Computational Efficiency Examination of a Regional Numerical Weather Prediction Model using KISTI Supercomputer NURION

Ji-Sun Kang[1]*, Sang-Kyung Lee[2], and Kiseok Choi[1]

[1] Convergence research center for data driven solutions, Korea Institute of Science and Technology Information, 41 Centum-dong ro, Haeundae-gu, Busan, 48059, Republic of Korea
[2]HPC Research Center, Moasys Corp., 4th floor, Il-Kwang Bldg, 220 Bawoomoe Road, Yangjae, Seocho, Seoul, 06746, Republic of Korea

Article History:Received:11 november 2020; Accepted: 27 December 2020; Published online: 05 April 2021

Abstract: For well-resolving extreme weather events, running numerical weather prediction model with high resolution in time and space is essential. We explore how efficiently such modeling could be, using NURION. We have examined one of community numerical weather prediction models, WRF, and KISTI's 5th supercomputer NURION of national HPC. Scalability of the model has been tested at first, and we have compared the computational efficiency of hybrid openMP + MPI runs with pure MPI runs. In addition to those parallel computing experiments, we have tested a new storage layer called burst buffer to see whether it can accelerate frequent I/O. We found that there are significant differences between the computational environments for running WRF model. First of all, we have tested a sensitivity of computational efficiency to the number of cores per node. The sensitivity experiments certainly tell us that using all cores per node does not guarantee the best results, rather leaving several cores per node could give more stable and efficient computation. For the current experimental configuration of WRF, moreover, pure MPI runs gives much better computational performance than any hybrid openMP + MPI runs. Lastly, we have tested burst buffer storage layer that is expected to accelerate frequent I/O. However, our experiments show that its impact is not consistently positive. We clearly confirm the positive impact with relatively smaller problem size experiments while the impact was not seen with bigger problem experiments. Significant sensitivity to the different computational configurations shown this paper strongly suggests that HPC users should find out the best computing environment before massive use of their applications

Keywords: High-resolution numerical weather prediction modeling, HPC, Parallel computing, Burst Buffer, KISTI supercomputer NURION

1. Introduction

Numerical weather prediction models have been developed since 1920s and have been practically applied to produce weather forecast based on computer simulations since 1950s [1]. Not only for predicting weather for the next days but also for understanding and analyzing weather phenomena, we employ the numerical models, based on the natural laws of atmospheric physics/dynamics, to produce the meteorological data of interest. Those models provide 4-dimensional atmospheric states (zonal, meridional, vertical, and temporal, 4 dimensions) for the period of interest, as users configure.

On the other hand, in recent years, high-impact weather such as heavy rainfall, severe storms, cold/heat waves, etc., tends to have relatively small spatial and temporal scales and hence high-resolution integrations of the numerical models are essential to analyze such extreme weather events over limited areas. Increasing resolution of the numerical weather prediction model is very demanding these days, but it could not be possible without the considerable computing power. This is why most popular supercomputers in the world are significantly consumed by the field of weather/climate modeling and simulation these days. Figure 1 shows the statistics of application area system share of TOP500 [2] supercomputer lists. It shows a significant portion of weather/climate research occupying the total supercomputer use over time.

Even though lots of computer resources have been devoted to the weather/climate research, most of atmospheric researchers need to consider the efficient use of the given limited computing resources. With the limited computing resources, therefore, one need to find out efficient parallelization strategies and utilization of available technology and equipment in the computing machine. In this study, we have tested one of community numerical weather prediction models, WRF (version 4.1.5) [3], using KISTI supercomputer NURION.

*Corresponding author: Ji-Sun Kang
Convergence research center for data driven solutions, Korea Institute of Science and Technology Information, 41 Centum-dong ro, Haeundae-gu, Busan, 48059, Republic of Korea

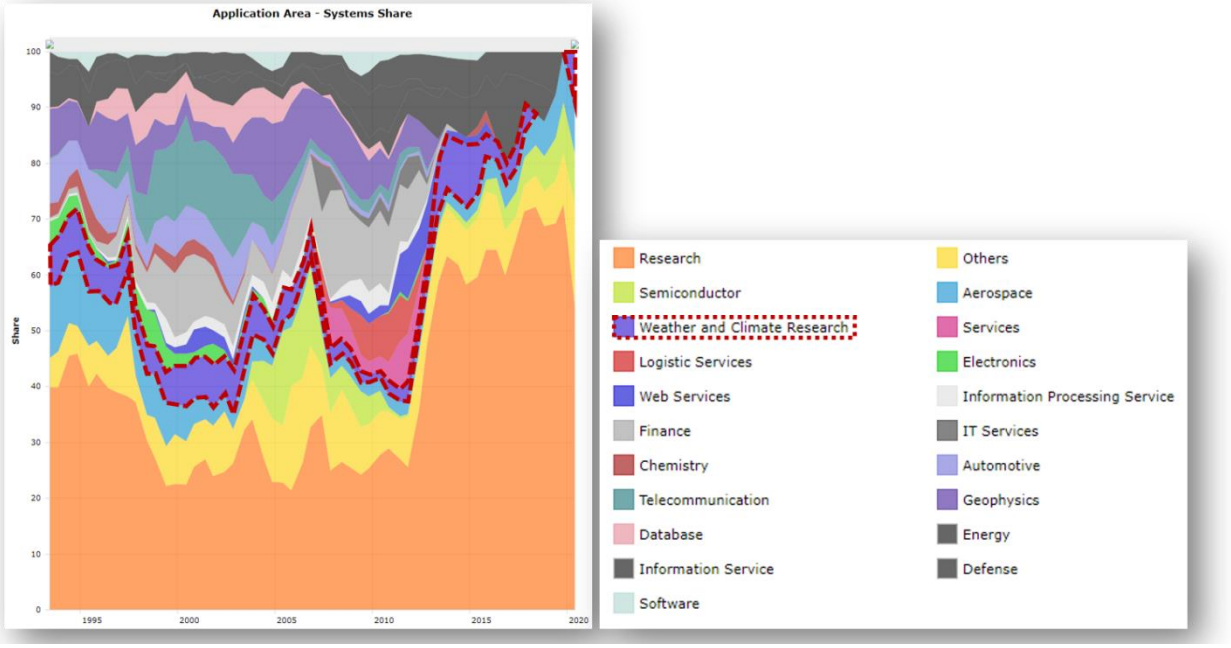

Figure 1. Statistics of Application Area-System Share over time from TOP500 Statistics[2]. Red dashed line highlights the portion of weather/climate research. (adopted from TOP500 statistics figures)

2. Related Works

Scalability is one of significant issues in operational numerical weather prediction models because it represents whether the state-of-art models with the increasing complexity and the resolution can be operated to produce necessary information in time with effective parallelization environment or not [4]. Thus, we need to examine how efficiently our application can be simulated under the given supercomputing environment, in a way to find out the optimal computational setting to obtain the simulation results as fast as possible.

In addition, massive amount of computing usually causes serious I/O performance bottleneck [5], [6]. As the resolution of the model becomes higher in time and space, data size generated by the model would be larger and/or the process of printing the result out could be more frequent. Unfortunately, however, I/O performance has not been improved as the floating-point operation per second (FLOPS) has been done [7]. Indeed, many researchers have been experienced slower I/O with super-high-resolution simulation that generates very big size of output, and then there are technical investigation to improve I/O performance (e.g. [8]). For improving I/O bottleneck problem with frequent I/O cases, NURION has introduced burst buffer storage that is expected to alleviate such bottleneck issue in High Performance Computing (HPC) systems, which uses fast non-volatile storage technologies [9]. Thus, we will investigate how well it works with the application of a regional numerical weather prediction model, WRF, for several experimental settings in this study. WRF is one of community numerical weather prediction models, which has been used for long by many researchers as well as operational centers in the world, to predict and understand atmospheric states mostly at regional scale [10-14].

3. Methods

We have examined a strong scalability of two-domain WRF configuration (Figure 2). There are 400X360 grids with 9 km horizontal resolution for the mother domain (D01) and 532X481 grids with 3 km resolution in the nested domain (D02). We set two-way nesting, which allows feedback of D02's high resolution results to D01's results. The number of vertical levels for both domains sets 50, and the timestep of the integration sets 12 seconds. Besides, we let the model write every hour output of D01, but do every 10 minutes output of D02. Hence, it gives more frequent model data for higher resolution states. This could be very useful setting for this numerical weather prediction model to analyze extreme weather events over South Korea.

With that fixed problem size, we have measured computing time for 1-day integrations of WRF with different MPI + openMP settings to see how well distributed the computations are with such different configurations. The period of experiment is from 06 UTC on Oct. 16, and the boundary/initial data of regional model integration are from NCEP GFS forecast data with 0.25° horizontal resolution [10]. Besides, we have tested burst buffer performance to see how much I/O can be accelerated. All experiments have been tried five times for each configuration to avoid abnormal performance due to any possible transient systematic errors.

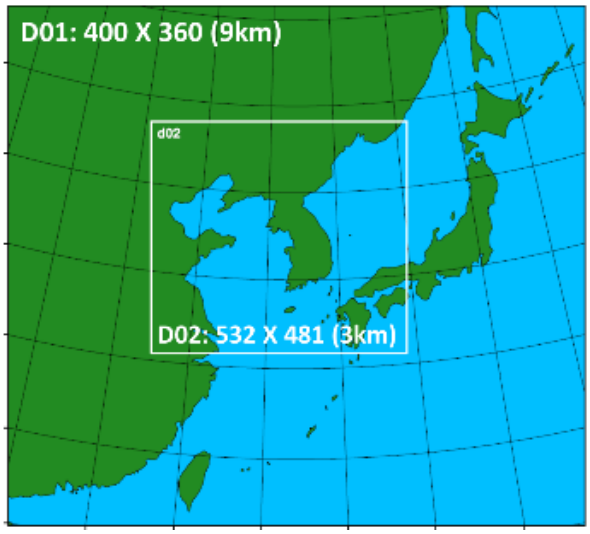

Figure 2. Domain setting of WRF experiments

Although KISTI NURION used for this study includes two different CPUs: Skylake and Knight Landing (KNL), we have examined only KNL cores in this study. It is because the majority of NURION is KNL architecture. Indeed, NURION has 8,305 KNL nodes and each node includes 68 cores (hyperthreads off), while NURION has only 132 nodes which have 40 Skylake cores per node. More detailed system spec can be found at the website of KISTI National Supercomputing Center [15]. Here, we built WRF codes with intel compiler (version 18.0.3) and intel MPI library (the same version with the compiler) that are pre-installed on NURION.

4. Experimental Results

First of all, we raise a question whether the computation would be fastest when using all cores per node. It would be easy to think that more CPU cores can give faster computation. But, we have tested the sensitivity of WRF simulation performance to the number of CPU cores per node, under the various settings of the number of nodes, such as 16/32/64/68 cores per node with 2/4/8/16 KNL nodes. Here, we first tested pure MPI runs while we will cover the performance tests of hybrid openMP + MPI runs next.

Table 1. Mean computing time (wall-time in sec) with various setting of the number of cores per node and the number of nodes.

# nodes	# cores per node	# total cores	Wall-time(sec)
2	16	32	71,057
	32	64	47,886
	64	128	**32,728**
	68	136	33,615
4	16	64	47,358
	32	128	**30,193**
	64	256	**42,691**
	68	272	43,075
8	16	128	30,006
	32	256	41,901
	64	512	**27,888**
	68	544	30,335
16	16	256	42,378
	32	512	26,794
	64	1024	**17,771**
	68	1088	21,082

The experimental results of Table 1 show that using as many cores per node as possible does not guarantee the best performance in practice. In fact, a system administrative work and/or WRF's memory-bounded process let the use of all cores per node less efficient, and hence using all cores per node rather reduces computational speed. Using 68 cores per KNL node never gives the best results for any cases of 2/4/8/16 nodes.

Interestingly, when using 4 nodes, the case with 32 cores per node gives the best computational speed, rather than 64 cores per node, while using 64 cores per node usually shows the best performance. As Figure 3 shows, however, 128 core computation has best combination of MPI distribution with grid-based computation size of current experimental setting. Still, we could have a general conclusion that full use of cores for the given nodes does not give the best result of computational application. That is, it would be better to leave several cores per node to avoid interruption from possible system administrative work and proper use of memory per node.

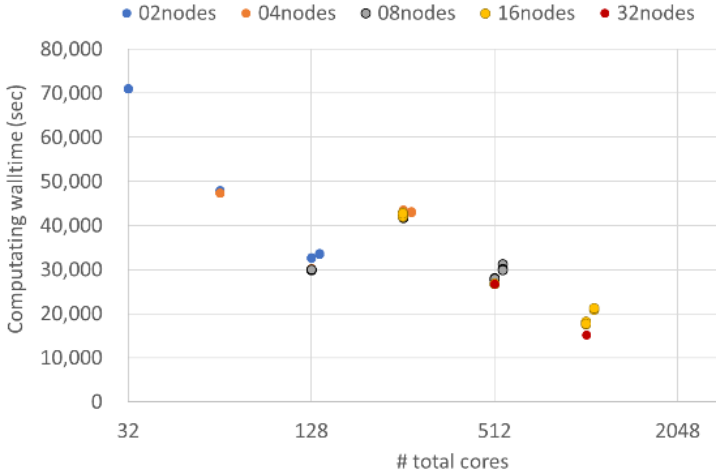

Figure 3. Computing wall-time w.r.t. the number of total cores used.

Figure 3 also tells us that the current problem of WRF is scalable up to 1,088 cores. We have tried to run the same problem with 32 nodes, but it turns out that the problem size is too small to be parallelized into 2,048 cores (64 cores per node with 32 nodes). Thus, we conclude that the WRF experiment configured for this study is well scalable up to about 1,024 cores and hence we decided to run the experiment effectively with 16 nodes and 64 cores per node.

Next, we have tested hybrid openMP + MPI simulation. Based on the experience from the previous pure MPI runs, we have tested the hybrid runs only for the case with 16 nodes and 64 cores per node. Using16 nodes, we have tested six different openMP thread experiments five times, and then each experiment is summarized by the average in Table 2. The problem size of current experimental settings does not show significant improvement of hybrid openMP + MPI configuration, compared to pure MPI computation. The results show that computational cost of hybrid runs gives worse performance than the pure MPI simulation in any combination of threads and cores. Rather, using more threads causes slower computation and thus the run with 32 threads shows about 3.9 times slower than the pure MPI run. There were several results of WRF benchmarking, consistent with our results [16-17]. Despite those results, one with a new configuration of WRF still needs to test hybrid openMP + MPI runs because the performance results could depend on the problem size and the system environment.

Table 2. Mean computing time (wall-time in sec) with various numbers of openMP threads**.**

# nodes	# MPI	# threads	Wall-time(sec)
16	64	1	**17,771**
	32	2	27,320
	16	4	41,995
	8	8	30,254
	4	16	48,757

2	32	69,746

Now, we take a look at the performance results from burst buffer utilization to see whether it really helps accelerating frequent I/O that often causes serious performance bottleneck. There are two experiments we have tried; CTRL_noBB that does not employ burst buffer, and EXP_BB that employs burst buffer for I/O. Both cases use 16 KNL nodes and 64 cores per node in pure MPI setting. Since log files from WRF integration give three different timing information such as "main" (standing for time integration of prognostic equation), "write" (printing out the results), and "process" (rest of processes, very small in general), we also extract such timing information to analyze the results.

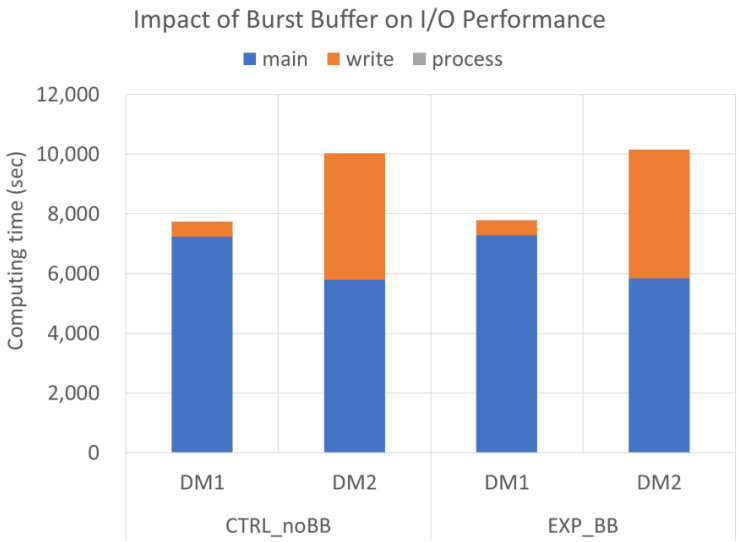

Figure 4. Computing time consuming for two domains of WRF from CTRL_noBB (left) and EXP_BB (right). Blue bar indicates wall-time for main integration of the model, and orange indicates time consuming for writing output. Gray bar indicating other processes is invisible due to its small portion of total wall-time.

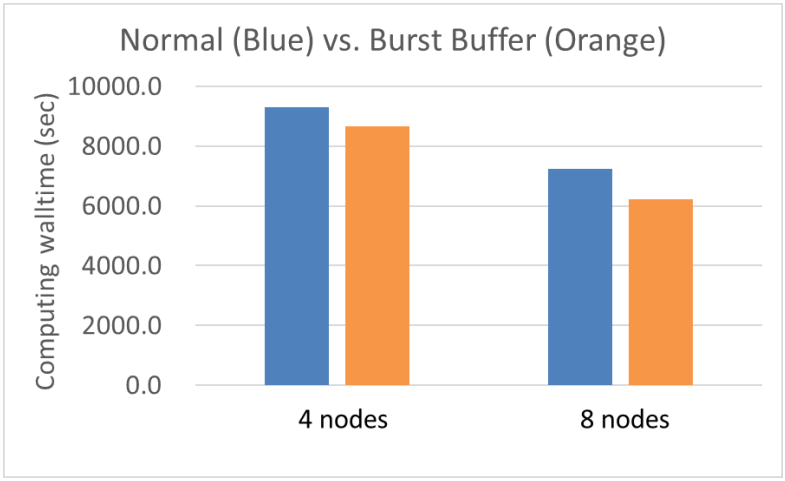

Figure 5. Computing wall-time results from experiment with/without burst buffer when using 8 KNL nodes.

We expected to see some improvement on computing time for "write" in EXP_BB compared to CTRL_noBB. However, both CTRL_noBB and EXP_BB show almost identical performance results (Figure 4), which does not confirm any advantages of burst buffer utilization unfortunately. Those results are also summarized by five times of each experimental setting. However, it could be possible that the results depend on the problem size and other systematic environment.

Indeed, we have another result showing that using burst buffer accelerates computing time significantly under somewhat different WRF experimental setting. We have ever tried WRF with smaller domain size such as 350X331 and 331X319 grids for D01 and D02, respectively. Those experiments still set the horizontal resolution same, 9 km and 3 km for D01 and D02, respectively, and other experimental setting including physics

package and the number of vertical levels was same with the experiment previously shown in this paper.

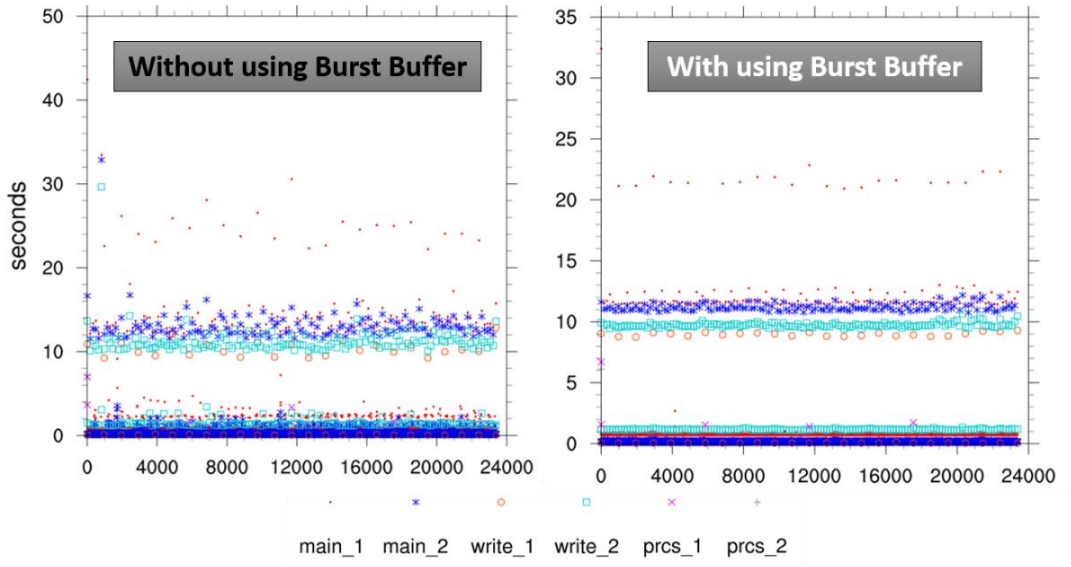

Figure 6. Time series of elapsed time for computing every timestep, from WRF log analysis of the experiments without (left)/with (right) burst buffer in one of 8 node cases.

Figure 5 shows the comparison of computing wall-time from the experiment with/without burst buffer. There are 7.4 % and 16.6% speed-up impact of burst buffer for 4 node case and 8 node case, respectively. Those results are also averaged by five times repetition of the same experiment for each, in order to avoid the data contamination due to occasional abrupt delay of supercomputing system.

Also, from the analysis of WRF log file, we could see that the use of burst buffer gives much more stable performance of all the steps including "main" and "write" for both domains. Figure 6 shows the time series of computational elapsed seconds at every step, written in WRF log file. It clearly shows that burst buffer gives very stable ~10 seconds elapsed time for not only writing output but also integration of D02. The integration timestep of D02 is very small (4 seconds) and the output of D02 is printed quite frequently (every 10 minutes). Therefore, such stable performance of the integration and the model output processes provided by burst buffer firmly improves the computational speed very effectively.

5. Conclusions

In order to generate high-resolution extreme weather data in time with the limited computing resources, we have investigated the scalability of a community numerical weather prediction model WRF on KISTI's 5th supercomputer NURION. In addition, we have tested computational performance with different MPI and/or openMP configuration. Lastly, we have examined burst buffer storage layer to see whether it could reduce I/O performance bottleneck, one of serious performance bottleneck issues.

First, we have tested whether it would be the best choice to use all cores per node for the computational efficiency. Results confirm that leaving several cores on each node would be better strategy to achieve better performance, rather using all cores per node. Those results are very consistent even with different configurations. And, we found that the current WRF experiment has good scalability up to 1,088 cores.

We also tested hybrid openMP + MPI run, but our WRF experiment with the current configuration does not give any better performance than pure MPI run. Rather, using more threads lets the computation slow down. Those results may not be consistent with respect to a problem size and a detailed configuration of the model simulation. Therefore, users still need to check which computational strategy would be optimal for ones' situation.

Lastly, we have tested burst buffer storage layer to see its impact on a frequent I/O of the model. Even though we experienced somewhat inconsistent results between two different problem sizes of WRF experiments, burst buffer storage layer shows a potential to stabilize I/O performance as well as model integration. In general, frequent small I/O is expected to be accelerated by burst buffer and we confirmed it with relatively smaller WRF experiment, although another bigger problem set does not benefit from burst buffer. It tells us that the performance improvement of burst buffer can also depends on experimental design and system environment. We

plan to see more details on application of burst buffer storage layer with respect to WRF simulation for the further study.

In this study, all of experimental results are summarized statistically and we have introduced how efficiently we could generate the numerical data of high resolution weather events from a HPC system. Our experimental results show that one needs to test their application into the system with various setting such as the number of cores per node, the best combination within hybrid openMP threads and MPI cores, and I/O acceleration technology, to find the best combination of one's application in terms of computational efficiency.

Acknowledgements

This research was conducted with the support of Data Driven Solutions(DDS) Convergence Research Program funded by the National Research Council of Science and Technology "Development of solutions for region issues based on public data using AI technology -Focused on the actual proof research for realizing safe and reliable society-"(1711101951). Furthermore, this work could be possible due to the support of National Supercomputing Center at KISTI. We also deeply thank Mr. Hunjoo Myung for his sincere discussion on our work.

References

1. Harper, Kristine, Louis W. Uccellini, Eugenia Kalnay, Kenneth Carey, and Lauren Morone. "50th anniversary of operational numerical weather prediction." *Bulletin of the American Meteorological Society* 88, no. 5 (2007): 639-650.
2. Top500-List Statistics on http://top500.org/ statistics/overtime
3. Skamarock, W. C., and J. B. Klemp. A Description of the Advanced Research WRF Model Version 4. Ncar Technical Notes, No. NCAR/TN-556+ STR, 2019.
4. Mizielinski, M. S., M. J. Roberts, P. L. Vidale, R. Schiemann, M-E. Demory, J. Strachan, T. Edwards et al. "High resolution global climate modelling; the UPSCALE project, a large simulation campaign." *Geoscientific Model Development* 7, no. 4 (2014): 1629-1640.
5. Xie, Bing, Jeffrey Chase, David Dillow, Oleg Drokin, Scott Klasky, Sarp Oral, and Norbert Podhorszki. "Characterizing output bottlenecks in a supercomputer." In *SC'12: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, pp. 1-11. IEEE, 2012.
6. Shantharam, Manu, Mahidhar Tatineni, Dongju Choi, and Amitava Majumdar. "Understanding I/O Bottlenecks and Tuning for High Performance I/O on Large HPC Systems: A Case Study." In *Proceedings of the Practice and Experience on Advanced Research Computing*, pp. 1-6. 2018.
7. From FLOPS to IOPS: The New Bottlenecks of Scientific Computing on http://sigarch.org
8. Yepes-Arbós, Xavier, Mario C. Acosta, Gijs van den Oord, and Glenn Carver. *Computational aspects and performance evaluation of the IFS-XIOS integration*. European Centre for Medium Range Weather Forecasts, 2018.
9. Han, Jaehyun, Donghun Koo, Glenn K. Lockwood, Jaehwan Lee, Hyeonsang Eom, and Soonwook Hwang. "Accelerating a burst buffer via user-level i/o isolation." In *2017 IEEE International Conference on Cluster Computing (CLUSTER)*, pp. 245-255. IEEE, 2017.
10. NCEP GFS 0.25 Degree Global Forecast Grids Historical Archive on http://rda.ucar.edu/datasets/ds084.1
11. Cardoso, R. M., P. M. M. Soares, P. M. A. Miranda, and M. Belo-Pereira. "WRF high resolution simulation of Iberian mean and extreme precipitation climate." *International Journal of Climatology* 33, no. 11 (2013): 2591-2608.
12. Zheng, Yue, Kiran Alapaty, Jerold A. Herwehe, Anthony D. Del Genio, and Dev Niyogi. "Improving high-resolution weather forecasts using the Weather Research and Forecasting (WRF) Model with an updated Kain–Fritsch scheme." *Monthly Weather Review* 144, no. 3 (2016): 833-860.
13. Cunden, TS M., A. Z. Dhunny, M. R. Lollchund, and S. D. D. V. Rughooputh. "Sensitivity Analysis of WRF Model for Wind Modelling Over a Complex Topography Under Extreme Weather Conditions." In *2018 5th International Symposium on Environment-Friendly Energies and Applications (EFEA)*, pp. 1-6. IEEE, 2018.
14. Case, Jonathan L., William L. Crosson, Sujay V. Kumar, William M. Lapenta, and Christa D. Peters-Lidard. "Impacts of high-resolution land surface initialization on regional sensible weather forecasts from the WRF model." *Journal of Hydrometeorology* 9, no. 6 (2008): 1249-1266.
15. NURION at KISTI National Supercomputing Center on https://ksc.re.kr/ eng/resource/nurion
16. Morton, Don, Oralee Nudson, and Craig Stephenson. "Benchmarking and evaluation of the Weather Research and Forecasting (WRF) Model on the Cray XT5." *Cray User Group Proceedings, Atlanta, GA* (2009): 04-07.
17. Weather Research and Forecast (WRF) Scaling and Performance Assessment NCAR SIParCS Program on https://akirakyle.com/ WRF_benchmarks/results.html#hybrid