Optimizing performance of the Weather Research and Forecasting model at large core counts:
a comparison between pure MPI and hybrid parallelism and an investigation into domain decomposition

Christopher G. Kruse[1] (christopher.kruse@yale.edu) and Davide Del Vento[2] (ddvento@ucar.edu)

[1]Yale University, New Haven, Connecticut [2]National Center for Atmospheric Research, Boulder, Colorado

Introduction

Numerical simulations of the atmosphere are extremely useful in operational, research, and educational settings. These simulations can require vast computational resources and optimal efficiency is desirable. This study investigated performance of the widely used Weather Research and Forecasting (WRF) model [1] by studying the effects of hybrid parallelism (distributed (dm) and shared memory (sm) parallelism), the relationship between performance and increased computational resources, and the effects of varied domain decompositions on model performance.

The WRF Model, Compilation, Cluster

- Used WRF version 3.5
- Compiled with Intel, PGI and GNU compilers
- Compiled for both distributed-only and hybrid (dm and sm) parallelism
- Additionally, two MPI implementations were tested: an IBM Parallel Environment (PE) [2] and Intel [3] (IMPI) implementation
- Simulations completed at NCAR, on the Yellowstone supercomputer - 4,518 nodes, 16-core/node, 72,288 total cores [4]

The Katrina Workload

- Two simulations of Hurricane Katrina
- Different resolution and time step
- Same physical parameterizations
- Same number of vertical levels (35)

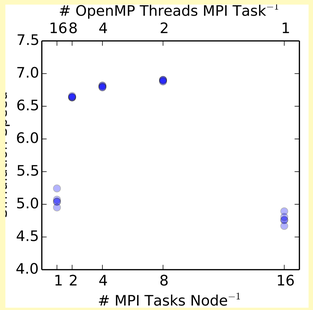
Resolution	1 km	3 km
Zonal grid points	3665	1396
Meridional grid points	2894	1384
Time step	3 s	10 s

- In addition to these to simulations, standard benchmarking workloads were also used in this study [5]

Acknowledgements

The authors would like to thank C. M. Patricola, P. Chang, and R. Saravanan for providing the Katrina workload [6], Dave Gill for extremely helpful feedback and suggestions, the Summer Internships in Parallel Computational Science program [7], Rich Loft, NCAR, and the NSF for financial support of this work.

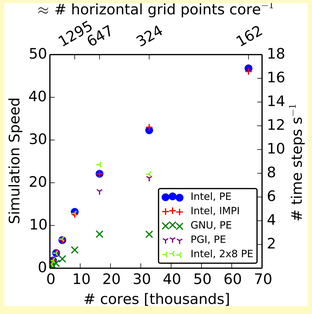
Hybrid Parallelism



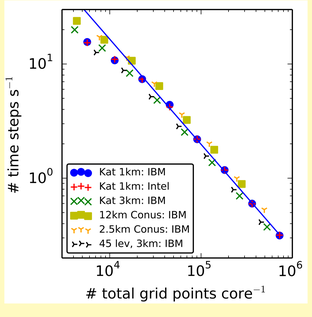
- MPI tasks/node and OpenMP threads per task were varied, keeping their product equal to the number of physical cores
- Average MPI-only simulation speed: 6.53
- Marginal increases (up to ≈5% relative to MPI-only) in simulation speed possible with hybrid parallelism using 2, 4, or 8 MPI tasks/node
- Performance decreased using 1 and 16 tasks/node

- $Simulation\ Speed = \frac{Simulated\ Time}{Wall\ Clock\ Time}$
- Processor binding environmental variable setting: "MP_TASK_AFFINITY=core:$OMP_NUM_THREADS"

Scaling



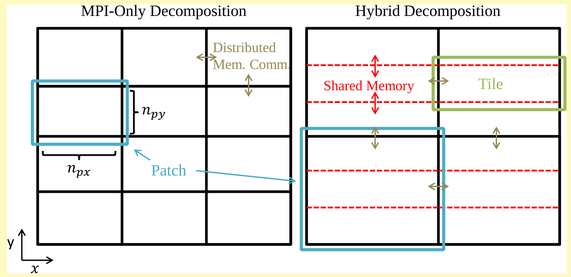
- WRF is known to scale well given enough work assigned to each core [5, 8]
- Scaled the 1-km Katrina workload with GNU, PGI, and Intel compilers
- Tested the two MPI implementations with the Intel compiler
- Scaling approximately linear through 16K cores for all compilers



- Compared performance when varying simulation size, resolution, parameterizations, vertical levels
- Linear scaling noted for all simulations when at least 20,000 grid points are assigned per core
- Deviations from linear scaling occur approximately at the same number of grid points/core for all workloads

- Non-linear scaling due to workload parallelization and not size of system utilized, since smaller workloads scale non-linearly similar to the larger workloads

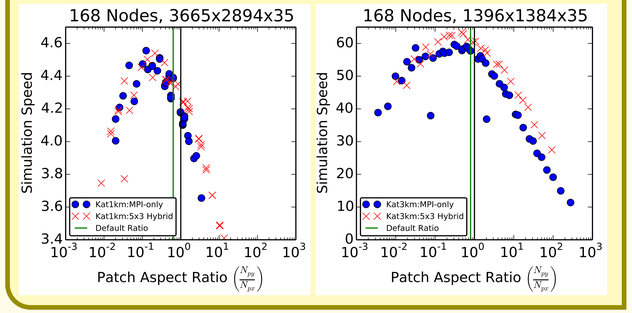
Domain Decomposition



- Given a number of MPI tasks, N, WRF decomposes the specified domain into N patches
- The number of patches in the x and y directions (N_x, N_y) must be factors of N: $N_x N_y = N$
- Number of grid points per patch: $N_p = N_{px} N_{py}$
- N_x, N_y are chosen to be as close to $\sqrt{N}$ as possible, defaulting to the decomposition that results in patches with aspect ratios $\frac{N_{py}}{N_{px}} < 1$
- Hybrid parallelism further decomposes patches into tiles, with default patch decompositions 1-D in the y direction

Decomposition Performance

- Grid points per patch constant (total cores constant), deviations from a patch aspect ratio of 1 increase perimeter points, MPI communication
- As a workload is scaled, MPI communication may become a non-negligible portion of time step completion
- Observed optimal performance occurs with aspect ratios less than one
- When N_p is large, (e.g. $N_p \approx 5800$ for MPI-only 1-km workload at left) the optimal aspect ratio is much less than one, small deviations in performance ($\sigma = 0.32$)
- When N_p is small, (e.g. $N_p \approx 770$ for MPI-only 3-km workload at right) the optimal aspect ratio closer to one, significant deviations in performance ($\sigma = 12.5$)
- The performance curves below are thought to be due to a trade off between optimal memory access (favoring low aspect ratios) and minimal MPI communication (achieved at an aspect ratio of unity)



Conclusions

Large WRF simulations can be scaled up to 64K cores. Deviations from linear scaling at large core counts occur due to non-negligible MPI communication times. Performance had some dependence on compiler choice, with best performance achieved using the Intel compiler on the Yellowstone system.

The use of hybrid parallelism can provide marginally higher performance than pure MPI parallelism, but high run to run variability was observed for certain processor binding settings on Yellowstone. More research is being conducted to ameliorate this variability.

When computational resources are not a constraint and time to solution must be minimized, these results show that care should be taken to not decompose the domain such that fewer than 20,000 grid points are assigned to each core/patch. Also, in this situation, performance is highly sensitive to domain decomposition, decreasing by as much as 83% with large aspect ratios. Hybrid parallelism can potentially increase performance of highly decomposed simulations since MPI communication is reduced.

At core counts larger than 4K, file I/O is a non-negligible (not shown) and optimizing I/O has the potential significantly reduce total run time.

References

[1] W. C. Skamarock, J. B. Klemp, J. Dudhia, D. O. Gill, D. M. Barker, M. G. Duda, X. Huang, W. Wang, and J. G. Powers. A description of the advanced research WRF version 3. 2008. NCAR Tech. Note NCAR/TN-4751STR, 113 pp. [Available online at `http://www.mmm.ucar.edu/wrf/users/docs/arw_v3.pdf`.].

[2] D. Quintero, A. Chaudhri, F. Dong, J. Higino, P. Mayes, K. Sacilotto de Souza, W. Moschetta, and X. T. Xu. IBM Parallel Environment (PE) Developer Edition. February 2013. http://www.redbooks.ibm.com/redbooks/pdfs/sg248075.pdf.

[3] Intel. Intel® MPI Library, 2013. http://www.intel.com/go/mpi.

[4] NCAR. NCAR-Wyoming Supercomping Center, 2013. http://www2.cisl.ucar.edu/resources/yellowstone.

[5] C. Eldred and J. Michalakes. WRF V3 parallel benchmark page, 2008. http://www.mmm.ucar.edu/wrf/WG2/benchv3.

[6] C. M. Patricola, P. Chang, R. Saravanan, and R. Montouro. The effect of the atmosphere-ocean-wave interactions and model resolution on Hurricane Katrina in a coupled regional climate model. *Geophys. Res. Abs.*, 14, 2012. http://meetingorganizer.copernicus.org/EGU2012/EGU2012-11855.pdf.

[7] NCAR. Summer internships in parallel computational science, 2013. http://www2.cisl.ucar.edu/siparcs.

[8] J. Michalakes, J. Hacker, R. Loft, M. O. McCracken, A. Snavely, N. J. Wright, T. Spelce, B. Gorda, and R. Walkup. WRF nature run. *J. of Physics: Conference Series*, 125, 2008. doi: 10.1088/1742-6596/125/1/012022.