Maximizing the Performance of the Weather Research and Forecast Model over the Hawaiian Islands

Kevin P. Roe
The Maui High Performance Computing Center, Kihei, HI 96753 USA

Duane Stevens
Meteorology Department, University of Hawaii at Manoa, Honolulu, HI 96822 USA

Abstract

The Hawaiian Islands consist of dramatic terrain changes over short distances, resulting in a variety of microclimates in close proximity. To handle these challenging conditions, weather models must be run at very fine vertical and horizontal resolutions to produce accurate forecasts. Computational demands require WRF to be executed in parallel on the Maui High Performance Computing Center's Mana system, a PowerEdge M610 Linux cluster. This machine has 1,152 compute nodes, each with two 2.8 GHz quad-core Intel® Nehalem processors and 24 GB RAM. Realizing maximum performance on Mana relied on the determination of an optimal number of cores to use per socket, the efficiency of an MPI only implementation, an optimal set of parameters for adaptive time stepping, a way to meet the strict stability requirements necessary for Hawaii, effective choices for processor and memory affinity, and parallel automation techniques for producing forecast imagery.

I. INTRODUCTION

The telescope operations on Haleakala are highly dependent on weather conditions around the Hawaiian island of Maui. The telescopes cannot be used under a variety of conditions, including high wind speeds, heavy clouds, rainfall, high relative humidity, and high levels of optical turbulence. Even when these conditions are within an acceptable range to allow operations, they can diminish the effectiveness of the telescopes. In order to efficiently schedule telescope operations, a timely accurate weather prediction is extremely valuable. Current forecasts that are available from the National Weather Service (NWS) give good indications of approaching storm fronts but only at the medium-coarse level (20-30 km resolution). Because of the location of the telescopes on Maui, this can be insufficient for their needs.

The additional benefit of having access to an accurate forecast is that they can perform some operational scheduling for the telescope facilities. For example, if unacceptable weather conditions are predicted, they can plan maintenance. This allows the facility to function more effectively by saving time and ultimately operating expense.

II. NUMERICAL WEATHER MODELING

The numerical weather model (NWM) used for this project is the Weather Research and Forecasting (WRF) Model [1], [2]. It was chosen because it has many desirable capabilities:

1. Handles multiple nested grids.
2. Excellent data assimilation routines.
3. Excellent initialization routines.
4. Operates in parallel for faster execution.

The nested grid capability allows a coarse grid to be run over a large area (of less interest over the open ocean) in less compute time while still being able to operate using finer grids on smaller areas (of great interest over the Hawaiian Islands where terrain changes can dramatically change in short distances), rather than using a fine grid over a very large area at a high computational cost. Observation data ingested by the WRF-Variational (WRF-Var) [3] data assimilation routine allows for a "hot start," which means less spin-up time for the WRF simulation. The WRF Preprocessing System (WPS) is the first step to initialize the model for real data simulations. Finally, the ability for the WRF model to be run in parallel is crucial because it allows the production of high-resolution output in a reasonable time frame (when the forecast produced by the simulation is still a prediction).

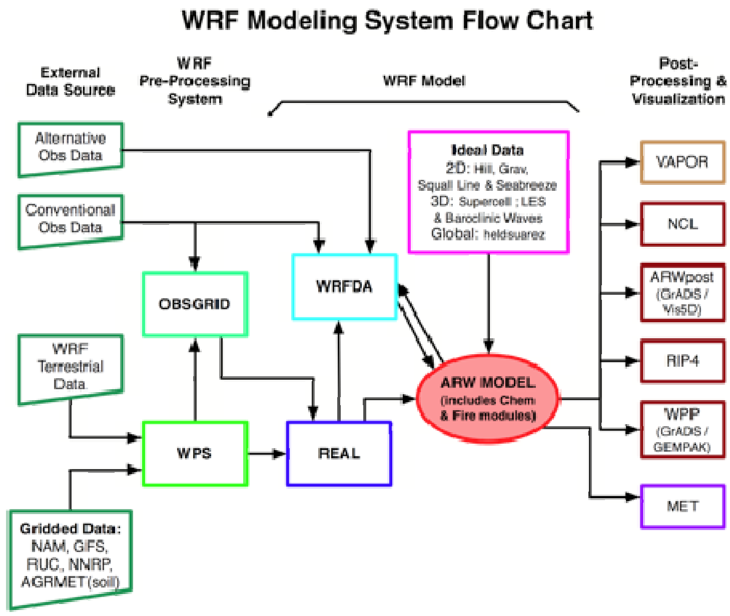

Figure 1: WRF Modeling System Flow Chart

The best way to understand how the WRF model operates is to explain the main routines it uses to accomplish a numerical simulation. Figure 1 is a flow chart of the main routines used in the WRF model. Focus will be given to WPS, REAL, ARW (Advanced Research WRF) model, and lastly the post-processing and visualization tools. WPS is a collection of programs: The static fields and grid domains are specified in the first program, “geogrid.” The external analysis and forecast data are decoded from the GRIB (GRIdded Binary) format with the “ungrib” program in WPS. The final program in WPS is “metgrid,” which horizontally interpolates the data from “ungrib.” The output data from WPS is passed to REAL, which converts the output of WPS into a format useable by the WRF model. The WRF model is then run to generate the numerical weather simulation over the desired area. The output data can then be post-processed and visualized with a variety of utilities.

III. Setup and Area of Interest

The WRF model is a fully compressible, non-hydrostatic model (with a hydrostatic option) utilizing terrain-following sigma vertical coordinates. In this simulation we will use:

1. 55 vertical levels from the surface to the 10-millibar (mb) level with a bias towards levels below a sigma of 0.9 (close to the surface). High vertical resolution is needed at the lowest levels to resolve the anabatic flow, katabatic flow and nocturnal inversion in the near surface layer [4], [5].
2. The Betts-Miller-Janjic cumulus parameterization scheme [6] is used for the 54 and 18 km resolution domains. For the rest of the finer resolution domains no parameterizations are used. It is an appropriate parameterization scheme for this level of resolution.
3. The Mellor-Yamada-Janjic Planetary Boundary Layer (PBL) scheme [6] for all domains. It is a one-dimensional prognostic turbulent kinetic energy scheme with local vertical mixing.
4. 4.The Monin-Obukhov (Janjic Eta) surface-layer scheme. Eta similarity; based on Monin-Obukhov with Zilitinkevich thermal roughness length and standard similarity functions from look-up tables.
5. The RRTM (Rapid Radiative Transfer Model) for long-wave radiation. An accurate scheme using look-up tables for efficiency. Accounts for multiple bands, trace gases, and microphysics species.

6. The Dudhia scheme for shortwave radiation: A simple downward integration allowing for efficient cloud and clear-sky absorption and scattering.
7. A 5-layer soil ground temperature scheme. Temperature is predicted in 1, 2, 4, 8, and 16 cm layers with fixed substrate below using the vertical diffusion equation.
8. The Ferrier (new Eta) microphysics: the operational microphysics in NCEP models; a simple efficient scheme with diagnostic mixed-phase processes.

The area of interest is the Hawaiian Islands because this prediction is intended for the operators of the telescopes on Haleakala. Two major items must be handled in order to produce a useful accurate forecast. First, the simulation must include a significant area surrounding the Hawaiian Islands in order to capture storm systems early on. Secondly, the Hawaiian Islands contain a variety of microclimates in a very small area. Some islands have rainforests within a few miles of deserts; some have 10,000+ feet summits only a few miles away from the coastline. To effectively model these conditions, the WRF simulation must be run at a very fine resolution. To satisfy both requirements, a nested grid approach must be used. The WRF model uses a conventional 3:1 nesting scheme for two-way interactive domains. This allows the finer resolution domains to feed data back to the coarser domains. The largest domain covers an area of approximately 7000 km by 7000 km at a 54 km grid resolution. Figure 2 displays how the grid is then nested down to 18 and 6 km around the Hawaiian Islands and then down to 2 km for each of the 4 counties.

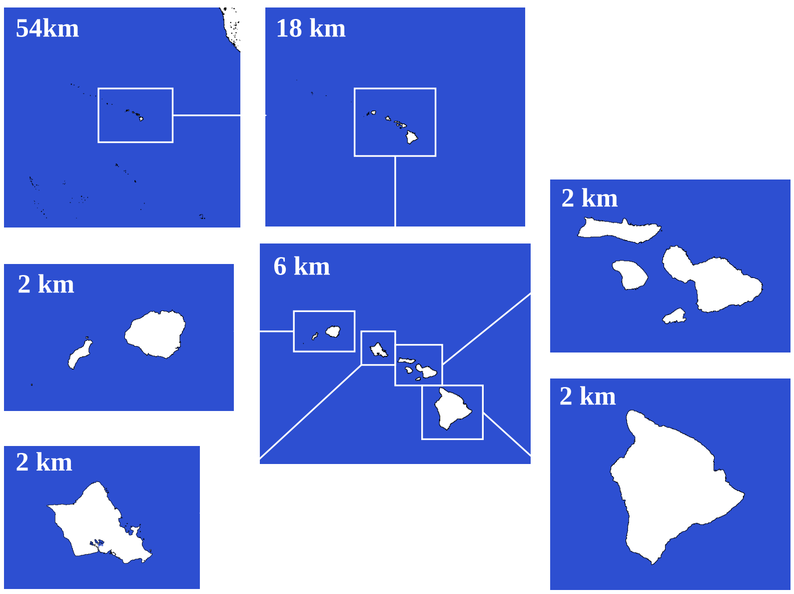

Figure 2: Nest Domain Structure

IV. Daily Operations

Every night at midnight Hawaiian Standard Time (HST), a PERL script is run to handle the entire operation necessary to produce a weather forecast and post it to the Haleakala Weather Center web page (*http://weather.mhpcc.edu*) hosted by MHPCC. The script executes the following steps:

1. Determine and download the latest global analysis files from NCEP for a 48-hour simulation.
2. WPS processes the downloaded data into format useable by REAL.
3. REAL processes the data files from WPS into a format useable by the WRF model.
4. Submit the parallel WRF run to MHPCC's 2.8 GHz Nehalem based Linux System ("Mana") for execution.
5. Average daily run on 2 nodes requires ~3.00 hours.
6. Simulation data is output in 1-hour increments.
7. Simulation data is processed in parallel to create useful images for meteorological examination.

8. Convert images to a web viewable format.
9. Create the web pages these images will be posted on.
10. Post web pages and images to MHPCC's web site.

Most of these stages are self-explanatory, but some require additional information. Step 1 can require some time as the script is downloading 9 distinct, 24 to 26 MB global analysis files from NCEP. This can affect the time it takes for the entire process to complete as the download time can vary based on the NCEP ftp site, web congestion, and MHPCC's connectivity. In addition, the data is posted to the NCEP ftp site starting at 11 P.M. (HST) and complete any time from 11:45 P.M. to 12:00 P.M. (HST); hence the script is setup with a means to check the "freshness" and completeness of the files to be downloaded. Step 4, job submission, is handled through a standing reservation for 2 nodes (4 processors, 16 cores) starting at 12:30 A.M. (HST). This ensures that the model will be run and completed at a reasonable time in the morning. Step 7, data processing, includes the choices of fields to be output to the web. Current choices are: temperature, wind speed & direction, relative humidity, and rainfall. A more detailed description is given below:

1. Surface temperature (° Fahrenheit at 2 meters).
2. Surface wind (Knots at 10 meters).
3. Relative Humidity (% with respect to water): This field provides the relative humidity at the lowest sigma level (.99). Sigma of .99 conforms to an Elevation of 96 meters (315 ft) above sea level (see Figure 3 to visualize the terrain conforming sigma levels).
4. Hourly accumulated rainfall (mm).

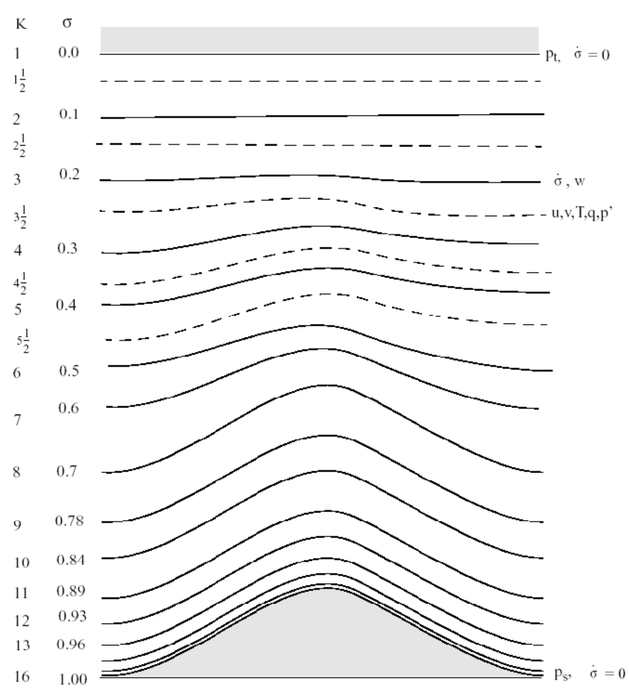

Figure 3: Terrain conforming sigma vertical levels

Additional capabilities have been added to the process of obtaining the forecasts [7] include:

1. Highly reliable (fault-tolerant) script.
2. Script retrieves the most recent pre-processing data (global analysis, observational data, etc).
3. Script handles parallel image and data post-processing for web posting.

The fault-tolerant script ensures that the operation will adjust and continue even in the face of an error or will report that there is a process ending error. The script has been written to be smart enough to retrieve the latest pre-processing data if it is not already present on the system; this ensures that the simulation will have the most recent data and/or avoid downloading data that is already present. Parallel image and data processing (through the use of child processes) has been shown to achieve a 6 times speedup on 1 node with 2 processors (8 cores). This type of parallelism allows the capability of plotting more fields without significantly increasing the total image processing time with the addition of more nodes/cores.

V. WEB OUTPUT

Now that the above processes have created images, they must be made available for the telescope operators [8], [9]. This is accomplished by posting to the MHPCC web page, http://weather.mhpcc.edu. The title page (Figure 4) gives the user the option of what model, domain, and resolution they would like to examine.

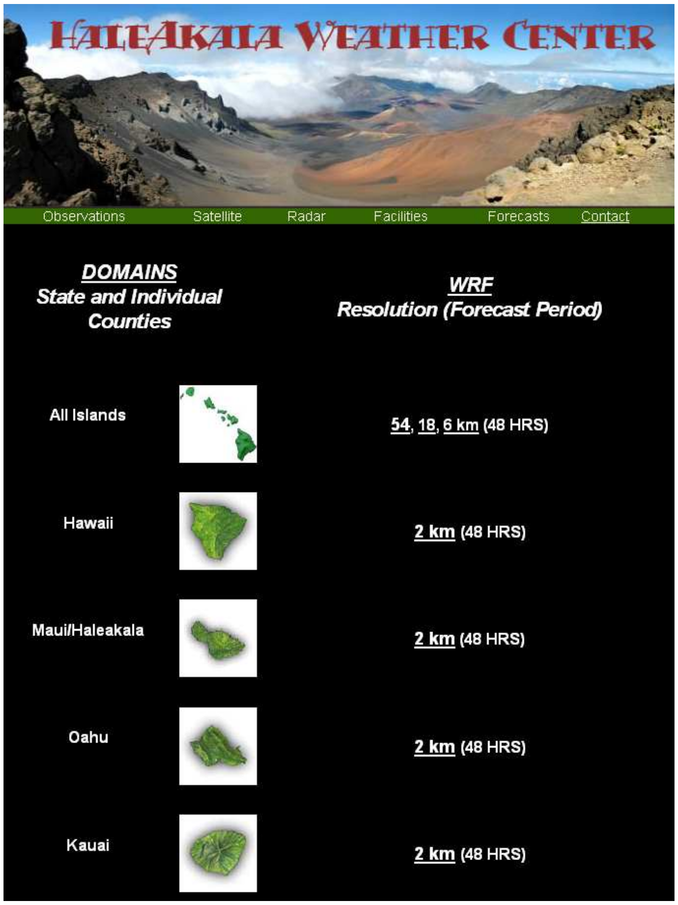

Figure 4: Haleakala Weather Center Homepage

From the title page, the user can select from WRF all-island forecasts at resolutions of 54, 18, and 6 km, as well as 2 km resolutions for all 4 counties (Hawaii, Maui, Oahu, and Kauai). Once one of the above has been selected, the user is transported to a regional web page that initially includes an image of the surface wind speed and direction in the selected area.

On the regional web pages (see Figure 5 and Figure 6), the viewer can select to see the previous or next image, through the use of small JavaScript. If the viewer prefers, an animation of the images (in 1 hour increments) can be started and stopped. Finally, the user can select any of the hourly images from a pull-down menu. If the viewer would like to change the field being examined, a pull-down menu on the left side of the page will transport the user to the main menu to choose a different field, domain, or domain from a different model.

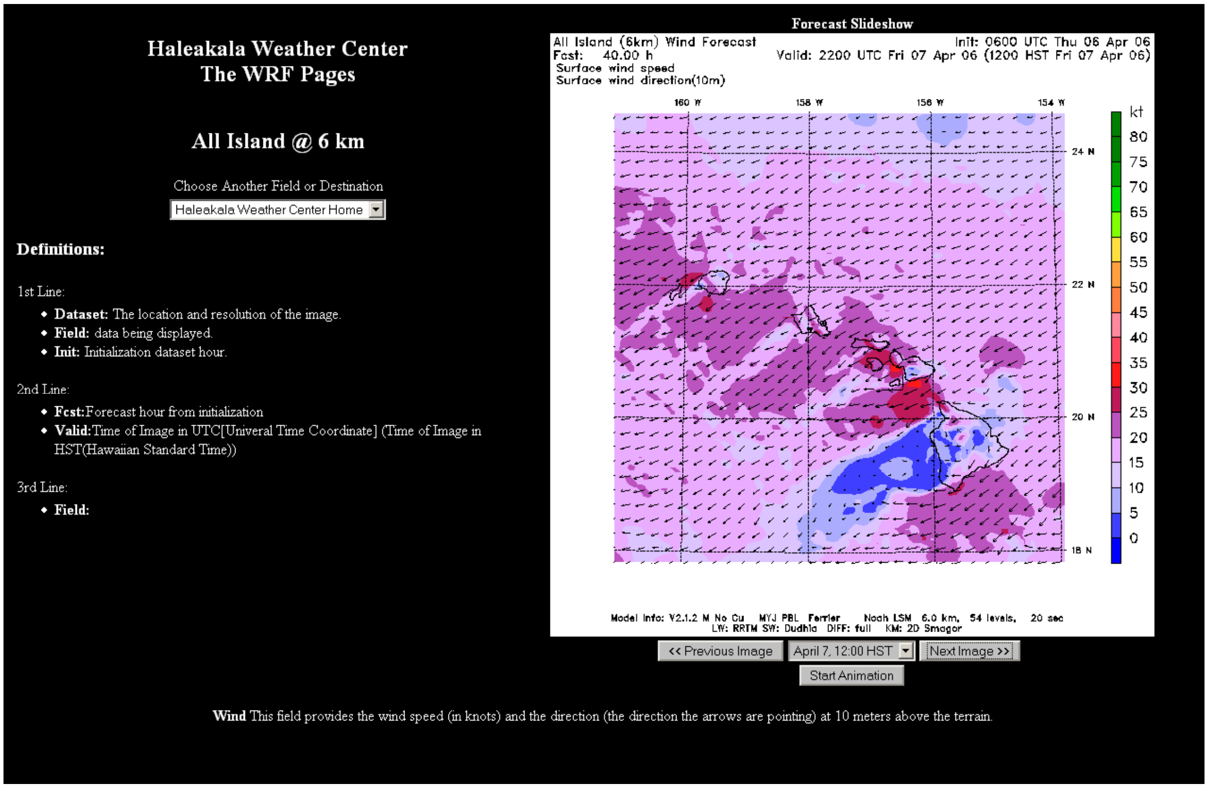

Figure 5: Wind Speed and Direction (Knots)

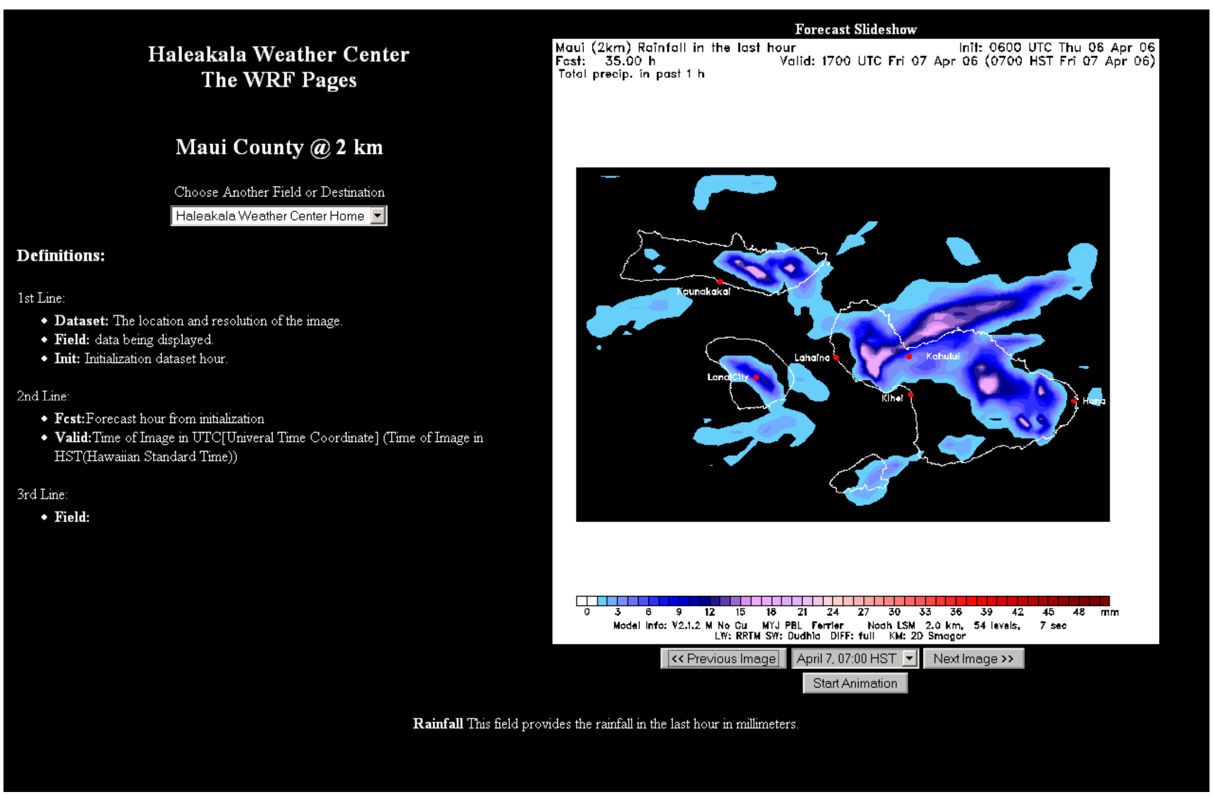

Figure 6: Hourly Accumulated Rainfall (mm)

VI. SCHEDULING AND BENCHMARKING

In order to produce daily operational forecasts, a strict schedule must be maintained and a choice must be made as to how many nodes (and cores) will be utilized for the model's execution. In order to determine this, benchmarks were done to determine the processing time, the parallel efficiency, and the node hour cost. Processing time was examined so that we may maintain the schedule needed to produce a timely daily forecast. The goal of having the forecast ready before 8 AM (HST) would be helpful for operators to determine a schedule for the current and following evening. Speed up and parallel efficiency were examined to determine the most cost-effective choice (see Table 1**Error! Reference source not found.** and Table 2). There are some details that must be clarified for these benchmarks:

1. The "Mana" Linux cluster consists of 1,152 compute nodes, each with two 2.8 GHz quad-core Intel® Nehalem processors and 24 GB RAM (3 GB/core), for a grand total of 9,216 compute cores. It uses a high speed interconnect dual data rate InfiniBand to provide low latency, high bandwidth, and low CPU overhead; this allows for excellent scalability between Mana's nodes.
2. Each node consists of 2 processors with 4 cores per processor for a total of 8 cores per node.
3. Parallel run times are implemented using MPI libraries (OpenMPI) only. Hybrid MPI and OpenMP runs performed significantly worse and were discarded as a viable option [10].
4. The run time is average of 5 runs.
5. Speed up is calculated from the sequential run time divided by the parallel run time.
6. Core cost is calculated by multiplying the run time by the number of cores used. This cost assumes that one is only being charged for the cores utilized, not for the exclusive use of the whole node.
7. Exclusive node core cost is calculated by multiplying the run time by the total number of cores for all nodes even if a subset of these cores is utilized. This metric is needed because exclusive use of the node is necessary for it to reach its maximum performance potential.
8. The runs that use 1 node, 1 processor, and 1 core are designated as 1 node, 1 core per processor, and 1 total core. This is different than the runs that are 1 node, 1 core per processor and 2 total cores; this run uses 2 processors per node with 1 core per processor.

The REAL code was run in parallel using MPI libraries and benchmarked (Table 1). Although it does not take long to process it was run in parallel since the nodes were already reserved and the total exclusive node core cost is less than a sequential run. The savings were only 25 seconds but the speedup was 2.92 times; more importantly, the node cost was only 1.733 minutes as opposed to the sequential runs cost of 5.067 minutes. Clearly there is a loss of efficiency as more cores are used per node, but the exclusive node core cost drops as more cores are used within a single node. When the code is run across nodes the performance takes a hit. The overhead associated with running this code across nodes is just too much to achieve any performance worth the node cost.

Table 1: REAL Benchmark

Nodes	Cores Per Proc	Total Cores	Time: Average (s)	Speed Up	Efficiency (%)	Cost: Core Minutes	Cost: Exclusive Node Core Minutes
1	1	1	40	0.95	95.0%	0.667	5.333
1	1	2	26	1.46	73.1%	0.867	3.467
1	2	4	21	1.81	45.2%	1.400	2.800
1	4	8	13	2.92	36.5%	1.733	1.733
2	1	2	46	0.83	41.3%	1.533	12.267
2	1	4	37	1.03	25.7%	2.467	9.867
2	2	8	43	0.88	11.0%	5.733	11.467
2	4	16	47	0.81	5.1%	12.533	12.533
Sequential Performance							
1	1	1	38	1.00	100%	0.633	5.067

The WRF code was run in parallel using MPI libraries and benchmarked (Table 2). A variety of node, processors per node, and cores per processor combinations were examined. The sequential time is displayed at the bottom of the table; note that although its core cost is the smallest of all runs (because of MPI library overhead) it has a much higher exclusive node core cost. A 1 node 1 processor 1 core run was done to show the overhead for using the MPI libraries; it showed 95% efficiency. Next, testing was done of single node performance using multiple cores. There is a drop in efficiency as more cores are used. The drop is minimal for 2 cores, but more significant for 4 and 8 cores. The most likely reason for this drop in performance is the increased synchronization overhead and less available memory bandwidth per process when high core counts are used [11], [12], [13]. Although the efficiency has drop to 48.7% for 1 node with 8 cores, the exclusive node core cost is still the lowest of all cases examined; At 37.12 core hours this yields a total annual cost of 13,215 core hours. Given that the goal of returning a forecast by 8 AM can still be reached with this configuration (take 4 hours and 38 minutes on average), it is the optimal choice remembering that run time and exclusive node core costs are the reason why parallel computing is useful and cost efficient to this project.

Table 2: WRF Benchmark

Nodes	Cores Per Proc	Total Cores	Time Avg. (s)	Speed up	Eff. (%)	Cost: Core Hrs	Cost: Excl. Node Core Hrs
1	1	1	68560	0.95	95.0%	19.04	152.56
1	1	2	35798	1.82	90.9%	19.89	79.55
1	2	4	24115	2.70	67.5%	26.79	53.59
1	4	8	16705	3.90	48.7%	37.12	37.12
2	1	2	36694	1.78	88.8%	20.38	163.08
2	1	4	22417	2.91	72.6%	24.90	99.63
2	2	8	13442	4.85	60.6%	29.87	59.74
2	4	16	12189	5.34	33.4%	54.17	54.17
3	1	3	26288	2.48	82.6%	21.91	175.25
3	1	6	16572	3.93	65.5%	27.62	110.48
3	2	12	11702	5.57	46.4%	39.00	78.01
3	4	24	9073	7.18	29.9%	60.49	60.49
4	1	4	22926	2.84	71.0%	25.47	203.79
4	1	8	13470	4.84	60.4%	29.93	119.73
4	2	16	10373	6.28	39.2%	46.10	92.20
4	4	32	7607	8.56	26.8%	67.62	67.62
8	1	8	13600	4.79	59.9%	30.22	241.78
8	1	16	9829	6.63	41.4%	43.68	174.74
8	2	32	6916	9.42	29.4%	61.48	122.95
8	4	64	6960	9.36	14.6%	123.73	123.73
Sequential Performance							
1	1	1	65138	1.00	100%	18.09	144.75

Although the optimal choice for this application is 1 node 8 cores, this could only be determined by examining a variety of other cases. Additionally, an examination of these cases give insight into the WRF model's performance if the time available to generate the forecast is decreased or if a higher resolution run is needed (requiring more resources to complete in the desired time frame). The 2 node cases were examined and still decrease the run time. Although efficiency has dropped from the single node performance, the run time has decreased by 1.37 times. Exclusive node core cost has become 1.46 times more, yielding an annual cost of 19,285 core hours. The 3 node cases were examined and still decrease the run time, although efficiency has further dropped. The run time has decreased by 1.84 times from the single node performance and the exclusive node core cost has increased by 1.63 times for an annual cost of 21,534 core hours. The 4 node cases were examined and further decrease the run time with a drop in efficiency. The run time has decreased by 2.20 times from single node performance and the exclusive node core cost has increased by 1.82 times for an annual cost of 24,072 core hours. Lastly, the 8 node cases were examined and a point of diminishing returns was reached. There was a greater improvement in the run time using 2 cores per processor over 4 cores per processor. The 8 node 2 cores per processor case decreased run time by 2.42 times over single node performance and increased the exclusive node core cost by 3.33 for an annual cost of 44,048 core hours.

In addition, cases were examined with a single core on a single processor per node. These were tested to show the code's performance if memory from both processors were available to a single core on a single processor. If the code had been memory bound (and not CPU bound), then it may have been beneficial. As the code was not memory bound, this showed no performance gain. In fact, the exclusive node core cost made this an even less desirable configuration.

The last item to be parallelized was the image generation routines. Images are created after the WRF model has completed with the RIP (which stands for Read/ Interpolate/ Plot) tool set so that they may be posted to the website. This contains sequential codes, but requires multiple runs for each domain and each field (i.e. surface winds, temperature, hourly rainfall accumulation, relative humidity, etc.). The output of RIP for each field is a collection of hourly images in a single NCAR Computer Graphics Metafile (NCGM). This file must be split into separate files for each hour and converted to a useable web format. The process for each field in each domain must be completed in successive order (similar to a pipeline). The average consecutive sequential time to finish all these runs for all domains is 29:17. Since there are 7 domains (not all of equal dimensions) this collection of runs can be run concurrently on the same node. Running the image processing routines for all domains on 1 node of 8 cores returned in 4:47. This is a speedup of ~ 6.2 times. For 7 domains, a linear speedup would yield a 7 times speedup. Because the domains are not of equal sizes the performance was less that linear. Aside from the obvious speed up obtained by running the image processing routines concurrently, if additional fields and domains were added at some future time there now exists a methodology to handle this in a nearly linear fashion.

VII. Future Work

There is additional work that can be done to improve the model's predictive capabilities. Some will help the reliability of the model in producing a forecast in the required time frame, while others will help the accuracy of the model. A list of future work includes:

1. Porting the code to another machine. Having a secondary machine available when the primary machine is unavailable (whether due to maintenance or higher priority users) will better ensure daily operational service.
2. Port to a hybrid GPU-based parallel architecture. CPU intensive portions of the WRF code can take great advantage of GPU [14].
3. Increase horizontal resolution of Hawaiian counties from 2 kilometer to sub-kilometer. This is dependent on inclusion of sub-kilometer terrain data. In addition, more research must be done to investigate the accuracy of model at this resolution. It is not entirely clear how the model will behave at a finer resolution and hence the physics packages used by the model may need to be improved and/or modified.
4. Inclusion of sub kilometer terrain data. Currently the model uses 30-second (~0.9 km) terrain data, which limits it from running with accurate terrain data at sub-kilometer resolutions.
5. Increase the vertical resolution. The current use 55 vertical level is already higher than commonly used, but there is the potential to improve the accuracy of the optical turbulence calculations [15].
6. Extend and validate the forecast from 48 to 72 simulation hours.

VIII. Conclusions

A methodology has been created that will produce high-resolution weather forecasts over the state of Hawaii utilizing the next generation WRF model. This methodology is focused on providing the required forecast in a minimal time so as to still be useful to telescope operators on Haleakala who are trying to determine if future weather conditions are within their operational limits. The web output has been chosen to given telescope operators the necessary fields needed to make operational decisions. This will allow better scheduling and improve the potential efficiency of telescope operations.

Acknowledgments

The authors would like to acknowledge the support received from the Maui High Performance Computing Center.

REFERENCES

[1] Michalakes, J., S. Chen, J. Dudhia, L. Hart, J. Klemp, J. Middlecoff, and W. Skamarock: Development of a Next Generation Regional Weather Research and Forecast Model. Developments in Teracomputing: Proceedings of the Ninth ECMWF Workshop on the Use of High Performance Computing in Meteorology. Eds. Walter Zwieflhofer and Norbert Kreitz. World Scientific, 2001, pp. 269-276.

[2] Michalakes, J., J. Dudhia, D. Gill, T. Henderson, J. Klemp, W. Skamarock, and W. Wang: The Weather Research and Forecast Model: Software Architecture and Performance. Proceedings of the Eleventh ECMWF Workshop on the Use of High Performance Computing in Meteorology. Eds. Walter Zwieflhofer and George Mozdzynski. World Scientific, 2005, pp 156 – 168.

[3] Barker, D. M., W. Huang, Y. R. Guo, and Q. N. Xiao. 2004: A Three-Dimensional (3DVAR) Data Assimilation System For Use With MM5: Implementation and Initial Results. Monthly Weather Review, 132, 897-914.

[4] Chen, Yi-Leng and Feng, J., “Numerical Simulation of Airflow and Cloud Distributions over the Windward Side of the Island of Hawaii. Part I: The Effects of Trade Wind Inversion.” American Meteorological Society, 1117-1134. May 2001.

[5] Chen, Yi-Leng and Feng, J., “Numerical Simulation of Airflow and Cloud Distributions over the Windward Side of the Island of Hawaii. Part II: Nocturnal Flow Regime.” *American Meteorological Society*, 1135-1147. May 2001.

[6] Janjic, Z. I., 1994: The step-mountain eta coordinate model: further developments of the convection, viscous sublayer and turbulence closure schemes. *Mon. Wea. Rev*., 122, 927-945.

[7] Roe, K.P., Stevens, D., “High Resolution Weather Modeling in the State of Hawaii,” *The Eleventh PSU/NCAR Mesoscale Model User’ Workshop*, Boulder, Colorado, 2001.

[8] Roe. K.P., Stevens, D. “One Kilometer Numerical Weather Forecasting to Assist Telescope Operations” *DoD High Performance Computing Modernization Program User Group Conference*. Bellevue, WA, 2003.

[9] Roe. K.P., Waterson, M., “High Resolution Numerical Weather Forecasting to Aid AMOS.” *AMOS Technical Conference*, Wailea, HI, 2003.

[10] Morton, D., O. Nudson, and C. Stephenson, "Benchmarking and Evaluation of the Weather Research and Forecasting (WRF) Model on the Cray XT5" in Cray User Group Proceedings, Atlanta, GA, 04-07 May 2009.

[11] Shainer, G., Liu, T., Michalakes, J., Liberman, J. “Weather Research and Forecast (WRF) Model Performance and Profiling Analysis on Advanced Multi-core HPC Clusters”. *The 10th LCI International Conference on High-Performance Clustered Computing*. Boulder, CO. 2009.

[12] Abdullah Kayi1, Edward Kornkven, Tarek El-Ghazawi1, Samy Al-Bahra, Gregory B. Newby. “Performance Evaluation of Clusters with ccNUMA Nodes - A Case Study”. *The 10th IEEE International Conference on High Performance Computing and Communications*. Dalian, China. 2008.

[13] Tyler Simon, James McGalliard. “Observation and analysis of the multicore performance impact on scientific applications”. Concurrency and Computation: Practice and Experience. 2009. Volume 21 Issue 17, Pages 2213 – 2231.

[14] Michalakes, J. and M. Vachharajani: GPU Acceleration of Numerical Weather Prediction. Parallel Processing Letters Vol. 18 No. 4. World Scientific. Dec. 2008. pp. 531-548.

[15] Ruggiero, F., Roe, K.P., DeBenedictis, D.A. “Comparison of WRF versus MM5 for Optical Turbulence Prediction.” *DoD High Performance Computing Modernization Program User Group Conference*. Williamsburg, VA, 2004.