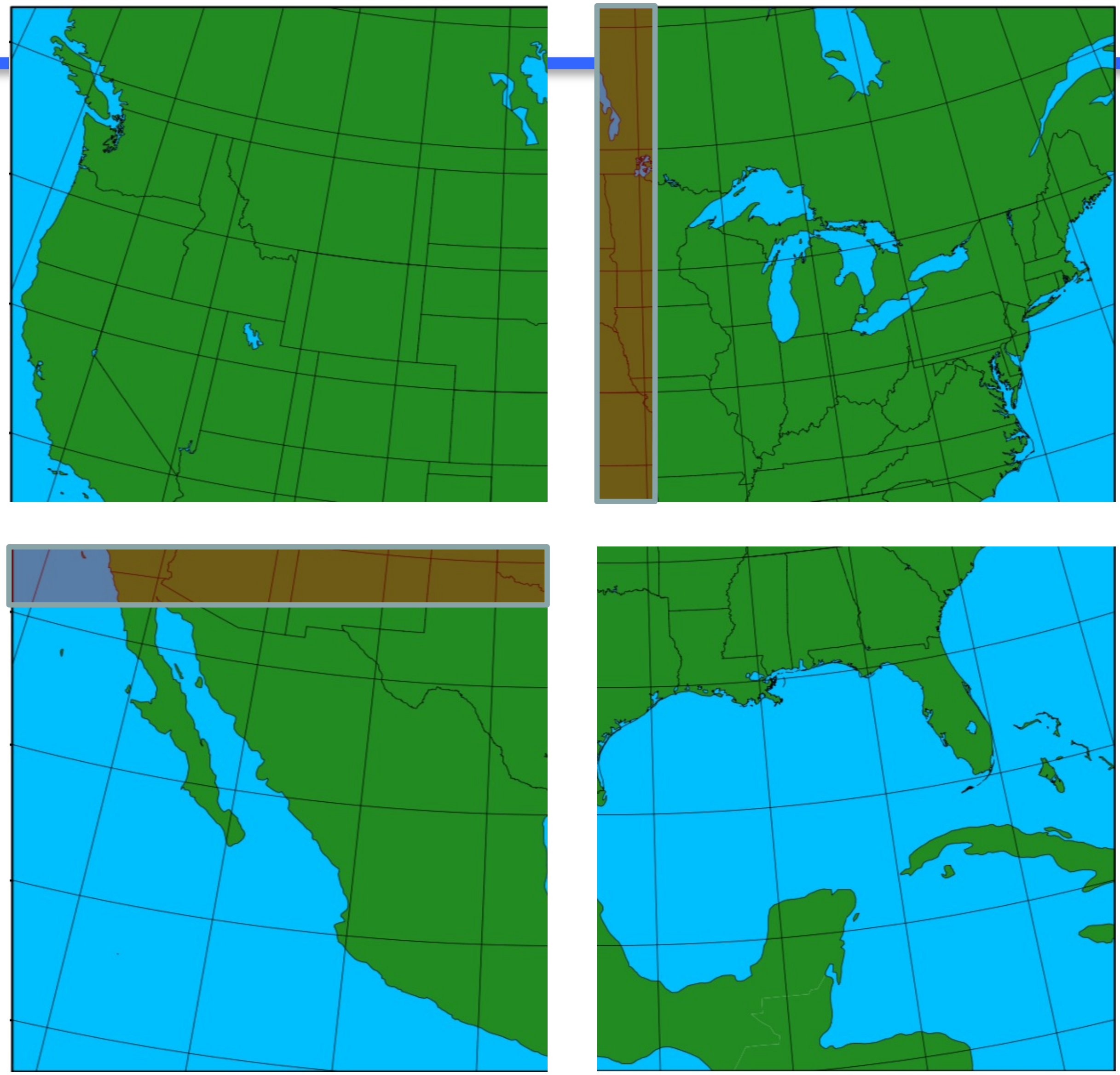
Halos to the right and above

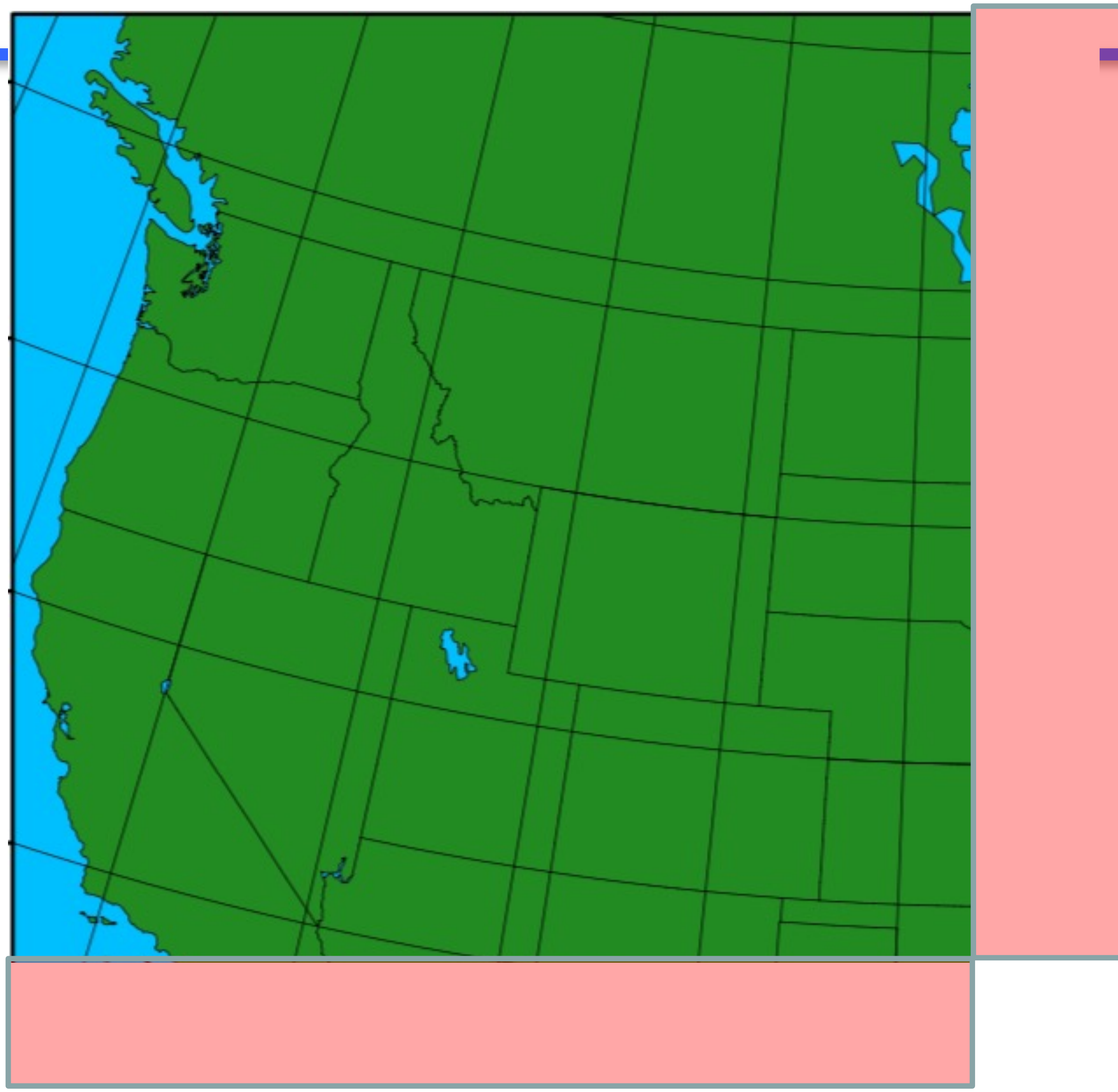


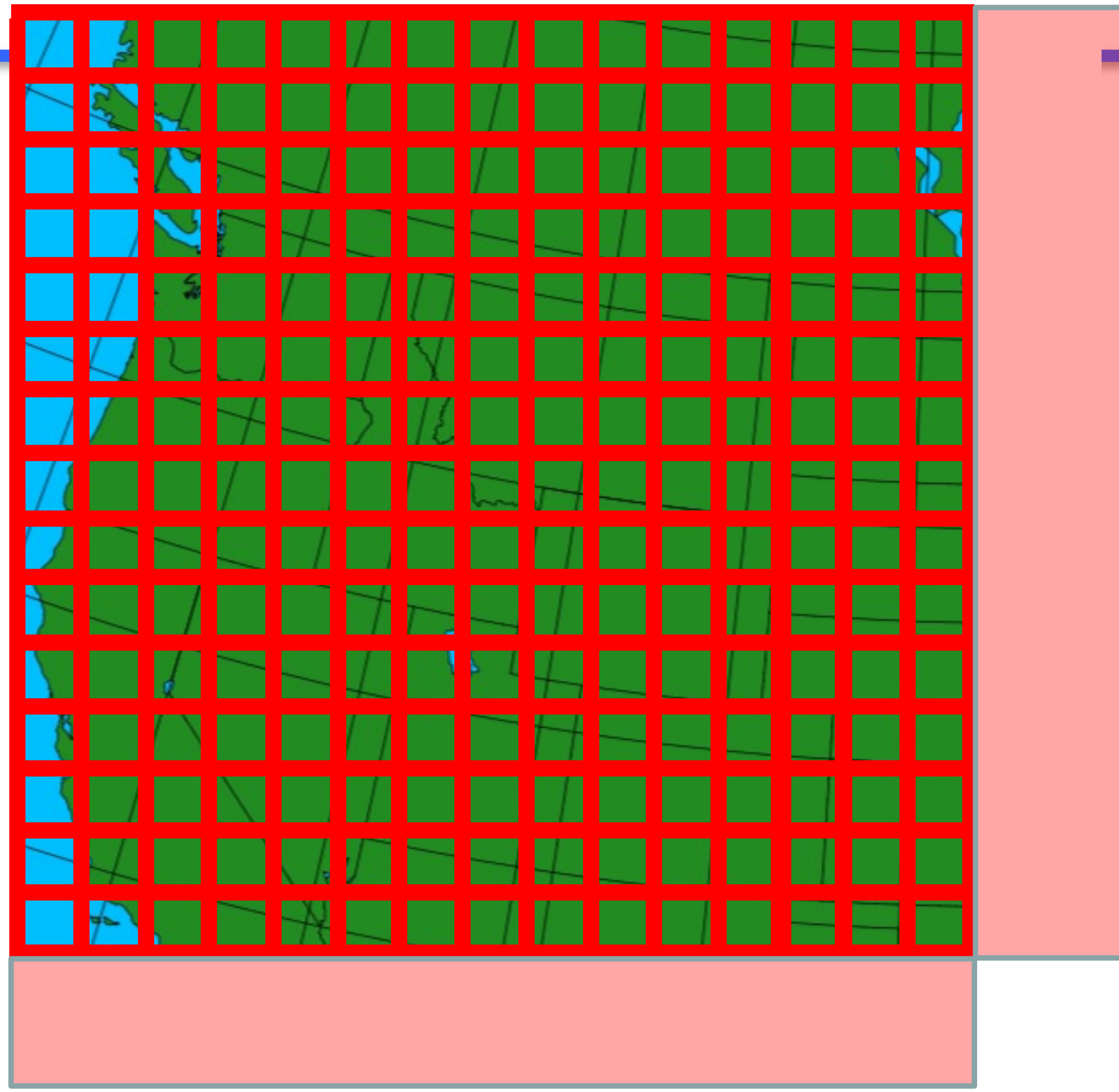


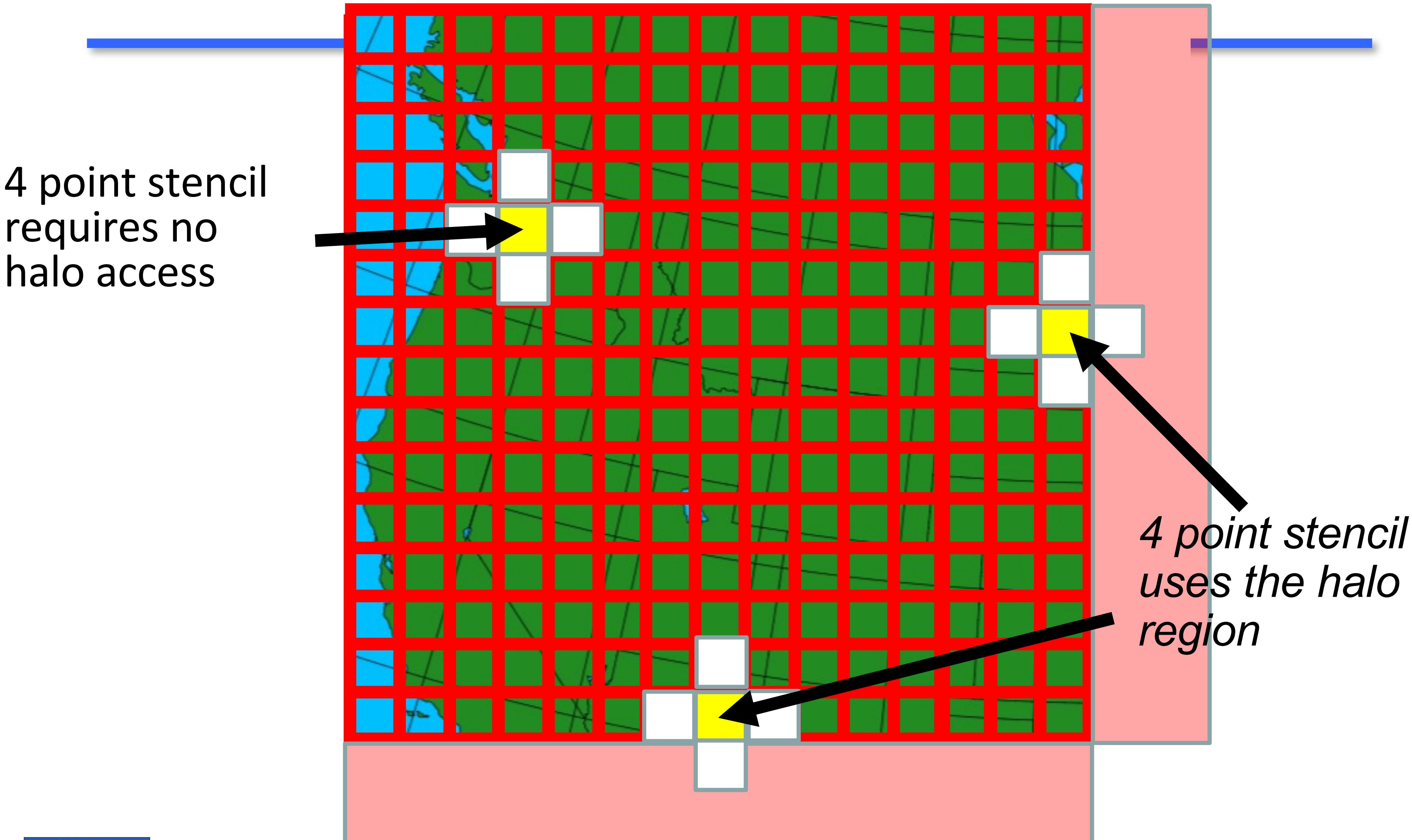
Halos to the left and above

Halos to the right and below








4 point stencil requires no halo access
4 point stencil uses the halo region

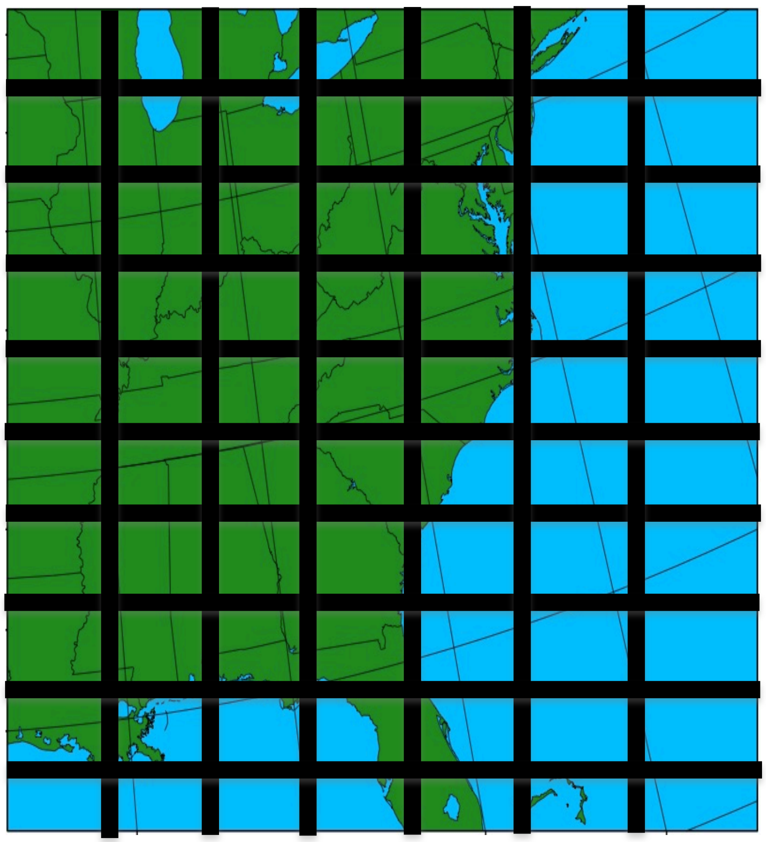
Quick Guide to WRF Parallelism

- How are the processor decompositions chosen

- By **default**, the decomposition of MPI tasks is computed as the two closest multiplicative factors
  - For example: 32 MPI tasks = 4x8 decomposition, NOT 2x16
  - For example: 144 MPI tasks = 12x12 decomposition, NOT 4x36
- The larger of the two factors decomposes the j-direction
- What to avoid: primes or large prime factors

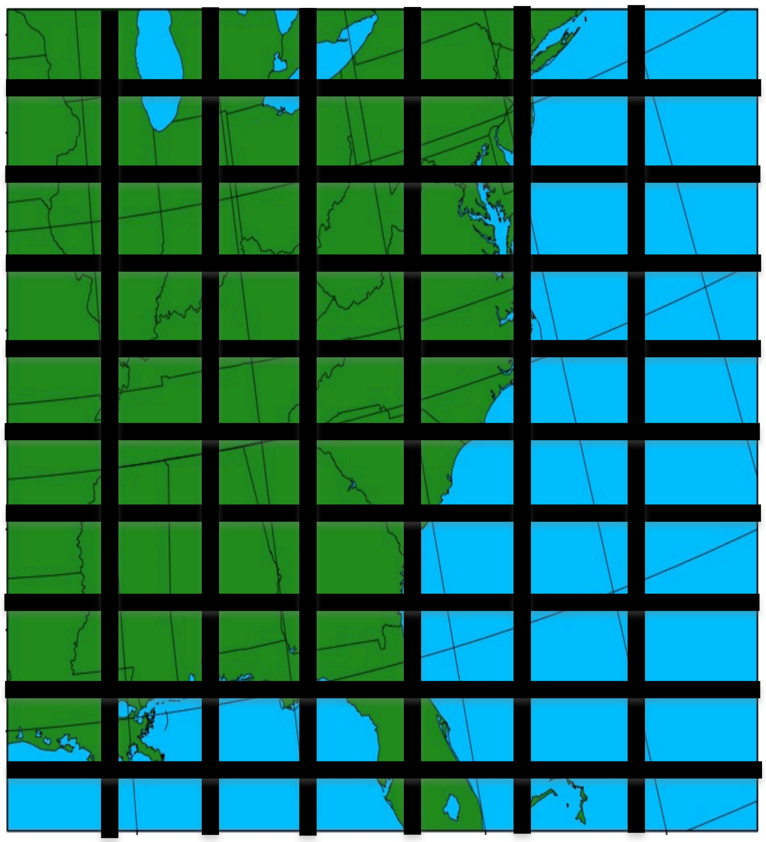
Quick Guide to WRF Parallelism

- 70 tasks
- 10 (j) x 7 (i)



Quick Guide to WRF Parallelism

- 70 tasks
- 10 (j) x 7 (i)



- 71 tasks
- 71 (j) x 1 (i)

Quick Guide to WRF Parallelism

- The maximum number of processors is based on the underlying stencil communications inside of the WRF model
- The model gracefully halts if you try to make a resultant distributed memory patch with < 10 grid cells on either side
- For the 1500x1500 benchmark case, we could have 150 units of patches that are 10 grid cells across (in the i- and j- directions)
- Therefore a maximum of 150x150 = 22,500 MPI processor cores

Overview of the ISC21 SCC WRF task

- Wiki page has EVERYTHING!

Basically

- Run WRF (we only care about the 3-domain case)
- Report the timings
- Verify the model results are OK
- Look at impact of compiler and parallelism options
- Run profiler on code

Optimization: What to Tune (and not!)

What is allowed to be modified:

- Compiler, build options
- Mix of OpenMP and MPI
- Flavors or versions of external libraries

What is NOT allowed to be modified:

- Source code
- Most of the run-time configuration file

WRF ISC21 Wiki pages

Single-domain practice case

https://hpcadvisorycouncil.atlassian.net/wiki/spaces/HPCWORKS/pages/1827438600/WRF+with+Single+Domain+-+Practice+case+for+ISC21+SCC

3-domain competition case

https://hpcadvisorycouncil.atlassian.net/wiki/spaces/HPCWORKS/pages/1827438607/WRF+-+3+Domain+Problem+for+ISC21+SCC

Questions?