Petascale Computing: Algorithms and Applications

To be published by Taylor & Francis Group, LLC
2007

Chapter 18

Numerical Prediction of High-Impact Local Weather: A Driver for Petascale Computing

Ming Xue and Kelvin K. Droegemeier
Center for Analysis and Prediction of Storms and School of Meteorology, University of Oklahoma, Norman OK

Daniel Weber
Center for Analysis and Prediction of Storms, University of Oklahoma, Norman OK

18.1 Introduction 1
18.2 Computational Methodology and Tools 4
18.2.1 Community Weather Prediction Models 4
18.2.2 Memory and Performance Issues Associated with Petascale Systems 5
18.2.3 Distributed-memory Parallelization and Message Passing 6
18.2.4. Load Balancing 9
18.2.5. Timing and Scalability 9
18.2.6 Other Essential Components of NWP Systems 10
18.2.7 Additional Issues 12
18.3 Example NWP Results 12
18.3.1 Storm-Scale Weather Prediction 12
18.3.2 Very High-Resolution Tornado Simulation 13
18.3.3 The Prediction of an Observed Supercell Tornado 14
18.4 Numerical Weather Prediction Challenges and Requirements 16
18.5. Summary 17
18.6. Acknowledgement 17
References 17

18.1 Introduction

The so-called mesoscale and convective scale weather events, including floods, tornadoes, hail, strong winds, lightning, hurricanes and winter storms, cause hundreds of deaths and average annual economic losses greater than $13 billion in the United States each year (Pielke 2002; Droegemeier et al. 2005). Although the benefit of mitigating the impacts of such events on the economy and society is obvious, our ability to do so is seriously constrained by the available computational resources which are currently far from sufficient to allow for explicit realtime numerical prediction of these hazardous weather events at sufficiently high spatial resolutions or small enough grid spacings.

Operational computer-based weather forecasting, or numerical weather prediction (NWP), began in the late 1910's with a visionary treatise by L.F. Richardson (Richardson

1922). The first practical experiments, carried out on the ENIAC (electronic numerical integrator and computer) some three decades later, established the basis for what continues to be the foundation of weather forecasting. An NWP model solves numerically the equations governing relevant atmospheric processes including fluid dynamics for air motion, thermodynamic processes, and thermal energy, moisture and related phase changes. Physical processes related to long-wave and short-wave radiation, heat and momentum exchanges with the land and ocean surfaces, cloud processes and their interaction with radiation, and turbulence processes often cannot be explicitly represented because of insufficient spatial and temporal resolution. In such cases, these processes are 'parameterized', i.e., treated in a simplified form by making them dependent upon those quantities which the model *can* explicitly resolve. Parameterization schemes are, however, often empirical or semi-empirical; they are the largest source of uncertainty and error in NWP models, in addition to resolution-related truncation errors.

Operational NWP has always been constrained by available computing power. The European Center for Medium-Range Weather Forecasting (ECMWF) operates the world's highest-resolution global NWP model with an effective horizontal resolution[1] of approximately 25 km. The model is based on a spherical harmonic representation of the governing equations, triangularly truncated at total wave number of 799 in the longitudinal direction with 91 levels in the vertical. The daily forecasts extend to 10 days and are initialized using the four-dimensional variational (4DVAR) data assimilation method (Rabier et al. 2000). At the same time, a 51-member Ensemble Prediction System (EPS) is also run daily at a reduced 40 km grid spacing. The EPS provides probabilistic information that seeks to quantify how uncertainty in model initial conditions can lead to differing solutions. Such information is very important for decision making.

The global deterministic (single high-resolution) prediction model operated by the U.S. National Weather Service currently has 382 spectral wave numbers and 64 vertical levels, while its probabilistic ensemble prediction system contains 14 members and operates at the spectral truncation of 126. Forecasts are produced four times a day. To obtain higher resolution over North America, regional deterministic and ensemble forecasts also are produced at 12 and 40 km horizontal grid spacings, respectively.

Even at 12 km horizontal grid spacing, important weather systems that are directly responsible for meteorological hazards including thunderstorms, heavy precipitation and tornadoes can not be directly resolved because of their small sizes. For individual storm cells, horizontal grid resolutions of at least 1 km grid are generally believed to be necessary (e.g., Xue et al. 2003), while even higher resolutions are needed to resolve less organized storms and the internal circulations within the storm cells. Recent studies have also shown that to resolve the inner wall structure of hurricanes and to capture hurricane eye wall replacement cycles that are important for intensity forecasts, 1-2 km resolution is necessary (Chen and Tenerelli 2006; Houze et al. 2006). Furthermore, because the smallest scales in unstable convective flows tend to grow the fastest, the resolution of convective structures will always benefit from increased spatial resolutions (e.g., Bryan et al. 2003), though the extent to which super fine resolution is necessary for non-tornadic

[1] The term "resolution" is used here in a general manner to indicate the ability of a model to resolve atmospheric features of certain spatial scales. In gridpoint models, the term "grid spacing" is more appropriate whereas in spectral or Galerkin models, "spectral truncation" more appropriately describes the intended meaning.

storms remains to be established. To predict one of nature's most violent phenomena, the tornado, resolutions of few tens of meters are required (Xue and Hu 2007).

Because most NWP models use explicit time-integration schemes with a time step size limited by the CFL (Courant-Friedrichs-Lewy) linear stability criterion, the allowable time step size is proportional to the effective grid spacing. Thus, a doubling in 3D spatial resolution, and the requisite halving of the time step, requires a factor of 2^4=16 increase in processing power and a factor of 8 increase in the memory. Data I/O volume is proportional to memory usage or grid size, and the frequency of desired model output tends to increase with the number of time steps needed to complete the forecast.

In practice, the vertical resolution does not need to be increased as much because it is already relatively high compared to the horizontal resolution. The time step size is currently constrained more by the vertical grid spacing than the horizontal one because of the relatively small vertical grid spacing therefore the time step size often does not have to decreased by a factor of two when the horizontal resolution doubles. However, physics parameterizations usually increase in complexity as the resolution increases. Taking these factors into account, a factor of 8 increase in the processing power when the horizontal resolution doubles is a good estimate. Therefore, to increase the operational North American model from its current 12 km resolution to 1 km resolution would require a factor of 12^3=1728 or nearly a factor of two thousand increase in raw computing power. It is estimated that a 30-hour continental-U.S.-scale severe thunderstorm forecast using a state-of-the-art prediction model and 1-km grid spacing would require a total of 3×10^{11} floating point calculations. Using 3000 of today's processors, this forecast will take 70 hours to complete (Weber and Neeman 2006) while for operational forecast, this needs to be done within about one hour. This assumes that the code runs at 10% of peak performance of the supercomputer and there are 10,000 floating-point calculations per grid point per time step. To operate a global 1-km resolution model will be an even greater challenge that is beyond the petascale, but such models are necessary to explicitly resolve convective storms and capture the mutual interaction of such events with short- and long-term climate. Furthermore, the need to run high-resolution ensemble forecasts will require a factor of ~100 increase in the computing power, assuming the ~100 ensemble members also have ~1 km grid spacing.

The development of new high-resolution nonhydrostatic[2] models, coupled with continued rapid increases in computing power, are making the explicit prediction of convective systems, including individual thunderstorms, a reality. Advanced remote sensing platforms, such as the operational U.S. WSR-88D (Weather Surveillance Radar – 1988 Doppler) weather radar network, provide 3D volumetric observations at fine scales for initializing convection-resolving models. Unfortunately, only one dimension of the wind, in the radial direction of radar electromagnetic wave radiation, is observed, along with a measure of precipitation intensity in terms of the power of the electromagnetic waves reflected by the precipitating particles (rain drops and ice particles). The latter is called radar reflectivity. From time series volumes, or radar scan volumes, of these quantities, along with other available observations and specified constraints, one must infer the complete state of the atmosphere.

[2] In the hydrostatic equations of fluid dynamics, vertical accelerations are assumed to be small. This approximation is valid for large-scale flows, where the atmosphere is shallow compared to its lateral extent. In thunderstorms, vertical accelerations are quite large and thus the non-hydrostatic equations must be used.

In this chapter, we address the computational needs of convection-resolving NWP, which refers to predictions that capture the most energetically relevant features of storms and storm systems ranging from organized mesoscale convective systems down to individual convective cells.

18.2 Computational Methodology and Tools

With upcoming petascale computing systems, routine use of kilometer-scale resolutions covering continent-sized computational domains, with even higher-resolution nests over subdomains, will be possible in both research and operations. Accurate characterization of convective systems is important not only for storm-scale NWP, but also for properly representing scale interactions and the statistical properties of convection in long-duration climate models. However, one cannot overlook that models are not perfect and thus even with advanced assimilation methods and excellent observations, model error needs to be minimized. One of the simplest methods is simply to increase model resolution, and for this reason, sub-kilometer grid spacing may be required during radar data assimilation cycles and for short-range convective storm forecasting.

18.2.1 Community Weather Prediction Models

Two of the community models used most frequently for storm-scale research and experimental forecasting are the Advanced Regional Prediction System (ARPS, Xue et al. 1995; Xue et al. 2000; Xue et al. 2001; Xue et al. 2003) and the Weather Research and Forecast (WRF) model (Michalakes et al. 2004; Skamarock et al. 2005). Both were designed to be scalable (e.g., Johnson et al. 1994; Droegemeier et al. 1995; Sathye et al. 1995; Sathye et al. 1996; Sathye et al. 1997; Michalakes et al. 2004) and have been run on numerous computing platforms. Owing to the variety of architectures available at the time of their design, and because of their research orientation and the need to serve a large use base, these systems are not specifically optimized for any particular platform.

Both ARPS and WRF solve a fully compressible system of equations, and both utilize finite difference numerical techniques and regular computational grids. The ARPS uses a generalized terrain-following curvilinear coordinate based on geometric height, and its horizontal grid is rectangular in map projection space but horizontally non-uniform in physical Earth coordinates (Xue et al. 2000; Xue et al. 2001). The same is true for the WRF model except that it uses a mass-based vertical coordinate (Skamarock et al. 2005) that is close to the hydrostatic pressure-based sigma-coordinate used in large-scale NWP models. However, the WRF does not invoke the hydrostatic assumption in the vertical so that a prognostic equation for the vertical equation of motion is solved.

Both ARPS and WRF employ the split-explicit approach of time integration (Skamarock and Klemp 1994) in which the fast acoustic modes[3] are integrated using a small time step while the terms responsible for slower processes, including advection, diffusion, gravity wave modes and physical parameterizations, are integrated using a

[3] Acoustic modes have no physical importance in the atmosphere, except possible in the most intense tornadoes where the Mach number could approach 0.5. They are contained in the compressible systems of equations, however, and affect the stability of explicit time integration schemes.

large time step. For most applications, the vertical grid spacing, especially that near the ground, is much smaller than the horizontal spacing. Therefore, an explicit integration of the vertical acoustic modes would impose a severe restriction on the small time step size. For this reason, both models use an implicit integration scheme in the vertical for terms responsible for vertically propagating acoustic waves. A solver for tri-diagonal systems of equations is used by the implicit scheme.

18.2.2 Memory and Performance Issues Associated with Petascale Systems

The upcoming petascale computing systems are expected to be comprised of hundreds of thousands of processor cores. These systems will rely on multi-core technology and in most cases will contain cache sizes similar to existing technology (< 10Mb). Most of today's supercomputers make use of scalar-based processor technology in a massively parallel configuration to achieve terascale performance. Such individual processors are capable of billions of floating point calculations per second but most applications, in particular weather prediction models and large CFD (computational fluid dynamics) codes, cannot fully realize the hardware potential, largely due to the fact that the memory storage hierarchy is not tuned to the processor clock rate. For example, the 3.8 GHz Pentium 4 CPU has a theoretical rating of 7.6 GFLOPs. To achieve this performance, the processor needs to be fed with data at a rate of 182.4 GB per second. The current RAM and memory bus (e.g., the 800 MHz Front Side Bus) can only move data at a theoretical peak of 6.4 GB per second, a factor of 28.5 slower than what is needed to keep the processor fully fed. This memory access – data supply mismatch will intensify with the proliferation of multi-core processors. To avoid severe penalties due to slow memory bus, the efficiency of the fast cache utilization has to be significantly improved, and to do so usually requires significant efforts at the level of application software design.

Relatively few supercomputers of today and of the near future use vector processors with fast memory technology due to economical reasons. An evaluation of the ARPS on scalar and vector-based computers indicates that the nested do-loop structures were able to realize a significant percentage of the peak performance of vector platforms, but on commodity-processor-based platforms the utilization efficiency is typically only 5-15% (Fig. 18.1) (Weber and Neeman 2006). The primary difference lies with the memory and memory access speeds. Since weather forecast models are largely memory bound, they contain far more loads/stores than computations and, as currently written, do not reuse in-cache data efficiently. One approach, called *supernoding* (Irigoin and Triolet 1989) or tiling, can reduce memory access overhead by structuring the computations so as to reuse data from the high level cache. It holds promise for increasing the scalar processing efficiency for weather models up to 30-40%. Tiling involves the further splitting of the original decomposed subdomains on each processor into smaller regions so that all of their data used in a series of calculations fit into the Level 2 and/or Level 3 cache. This approach allows for the reuse of data residing in the much faster cache memory and reduces processor wait states while accessing the main memory. Tiling usually requires changing loop bounds in order to perform more calculations on a sub-domain and it is possible to specify the tiling parameters at runtime to match the cache size. Tiling has been implemented into a research version of the ARPS and shows approximately a 20%

improvement in performance for the small time step solver (Weber and Neeman 2006). Recognizing that the scalability of software on a massively parallel computer is largely tied to the parallel processing capability, the tiling concept must be used in conjunction with efficient message passing techniques to realize as much of the peak performance as possible.

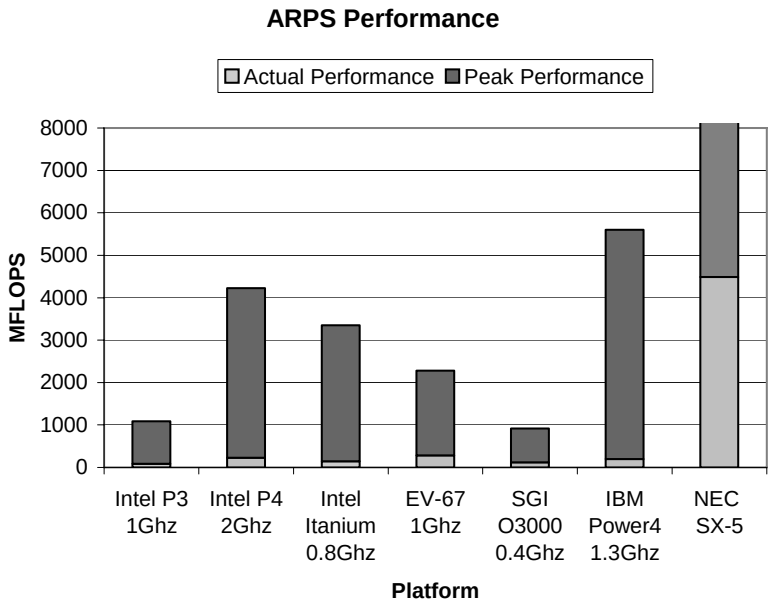

Fig. 18.1. ARPS MFLOP ratings on a variety of computing platforms. Mesh sizes are the same for all but the NEC-SX, which employed a larger number of grid points and with radiation physics turned off (Weber and Neeman 2006).

18.2.3 Distributed-memory Parallelization and Message Passing

Because of the use of explicit finite difference schemes in the horizontal, it is relatively straightforward to use 2D domain decomposition for the ARPS and WRF on distributed memory platforms. No domain decomposition is done in the vertical direction because of the implicit solver, and because a number of physical parameterization schemes, including those involving radiation and cumulus convection, are column based, i.e., their algorithms have vertical dependencies. MPI is the standard strategy for inter-domain communication, while mixed distributed and shared memory parallelization with MPI and OpenMP, respectively, are supported by WRF.

Domain decomposition involves assigning subdomains of the full computational grid to separate processors and solving all prognostic and/or diagnostic equations for a given subdomain on a given processor; no global information is required at any particular grid point and inter-processor communications are required only at the boundaries of the subdomains (Fig. 18.2). The outer border data resident in the local memories of adjacent processors are supplied by passing messages between processors. As discussed earlier, grid-point-based meteorological models usually employ two-dimensional domain

decomposition because of common column-based numerics and physical parameterizations that have global dependencies in the vertical.

In the ARPS, even though the model contains forth-order advection and diffusion options that are commonly used, only one extra zone of grid points is defined outside the non-overlapping subdomain. This zone contains values from the neighboring processors when this subdomain is not at the edge of the physical boundary. Such a zone is often referred to as the 'fake zone'. With a leapfrog time integration scheme used in the large time steps, advection and diffusion terms are evaluated once every time step but their calculations involve 5 grid points in each direction. This is illustrated for the forth-order computational diffusion term (Xue 2000) in the x-direction, $-K\,\partial^4\phi/\partial x^4$, where K is the diffusion coefficient. In standard centered difference form this term becomes

$$-K\delta_{xxxx}\phi \equiv -K\delta_{xx}\left[\delta_{xx}\phi\right] \equiv -K\left(\phi_{i-2} - 4\phi_{i-1} + 6\phi_i - 4\phi_{i+1} + \phi_{i+2}\right)/\left(\Delta x\right)^4, \tag{1}$$

where we define the finite difference operator

$$\delta_{xx}\phi \equiv \left(\phi_{i-1} - 2\phi_i + \phi_{i+1}\right)/\left(\Delta x\right)^2, \tag{2}$$

and i is the grid point index.

For calculating this term on the left boundary of the subdomain, values at *i-1* and *i-2*, which reside on the processor to the left, are needed. However, the ARPS has only one fake zone to store the *i-1* value. This problem is solved by breaking the calculations into two steps. In the first step, the term $\delta_{xx}\phi$ is evaluated at each interior grid point and its value in the fake zone is then obtained from neighboring processors via MPI. In the second step, the finite difference operation is applied to $\delta_{xx}\phi$, according to

$$-K\delta_{xxxx}\phi \equiv -K\delta_{xx}\left[\delta_{xx}\phi\right] \equiv -K\left(\left[\delta_{xx}\phi\right]_{i-1} - 2\left[\delta_{xx}\phi\right]_i + \left[\delta_{xx}\phi\right]_{i+1}\right)/\left(\Delta x\right)^4. \tag{3}$$

After the entire diffusion term is calculated at interior points, it is used to update the prognostic variable, ϕ. Usually, the update of fake zone values of individual terms in the prognostic equation, including this diffusion term, is not necessary. The update of the prognostic variable is necessary only after completion of one time step of the integration. However, for the reason noted above, one additional set of messages from the neighboring processors is needed to complete the calculation of forth-order horizontal diffusion and advection terms. An alternative is to define more fake zones and fill them with values from neighboring processors. In this case, the amount of data communicated is about the same, but the number of associated message passing calls is reduced. In the case of even higher-order diffusion and/or advection schemes, such as the 6th-order diffusion scheme recommended by Xue (2000) and the 5th- and 6th-order advection schemes commonly used in the WRF, more communications are needed. Furthermore, the WRF model commonly uses the 3rd-order Rouge-Kutta time integration scheme, which involves three evaluations of the advection terms during each large time step

(Wicker and Skamarock 2002). As a result, the associated MPI communications are tippled. Fortunately, this time integration scheme allows for a larger large time step size.

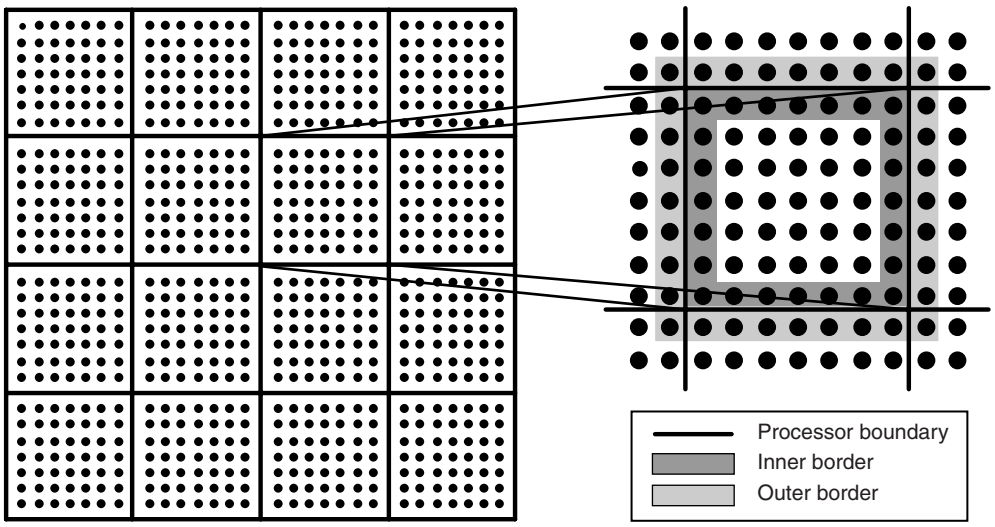

Fig. 18.2. Two-dimensional domain decomposition of the ARPS grid. Each square in the upper panel corresponds to a single processor with its own memory space. The lower panel details a single processor. Grid points having a white background are updated without communication, while those in dark stippling require information from neighboring processors. To avoid communication in the latter, data from the outer border of neighboring processors (light stippling) are stored locally.

An issue unique to the split-explicit time integration scheme is the need to exchange boundary values within small time steps, which incurs additional MPI communication costs. Most of the time, message sizes are relatively small and thus communication latency is a larger issue than bandwidth. Because boundary zone values of dependent variables cannot be communicated until the time step integration is completed, frequent massage passing can incur a significant cost. One possible solution is to sacrifice memory and CPU processing in favor of fewer but larger messages. This can be done by defining a much wider fake zone for each subdomain. For example, if 5 small time steps are needed in each large time step, one can define a fake zone that is five grid point wide instead of only one. These fake zone points then can be populated with values from neighboring processors at the beginning of each large time step through a single message passing call for each boundary. The small time step integration then will start from the larger expanded subdomain, and decrease by one grid zone at each boundary after each small step integration. This way no additional boundary communication is required throughout the 5 small steps of integration. This strategy has been tested with simple codes but is yet to be attempted with the full ARPS or WRF codes. The actual benefit will depend on the relative performance of the CPU versus network, and for the network also on the bandwidth and latency performance ratio. The subdomain size and the large-small time step ratio also influence the effectiveness of this strategy. To further reduce communication overhead, more sophisticated techniques, such as asynchronous communications and message overlap or hiding, may be exploited. Computations,

performed between message passing points, are used to 'hide' the network processes, through operation overlapping. To fully utilize the tiling and message hiding techniques, it is best that the software is designed from the beginning to accommodate them; otherwise, the 'retrofitting' efforts will be very significant, especially for software that contains a large number of loops with hardcoded loop bounds. Most applications, including the ARPS and WRF, will need significant restructuring to take full advantage of these techniques.

18.2.4. Load Balancing

Even though all processors run the same code with the ARPS and WRF, domain decomposition is subject to load imbalances when the computational load is data dependent. This is most often true for spatially-intermittent physical processes such as condensation, radiation, and turbulence in atmospheric models. Because atmospheric processes occur non-uniformly within the computational domain, e.g., active thunderstorms may occur within only a few subdomains of the decomposed domain, the load imbalance across processors can be significant. For subdomains that contain active thunderstorms, some 20-30% additional computational time may be needed. Most implementations of atmospheric prediction models do not perform dynamic load balancing, however, because of the complexity of the associated algorithms and because of the communication overhead associated with moving large blocks of data across processors.

18.2.5. Timing and Scalability

The scalability of the WRF on several large parallel systems is shown here for a relatively simple thunderstorm simulation case. The size of the subdomain on each processor is held fixed at $61 \times 33 \times 51$ points as the number of processors is increased, i.e., the global domain size increases linearly with the number of processors. When 1000 processors are used, the entire model contains approximately 92 million grid points. With the WRF and ARPS using some 150 3-D arrays, the total memory usage is approximately 60 GB for this problem.

Fig. 18.3 shows the timings of the WRF tests on the Pittsburgh Supercomputing Center (PSC) Cray XT3 and Terascale Computing System (TCS), and the Datastar at the San Diego Supercomputing Center (SDSC). The PSC Cray XT3 system contains 2068 compute nodes linked by a custom-designed interconnect, and each node has two 2.6 GHz AMD Opteron processors with 2 GB of shared memory. The TCS consists of 764 Compaq ES45 Alphersevrver nodes with four 1-GHz Alpha processors and 4 GB of shared memory. The nodes are interconnected using a Quadrics network. The SDSC Datastar has 272 IBM P655+ nodes with eight 1.5 or 1.7 GHz CPUs on each node. It is clear from the plot that the scaling performance[4] is reasonable for processor counts ranging from tens to a couple of hundreds, but it becomes poor for more than 256

[4] Ideally, the run time should remain constant as the number of processors is increased (i.e., a horizontal line). Any deviation is due to communication/memory access overhead and load imbalances.

processors for the Datastar and XT3. The scalability deterioration is particularly severe on the XT3 for more than 512 processors. Interestingly, the scalability after 512 processors is excellent on PSC TCS, with the curve remaining essentially flat, indicating perfect scaling when processor the count increases from 512 to 1024.

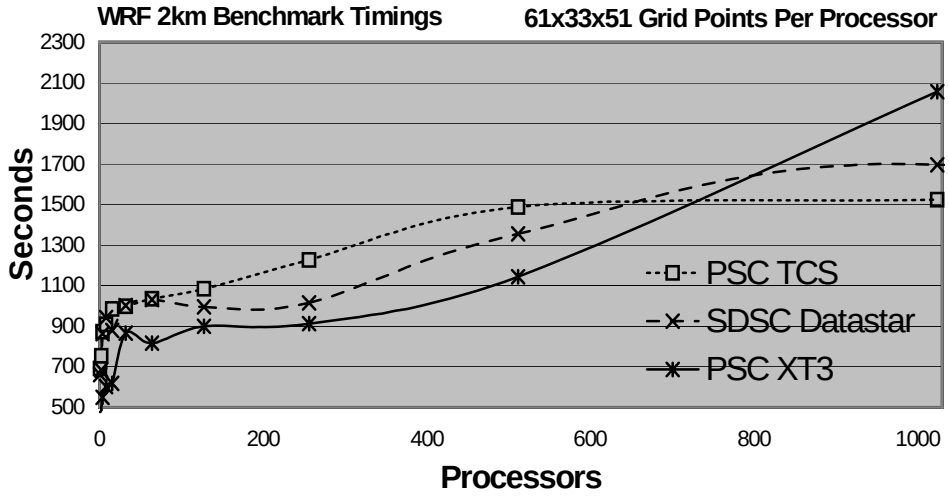

Fig. 18.3. Performance of the WRF model on various TeraGrid computing platforms. The WRF is executed using 61x33x51 grid points per processor such that the overall grid domain increases proportionally as the number of processors increase. For a perfect system, the run time should not increase as processors are added, resulting in a line of zero slope.

For all three systems, rather poor scaling performance occurs when the processor counts increase from one to a few tens, as indicated by the steep slopes of the timing curves for a small number of processors. In all cases, the performance is best when running a single subdomain on the single node, for which the code has exclusive access to the CPU-memory channel and all levels of CPU cache. When every processor on a given node is allocated a subdomain, the memory access contention through the shared memory bus degrades performance, and when the program spans multiple nodes, the MPI communications cause further degradation. Except for PSC TCS, which has slower processors but a faster network, the other two systems clearly do not provide the needed scalability for larger numbers of processors. For future petascale systems having orders of magnitude larger numbers of processor cores, significantly faster network interconnects as well as aggressive algorithm optimization and/or redesign will be needed to achieve useful performance. The expected use of CPUs with dozens of processor cores that share the same path to memory will make the scaling issue even more challenging.

18.2.6 Other Essential Components of NWP Systems

An equally important component of NWP systems is the data assimilation system, which optimally combines new observations with a previous model forecast to arrive at a best estimate of the state of the atmosphere that is also dynamically compatible with the model's equations. The four dimensional variational (4DVAR) and ensemble-Kalman filter (EnKF) techniques are the leading methods now available but are also

computationally very expensive (Kalnay 2002). 4DVAR obtains an initial condition for the prediction model by minimizing a cost function that measures the discrepancy between observations collected within a defined time window (called the assimilation window) and a model forecast made within the same window. The process involves setting to zero the gradient of the cost function with respect to control variables, usually the initial conditions, at the start of the assimilation window. To do so, the adjoint of the prediction model (mathematically, the adjoint is the transpose of the linear tangent version of the original nonlinear prediction model, see, Kalnay 2002) is integrated 'backward' in time and an optimization algorithm is used to adjust the control variables using the gradient information. Such an iterative adjustment procedure usually has to be repeated 50 to 100 times to find the cost function minimum, and for problems with larger degrees of freedom, stronger nonlinearity and/or more observations, the iteration count is correspondingly larger. The adjoint model integration is usually 2-3 times of the cost of the forward model integration, and domain decomposition strategies appropriate for the forward model usually apply. Because of the high computational cost of 4DVAR, operational implementations usually use coarser resolution for the iterative minimization and add the resulting information is added to the high-resolution prior estimate to obtain an initial condition for the high-resolution forecast.

The development and maintenance of an adjoint code are extremely labor intensive though mostly straightforward. 4DVAR is known to exhibit difficulty with high nonlinearity in the prediction model and/or in the operators used to convert model dependent variables (e.g., rain mass) to quantities that are observed (e.g., radar reflectivity). A promising alternative is the ensemble Kalman filter (Evensen 2003; Evensen 2006) whose performance is comparable to 4DVAR. EnKF has the additional advantage of being able to explicitly evolve the background error covariance, which is used to relate the error of one quantity, such as temperature to the error of another, such as wind, and provide a natural set of initial conditions for ensemble forecasts. The probabilistic information provided by ensemble forecasting systems that include tens to hundreds of forecast members has become an essential part of operational NWP (Kalnay 2002).

Efficient implementation of the EnKF analysis algorithms on distributed memory systems is non-trivial. With the commonly used observation-based ensemble filter algorithms, the observations need to be sorted into batches, with those in the same batch not influencing the same grid point (Keppenne 2000; Keppenne and Rienecker 2002). Such implementations are rather restrictive and may not work well for a large number of processors and/or when spatially inhomogeneous data have different spatial influence ranges. A variant of the EnKF, the local ensemble transport Kalman filter (LETKF) (Hunt et al. 2005), is more amenable to parallelization because of the use of independent local analysis subdomains. This method does not, however, ensure processor independence of the results, although it appears to be the most viable approach to date for parallelization. Finally, all ensemble-based assimilation algorithms require the covariance calculations the model state variables from all ensemble members that are within a certain radius of an observation or within a local analysis domain. This requires global transpose operations when individual ensemble members are distributed to different processors. The cost of moving very large volumes of 3D gridded data among processors can be very high. Further, owing to the non-uniform spatial distribution of observational

data, load balancing can be a major issue. For example, ground-based operational weather radar networks provide the largest volumes of weather observations yet they are only available over land and the most useful data are in precipitation regions. Orbiting weather satellites provide data beneath their flight paths, creating non-uniform spatial distributions for any given time.

Other essential components of NWP systems include data quality control and preprocessing, data post-processing and visualization, all of which need to scale well for the much larger prediction problems requiring petascale systems.

18.2.7 Additional Issues

In addition to processor speed, core architecture and memory bandwidth, I/O is a major issue for storm-scale prediction using petascale system. As the simulation domains approach sizes of order 10^{12} points, the magnitude of the output produced will be enormous, requiring very high performance parallel file systems and distributed parallel post-processing software for analysis, mining and visualization. Parallel I/O strategies are needed where the subdomains in a simulation are stored separately instead of being gathered into full three-dimensional arrays. However, most analysis and visualization software assumes access to full three-dimensional arrays. Such software will have to be adapted to utilize the subdomains directly. In fact, the ARPS graphics plotting software, ARPSPLT, reads ARPS output stored in subdomains and the software itself supports MPI. In addition, very high-resolution graphical displays will be needed for display at scales of one pixel per grid point. Interactive visualization software with parallel computational and rendering backends also will be essential. To aid in analyzing nested-grid simulations, visualization methods for viewing simultaneously all nested grids within a simulation will be needed (WRF-RAB 2006).

As petascale systems are expected to contain hundreds of thousands of processors, the likelihood of node failure increases exponentially. Although hardware and operating systems, and MPI implementations, should account for most of such failures, complete dependency on the system is idealistic and NWP models will require built-in fault tolerance. This includes enhancing model codes to take advantage of the fault tolerance in the system software as well as in MPI implementations.

18.3 Example NWP Results

18.3.1 Storm-Scale Weather Prediction

For the purpose of demonstrating and evaluating convective storm-scale prediction capabilities in a quasi-operational setting, the Center for Analysis and Prediction of Storms at the University of Oklahoma produced daily 30-hour forecasts using the WRF model at an unprecedented 2-km resolution, over 2/3 of the continental US, during the spring 2005 southern Great Plains storm season. The project addressed key scientific issues that are vital for forecasting severe weather, including the development and movement of thunderstorm complexes, quantitative precipitation forecasting, and convective initiation. Fig. 18.4 presents the 24-hour model forecast radar reflectivity (proportional to precipitation intensity, with warm colors indicating heavier rates) at 00

UTC on 5 June 2005 (left) which compares favorably with the radar-observed reflectivity (right). The forecast required 8 hours of wall clock time using 1100 processors on the TCS at PSC.

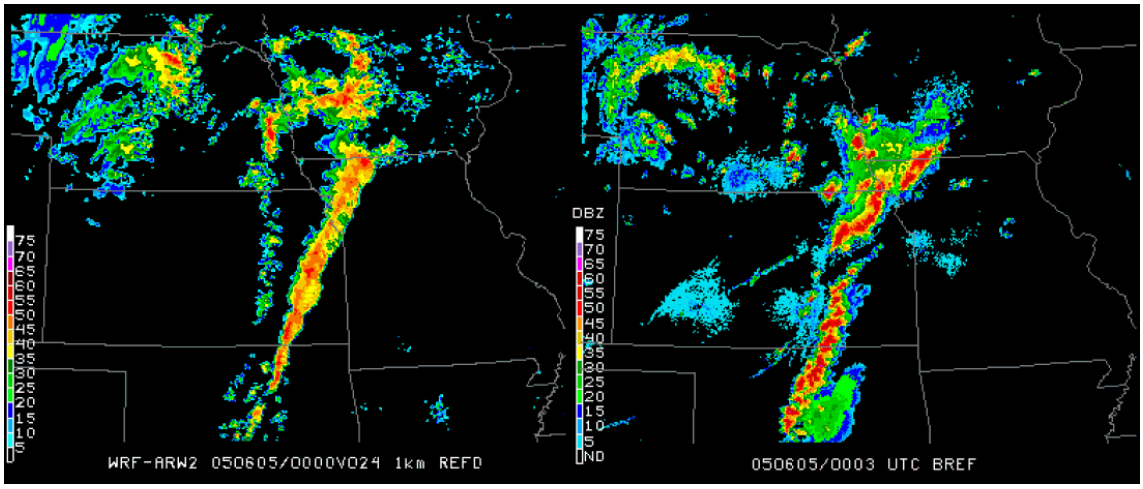

Fig. 18.4. 24-hour WRF-predicted (left) and observed (right) radar reflectivity valid at 00 UTC on 5 June 2005. Warmer colors indicate higher precipitation intensity. The WRF model utilized a horizontal grid spacing of 2 km and forecasts were produced by CAPS on the Terascale Computing System at the Pittsburgh Supercomputing Center as part of the 2005 SPC Spring Experiment.

During the spring of 2007, the above forecast configuration was further expanded to include 10-member WRF ensemble forecasts at 4 km grid spacing (Xue et al. 2007). Using 66 processors of the PSC Cray XT3 system, each 33-hour forecast ensemble member took 6.5 to 9.5 hours, with the differences being caused by the use of different physics options in the prediction model. A single forecast in the same domain using a horizontal grid spacing of 2 km (1501 × 1321 × 51 points) and 600 Cray XT3 processors took about 9 hours for a 33-hour, including full data dumps every 5 minutes. The data dumps into a parallel Luster file system accounted for over 2 hours of the total time. For truly operational implementations, such forecasts will need to be completed within one hour.

18.3.2 Very High-Resolution Tornado Simulation

Using the Terascale Computing System at PSC and the ARPS model, the first author obtained the most intense tornado ever simulated within a realistic supercellular convective storm. The highest resolution simulations used 25 m horizontal grid spacing and 20 m vertical spacing near the ground. The simulation domain was 48 × 48 × 16 km^3 in size and included only non-ice phase microphysics. No radiation or land surface processes were included. The use of a uniform-resolution grid large enough to contain the entire parent storm of the tornado eliminates uncertainties associated with the typical use of nested grids for this type of simulation. The maximum ground-relative surface wind

speed and the maximum pressure drop in the simulated tornado were more than 120 ms^{-1} and 8 $\times 10^3$ Pascals, respectively. The peak wind speed places the simulated tornado in the F5 category of the Fujita tornado intensity scale, the strongest of observed tornadoes. This set of simulations used 2048 Alpha processors and each hour of model simulation time required 15 hours of wall clock time, producing 60 terabytes of data dumped at 1 second intervals. These output data were used to produce extremely realistic 3D visualizations of the storm clouds and the tornado funnel. Fig. 18.5 shows a 3D volume rendering of the model simulated cloud water content in a 7.5 x 7.5 x 3 km domain, with a tornado condensation funnel reaching the ground.

Fig. 18.5. Three dimensional volume rendering of model simulated cloud water content in a 7.5 x 7.5 x 3 km domain, showing the tornado condensation funnel reaching the ground. The lowered cloud base to the left of the tornado funnel is known as the wall cloud (Rendering courtesy Greg Foss of the Pittsburgh Supercomputing Center).

In a 600 second long restart simulation using 2048 Alpha processors on 512 nodes, the message passing overhead consumed only 12% of the total simulation time. The small time step integration used more than 26%, and the subgrid-scale turbulence parameterization used 13%. The initialization of this restart simulation that involves the reading of 3D initial condition fields from a shared file system took over 11%. The 1-second output written to local disks of the each compute node took only 8% but the copying of output from the node disks to a shared file system using a PSC-built parallel copying command at the end of the simulation took as much as 1/5 of the total simulation time. Clearly I/O performance was a major bottleneck.

18.3.3 The Prediction of an Observed Supercell Tornado

Using the ARPS and its 3DVAR[5] and cloud analysis data assimilation system, Xue and Hu (2007) obtained a successful prediction of an observed thunderstorm and embedded tornado that occurred on 8 May 2003 in the Oklahoma City area. The horizontal grid had a spacing of 50 m and was one-way nested, successively, within grids of 100 m, 1 km and 9 km horizontal spacings. The 80 × 60 km^2 horizontal grid of 50 m spacing had 1603 × 1203 grid points and the 40 minute forecast took 2 days to complete using 1600 processors on the PSC TCS. A very significant portion of the time was consumed by writing output at 15 s intervals to a slow shared file system. A single processor was gathering data from all processors and writing out to this shared file system.

During the 40-minute forecast, two successive tornadoes of F1-F2 intensity with life spans of about 5 minutes each were produced within the period of the actual tornado outbreak, and the predicted tornadoes traveled along a path about 8 km north of the observed damage track with correct orientations. A half-hour forecast lead time was achieved by the 50 m forecast nested within the 100 m grid.

The surface reflectivity at 13.75 minutes for the 50 m forecast (valid at about 2214 UTC) is plotted in Fig. 18.6, together with observed reflectivity at the 1.45° elevation of the Oklahoma City operational weather radar at 2216 UTC. At this time, the predicted tornado shows a pronounced hook-shaped reflectivity pattern containing a inwardly spiraling reflectivity feature at the southwest end of the precipitation region (Fig. 18.6a). The observed low level reflectivity approximately 2 minutes later also contains a similar hook echo pattern (Fig. 18.6b). Due to resolution limitations of the radar, the observed reflectivity shows fewer structural details than the prediction.

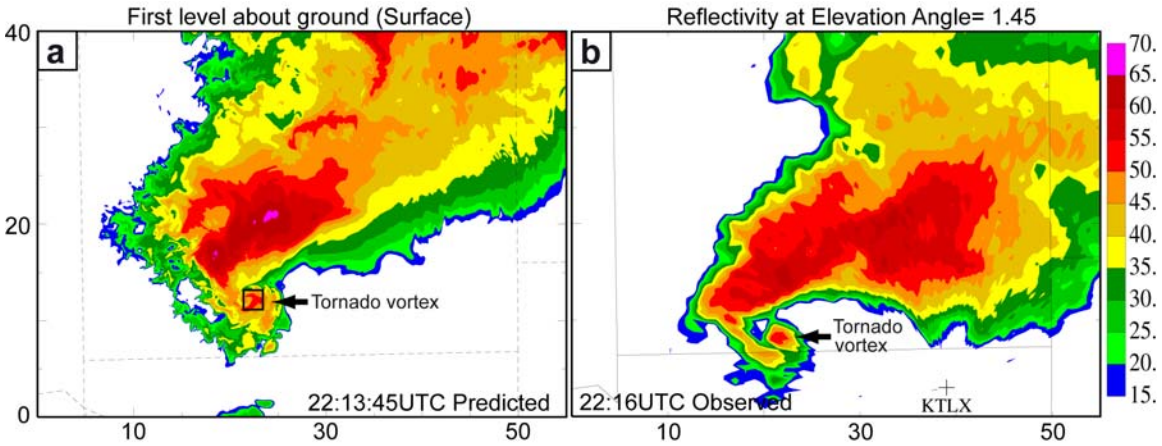

Fig. 18.6. Predicted surface reflectivity field at 13.75 minutes of the 50 m forecast valid at 2213:45 UTC and (b) observed reflectivity at the 1.45° elevation of the Oklahoma City radar observation at 2216 UTC. The domain shown is 55 km × 40 km in size, representing the portion of the 50 m grid between 20 and 75 km in the east-west direction and from 16 to 56 km in the north-south direction.

[5] 3DVAR is a subset of 4DVAR that does not include a time integration component.

18.4 Numerical Weather Prediction Challenges and Requirements

A grand vision of NWP for the next 20 years is a global NWP model running at 1 km horizontal grid spacing with over 200 vertical levels. Such resolution is believed to be the minimum required to resolve individual thunderstorms, which are the main causes of hazardous local weather. To cover the roughly 510 million square kilometers of Earth's surface, about half a billion model columns will be needed, giving rise to 100 billion grid cells when 200 levels are used. With each grid cell carrying about 200 variables so as to accommodate more sophisticated physical processes, a total of 80 terabytes of memory will be needed to run a single prediction. If this model is distributed to a petascale system with 1 million processor cores, each core will have 510 model columns with which to work. This corresponds to roughly a 26 × 20 × 200 subdomain, a relatively small domain in terms of the interior domain – boundary interface ratio, yet relatively large in terms of fitting required information into the CPU cache of expected sizes. If 100 forecasts are to be run simultaneously as part of an ensemble system, each forecast can be run on a subset of the processors. In this case, each processor will get 100 times more columns to work with, i.e., the subdomain size will be 100 times larger, at 255 × 200 × 200. In this case, the ratio of MPI communication to computation is significantly reduced, and the memory access will become the dominant performance issue.

Before 100-member global ensembles of 1-km spacing can be carried out in a timely manner, regional models can be run for smaller domains, say covering the North America. The ARPS and WRF models are mainly designed for such purposes. Further increases in horizontal resolution are possible and can be very desirable in order to resolve smaller, less organized convective storms. To be able to explicitly predict tornadoes, resolutions less than 100 m are essential and a better understanding of storm dynamics, coupled with better observations and parameterizations of land-atmosphere interactions, is needed. Because small-scale intense convective weather can develop rapidly and can be of short duration (e.g., 30 minutes), numerical prediction will require very rapid model execution. For example, a tornado prediction having a 30 minute lead time should be produced in a few minutes so as to provide enough time for response.

The ensemble data assimilation system (Tong and Xue 2005; Xue et al. 2006) that assimilates full volumes of weather radar data poses another grand challenge problem. The current U.S. operational Doppler weather radar network consists of over 120 radars. Producing full volume scans every 5 five minutes, roughly 600 million observations of radial wind velocity and reflectivity are collected by the entire network every 5 minutes assuming that all radars operate in precipitation mode. The planned doubling of the azimuthal resolution of reflectivity will further increase the data volume, as will the upgrade of the entire network to gain polarimetric[6] capabilities. These, together with high-resolution satellite data, pose a tremendous challenge to the data assimilation problem, and there is no limit to the need for balanced computing resources in the foreseeable future.

In addition to basic atmospheric data assimilation and prediction problems, the inclusion of pollution and chemistry processes and highly detailed microphysical

[6] Polarimetric refers to radars that transmit an electromagnetic pulse in one polarization state (e.g., horizontal) and analyze the returned signal using another state (e.g., vertical). This allows the radar to discriminate among numerous frozen and liquid precipitation species.

processes, and the full coupling of multi-level nested atmospheric models with ocean and wave models that are important for, e.g., hurricane prediction (Chen et al. 2007), will further increase the computational challenge.

18.5. Summary

The challenges faced in numerically predicting high-impact local weather were discussed, with particular emphasis given to deep convective thunderstorms. Two community regional numerical weather prediction (NWP) systems, the ARPS and WRF, were presented and their current parallelization strategies described. Key challenges in applying such systems efficiently on petascale computing systems were discussed, along with computing requirements for other important components of NWP systems including data assimilation with 4D variational or ensemble Kalman filter methods. Several examples demonstrating today's state of the art simulations and predictions were presented.

18.6. Acknowledgement

This work was mainly supported by NSF grants ATM-0530814 and ATM-0331594. Gregory Foss of PSC created the 3D tornado visualization with assistance from the first author. Kevin Thomas performed the timing tests shown in Fig. 18.3.

References

Bryan, G. H., J. C. Wyngaard, and J. M. Fritsch, 2003: Resolution Requirements for the Simulation of Deep Moist Convection. *Mon. Wea. Rev.*, **131,** 2394-2416.

Chen, S. S. and J. E. Tenerelli, 2006: Simulation of hurricane lifecycle and inner-core structure using a vortex-following mesh refinement: Sensitivity to model grid resolution. *Mon. Wea. Rev.***,** Submitted.

Chen, S. S., J. F. Price, W. Zhao, M. A. Donelana, and E. J. Walshc, 2007: The CBLAST-Hurricane program and the next-generation fully coupled atmosphere–wave–ocean models for hurricane research and prediction. *Bull. Amer. Meteor. Soc.*, **88,** 311–317.

Droegemeier, K. K., coauthors, 1995: Weather prediction: A scalable storm-scale model. *High Performance Computing*, G. Sabot, Ed., Addison-Wesley, 45-92.

Droegemeier, K. K., coauthors, 2005: Service-oriented environments for dynamically interacting with mesoscale weather. *Comput. Sci. Engineering*, **7,** 12-27.

Evensen, G., 2003: The ensemble Kalman filter: Theoretical formulation and practical implementation. *Ocean Dynamics*, **53,** 343-367.

Evensen, G., 2006: *Data Assimilation: The Ensemble Kalman Filter*. Springer, 280 pp.

Houze, R. A., Jr., coauthors, 2006: The Hurricane Rainband and Intensity Change Experiment: Observations and Modeling of Hurricanes Katrina, Ophelia, and Rita. *Bull. Amer. Meteor. Soc.*, **87,** 1503–1521

Hunt, B. R., E. J. Kostelich, and I. Szunyogh, 2005: Efficient data assimilation for spatiotemporal chaos: a Local Ensemble Transform Kalman Filter *Physics***,** 11236H

Irigoin, F. and R. Triolet, 1989: Supernode partitioning. *Proceeding, 5th Annual ACM SIGACT-SIGPLAN Symp. Principles Programming Languages*.

Johnson, K. W., G. A. J. Bauer, Riccardi, K. K. Droegemeier, and M. Xue, 1994: Distributed processing of a regional prediction model. *Mon. Wea. Rev.*, **122,** 2558-2572.

Kalnay, E., 2002: *Atmospheric modeling, data assimilation, and predictability*. Cambridge University Press, 341 pp.

Keppenne, C. L., 2000: Data Assimilation into a Primitive-Equation Model with a Parallel Ensemble Kalman Filter. *Mon. Wea. Rev.*, **128,** 1971-1981.

Keppenne, C. L. and M. M. Rienecker, 2002: Initial testing of a massively parallel ensemble Kalman filter with the Poseidon isopycnal ocean general circulation model. *Mon. Wea. Rev.*, **130,** 2951-2965.

Michalakes, J., coauthors, 2004: The Weather Research and Forecast Model: Software Architecture and Performance. Proceedings, 11th ECMWF Workshop on the Use of High Performance Computing In Meteorology, , Reading U.K.

Pielke, R. A. a. R. C., 2002: Weather impacts, forecasts, and policy. *Bull. Amer. Meteor. Soc.*, **83,** 393-403.

Rabier, F., H. Jarvinen, E. Klinker, J.-F. Mahfouf, and A. Simmons, 2000: The ECMWF operational implementation of four-dimensional variational assimilation. I: Experimental results with simplified physics. *Quart. J. Roy. Met. Soc.*, **126,** 1143-1170.

Richardson, L. F., 1922: *Weather Prediction by Numerical Precess*. Cambridge University Press. Reprinted by Dover Publications, 1965, 236 pp.

Sathye, A., G. Bassett, K. Droegemeier, and M. Xue, 1995: Towards operational severe weather prediction using massively parallel processing. *High Performance Computing*, Tata McGraw Hill.

Sathye, A., M. Xue, G. Bassett, and K. K. Droegemeier, 1997: Parallel weather modeling with the Advanced Regional Prediction System. *Parallel Computing*, **23,** 2243-2256.

Sathye, A., G. Bassett, K. Droegemeier, M. Xue, and K. Brewster, 1996: Experiences using high performance computing for operational storm scale weather prediction. *Concurrency: Practice and Experience, special issue on Commercial and industrial Applications on High Performance Computing*, John Wiley & Sons, Ltd., 731-740.

Skamarock, W. and J. B. Klemp, 1994: Efficiency and accuracy of the Klemp-Wilhelmson time-splitting technique. *Mon. Wea. Rev.*, **122,** 2623-2630.

Skamarock, W. C., coauthors, 2005: A Description of the Advanced Research WRF Version 2, 88 pp.

Tong, M. and M. Xue, 2005: Ensemble Kalman filter assimilation of Doppler radar data with a compressible nonhydrostatic model: OSS Experiments. *Mon. Wea. Rev.*, **133,** 1789-1807.

Weber, D. B. and H. Neeman, 2006: Experiences in optimizing a numerical weather prediction model: An exercise in futility? 7th Linux Cluster Institute Conference.

Wicker, L. J. and W. C. Skamarock, 2002: Time-Splitting Methods for Elastic Models Using Forward Time Schemes. *Mon. Wea. Rev.*, **130,** 2088-2097.

WRF-RAB, 2006: Research community priorities for WRF-system development. WRF Research Applications Board Report, 35 pp.

Xue, M., 2000: High-order monotonic numerical diffusion and smoothing. *Mon. Wea. Rev.*, **128,** 2853-2864.

Xue, M. and M. Hu, 2007: Numerical prediction of 8 May 2003 Oklahoma City supercell tornado with ARPS and radar data assimilation. *Geophys. Res. Letters***,** In review.

Xue, M., K. K. Droegemeier, and V. Wong, 2000: The Advanced Regional Prediction System (ARPS) - A multiscale nonhydrostatic atmospheric simulation and prediction tool. Part I: Model dynamics and verification. *Meteor. Atmos. Physics*, **75,** 161-193.

Xue, M., M. Tong, and K. K. Droegemeier, 2006: An OSSE framework based on the ensemble square-root Kalman filter for evaluating impact of data from radar networks on thunderstorm analysis and forecast. *J. Atmos. Ocean Tech.*, **23,** 46–66.

Xue, M., K. K. Droegemeier, V. Wong, A. Shapiro, and K. Brewster, 1995: *ARPS Version 4.0 User's Guide*. [Available at http://www.caps.ou.edu/ARPS], 380 pp.

Xue, M., D.-H. Wang, J.-D. Gao, K. Brewster, and K. K. Droegemeier, 2003: The Advanced Regional Prediction System (ARPS), storm-scale numerical weather prediction and data assimilation. *Meteor. Atmos. Physics*, **82,** 139-170.

Xue, M., coauthors, 2001: The Advanced Regional Prediction System (ARPS) - A multi-scale nonhydrostatic atmospheric simulation and prediction tool. Part II: Model physics and applications. *Meteor. Atmos. Phys.*, **76,** 143-166.

Xue, M., coauthors, 2007: CAPS realtime storm-scale ensemble and high-resolution forecasts as part of the NOAA Hazardous Weather Testbed 2007 Spring Experiment. Preprint, 18th Conf. Num. Wea. Pred., Utah, Ameri. Meteoro. Soc.