

# WRF Quilting and Decomposition Notes

Eric Kemp

2 March 2015

# Quilting Preliminaries

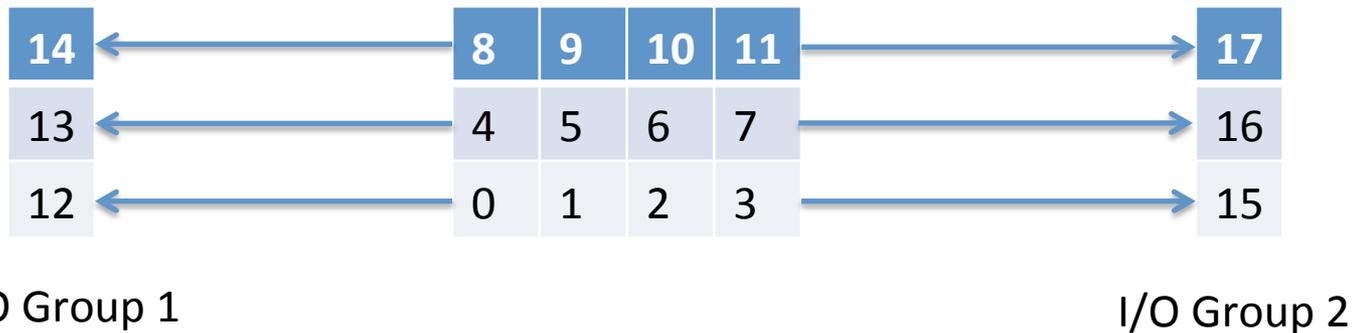
- Two types of MPI tasks: *compute* (client) and *I/O* (server)
- Compute tasks:
  - Total number =  $nproc\_x * nproc\_y$  (number of processors along x and y axes for decomposition)
  - First number is zero
- I/O tasks:
  - Total number =  $nio\_groups * nio\_tasks\_per\_group$
  - $Nio\_tasks\_per\_group$  cannot exceed  $nproc\_y$
  - First I/O task number immediately follows last compute task number
- Code will attempt to match each I/O task with compute tasks in east-west rows
  - Ideally  $nproc\_y$  should be exact multiple of  $nio\_tasks\_per\_group$

Note: Only output is currently supported by I/O quilting, not input.

# Sample Task Layouts



If nio\_groups=2 and nio\_groups\_per\_task=3, then:



- Each compute node in a row (e.g., tasks 0-3) will send data to a I/O server (e.g., 12 or 15) at output time.
- One I/O group is selected on-the-fly at output time to handle output to a particular file.
- Within an I/O group, the servers will forward data to designated “root” server which will perform the actual write.

Note: Above is based on running the code. The example in the source code comments does not agree!

# Quilting Performance

- An I/O group can only handle one file write at a time
  - If too few I/O groups available, WRF will stall

	No Quilting	1 Groups	2 Groups	3 Groups
wrfout	20.08 s	1.18 s	1.05 s	1.20 s
wrfdiag	0.64 s	63.88 s	0.50 s	0.62 s
wrf2dout	1.48 s	1.49 s	19.92 s	0.68 s
wrfpress	0.56 s	6.67 s	0.40 s	0.74 s

Sample output times for A24 grid with nproc\_x=53, nproc\_y=12 compute tasks, varying I/O groups with nio\_tasks\_per\_group=12

Note: Reported output times are what compute tasks “see” when communicating and waiting – does not necessarily reflect time spent writing by a I/O group

# Quilting Recommendations & Comments

- Set `nio_groups = 5` (for `wrfout`, `wrf2dout`, `wrfpress`, `wrfdiagnostics`, `wrfrst` files)
- Try setting `nio_tasks_per_groups` so it can exactly divide `nproc_y`
  - Otherwise some I/O server(s) will have more compute tasks to handle than others
  - Don't know how performance scales with tasks per group
- Race condition: Multiple groups can't write to same file at time
- Simple test runs still showed some strange stalls during computational steps (not output times)

# Decomposition Notes

- Found several third-party discussions of running WRF on HPC systems (Cray, STFC, Lenovo, Barcelona Supercomputing Center)
- All recommend setting  $nproc\_x < nproc\_y$ , but “sweet spot” is problem and hardware dependent
- Arguments (from Cray):
  - Give inner loops larger lengths for better SSE vector and register reuse
  - Shorten outer loop lengths for better cache use and register reuse
  - Get more favorable halo exchange communications pattern
- Suggest testing this!
- Cray also points out a “reorder\_mesh” namelist option to try keeping adjacent compute tasks on the same node, but this doesn’t work with quilting.

# WRF HPC References

- <http://weather.arsc.edu/Events/AWS10/Presentations/Johnsen.pdf>
- [https://cug.org/5-publications/proceedings\\_attendee\\_lists/CUG10CD/pages/1-program/final\\_program/CUG10\\_Proceedings/pages/authors/01-5Monday/3C-Porter-paper.pdf](https://cug.org/5-publications/proceedings_attendee_lists/CUG10CD/pages/1-program/final_program/CUG10_Proceedings/pages/authors/01-5Monday/3C-Porter-paper.pdf)
- <http://www.ecmwf.int/sites/default/files/HPC-WS-Christidis.pdf>
- [http://www.markomanolis.com/Research/Talks/2013\\_BSC-ES-Course.pdf](http://www.markomanolis.com/Research/Talks/2013_BSC-ES-Course.pdf)