A preliminary evaluation of high-performance advanced regional eta-coordinate model (H-AREM)

CHENG Yu-Feng, XU You-Ping, LI Li-Juan, and WANG Bin

ABSTRACT
This paper preliminarily evaluates the speedup, scalability, and prediction skill of the high-performance advanced regional eta coordinate model (H-AREM), which is based on several parallel processing methods and decomposition strategies. Results show that the parallel version of the model that is based on a modular parallel framework and a multidimensional domain decomposition strategy performs better overall, e.g. it is faster and more scalable than the version based on a message passing interface and a one-dimensional decomposition strategy. In particular, the scalability of the H-AREM with a resolution of 8 km approaches 8099 cores. Moreover, in the H-AREM, higher resolutions result in more realistic precipitation predictions without remarkable increases in simulation time.

1. Introduction
Regional numerical weather prediction models are among the most important tools for atmospheric research and weather forecasting. The flexible prediction domain allows a regional model to refine its grid in an area of interest while avoiding the need for additional memory and computation time. Therefore, regional models are suitable for simulating and forecasting weather disasters that occur on sub-synoptic scales, such as local strong convection and heavy rain (Dudhia 1999; Michalakes et al. 2001; Xue, Droegemeier, and Wong 2000; Yu 1989). However, when the model's resolution is increased and full physical processes are included, the amount of memory and the computation time required increase exponentially (Barros and Kauranne 1994). Therefore, parallel computation technology becomes essential.

Two parallel programming methods are used in weather forecasting models. The first involves mixing the message passing processes with the integration codes, i.e. using a message passing interface (MPI) and/or an Open Multi-Processing (OpenMP) library to parallelize a serial model; this method is used in the Pennsylvania State University/ National Center for Atmospheric Research (NCAR) mesoscale model (MMS) (Dudhia 1999), the Advanced Regional Prediction System (ARPS) (Xue et al. 1995), and an early version of the Weather Research and Forecasting (WRF) model (Xue et al. 1995), among others. This method requires reconstructing the entire numerical calculation processes; therefore, developing and/or maintaining the code is complex, and the code is poorly reusable. The other method is to build an individual and reusable supporting layer for a parallel algorithm in software by using a parallel framework; parallel communication does not occur in the numerical components to simplify parallel function calling and to implement highly efficient parallel computation. This method is used in the Earth System Modeling Framework of the National Aeronautics and Space Administration (Hill et al. 2004), the latest version WRF model of NCAR (Michalakes et al. 2004) and the Global and Regional Assimilation and Prediction System of the China Meteorological Administration (CMA).
(Wu et al. 2007), among others. This method separates atmospheric science problems from computer science problems to allow the model's developers to focus on their own fields and improve the efficiency of the deployed model. This method is the main trend in parallelizing numerical weather prediction models, and the key point is the parallel framework. J Adaptive Structured Meshes applications Infrastructure (JASMIN) (Mo et al. 2010) is a universal parallel framework that supports the massively parallel computation of structure mesh or unstructured joint structured mesh applications, including fluid dynamics (Cheng, Mo, and Shao 2013), radiation and neutron transportation simulation (Ren, Wei, and Cao 2012), and computational electromagnetics (Cao et al. 2011). It is developed by the Beijing Institute of Applied Physics and Computational Mathematics. JASMIN employs highly modular and hierarchical architecture. Therefore, users could separate the model into a supporting layer for parallel algorithms, a middle layer for universal numerical algorithms, and a top layer for interfaces and specific numerical algorithms, such as those for modeling dynamic and physical processes. It assists the development of parallel code scaling to thousands of processors.

With the development of computation platforms and parallel skills, most models could operate with increasing cores with higher efficiency. WRF uses 768 cores on Dell HPC clustering (Shainer et al. 2009). GRAPES can utilize 2048 cores with MPI acceleration (Wu 2011) and 8192 cores with hybrid MPI and OpenMP acceleration (Jiang 2014). GAMIL (0.25 resolution) achieves 57.5% parallel efficiency using 960 cores (Liu et al. 2014). FAMIL could effectively utilize 1536 and 3456 cores for 12.5 and 6.25 km resolutions, respectively (Zhou et al. 2012). Parallel skills are also becoming increasingly advanced and diversified. Despite the proven ability of MPI and OpenMP, graphics processing units and Many Integrated Core technology have emerged to accelerate numerical weather prediction model, i.e. WRF (Huang et al. 2015; Mielikainen, Huang, and Huang 2013, 2014), ARPS (Whetstone, Limpasuvan, and Larkins 2014), and GRAPES (Wang et al. 2013; Xiao et al. 2013). Rebuilding code is a monumental task for the model given the rapid development of parallel technology. Therefore, modular and hierarchical architecture, which separates atmospheric science problems from computer science problems, is important.

However, as an important domestic mesoscale numerical model for predicting heavy rain, the Advanced Regional Eta-coordinate Model (AREM), which has been used for precipitation forecasting and disastrous weather research in many institutions, encounters bottlenecks in parallel performance. The current parallel version of AREM, which is AREM3.4P (Pu et al. 2008), is based on MPI library. With up to one or two hundred cores, AREM3.4P exhibits good parallel performance, but it is unable to use more cores because of the limitations of the parallel algorithm. Therefore, the current version of AREM cannot complete predictions within a limited period at a higher resolution. To solve this problem, the model's code is rebuilt based on the JASMIN framework, and the high-performance version of AREM is developed and named H-AREM. H-AREM, is highly modular and hierarchical. However, whether H-AREM performs better (i.e. faster and more scalable) than AREM3.4P must be determined.

This paper provides a preliminary evaluation of H-AREM in terms of both parallel and simulation performance. In Section 2, the model and the experiments are described. The results are analyzed in Section 3. In the final section, the conclusions and discussion are presented.

2. Methods

2.1. Model description

AREM, which is based on the framework of the regional eta coordinate model (REM) (Yu 1989), was jointly developed by the State Key Laboratory of Numerical Modeling for Atmospheric Sciences and Geo-physical Fluid Dynamics/Institute of Atmospheric Physics, the Beijing Institute of Applied Meteorology (BIAM), and the Institute of Heavy Rain, CMA, Wuhan. Given its appropriate treatment of complex topography and its reasonable representation of particular features of East Asian weather and climate, AREM simulates heavy rainfall in the East Asian monsoon region well (Gong 2007; Li et al. 2005a, 2005b; Rui et al. 2011a, 2011b; Yu and Xu 2004).

The model employs a water-bearing numerical framework (Xia and Xu 1998; Xu 2009; Xu, Xia, and Qian 1996, 1998) and a two-step shape-preserving advection scheme (Yu 1994a, 1995) with an Arakawa E-grid and η as the vertical coordinate (Yu 1989, 1994b). The physical parameterization schemes used in AREM include a nonlocal planetary boundary layer scheme (Holtslag and Boville 1993; Holtslag and Moeng 1991), an explicit warm-cloud/cold-cloud prediction scheme (Xu 2009), a multi-stratification surface flux parameterization scheme (Zeng, Zhao, and Dickinson 1998), a Betts–Miller moist convection adjustment scheme (Betts 1986), a biosphere-atmosphere transfer scheme (Dickinson, Kennedy, and Henderson-Sellers 1993; Yang et al. 2008), and a Benjamin surface radiation scheme (Benjamin 1986; Benjamin and Carlson 1986), among others.

The previous parallel version of AREM is AREM3.4P. Its parallel processing method is based on the MPI library and employs one-dimensional (1D) longitudinal decomposition.
The new parallel version of AREM is H-AREM. Its parallel processing method is based on the JASMIN framework. The software architecture of H-AREM is presented in Table 1. H-AREM builds up three layers using the JASMIN framework.

The decomposition strategy of H-AREM is based on the patches. Figure 1 depicts a typical decomposition case. The dashes lines stand for the grid cells; the solid lines stand for the patches and physical boundaries. The computational domain is decomposed into 15 patches. In practice, the patch size can be adjusted by setting parameters to fit the cache memory size of CPU. In particular cases, the patches could decompose in one or two dimensions.

To store the data transferred from neighboring patches, each patch extends a specific width called ghost area, i.e. the gray zone of Patch-07 in Figure 1. Communication is carried out by filling or updating the ghost area of each patch before the numerical components. Thus, the communication will not be involved in the numerical computation.

2.2. Experimental design

To evaluate both the computational performance and the forecasting performance, the real case of Typhoon Matmo on 23 July 2014 is simulated; the parallel computation and weather prediction results are presented. The simulation includes all the physical processes in the model, including a CCM3 nonlocal planetary boundary-layer scheme, an MM5 surface radiation scheme, a Zeng surface turbulent flux scheme, and a BIAMC explicit cold-cloud prediction scheme. The initial and boundary conditions are based on T799 analysis data (accurate to 0.1°). The simulation is initialized at 0000 UTC 23 July 2014. The features of two versions of the model used in this study are presented in Table 2.

The combination of AREM and JASMIN framework introduces several advanced parallel algorithms to the model, including optional decomposition, dynamic load balance, and multilayer data structures, all of which benefit the computation performance. To verify the performance, the first group of experiments focuses on the decomposition strategy by comparing AREM3.4P and H-AREM.

The architecture of H-AREM has been redesigned. Thus, H-AREM provides an optional decomposition strategy. Its patch size is controlled by two parameters, namely, largest_patch_size and smallest_patch_size. The domain can be decomposed either in one or two dimensions (2D) by setting the two relevant parameters appropriately in response to the demands of users. In this experiment, H-AREM (1D) adopts a 1D decomposition strategy and ensures that the size of the subdomains are the same as those of AREM3.4P. H-AREM (2D) adopts a 2D decomposition strategy. The domain of the integration extends from 60°E to 140°E and from 10°N to 60°N; the grid has a size of 401 × 501 for a horizontal resolution of 15 km. Keeping the same domain settings ensures that the same calculation is performed by both versions of the model. Each version of the model simulates three model hours, and the time required, according to the wall clock, is recorded.

The three-layer software architecture makes updating the model’s code easy. A high resolution can be implemented by adjusting part of the code in the top layer and resetting the parameters in the middle layer. The high resolution requires additional computations, which is closely related to the parallel performance. To verify this finding, the second group of experiments focuses on speed and scalability by comparing different versions of H-AREM with different resolutions.

The domain considered is the same as in the first group of experiments, but the horizontal resolution is 15 km/8 km with 401 × 501/801 × 1001 grid points. The two different versions of H-AREM simulate 100 time steps using

![Figure 1. Typical decomposition case of computation domain.](image-url)
AREM3.4P is most efficient with 80 cores and takes 186 s. By contrast, H-AREM (1D) and H-AREM (2D) are scalable to 500 and 1500 cores, respectively. They are most efficient with 320 and 1000 cores, respectively, and take 126 and 28 s, respectively.

Furthermore, a comparison between H-AREM (1D) and AREM3.4P shows that the former performs better than the latter for 40 or more cores. For example, for 10 cores, H-AREM (1D) takes 961 s, and AREM3.4P takes 813 s. H-AREM creates an individual supporting layer on the basis of the JASMIN framework to deal with parallel algorithms, thereby increasing the system overhead to a certain extent. But this modularized structure of H-AREM also optimizes parallel communication and decreases the program’s complexity. In addition, when more than 160 cores are present, AREM3.4P is not only unable to further accelerate but also cannot run properly, whereas H-AREM (1D) accelerates the most with 320 cores and runs properly with 500 cores. This finding indicates the influence of the parallel algorithm is more remarkable with more than hundreds of cores.

A comparison between H-AREM (1D) with H-AREM (2D) for the same number of cores shows that H-AREM (2D) always runs faster than the H-AREM (1D) does. This advantage becomes more significant on larger parallel scales. However, H-AREM (1D) cannot accelerate when the number of cores is greater than 320, whereas H-AREM (2D) continues to accelerate until the number of cores is greater than 1000. H-AREM (2D) is much more scalable than H-AREM (1D) is.

The runtime of the different decomposition strategy may be mainly due to the amount of data communication. Table 3 presents the subdomain size of different decomposition strategies (the numbers in parentheses are the result of dealing with the remainder to maintain static load balance).

| Cores | AREM3.4P | H-AREM (1D) | H-AREM (2D) |
|-------|----------|-------------|-------------|
| 10    | 401 × 50 (51) | 401 × 50 (51) | 200 (201) × 100 (101) |
| 20    | 401 × 25 (26) | 401 × 25 (26) | 100 (101) × 100 (101) |
| 40    | 401 × 12 (13) | 401 × 12 (13) | 50 (51) × 100 (101) |
| 80    | 401 × 6 (7) | 401 × 6 (7) | 50 (51) × 50 (51) |
| 160   | 401 × 3 (4) | 401 × 3 (4) | 25 (26) × 50 (51) |
| 320   | 401 × 2 (1) | 401 × 2 (1) | 25 (26) × 25 (26) |
| 500   | 401 × 1 (2) | 401 × 1 (2) | 25 (26) × 25 (26) |
| 1000  | 100 (11) × 20 (21) | 100 (11) × 20 (21) | 20 (21) × 20 (21) |
| 1500  | 100 (11) × 1 (2) | 100 (11) × 1 (2) | 100 (11) × 1 (2) |

The above finding suggests that the 2D decomposition strategy has a higher degree of parallelism than 1D decomposition strategy, which implies that the computational domain can be decomposed into and distributed across more processes. Even when it runs on the same number of thousands of cores, and the speedup of the computation is computed.

The speedup ($S_p$) of the model is defined as

$$S_p = \frac{T_s}{T_p},$$

where $T_s$ is the serial running time of the code, and $T_p$ is the parallel running time of the code for $p$ cores. In this study, the code cannot be run serially because of its memory requirement. Therefore, $T_s$ is replaced by an approximate value that is deduced from the running time for the fewest cores.

### 3. Results

The model’s running times for different parallel processing strategies are shown in Figure 2. As the parallel scale increases, the model’s running time decreases. The parallel algorithm used in AREM3.4P uses at least three grid rows per process. Thus, the scalability is limited. The largest parallel scale supported by AREM3.4P is only 160 cores.
of cores, 2D decomposition produces a more reasonable patch size than 1D decomposition does, which results in a higher acceleration and a shorter running time. Selecting an appropriate decomposition strategy strongly influences the speedup and scalability.

Figure 3 depicts the scalability of H-AREM for different resolutions. The simulation results are obtained in the same computational domain by models with different resolutions, which means that the numbers of computations were different. The speedup of H-AREM (15 km) is shown for the range from 100 cores to 1500 cores. The two largest speedups are 5.12 times for 1000 cores and 5.09 times for 1500 cores. The speedup is no more than 5.09 times. Thus, the results for 1500 more cores are not provided even though these parallel scales are supported by the H-AREM (15 km). The speedup of the H-AREM (8 km) is shown for the range from 100 cores to 8099 cores; its maximum value is 16.35 times for 8099 cores. Additional cores are supported in theory but are not tested because of limited computing resources. A comparison between the results of H-AREM (15 km) and H-AREM (8 km) shows that the acceleration can be extended to larger parallel scales as the number of computations increases. This result occurs because when the resolution is improved, the proportion that can be parallelized increases faster than the proportion that cannot be parallelized in the total number of computations performed by H-AREM. H-AREM’s performance is highly scalable, which suggests that developing higher resolutions is feasible and considering problems with the number of computations is unnecessary.

All the experiments are based on the real case of Typhoon Matmo on 23 July 2014. The simulation time is limited to less than three hours because of the available computational resources. To verify the prediction performance, Figure 4 presents the 24-h accumulated precipitation for 23 to 24 July 2014 for comparison with the results of two versions of

Figure 3. Model speedup or different resolutions: H-AREM at a resolution of 8 km (dashed line) and H-AREM at a resolution of 15 km (solid line).

Figure 4. 24-h accumulated precipitation from 23 to 24 July 2014: observation and forecast (mm/24 h). Initial forecast time: 0000 GMT 23 July 2014.
Notes: Top: observation; Middle: H-AREM (8 km); Bottom: H-AREM (15 km). Rmse: root-mean-square error between observation and forecast. Cor_index: spatial correlation index between observation and forecast.
H-AREM: 8 and 15 km. The observation data from the China Hourly Merged Precipitation Analysis combine observations from automatic weather stations with CMORPH (Shen et al. 2014) on a 0.1° × 0.1° grid (CMPA-Daily).

As shown in the observation (top of Figure 4), the precipitation is mainly distributed along the coast of Fujian province and most of Taiwan, and the intensity of the precipitation center is greater than 250 mm. The results of H-AREM (8 km) are similar in intensity and location to the observation, and the results of H-AREM (15 km) show a smaller range, a greater intensity of the precipitation center on the coast of Fujian province, and a northward shift over Taiwan. For the root-mean-square error between the observation and forecast, the results of H-AREM (8 km) and H-AREM (15 km) are 23.80 and 27.35 mm, respectively. As for the spatial correlation index, the results of H-AREM (8 km) and H-AREM (15 km) are 0.66 and 0.56, respectively. The results demonstrate that H-AREM performs better in this case when it has a higher resolution and that its parallel performance controls the running time well. For example, simulating 24 h takes nearly 530 s with H-AREM (15 km) on 700 cores and nearly 2500 s with H-AREM (8 km) on 1000 cores. By contrast, the previous parallel version of the AREM, which is based on MPI and 1D decomposition, takes nearly 1800 s at a resolution of 15 km and 9000 s at a resolution of 8 km to simulate 24 h. In general, the H-AREM that is based on the JASMIN framework supports high-efficiency, highly scalable parallel computation, which allows it to overcome the bottleneck due to the number of computations required at high resolutions.

4. Discussion and conclusions

In this paper, the performance of H-AREM is verified using two groups of experiments: decomposition strategy and high scalability experiments. All the experiments are based on a simulation of Typhoon Matmo on 23 July 2014, and the predictions of the precipitation accumulated over 24 h at different resolutions are used to evaluate the models’ performances.

On the basis of the wall clock time for three runs, AREM3.4P takes at least 186 s with 80 cores, H-AREM (1D) takes at least 126 s with 320 cores, and H-AREM (2D) takes at least 28 s with 1000 cores. AREM3.4P, AREM (1D), and AREM (2D) are most scalable at fewer than 160, 500, and 1500 cores, respectively. Furthermore, the scalability of the H-AREM is tested; it is 8099 cores for a resolution of 8 km. Results demonstrate that a parallel implementation method based on a modular parallel framework and a multidimensional domain decomposition strategy enables H-AREM to perform better overall, including in terms of speedup and scalability, than the AREM3.4P. As for the simulation performance, higher resolutions result in more realistic precipitation predictions by the H-AREM. Moreover, the increase in running time caused by a high resolution is controlled well by the H-AREM’s parallel performance.

Although the JASMIN framework increases the system overhead, it significantly extends the H-AREM’s scalability from hundreds to thousands of cores and shortens the wall clock running time. Further development and operation are possible for the model because the computational performance bottleneck has been resolved. Further efforts will focus on simulation performance, and more weather forecasting experiments will be conducted using high-resolution version.

We evaluate H-AREM only with MPI in this study, but room for improvement still exists. For example, a hybrid MPI and OpenMP could be used in H-AREM for further acceleration. Updating the algorithm based on the JASMIN framework is convenient, which also provides good support on OpenMP; all that is needed is an adjustment of the supporting layer of the model and without the rebuilding the code of dynamic and physical processes.

Acknowledgements

We would like to thank Rui CHENG of the Beijing Institute of Applied Meteorology and Ye PU of LASG/IAP for help with AREM and parallel computation, and Hong GUO of the Beijing Institute of Applied Physics and Computational Mathematics for the parallel computation test and help with the JASMIN framework.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work is jointly supported by the National Basic Research Program of China (973 Program) [grant number 6131270305]; the Ministry of Water Resources’ special research grant for non-profit public service [grant number 201301062-02]; the National Natural Science Foundation of China [grant number 61572058]; the Strategic Priority Research Program of the Chinese Academy of Sciences [grant number XDA05110304].

References

Barros, S. R., and T. Kauranne. 1994. “On the Parallelization of Global Spectral Weather Models.” Parallel Computing 20 (9): 1335–1356.

Benjamin, S. G. 1986. “Some Effects of Surface Heating and Topography on the Regional Severe Storm Environment.” Part II: Two-Dimensional Idealized Experiments, Monthly Weather Review 114 (2): 303–343.

Benjamin, S. G., and T. N. Carlson. 1986. “Some Effects of Surface Heating and Topography on the Regional Severe Storm Environment.” Part I: Three-Dimensional Simulations, Monthly Weather Review 114 (2): 307–329.
Betts, A. K. 1986. “A New Convective Adjustment Scheme.” Part I: Observational and Theoretical Basis, Quarterly Journal of the Royal Meteorological Society 112 (473): 677–691.

Cao, X. L., Z. Y. Mo, X. Liu, X. W. Xu, and A. Q. Zhang. 2011. “Parallel Implementation of Fast Multipole Method Based on JASMIN.” Science China Information Sciences, 54 (4), 757–766.

Cheng, T. P., Z. Y. Mo, and J. L. Shao. 2013. “Jasmin-based Massive Parallel Computing of Large Scale Groundwater Flow (in Chinese).” Chinese Journal of Computational Physics 30 (3): 317–325.

Dickinson, R. E., P. Kennedy, and A. Henderson-Sellers. 1993. Biosphere-atmosphere Transfer Scheme (Bats) Version 1e as Coupled to the Ncar Community Climate Model, National Center for Atmospheric Research, Climate and Global Dynamics Division, 1.

Gong, Y. 2007. Evaluation and Analysis of the Rainfall Prediction of Arem in Flood Season of 2007 (in Chinese) 26(4), 372–380.

Dudhia, J. 1999. Psu/Ncar Mesoscale Modeling System Tutorial Class Notes and User’s Guide: Mms Modeling System Version 2. Mesoscale and Microscale Meteorology Division, 1.

Hill, C., C. DeLuca, M. Suarez, and A. D. Silva. 2004. “The Architecture of the Earth System Modeling Framework.” Computing in Science & Engineering 6 (1): 18–28.

Holtslag, A., and B. Boville. 1993. “Local versus Nonlocal Boundary-layer Diffusion in a Global Climate Model.” Journal of Climate 6 (10): 1825–1842.

Holtslag, A., and C. Moeng. 1991. “Eddy Diffusivity and Countergradient Transport in the Convective Atmospheric Boundary Layer.” Journal of the Atmospheric Sciences 48 (14): 1690–1698.

Huang, M., J. Mieliikainen, B. Huang, H.-L. A. Huang, and M. D. Goldberg. 2015. “Development of Efficient GPU Parallelization of WRF Yonsei University Planetary Boundary Layer Scheme.” Geoscientific Model Development 8 (9): 2977–2990.

Jiang, Q. G. 2014. “The Hybrid Mpi+Openmp Parallel Method of Grapes-global Model (in Chinese).” Master’s thesis, Beijing.

Li, K., Y. P. Xu, R. C. Yu, and R. Cheng. 2005a. “Comparative Studies of Three Types of Heavy Rainstorms Associated with the Meiuyu Front By Numerical Simulations (in Chinese).” Chinese Journal of Atmospheric Sciences 29 (2): 13.

Li, Y. Y, R. C. Yu, Y. P. Xu, and T. J. Zhou. 2005b. “Arem Simulations of Cloud Features Over Eastern China in February 2001.” Advances in Atmospheric Sciences 22 (2): 260–270.

Liu, L., R. Z. Li, G. W. Yang, B. Wang, L. J. Li, and Y. Pu. 2014. “Improving Parallel Performance of a Finite-difference AGCM on Modern High-performance Computers.” Journal of Atmospheric and Oceanic Technology 31 (10): 2157–2168.

Michalakes, J., S. Chen, J. Duddia, L. Hart, J. Klem, J. Middlecoff and W. Skamarock 2001. “Development of a Next Generation Regional Weather Research and Forecast Model.” Developments in Teracomputing: Proceedings of the Ninth Ecwmf Workshop on the Use of High Performance Computing in Meteorology, Reading, January 1.

Michalakes, J., J. Duddia, D. Gill, T. Henderson, J. Klem, W. Skamarock, and W. Wang 2004. “The Weather Research and Forecast Model: Software Architecture and Performance.” Proceedings of the 11th Ecwmf Workshop on the Use of High Performance Computing in Meteorology, Reading, October 25.

Mieliikainen, J., B. Huang, and H. A. Huang. 2013. “Speeding up the Computation of WRF Double-moment 6-Class Microphysics Scheme with GPU.” Journal of Atmospheric and Oceanic Technology 30 (12): 2896–2906.

Mieliikainen, J. B. Huang, and A. H. Huang. 2014. “Initial Results on Computational Performance of Intel Many Integrated Core (Mic) Architecture: Implementation of the Weather and Research Forecasting (Wrf) Purdue-Im Microphysics Scheme.” High-performance Computing in Remote Sensing IV, Amsterdam, September 22.

Mo, Z. Y., A. Q. Zhang, X. L. Cao, Q. K. Liu, X. W. Xu, H. B. An, W. B. Pei, and P. S. Zhu. 2010. “JASMIN: A Parallel Software Infrastructure for Scientific Computing.” Frontiers of Computer Science in China 4 (4): 480–488.

Pu, Y., B. Wang, Y. P. Xu, R. Cheng, and W. Cheng. 2008. “Application of Mpi Parallel Technique in Arem (in Chinese).” Climatic and Environmental Research 13 (5): 675–680.

Ren, J., J. X. Wei, and X. L. Cao. 2012. “Composition Computation of Radiation Hydrodynamics and Particle Transport Based on Jasmin (in Chinese).” Chinese Journal of Computational Physics 29 (2): 205–212.

Rui, C., Y. Rucong, X. Youping, et al. 2011a. “Impact of Cloud Microphysical Processes on the Simulation of Typhoon Rananim Near Shore.” Part II: Typhoon Intensity and Track, Acta Meteorologica Sinica 25 (4): 456–466.

Rui, C., Y. Rucong, F. Yunfei, et al. 2011b. “Impact of Cloud Microphysical Processes on the Simulation of Typhoon Rananim Near Shore.” Part I: Cloud Structure and Precipitation Features, Acta Meteorologica Sinica 25 (4): 441–455.

Shainer, G., T. Liu, J. Michalakes, J. Liberman, J. Layton, O. Celebioglu, S. A. Schultz, J. Mora, and D. Cownie. 2009. “Weather Research and Forecast (Wrf) Model Perfermance and Profiling Analysis on Advanced Multi-core Hpc Clusters.” The 10th Lci International Conference on High-performance Clustered Computing, Boulder, April 9.

Shen, Y., P. Zhao, Y. Pan, and J. J. Yu. 2014. “A High Spatiotemporal Gauge-satellite Merged Precipitation Analysis over China.” Journal of Geophysical Research Atmospheres 119 (6): 3063–3075.

Wang, Z. W. X. B. Xu Q. Zhao., S. B. He, and Y. P. Zhang. 2013. “Parallel Acceleration and Performance Optimization for Grapes Model Based on Gpu (in Chinese).” Journal of Computer Research and Development 50 (2): 401–411.

Whetstone, B. A., V. Limpasuvan, and B. D. Larkins. 2014. “Gpu Acceleration of the Advanced Regional Prediction System (Arps).” Proceedings of the 2014 Acm Southeast Regional Conference, New York, April 28.

Wu, X. J. 2011. “Study on the Parallel Computing in Grapes High Resolution Numerical Weather Prediction Model (in Chinese).” PHD’s thesis, Changsha.

Wu, X. J., Z. Y. Jin, D. H. Chen, J. Q. Song, and X. S. Yang. 2007. “A Parallel Computing Algorithm and Its Application in New Generation of Numerical Weather Prediction System (grapes) (in Chinese).” Journal of Computer Research and Development 44 (3): 510–515.

Xia, D. Q., and Y. P. Xu. 1998. “The Water-bearing Numerical Model and Its Operational Forecasting Experiments Part I: The Water-bearing Numerical Model.” Advances in Atmospheric Sciences 15 (2): 221–232.

Xiao, H. D., J. Sun, X. F. Bian, and Z. J. Dai. 2013. “Gpu Acceleration of the WSM6 Cloud Microphysics Scheme in GRAPES Model.” Computers & Geosciences 59 (1): 156–162.

Xu, Y. P. 2009. Development and Estimation of Water-bearing Numerical Framework for Arem (in Chinese), Beijing, 1–149.

Xu, Y. P., D. Q. Xia, and Y. Y. Qian. 1996. “The Real-time Forecasting Experiments with the Water-bearing Numerical Model (in Chinese).” Quarterly Journal of Applied Meteorology 7 (3): 257–266.
Xu, Y. P., D. Q. Xia, and Y. Y. Qian. 1998. “The Water-bearing Numerical Model and Its Operational Forecasting Experiments Part II: The Operational Forecasting Experiments.” *Advances in Atmospheric Sciences* 15 (3): 321–336.

Xue, M., K. K. Droegemeier, V. Wong, A. Shapiro, K. Brewster, F. Carr, D. Weber, Y. Liu, and D. Wang 1995. *Advanced Regional Prediction System (Arps) Version 4.0 User’s Guide.* Center for Analysis and Prediction of Storms, 1.

Xue, M., K. K. Droegemeier, and V. Wong. 2000. “The Advanced Regional Prediction System (Arps)–a Multi-scale Nonhydrostatic Atmospheric Simulation and Prediction Model.” *Part I: Model Dynamics and Verification, Meteorology and Atmospheric Physics* 75 (3): 161–193.

Yang, L., Y. P. Xu, R. Cheng, K. Li, and W. Zhao. 2008. “Improvement of the Land Process in Arem Model and Its Experiments (in Chinese).” *Torrential Rain and Disasters* 27 (1): 3–10.

Yu, R. C. 1989. “Design of Abrupt Terrain Finite Regional Numerical Prediction Model (in Chinese),” *Chinese Journal of Atmospheric Sciences* 13 (2): 11.

Yu, R. C. 1994a. “A Two-step Shape-preserving Advection Scheme.” *Advances in Atmospheric Sciences* 11 (4): 479–490.

Yu, R. C. 1994b. “A Test for Numerical Weather Prediction of Real-time for China Flood Seasou Precipitation in 1993 by a Regional η-coordinate Model (in Chinese).” *Chinese Journal of Atmospheric Sciences* 18 (3): 284–292.

Yu, R. C. 1995. “Application of a Shape-preserving Advection Scheme to the Moisture Equation in an E-grid Regional Forecast Model.” *Advances in Atmospheric Sciences* 12 (1): 13–19.

Yu, R., and Y. P. Xu. 2004. “Arem and Its Simulations on the Daily Rainfall in Summer in 2003 (in Chinese).” *Acta Meteorologica Sinica* 62 (6): 715–724.

Zeng, X. B., M. Zhao, and R. E. Dickinson. 1998. “Intercomparison of Bulk Aerodynamic Algorithms for the Computation of Sea Surface Fluxes Using TOGA COARE and TAO Data.” *Journal of Climate* 11 (10): 2628–2644.

Zhou, L. J., Y. M. Liu, Q. Bao, H. Y. Yu and G. X. Wu. 2012. “Computational Performance of the High-resolution Atmospheric Model Family.” *Atmospheric and Oceanic Science Letters* 5 (5): 355–359.