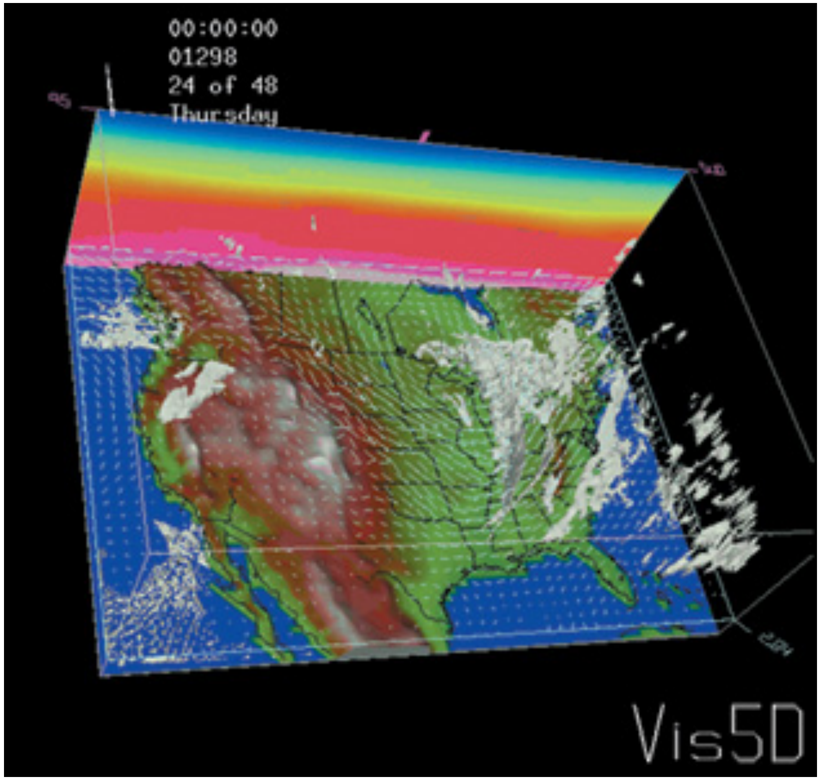
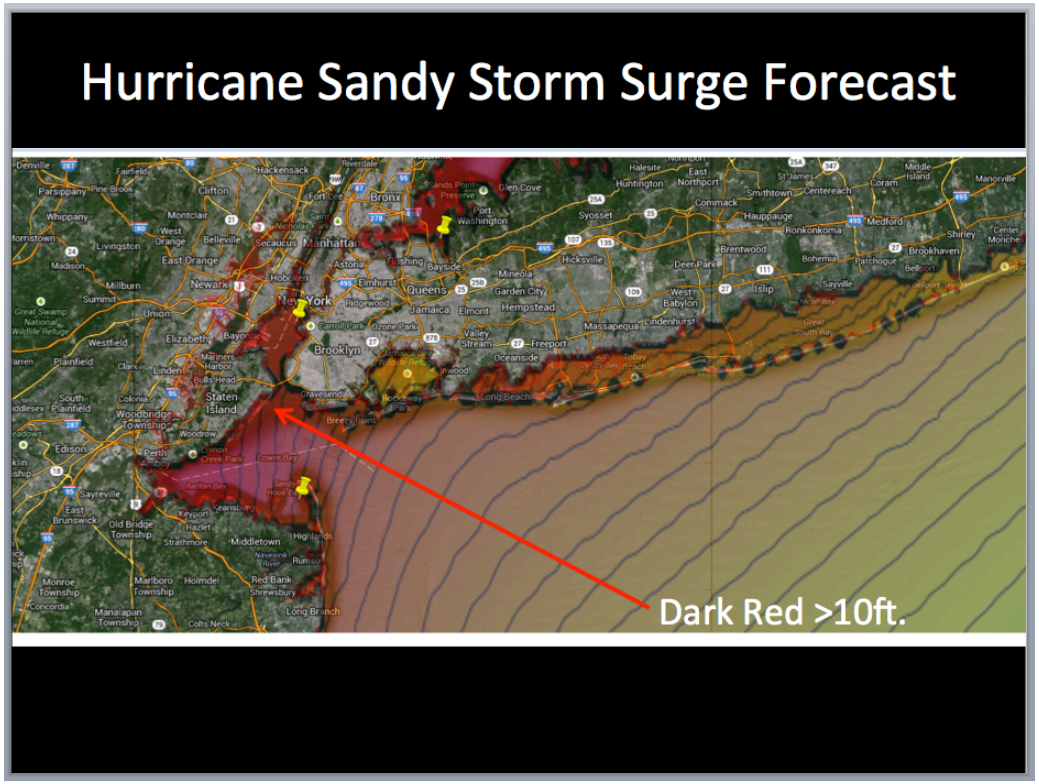
Running WRF on Yellowstone

Dave Gill
MMM/NESL



Courtesy John Michalakes, 2001
250 processors



Hurricane Sandy Storm Surge Forecast

Dark Red >10ft.

Courtesy Mel Shapiro, Alan Norton, 2013
500k cores

Gallia, and Running WRF on Yellowstone, Est Omnis Divisa in Partes Tres

- What does one *feed* **The WRF**

- Output in Itty-Bitty Pieces
- Szip or Extended-Rice Algorithm
- Whacking that which be Deem-ed Unnecessary
- How to shut down your division in one easy step: core counts
- bsub is my BFF
- Compiler tolerance via diversity training

- Society of the Secret Handshake

Gallia, and Running WRF on Yellowstone, Est Omnis Divisa in Partes Tres

- What does one *feed* **The WRF**

- Output in Itty-Bitty Pieces
- Szip or Extended-Rice Algorithm
- Whacking that which be Deem-ed Unnecessary
- How to shut down your division in one easy step: core counts
- bsub is my BFF
- Compiler tolerance via diversity training

- Society of the Secret Handshake

Too Much Data

Gallia, and Running WRF on Yellowstone, Est Omnis Divisa in Partes Tres

- What does one *feed* **The WRF**

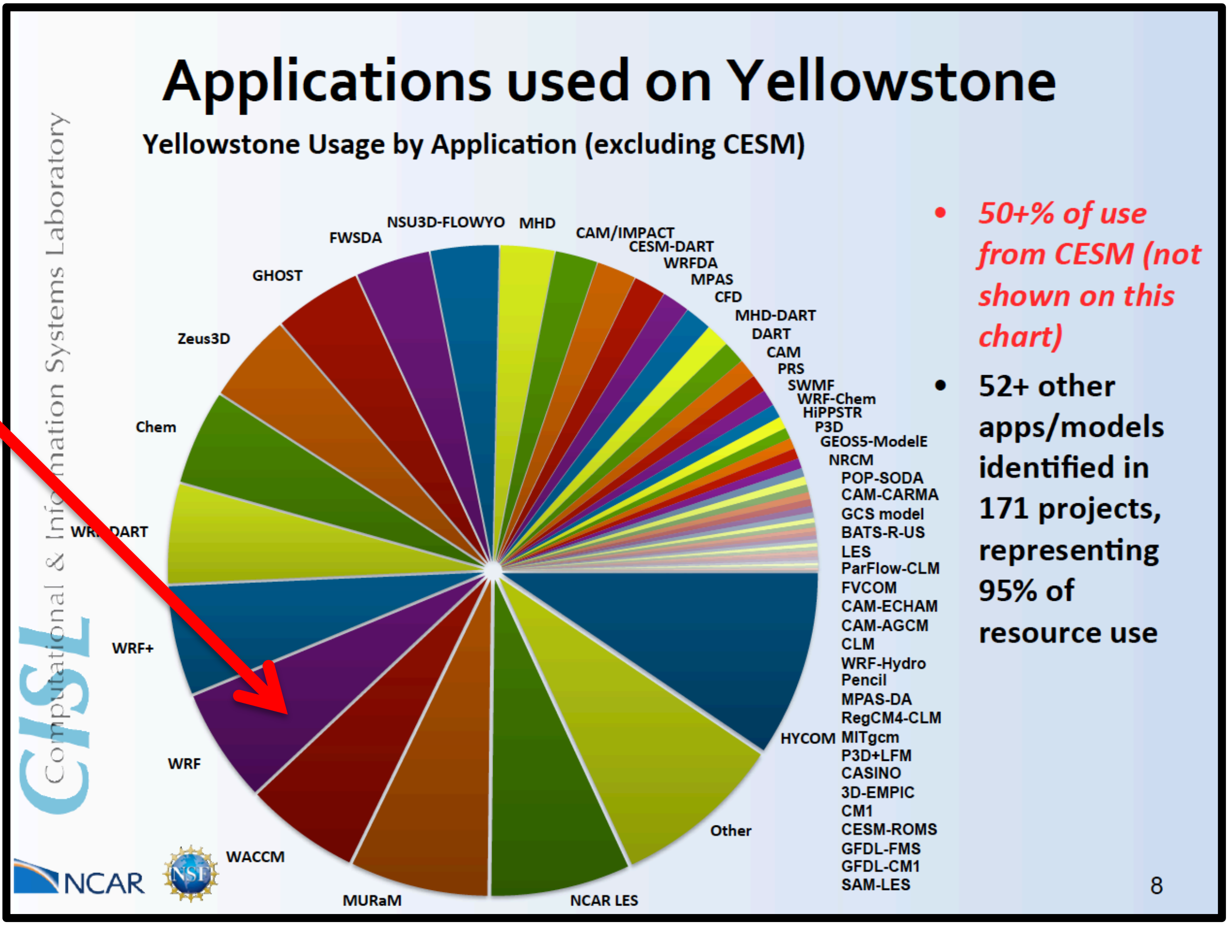
- Output in Itty-Bitty Pieces
- Szip or Extended-Rice Algorithm
- Whacking that which be Deem-ed Unnecessary
- How to shut down your division in one easy step: core counts
- bsub is my BFF
- Compiler tolerance via diversity training

- Society of the Secret Handshake

Yellowstone

Small is the new Big

WRF weighs in at approximately 3% of yellowstone use



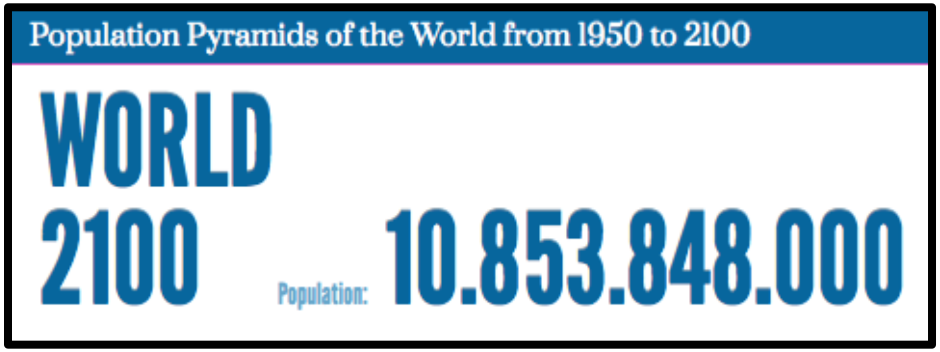
Applications used on Yellowstone

Yellowstone Usage by Application (excluding CESM)

- *50+% of use from CESM (not shown on this chart)*
- **52+ other apps/models identified in 171 projects, representing 95% of resource use**

Figure courtesy Shawn Strande, Yellowstone Workload Analysis
https://www2.cisl.ucar.edu/sites/default/files/YS%20workload%20analysis%20v4.2.pdf

WRF World Domination

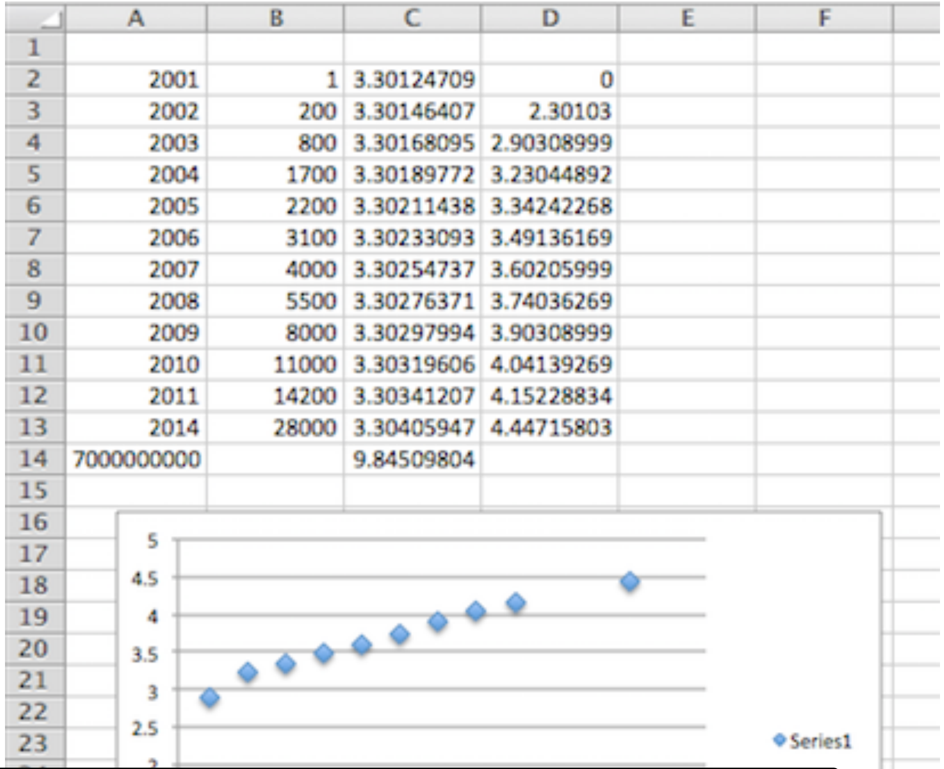


Population Pyramids of the World from 1950 to 2100

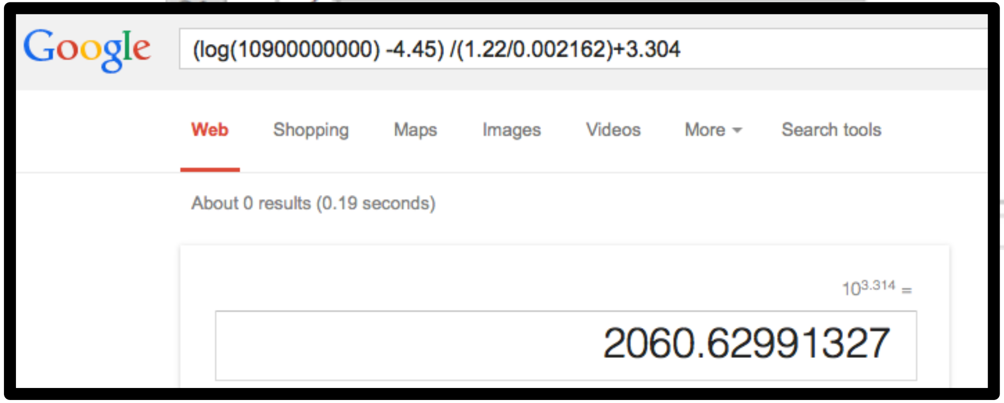
WORLD

2100 Population: 10.853.848.000

Assuming WRF current exponential growth continues (we all know long-term models don't lie), 100% of the world's population will run WRF by 2060 … ***including CGD***



	A	B	C	D	E	F
1						
2	2001	1	3.30124709	0		
3	2002	200	3.30146407	2.30103		
4	2003	800	3.30168095	2.90308999		
5	2004	1700	3.30189772	3.23044892		
6	2005	2200	3.30211438	3.34242268		
7	2006	3100	3.30233093	3.49136169		
8	2007	4000	3.30254737	3.60205999		
9	2008	5500	3.30276371	3.74036269		
10	2009	8000	3.30297994	3.90308999		
11	2010	11000	3.30319606	4.04139269		
12	2011	14200	3.30341207	4.15228834		
13	2014	28000	3.30405947	4.44715803		
14	7000000000		9.84509804			
15						



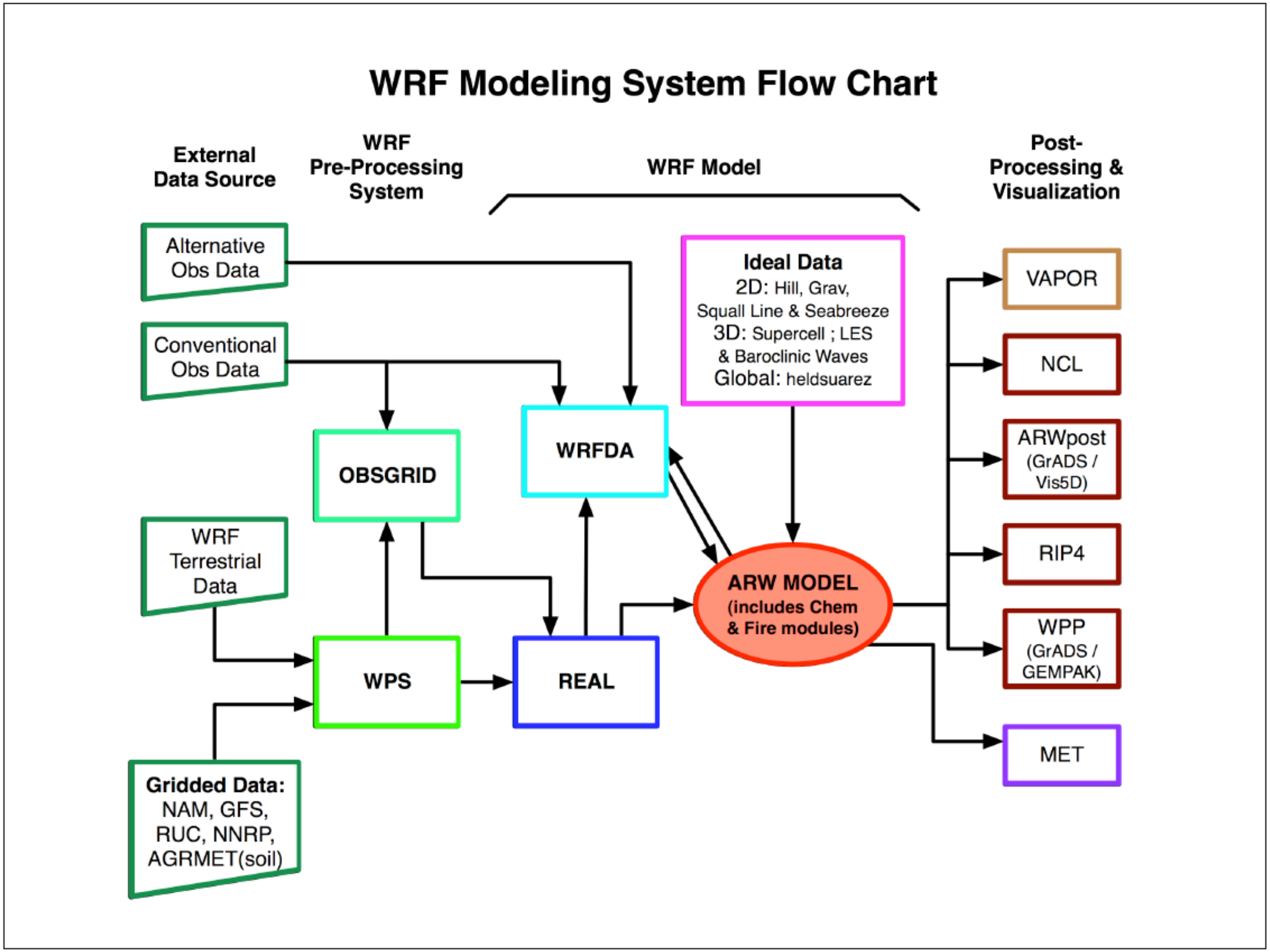
Google (log(10900000000) -4.45) /(1.22/0.002162)+3.304

About 0 results (0.19 seconds)

$10^{3.314}$ =

2060.62991327

Input to the WRF System



WRF Modeling System Flow Chart
External Data Source
WRF Pre-Processing System
WRF Model
Post-Processing & Visualization
Alternative Obs Data
Conventional Obs Data
WRF Terrestrial Data
Gridded Data: NAM, GFS, RUC, NNRP, AGRMET(soil)
OBSGRID
WPS
WRFDA
REAL
Ideal Data
2D: Hill, Grav, Squall Line & Seabreeze
3D: Supercell ; LES & Baroclinic Waves
Global: heldsuarez
ARW MODEL (includes Chem & Fire modules)
VAPOR
NCL
ARWpost (GrADS / Vis5D)
RIP4
WPP (GrADS / GEMPAK)
MET

Input to the WRF System

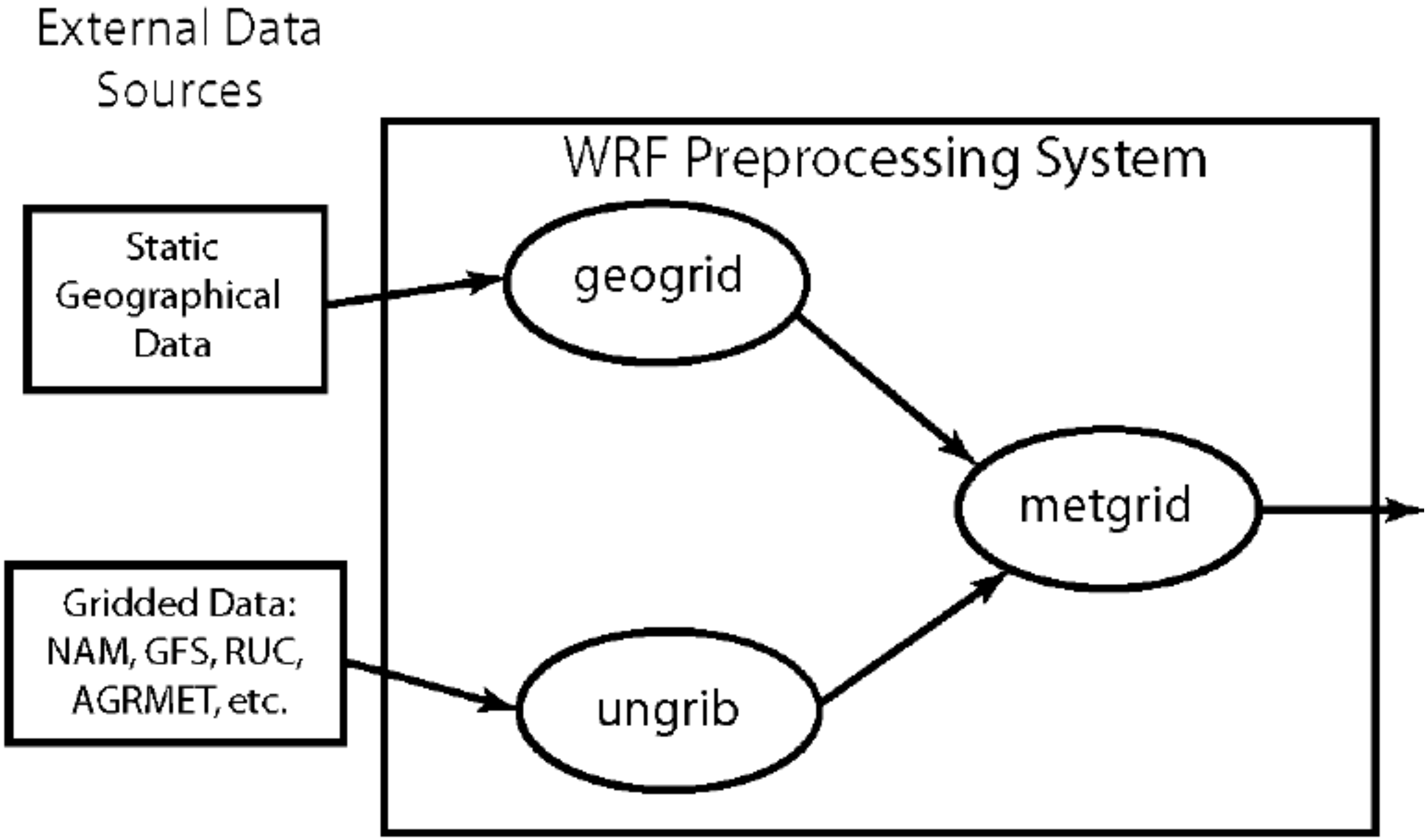


Figure courtesy Michael Duda

Input to the WRF System

- Static data:
  - Stored in /glade/u/home/wrfhelp

- Meteorological data – real time and historical:
  - http://www2.mmm.ucar.edu/wrf/users/FAQ_files/FAQ_wps_input_data.html
  - NCEP, JMA, ECMWF, UM, NOAA (*CAM, sort of*)
  - Analysis, re-analysis, forecast, global, regional, spectral, grid-point, isobaric, hybrid, projected
  - Grib Edition 1 or Grib Edition 2

Data, data every where, nor any drop to drink

Samuel Coleridge

Rime of the Ancient Modeler

- Output data more quickly
- Output data more smallly
- Output data more lessly

Stitching Model Output Together

- Yunheung Wang (CAPS) developed and Kevin Manning improved a scheme that joins "split data" back together

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

- Download **`joinwrf.tar.gz`**
- Edit the Makefile to choose the compiler and parallel preferences
- Edit the namelist file (namlist.join)

Stitching Model Output Together

- Usually the joiner program is called after a time period is available, i.e. multiple times per simulation.
- The joiner program runs within a few minutes, and requires only a small number of cores (1-4)
- A script is typically written to edit the namelist (namelist.join), and run the joiner code
- Execution: `./joinwrf < namelist.join`

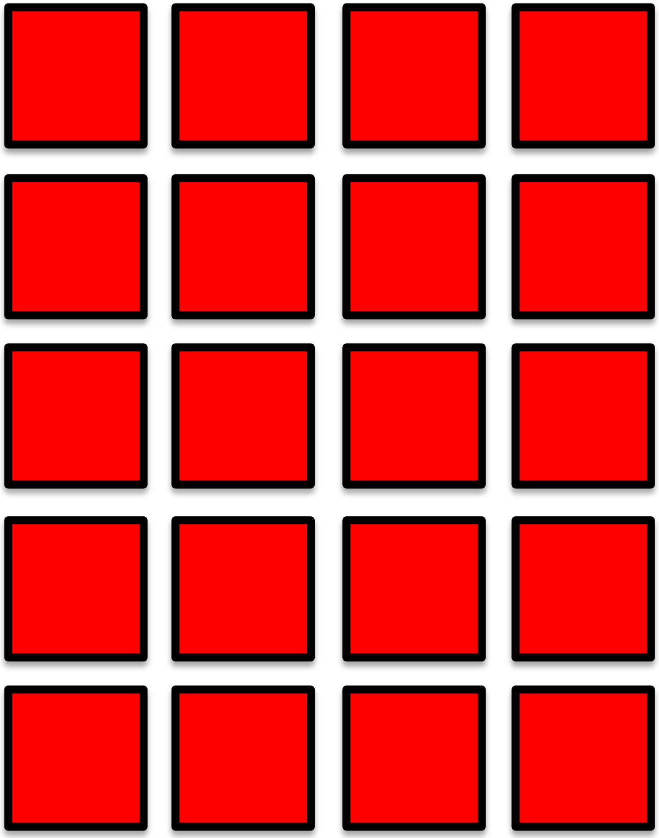
Stitching Model Output Together

- The WRF model is able to provide data split along processor boundaries using the io_form options.

```
&time_control
history_interval_s = 150,  60,   60,
io_form_history    = 102
/
```

Stitching Model Output Together

- Running on 20 cores could produce the following WRF model decomposition and output:



Stitching Model Output Together

- With large domains, model output can dominate the total wall clock time
- With the “102” option, when running on 800 cores, there are 800 files
- Files get constructed with names such as

```
wrfout_d01_2010-06-23_15:00:00_0000
wrfout_d01_2010-06-23_15:00:00_0001
…
wrfout_d01_2010-06-23_15:00:00_0799
```

Stitching Model Output Together

- The only purpose is timing performance
- Works well with multiple domains and when restarts overlap with model output times
- The joining program is DM parallel
- For a 2000x2000x100 WSM6 domain, 2 minutes per time period with 8 cores manufactured the single file
- The joiner program only works with model output; NOT with IC or restart files

Stitching Model Output Together

- Single file input:

```
Timing for processing wrfinput file
(stream 0) for domain        1:
320.15085 elapsed seconds
```

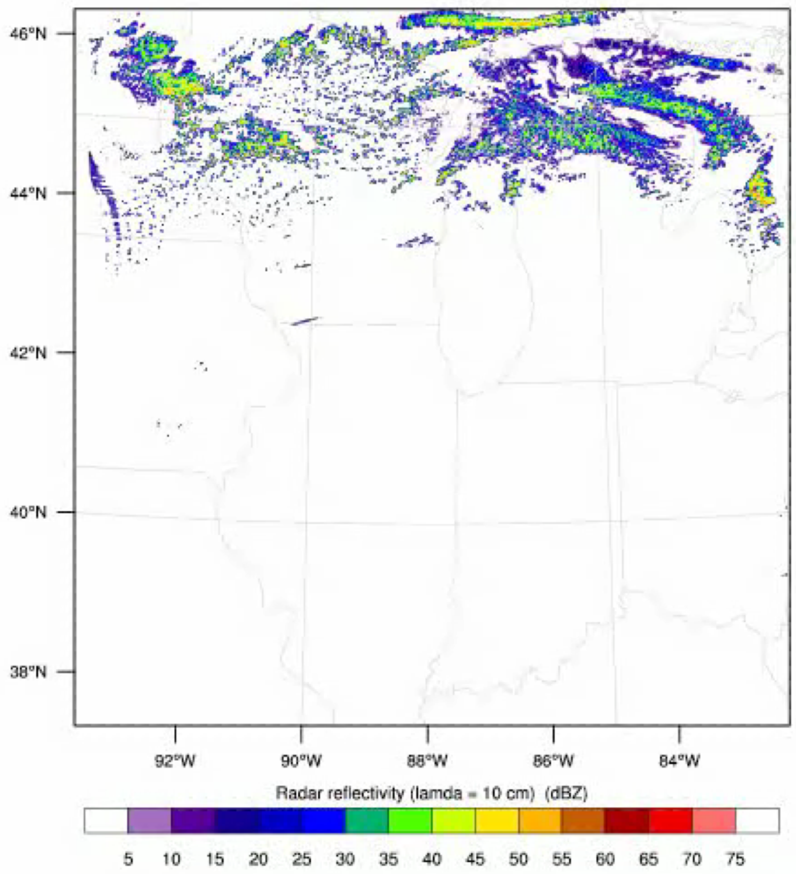
- Multiple file output:

```
Timing for Writing wrfout_d01_2010-06-
23_12:00:00 for domain        1:
0.90883 elapsed seconds
```

WRF: OHARE

Init: 2010-06-23_12:00:00
Valid: 2010-06-23_18:00:00

Radar reflectivity (lamda = 10 cm) (dBZ)



OUTPUT FROM WRF V3.5 MODEL
WE = 2001 ; SN = 2001 ; Levels = 105 ; Dis = 0.5km ; Phys Opt = 6 ; PBL Opt = 1 ; Cu Opt = 1

NETCDF4 Compression

- Huang Wei and Jianyu Liu have put in a simple way to get impressive NETCDF4 compression with WRF model output

- If the user has NETCDF4 libraries that have HDF5 compression included, then a single “env” variable is all that is required

NETCDF4 Compression

- Prior to running `./configure` …

```
setenv NETCDF4 1
export NETCDF4=1
```

NETCDF4 Compression

- This is fully supported in WRF 3.5 and beyond
- File sizes tend to be about half of the original size
- The compression works well with fields which contain similar values (such as near-zero quantities for many of the hydrometeor fields)
- YS NETCDF tools support this compression: ncview, ncl, nco

(De)Selecting Model Output Fields

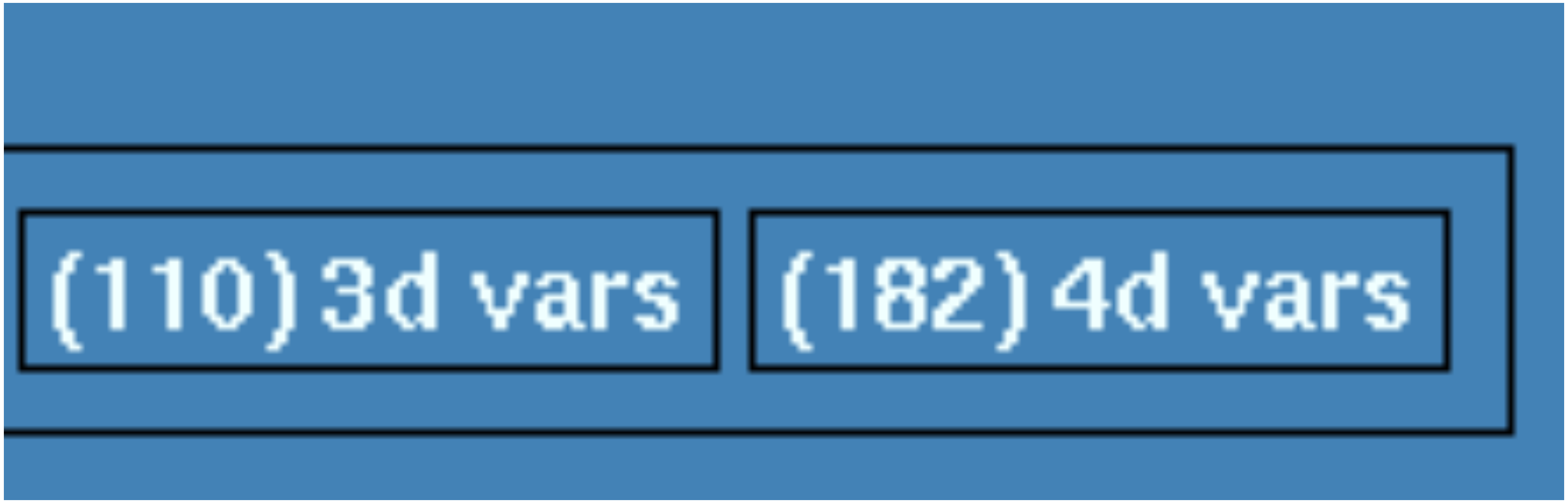
- Several years ago John Michalakes provided a simple run-time option to add and remove fields from WRF streams

```
&time_control
iofields_filename = "myoutfields.txt”
/
```

```
-:h:0:W,PB,P
```

(De)Selecting Model Output Fields

- Particularly helpful when ncview shows:



(De)Selecting Model Output Fields

- Removing half of the unwanted or never used 3d arrays cuts your file sizes in half
- Default values for “history” that are in the Registry do not obligate users

There once was a machine next to Wal-Mart
Which before the recabling, would fall apart
    Day after day
    I always seem to say
Have this job finish ‘ere this life I depart

Samuel Coleridge
Rime of the Ancient Modeler

- How many
- How to use
- How to choose

Just How Many Cores Do I Need

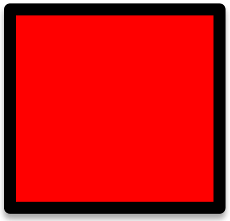
- The WRF model timing is sensitive to the selection of model options and various domain configurations.
- For a fixed grid size (km) and number of grid cells, the choice of microphysics and radiation impact the model run-time.
- For non-chemistry runs and for non-bin MP runs, the WRF model is not a memory hog.
- Horizontal domain decomposition is used.

Just How Many Cores Do I Need

- Assume three different domains:

A

1000x1000

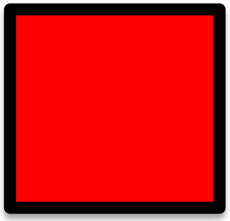


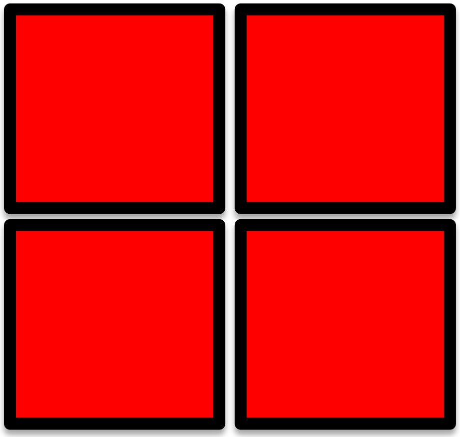
Just How Many Cores Do I Need

- Assume three different domains:

A B

1000x1000 2000x2000



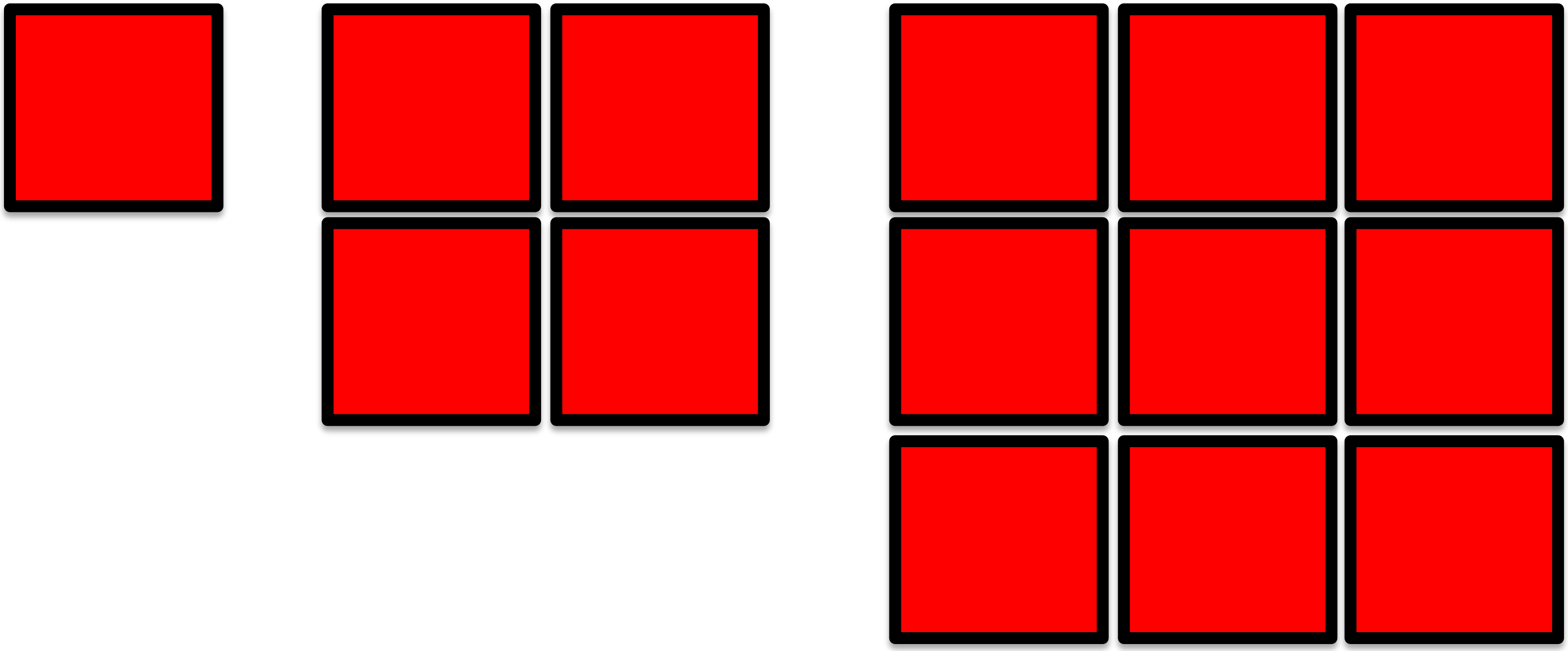


Just How Many Cores Do I Need

- Assume three different domains:

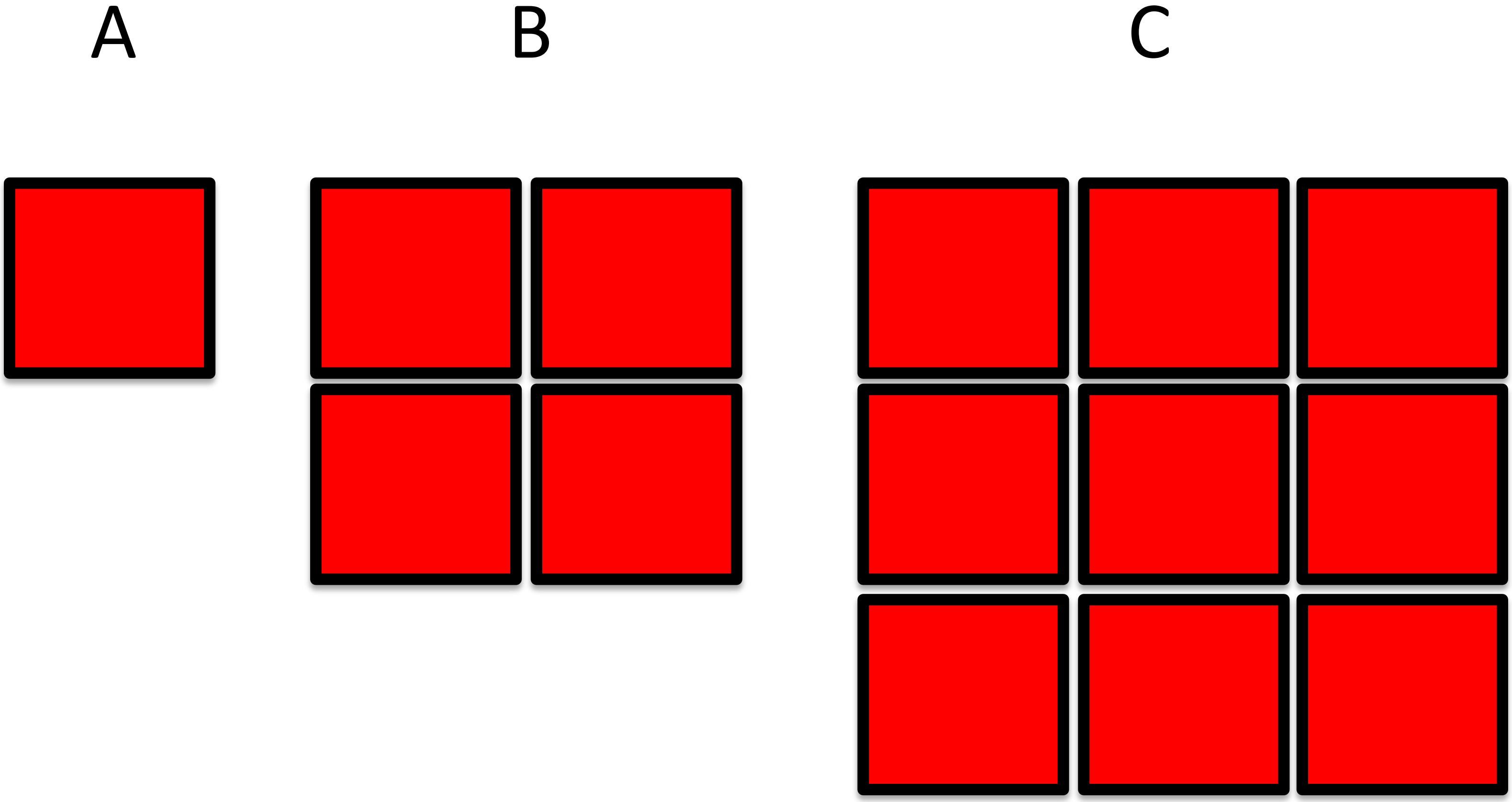
A B C

1000x1000 2000x2000 3000x3000



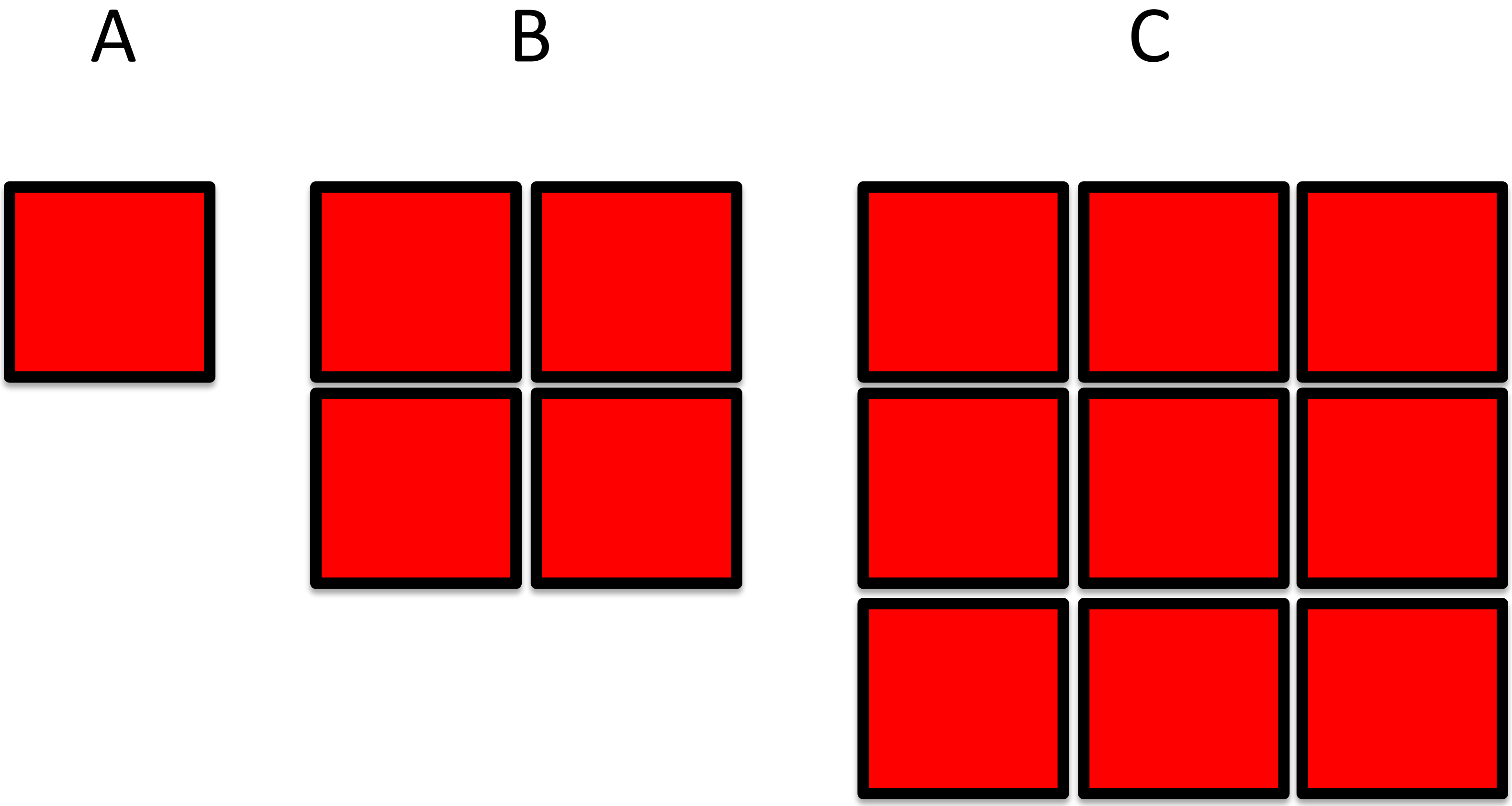
Just How Many Cores Do I Need

- If domain A fits on a *n* cores, then domains B and C fit on 4*n* and 9*n* cores, respectively



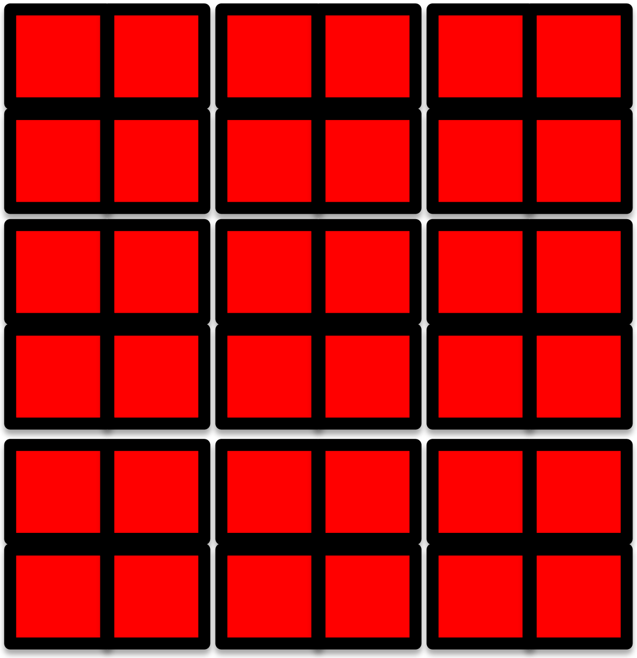
Just How Many Cores Do I Need

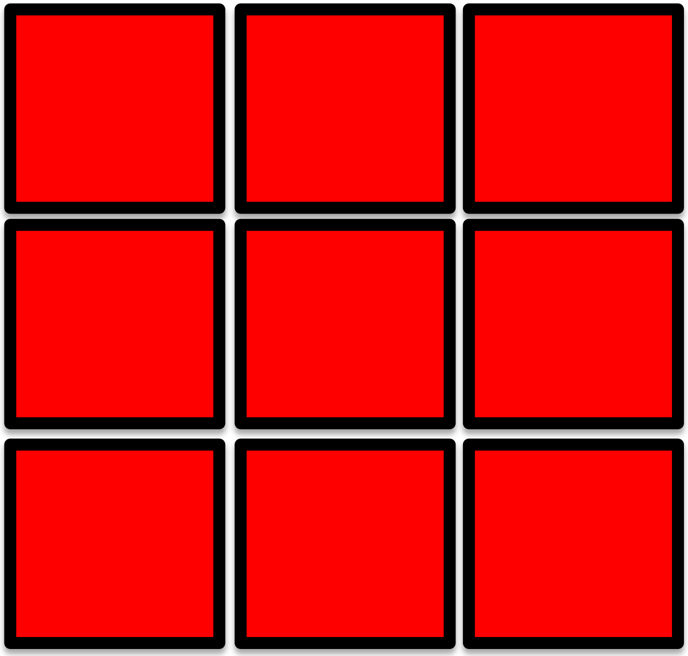
- The amount of wall-clock time for domain B (using 4***n*** cores) ~= domain C (using 9***n*** cores)



Just How Many Cores Do I Need

- The larger the decomposed sub-domain, the more efficiently it uses the core (more work with similar-ish amounts of communication).





Just How Many Cores Do I Need

- Scaling a WRF job is straightforward.
- If a 3000x2000 domain is the desired domain size of the eventual WRF simulation …
  - 600 (600=30x20) 100x100 sub-domains could be manufactured
  - Short timings on a 200x200 domain (four sub-domains of 100x100) would yield similar performance characteristics

Just How Many Cores Do I Need

- WRF default decompositions take the two closest factors.
  - 144 total cores is a 12x12 core set up, not 72x2
- Be careful with core counts with accidentally large prime factors.
  - 128 cores using 6 I/O processors gives 122 total computational cores, decomposing as 61x2
- Sub-domains should be larger than 10 cells on a side, and ***larger still for performance***.

Suggestions for bsub’ing

- WRF does well with MPI. For ease of use, stay away from OpenMP.

- Moderate performance gains, but test it out.
  - #BSUB -R "span[ptile=**32**]"

- Thousands of test WRF runs are made weekly, with no troubles:
  - unsetenv MP_PE_AFFINITY

Suggestions for module

- WRF is tested with GNU, PGI, and Intel weekly (most large jobs use Intel).
- Over the past couple of years, we have tested the GNU 4.[789].x, PGI 1[234].x, and Intel 1[2345].x compiler versions with WRF for the regression suite.

- There is no appreciable part of WRF that is setup for accelerators.

Helpful Stuff

The WRF group has scads of helpful resources

- Send email to wrfhelp@ucar.edu
- www.mmm.ucar.edu/wrf/users
- Online Tutorials
  - http://www2.mmm.ucar.edu/wrf/OnLineTutorial/index.htm
- Presentations from previous tutorials
  - http://www2.mmm.ucar.edu/wrf/users/tutorial/tutorial_presentation_summer_2014.htm

Helpful Stuff

The WRF group has scads of helpful resources

- User’s Guide
  - http://www2.mmm.ucar.edu/wrf/users/docs/user_guide_V3/contents.html
- Technical Description
  - http://www2.mmm.ucar.edu/wrf/users/docs/arw_v3.pdf
- FAQs
  - http://www2.mmm.ucar.edu/wrf/users/FAQ.html

Helpful Stuff

The WRF group has scads of helpful resources

- As a registered user you get notifications (and lots of job offers!).
- You have access to a large community of users who may be working in a similar research area.
- A week-long annual workshop in June with approximately 250 attendees.

Caesar’s Big Three Issues with WRF

- What goes in
- Helpful hints on WRF in general, and on YS
  - Make data as small as possible
  - Choose cores wisely
  - Simplify your life and your LSF scripts
- Who ya gonna call
  - Take advantage of available resources

Caesar's Big Three Issues with WRF

- What goes in

Food

- Helpful hints on WRF in general, and on YS
  - Make data as small as possible
  - Choose cores wisely
  - Simplify your life and your LSF scripts

- Who ya gonna call
  - Take advantage of available resources

Caesar’s Big Three Issues with WRF

- What goes in

Food

- Helpful hints on WRF in general, and on YS
  - Make data as small as possible
  - Choose cores wisely
  - Simplify your life and your LSF scripts

Digestion

- Who ya gonna call
  - Take advantage of available resources