UTILIZING MODEL INTEROPERABILITY AND SPATIAL CLOUD COMPUTING TO ENABLE THE COMPUTABILITY OF DUST STORM FORECASTING

by

Qunying Huang
A Dissertation
Submitted to the
Graduate Faculty
of
George Mason University
in Partial Fulfillment of
The Requirements for the Degree
of
Doctor of Philosophy
Earth Systems and Geoinformation Sciences

Committee:

_______ Dr. Chaowei Yang, Dissertation Director

_______ for Dr. Taylor Dr. George Taylor, Committee Member

_______ Dr. Donglian Sun, Committee Member

_______ Dr. Songqing Chen, Committee Member

P. Agouris _______ Dr. Peggy Agouris, Department Chairperson

_______ Dr. Timothy Born, Associate Dean for Academic and Student Affairs, College of Science

_______ Dr. Vikas Chandhoke, Dean, College of Science

Date: July 28. 2011. Summer Semester 2011
George Mason University
Fairfax, VA

UTILIZING MODEL INTEROPERABILITY AND SPATIAL CLOUD COMPUTING TO ENABLE THE COMPUTABILITY OF DUST STORM FORECASTING

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at George Mason University

By

Qunying Huang
Earth Science and Geoinformaiton
George Mason University, 2011

Director: Chaowei Yang, Professor
Geography and Geoinformation Science

Summer Semester 2011
George Mason University
Fairfax, VA

DEDICATION

This is dedicated to my wonderfully family. This thesis is dedicated to my mother, who inspires and encourages me to set high goals and gives me the confidence to achieve them, and who is always my spiritual support, especially during the hard time. It is also dedicated to my father, who gives me positive and optimistic personality to overcome the difficulties. It is also dedicated to my sisters and young brother, they help me take care of my parents while I was focus on my research and support me all those years.

ACKNOWLEDGEMENTS

I wish to thank Dr. Chaowei Yang, who has been the thesis supervisor and my committee chairman. His wise advice, insightful criticisms, and patient encouragement aided the writing of this thesis in innumerable ways. I would also like to thank my committee members, Dr. George Talyor, Dr. Donglian Sun, and Dr. Songqing Chen, who were more than generous with their expertise and precious time in commenting, reading, encouraging, and most of all patience throughout the entire process. I would also like to thank Dr. Karl Benedict whose assists in this project was greatly needed and deeply appreciated. Special thanks to Dr. Abdelmounaam Rezgui for the many hours of proofreading and valuable advice. Finally, I would like to thank my dear friends, Yunfeng Jiang, Jing Li, Min Sun, Kai Liu, Jizhe Xia, Chen Xu, Zhipeng Gui, Peter Lostritto, who have supported me throughout the process. I will always appreciate all they have done.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

UTILIZING MODEL INTEROPERABILITY AND SPATIAL CLOUD COMPUTING TO ENABLE THE COMPUTABILITY OF DUST STORM FORECASTING

Qunying Huang, MS

George Mason University, 2011

Dissertation Director: Chaowei Yang

Both environmental and human challenges, such as deforestation and desertification, require scientifically sound simulations of physical phenomena to better understand the past and to better predict future trends for improved decision support. However, many scientific problems cannot be processed using a single computer and require computing capability from many distributed computers. The problems should be solved by interdisciplinary efforts instead of by a single science community. Using dust storm forecasting as a case study, I investigate how interoperability technologies can facilitate data access service, model input integration, model coupling, and output utilization and dissemination. Additionally, the research will explore how to utilize spatiotemporal patterns of phenomena, models and computing resources to improve the performance of dust storm forecasting. Finally, I adopt and optimize cloud computing platforms through spatiotemporal patterns to enable the computability of dust storm forecasting over a large area with high resolution to support geospatial decision-making. This research eventually reduce the execution time and communication for two heterogeneous models, Eta-8bin

and NMM- dust storm models by enabling the interoperable and loosely-coupling execution of the two models.

CHAPTER 1

INTRODUCTION

Both environmental and human challenges, such as deforestation and desertification, require scientifically sound simulations of physical phenomena to better understand the past and to better predict future trends for improved decision support. Simulating geospatial phenomena is especially complex and time consuming when considering the dynamics of Earth system phenomena, for example, modeling and predicting cyclic processes (Donner et al., 2009) including ocean tides (Cartwright, 2000), earthquakes (Schuster, 1897), and dust storms (Xie et al., 2010). Such periodic phenomena simulation requires the iteration of the same set of intensive computations for many times. Therefore, they cannot be processed by a single computer and high-performance computing is usually adopted to speed up the computing process.

Large amount of multi-dimensional data are available from multiple sources, such as satellites observations and model simulations. Ingesting widely available data resources for model initiation, instead of using restricted data resources, would greatly improve the model integration. The capability of disseminating model output would greatly facilitate the sharing of data and model results among science communities. In addition, enabling the communication of multiple heterogeneous models, developed by different organizations, has attracted a lot of interest and always been a great challenge (Nativi et al., 2004; Zhou, 2006; Hu and Bian, 2009).

This chapter will introduce dust storm generation and impact, challenges of dust storm modeling, computing, and interoperability from dust storm forecasting. We will also discuss and present the potential computing and interoperability solutions that are needed to address the challenges and to enhance dust storm research and forecasting by facilitating data access, model input integration, model coupling, output utilization and dissemination, and enabling the computability of dust storm forecasting. We will conclude the chapter by discussing the proposed research area.

1.1 Dust Storm

Dust storms are the result of strong turbulent wind systems entraining particles of dust into the air, reducing visibility down to several meters (Goudie and Middleton, 1992). Global climate change has driven up the frequency and intensity of dust storms in the past decades with negative impacts on the environment, human health, and assets. For example, dust storms 1) contain marine nutrients, such as active iron and phosphorus, which can result in algal blooms over the ocean surface when decomposing into the ocean water (Dulac et al., 1996), 2) act as a pollutant which reduces air quality and affects the public health by causing allergies, respiratory diseases, and eye infections(Nickling and Gillies, 1993); 3) impact both regional and global environment and climate (Gong et al., 2003; Sokolik and Toon, 1996; Tegen and Fung, 1994) in cooling the oceans by reflecting solar radiation back to space (Gong et al., 2003) and contributing to global aerosol mass load and optical thickness (Gong et al., 2003). A dust-laden atmosphere with an average optical thickness of 0.5 would cause net radiative forcing of +20 to 40 W/m^2 over arid regions and -5 ~-15 W/m^2 over the ocean (Sokolik and Toon, 1996).

The severe impacts of dust storms on the environment have motivated scientists to better understand and predict the distribution and intensity of dust emission, deposition, and structure by developing dust models to 1) predict dust storms; 2) understand dust processes; 3) quantify the global dust cycle; and 4) re-construct past climates (Shao and Dong, 2006). Since the late 1980s, several research groups have developed dust models that can correctly predict spatiotemporal patterns, evolution, and order of magnitude of dust concentration, emissions and deposition (e.g., Westphal et al. 1988; Gong et al. 2003; Shao et al. 2003; and Han et al. 2004).

1.2 Dust Storm Forecasting and Computing Challenges

1.2.1 Distributed Data Resources and Heterogonous Models

Geospatial datasets are characterized by 1) large amount, 2) multiple dimensions, 3) geographically distributed over several servers and 4) various formats. They are, therefore, too complex to be easily browsed or combined with other information. Datasets from various research studies have become increasingly available because of the advancements in data collection technologies, data storage facilities, and data production, processing and retrieval algorithms and models. However, scientists still face the embarrassing situations that datasets may be available but not readily usable (Yang and Raskin, 2009). Data interoperability has been seen as a solution for sharing and integrating geospatial datasets by solving the syntactic, schematic, and semantic as well as the spatial and temporal heterogeneities among various representations of real–world phenomena (Brodeur et al., 2003).

Research on data interoperability to facilitate data discovery, access, and utilization is more focused on the aspects of integrating distributed heterogeneous datasets for geovisulization and performing some simple spatial analysis based on open standards (Cao et al., 2009). However, the workflow of how to utilize those massive available datasets to facilitate model integration and interoperability is not fully researched (Argent, 2004). Several aspects can be explored for model interoperability:

1) Utilizing distributed datasets as model input. Traditionally, in order to run a model, users need to access specific data centers to download the model input data with specific data formats as model input. For regional dust storm simulation or weather forecasting models, model input should be obtained from the NCAR data center, produced from NCEP's NAM (http://stu-in-flag.net/nam.php) or GFS model (http://www.emc.ncep.noaa.gov/gmb/moorthi/gam.html) to extract meteorological parameter to initialize the model. At the same time, many other data centers also provide the meteorological datasets, which are ready for atmospheric model initialization;

2) Enabling the sharing of model outputs and intermediate products. For a public oriented real time dust storm forecasting systems, users are expected to receive the results in a vivid fashion. A good strategy should be devised to enable users to easily access the temporarily predicted results during model execution and share the model output after forecasting is completed. The model usually produces temporary results in 3-hourly intervals. The model outputs and intermediate products should be available through the automated exposure of those products through service interfaces that are broadly supported by client analytic and visualization applications.

3) Communications of models. Most models have their own semantics, schemas, tools, and interfaces, which are usually not effectively interoperable (Nativi et al., 2004). Due to the development of Internet technology, the information, processing tools and application models for various disciplines and fields should be able to communicate and interact with each other via Internet. Non-experts should be able to utilize, through the Internet, the models without understanding their details. The solution to this problem is to unify or enable models to interoperate to provide a single information template across applications, thus, enabling the communication and reuse of the model components.

1.2.2Computational Intensity

For forecasting purposes, the entire computing time is normally limited. For example, a 2 hour period is recommended for one day forecasting (Lenz et al., 1995). Reducing the geographic scope and resolutions is usually used to complete the simulations within the time limit (Wolters et al., 1995). However, a zip code level resolution is needed for dust storm forecasting to support public health decision making (Yang et al., 2008). This poses significant computational challenges for dust storm simulations to be enhanced by a) reducing computing time, b) supporting high resolution, and c) lengthening the prediction time period.

Simulating geospatial phenomena is especially complex and computing intensive when considering the full dynamics of Earth system phenomena, for example, modelling and predicting cyclic processes (Donner et al., 2009), including ocean tides (Cartwright, 2000), earthquakes (Schuster, 1897), and dust storms (Xie et al., 2010). Such periodic phenomena simulation requires the iteration of the same set of intensive computations for a series of equations for many times. In the past decades, the increasing need for computing power for such phenomena has been addressed by using either High Performance Computing (HPC) (Armstrong et al., 2005, Yang et al., 2008, Huang and Yang, 2010, Clematis et al., 2003, Baillie et al., 1997, Reed, 2008).

While the parallelization approach is used to solve large scale complex computing problems in Geosciences, communication and synchronizations involved among computing units has limited the scalability of massively parallel computers(Drake and Foster, 1995). While size and memory requirements are essential factors in parallel computing, communication overhead and load balancing substantially contribute to the overall system execution time. In general, the processing throughput decreases while communication overhead increases with the numbers of processes involved for a parallel task (Sterling et al., 1995). Both load balancing and communication may impact performance considerably. How to better leverage HPC to improve the performance of parallel systems, therefore enabling the computability of compute intensive geosciences problems, has always been a research issue (Yang et al., 2011).

1.3 Potential Solutions with Spatial Cloud Computing

Computing intensive challenges drive the evolution of distributed computing paradigms from cluster computing, grid computing, to cloud computing, which can provide more powerful and scalable computing capabilities to enable the computability of geoscience applications. In addition, the performance of computing could be further improved and optimized through utilizing spatiotemporal patterns of phenomena, data, services, models and computing resources.

1.3.1 Optimization through spatiotemporal patterns

Spatiotemporal principles widely exist in all science domains, such as geographic sciences, biological sciences, and social sciences. It is very important for a parallel computing system to leverage the spatiotemporal principles, such as space and time constraints/drives, in computing arrangements, selection and utilization to enable the computability of science problems (Yang et al., 2011b). Spatiotemporal principles should be considered in algorithms, methodology and phenomena simulations. For example, in atmospheric sciences, the actual number of grid points selected for buffering would greatly impact both computation and forecasting accuracy (Nanjundiah, 1998; Xie et al., 2010). In addition, when forecasting dust storm as a weather component, we will consider the time and space interaction, i.e., how time changes impact the space distribution of the dust in the atmosphere. A lot of studies have demonstrated that Geoscience and HPC can benefit from each other (Wang and Liu, 2009). Guided by the spatiotemporal principles, we can develop scheduling strategies more efficiently, and therefore the performance of the parallel system could be improved and optimized (Yang et al., 2011a). However, leveraging HPC to achieve better performance requires understanding and balancing spatiotemporal patterns and constraints (Yang et al., 2011a).

1.3.2 Model Interoperability

The solution to fast access to science products, and appropriate represent the products using existing tools are in developing interoperable systems that can facilitate efficient use and reuse of model products (Argent 2004).A unify or interoperable model can also be used through open information template for data ingesting and output dissemination cross applications, therefore, to enable the communication and reuse of the model components. One solution to achieve model interoperability is the integration of Geographic Information System (GIS) capabilities into model integration workflows, either as data management, mapping and analysis tools that are largely independent of the modeling systems, or as the primary system through which both data and model execution are controlled (Argent 2004).

In this dissertation, open interoperability standards for data access and dissemination will be explored to build an integrated and interoperable dust storm modeling system. Such an interoperable model could address the general issues of accessibility, compatibility, and timeliness for both researchers and decision makers.

1.3.3 Nested Models

Model results are required with high resolution, larger domain size and longer duration for specific applications, such as public health decision supporting. However, current constraints on computing power and the scalability of parallel systems preclude an immediate solution to satisfy the requirements. More strategies should be explored to resolve these problems (Kuligowski and Barros, 1999, Yang et al., 2010). One approach is using adaptive multilevel modeling (AMM), in which different physical processes are resolved independently at the relevant spatial scales, thus leading to a suite of simple models operating on grids of different spatial resolutions (Barros, 1995; Kuligowski and Barros, 1999). Another approach is to nest a finer-scale grid or succession of grids within a model to enhance the resolution over specific areas of interest while moderating the required computational cost (Anthes, 1983, Kuligowski and Barros et al., 1999).Such an approach is called nested models, or more commonly named as high resolution limited-area models. They have the capability to generate meaningful small-scale features from low-resolution information, provided as initial lateral boundary conditions (Ramón et al., 2002). In the last two decades, nested models have been gaining a wide acceptance in the scientific community and have been considered as powerful methodology for predicting and studying weather and climate patterns (White et al., 1999).

The model nesting approach is especially useful because the dust storm has the characteristics of special spatial patterns and time variability which correlate with the soil type and geographic locations. For a large geographic area, dust storm occurs only in small local regions. Therefore, a nested dust storm model with the coarse results for the overall domain and higher resolution results for small regions with high dust concentration, would enable the predictions for a large area, with high resolution while complying with the two-hour time constraints for one-day predictions imposed on the model computation.

1.3.3 Spatial Cloud Computing

Cloud Computing is a new computing paradigm in which dynamic and often virtualized resources are provided as a service over the Internet. Within such a platform, computing resources can be scaled up or down based on computing requirements in real time. As geospatial processing and analysis is often complex with the intensities of data, computing, concurrent access, and spatiotemporal patterns, a geospatial application may need only a single CPU at one time but may need hundreds of CPUs at other times. For instance, dust storm simulation may need only one computing instance for low resolution and small area forecasting while a large computing pool is required for high resolution and large area forecasting. Therefore, such scalable cloud services would facilitate geospatial applications, elastically satisfying changes of computing requirements.

In addition, cloud computing users do not have to manage the computing infrastructure themselves(Vaquero et al., 2009).Therefore, cloud computing is able to provide a transparent platform for intensive processing so that users do not need to consider the underlying mechanism of data and service delivery and the correspondingly complex and time-consuming computational tasks required (Evangelinos and Hill, 2009). This is a pay as you go long held dream from distributed computing users (Yang et al., 2011b).

1.3.3.1 Cloud Computing Background

Technological advancements, such as multi-core processors and networked computing environments, drive computing platforms through several paradigms including cluster computing, Grid computing (Foster and Kesselman, 1998), P2P computing (Oram, 2001), and most recently, cloud computing (Armbrust et al., 2009). The basic concept of cloud computing is the use of the network, storage and computing power through the Internet. This computing model integrates Web 2.0, on-demand deployment, Internet delivery of services and open source software, and visualization, and visualization is considered as a key component to satisfy the computing needs of the users (Buyya, 2008).

The National Institute of Standards and Technology (NIST) defines cloud computing as "...a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction"(Mell and Grance, 2009). NIST also defines five essential characteristics that differentiate cloud computing from other distributed computing paradigm: 1) on-demand self-service, 2) multi-tenancy, 3) measured services, 4) device and location independent resource pooling, and 5) rapid elasticity (Mell and Grance, 2009).

The forms of cloud service models include Infrastructure as a Service (IaaS), Platform as a service (PaaS), Software as a Service (SaaS) and Data as a Service (DaaS). The first three are defined by NIST and DaaS is essential to geospatial sciences. These four services are referred to collectively as XaaS.

- IaaS is the most popular cloud service, which delivers computer infrastructure, including physical machines, networks, storage and system software, as virtualized computing resources over computer networks. IaaS enables users to configure, deploy, and run operating systems (OS) and applications based on the OS. IaaS users should have system administrative knowledge about the OS and they have full control over the virtualized machine. The most notable commercial product is the Amazon Elastic Compute Cloud (EC2, http://aws.amazon.com/ec2/).

- PaaS is a higher level service than IaaS and provides a platform service for software developers to develop applications. In addition to computing platforms, PaaS provides a layer of cloud-based software and APIs that can be used to build higher-level services. Microsoft Azure (www.microsoft.com/windowsazure) and Google App Engine are the most notable examples of PaaS. Users can develop or run existing applications on such a platform and do not need to consider maintaining the OS, server hardware, load balancing or computing capacity. PaaS provides all the facilities required to support the complete lifecycle of building and deploying applications and services entirely through the Internet.
- SaaS is the most used type of cloud computing service and provides various capabilities of sophisticated applications that are traditionally provided through the Web browser to end users. Notable examples are Salesforce.com and Google's gmail and apps. The ArcGIS implementation on the cloud is another example of SaaS.
- DaaS is the least well defined of the four types of cloud services. It supports data discovery, access, and utilization and delivers data and data processing on demand to end users regardless of geographic or organizational location of provider and consumer (Olson, 2010). Integrating a layer of middleware that collocates data and processing and optimizes cloud operations (Jiang 2011), DaaS is able to facilitate data discoverability, accessibility, and utilizability on the fly to support science on demand. We are currently developing a DaaS based on several cloud platforms.

Based on the characteristics of different cloud models, each cloud model can be utilized for different geospatial sciences and applications including storing and acquiring Earth Observation data, extracting parameters, configuring and running models, obtaining knowledge, making decisions, and collecting users' feedback.

1.3.3.2 Spatial Cloud Computing for Earth Sciences

There is an urgent need to investigate how geospatial sciences and applications can leverage cloud computing to improve the performance and enable the computability of scientific problems, and hide the complexity of the computing infrastructure so scientists can focus on scientific problems. Earth science applications have special requirements that cannot be automatically supported by generic cloud computing platforms because most geospatial algorithms and applications are not designed to leverage multiple CPUs and be delivered through the Internet as a service. Most importantly, both the geospatial sciences and the cloud computing environment are spatiotemporal intensive. Earth science phenomena are complex processes and Earth science applications often take a variety of data as input with a long and complex workflow. It becomes then a critical challenge to deliver such complex applications to the cloud as a transparent service to support massive numbers of users. For example, the middleware used to schedule computing tasks on a cloud computing platform is mostly not developed for Earth science applications and does not take the spatiotemporal principles and patterns into consideration. Such middleware should be reengineered to support spatiotemporal processing. Also, spatiotemporal patterns of phenomena, data, services, models and computing resources must be utilized to optimize the performance of geospatial processing and applications.

1.4 Proposed Research

This research reports on a proof-of-concept study of the characterization of dust storms in a near-real-time execution scenario, executed on spatial cloud computing platforms. The following research is proposed:

1) Utilize spatiotemporal patterns and principles to improve the performance of dust storm simulations.

2) Access and utilize distributed data resources, model communication, and open standards to enable the interoperability of heterogeneous models.

3) Explore nested models to enable dust storm forecasting with finer resolution, larger domain, and longer time scales.

4) Utilize cloud computing platforms optimized through spatiotemporal principles to support the execution of near real-time dust storm forecasting.

1.5 Thesis Outline

The thesis consists of seven chapters. This chapter introduces the study by analyzing challenges, potential solutions, and proposed research area. Chapter 2 reviews related work, including dust storm simulation, parallelization of atmospheric models, model interoperability, nested model, spatiotemporal patterns of physical phenomena, model, and computing resources, and spatial cloud computing. Based on the existing research efforts, Chapter 3 states the research objectives. Chapter 4 presents how to utilize spatiotemporal patterns in thinking and computing to improve the performance of dust storm simulation, through better parallelizing the dust storm model, arranging the computing resources and utilizing the spatiotemporal pattern of dust storm. In Chapter 5, we investigate more approaches to enable the computability of dust storm for large areas and high resolution through loosely-coupled interoperable models. Chapter 6 discusses the deployment of dust storm onto cloud computing platforms to enable the nested execution of dust storm models. Chapter 7 summarizes research and discusses future research.

CHAPTER 2
LITERATURE REVIEW

For years, Earth scientists have been working on modelling and predicting the dust storm phenomena and improving the prediction accuracy to better understand and prepare for dust storm events(Westphal et al. 1988; Toon et al. 1988;). However, to support public health decision making, and enable the public to make quick responses to severe dust storm events, computing issues and challenges must be addressed to enable the computability of dust storm forecasting with large spatiotemporal scope and high resolution requirement, and to disseminate the model results in an easy-to-access manner (Benedict et al., 2011). While scalability, communication, and synchronization issues prevent the parallel system to support such requirements, spatiotemporal principles and patterns are proposed to address those issues with optimized computing resources selection and arrangement strategies (Yang et al., 2011a; Yang et al., 2011b). With the capability to run regional high-resolution simulations without the enormous computational cost of a global model at the same resolution, nested coupling models have been considered as powerful tools to predict and study weather and climate patterns (White et al., 1999).

This chapter will review related works that have been done in areas of dust storm simulation, parallelization of dust storm models, nested models, model interoperability to enable the loosely-coupled nested model and model result dissemination, and cloud computing.

2.1 Dust Storm Simulation

Usually, dust simulation models are developed by coupling dust process modules to atmospheric process models for global, regional and local dust problems (Shao et al., 2007). Westphal et al. (1988) models dust storms using the limited area tropical model (KC) with horizontal grid spacing 220km and 13 vertical layers, coupled with the NASA Ames dust processes model (Toon et al. 1988) with 30 aerosol bins ranging from 0.1 to 80 um in radius. The US Naval Research Laboratory (NRL) has a Navy Operational Global Atmospheric Prediction System (NOGAPS) (Rosmond et al., 2002), which includes a dust module modified from CARMA developed by Christensen (1997). Gong et al., (2003) developed NARCM (Northern Aerosol Regional Climate Model) to model and simulate atmospheric soil dust aerosol processes by coupling CAM (Canadian Aerosol Module, Gong et al., 2002) and Canadian Regional Climate Model (RCM). The United States Air Force Weather Agency (AFWA) supported the Community Aerosol and Radiation Model for Atmospheres (CARMA) (Barnum et al., 2004) for the daily forecasting of dust, and the model is modified to assimilate meteorological forecast data from the Penn State fifth generation Mesoscale Meteorology Model (MM5) (Anthes and Warner, 1978). Shao et al. (2007) developed the Computational Environmental Modelling System 5 (CEMSYS5,) to couple a limited area atmospheric model (Leslie and Wightwick, 1995), and Atmosphere and Land Surface Interaction Scheme (ALSIS) module (Irannejad and Shao, 1998).

The Dust Regional Atmospheric Model (DREAM, Nickovic et al., 2001) is one of the most used for dust cycle modeling and designed to simulate dust entrainment and transport on a regional scale by incorporating the effects of particle size distribution on aerosol dispersion and all the major phases of the atmospheric dust life from dust production, diffusion and advection, to dust decomposition (Nickovic et al., 2001). DREAM can be easily configured and incorporated into many other atmosphere models, such as National Centers for Environmental Prediction (NCEP)'s Eta (the SKIRON weather forecasting system, Kallos et al. 1997; Nickovic et al., 1997) as Eta-4bin with 4 classes of particle size and Eta-8bin with 8 classes of particle size. Eta-4bin and Eta-8bin have been tested for various dust storm episodes in various places and resolutions and have been operational for three years to provide 72 hour forecasts for the Mediterranean region (http://www.icod.org.mt and http://forecast.uoa.gr). However, the Eta-8bin model has a coarse spatial resolution of 1/3 of a degree that cannot be used for many potential applications (Xie et al., 2010), and Numerical Weather Prediction (NWP) models run in sequence and reach validity limits for increasing resolution. Therefore, the Eta/NCEP model was replaced in NWS operations by a Non-hydrostatic Mesoscale Model (NMM), which could have high resolution up to one kilometer (KM) and runs in parallel (Janjic et al., 2001, Janjic, 2003). The coupling of DREAM and NMM (NMM-dust) is used in this dissertation for parallel processing and a higher resolution up to zip code level or about 3KM.

2.2 Parallelization of Models

Dust storm models are developed by adding dust solvers into the regional atmospheric models (Shao et al., 2007, Xie et al., 2010), and dust storm models parallelization is to parallelize the core atmospheric modules. Atmosphere is modeled by dividing the studied area into three-dimensional cells and atmospheric modeling is to solve a system of coupled nonlinear partial differential equations on each cell with appropriate boundary conditions (Purohit et al., 1999). The calculations of the equations on each cell are repeated with a time step to model phenomena evolution. Therefore, the computational cost of an atmospheric model is a function of the number of cells in the domain and the number of time steps (Baillie et al., 1997). For a given domain size, the cost of explicit three-dimensional hydrodynamics codes behaves like a function of n^4, where n is a grid dimension, including two horizontal dimension, one vertical dimension and time dimension (Baillie et al., 1997). Doubling the geographic scope on the horizontal direction would result in a four-fold increase. Halving the spatial resolution could result in an eight-fold increase in computational cost as halving the spatial resolution would require halving the time step. If the vertical resolution is also doubled, the computing complexity could increase sixteen-fold.

Parallel architectures are greatly used as an instrumental mechanism for the execution of computing intensive applications (Jin et al., 2003; Yang et al., 2005), such as Eta Models (Henderson et al., 1994), Rapid Update Cycle (RUC, Rodriguez et al.1995), QNH (Baillie et al., 1995), HIRLM (High Resolution Limited Area Model, Wolters et al.,1995), Fifth-generation Mesoscale Model (MM5, Davis et al. 1999), Advanced Regional Prediction System(ARPS, Xue et al., 2003) and RAMS(Cotton et al., 2003). The advances in computing capacity in CPU frequency and HPC allow modern NWP models to reach the resolution of a town level (e.g., Davis et al. 1999). For example, MM5 has been used for real-time weather prediction with grid spacing achieving 1 km by the U.S. Army Test and Evaluation Command (Davis et al. 1999).

Each cell in an atmospheric model performs essentially the same set of computations in, normally, a SPMD (single program, multiple data stream) domain decomposition approach and nearest neighbor communication in the physical domains is required (Nanjundiah, 1998). However, data dependencies between neighboring cells in the vertical directions are much larger than those in the horizontal directions. Horizontal decomposition in the model should also consider minimizing communication overhead (Wolters et al., 1995). The communication includes halo exchanges, periodic boundary updates, parallel transposes, and others. The halo region is the part of the local memory allocated around a cell for exchanging information with neighboring cells using message passing. The process of data decomposition should define the halo regions assigned to each processor and also define a virtual array of processors used to execute these regions and create neighboring relations between regions.

Optimizations have been conducted by improving the data structure, algorithm design, libraries for parallelization, and compiler for compiling the code (Rodriguez et al.1995, Rodriguez et al.1996). Rodriguez et al. (1995) studied and compared the issues of performance in the parallelization of weather prediction models using two different parallelization libraries. The authors also discussed the optimization strategies like the minimization of data exchanges through the use of redundant computations in another region (Rodriguez et al.1996). Baer and Zhang (1998) proposed to reconstruct the prediction equations in a format that will allow a larger time step without loss of accuracy. Since the total model run time is determined by the number of cells and time step, an increase of the time step can reduce the model simulation time linearly. In this dissertation, we will conduct a holistic study on how to improve the HPC performance for enabling computing capability of dust storm forecasting from the perspectives of discovering and employing the spatiotemporal patterns that exist in dust storm phenomena, models, and computing resources and through determining the parallelization degree, method, and how to select and arrange computing resources.

2.3 Spatiotemporal Patterns of Dust Storm, Models And Computing Resources

Dust generation and the parameterization of dust deposition processes show high variability spatiotemporal scales and respond in a non-linear way to a variety of environmental factors, such as soil moisture, land cover, and surface atmospheric turbulence (Basart et al., 2009). Spatiotemporal patterns exist in atmosphere phenomena and dust storm process in several aspects (Yang et al., 2011): 1) atmospheric phenomena and dust storm process are heterogeneous so that there may be more activity in some physical regions than in other regions (Koziar et al., 2001). 2) Increasing the spatiotemporal resolution of the models would result in large memory requirements and longer execution time. The cost of explicit 3D hydrodynamics codes is in the order of n^3, where n refers to the grid dimension. Doubling the resolution results in a minimum of an eight-fold increase in computational cost (Baillie et al, 1997). 3) For parallel processing, buffer or halo regions produced by neighboring processors are needed for local computation (Rodriguez et al., 1996). However, the actual number of grid cells selected for overlap and communication depends on the order of the finite difference scheme, which would greatly impact both computation and forecasting accuracy (Nanjundiah, 1998). For example, Equation 2.1 (Rodriguez et al., 1996) indicates that the computation of *df* in any grid cell depends on the values of array *f* on the four neighboring points. 4) Different domain sizes along W-E and S-N directions require different numbers of grid cells along these two directions, and result in different amounts of grid cells to exchange boundary conditions along West-East (W-E) and South– North (S-N) directions. Thus, for the same degree of parallelization, different decompositions can result in different communication overheads (Yang et al., 2011a). 5) Increasing the forecasting period will increase the computation of the dust storm model. 6) Time changes impact the space distribution of the dust in the atmosphere.

$$df(i,j) = \frac{1}{4}.\left(f(i-1,j) + f(i+1,j) + f(i,j-1) + f(i,j+1)\right) - f(i,j) \qquad (2.1)$$

For computing resources, spatiotemporal patterns are primarily present in three aspects: 1) the location of the computing nodes, 2) the network connection between computing nodes, and 3) the storage location and speed.

In this research, the spatiotemporal patterns of dust storm phenomena, models, and computing resources will be utilized to optimize the performance and therefore enable the computability of large area and high resolution dust storm forecasting.

2.4 Earth Science Model Interoperability

The heterogeneity of existing protocols and data models have gained wide acceptance among the Earth Science community (Nativi et al, 2006). Data interoperability are greatly discussed and resolved through the open standards, such as Open Geospatial Consortium (OGC)'s specifications, e.g., Web Feature Service (WFS), Web Map Service (WMS) and Web Coverage Service (WCS, Evans 2003), and a variety of tools and services, e.g., Mapserver, THREDDS (Thematic Real-time Environmental Distributed Data Services). These standards have greatly promoted data interoperability by enabling the Earth science community to share massive datasets. Section 2.1 introduces several popular data models used within the Earth science communities, some protocols and standards (e.g., WFS, WMS, and WCS), and some popular tools used to disseminate the massive Earth science data.

2.4.1 Data Formats

NetCDF is a very popular data format that is used within the Earth sciences community to store the output of weather and climate forecast models. The output of Earth science models is different from many other datasets currently used by the geospatial community. Generally, the data format supports several parameters (e.g., temperature, pressure, wind speed and direction) that vary in three spatial dimensions and involve two distinct time scales (model run time and forecast times) used in these model outputs (Nativi et al, 2006). Through several different client/server protocols, e.g., OPeNDAP (Open-source Project for a Network Data Access Protocol), ADDE (Abstract

Data Distribution Environment) and HTTP, which are already established in the atmospheric and ocean sciences data provider community, the netCDF interface is capable to supporting access to many different file formats , e.g. HDF5, GRIB, GINI, McIDAS AREA, NEXRAD, netCDF-3, netCDF-4, etc.

GRIB (GRIdded Binary) is a mathematically concise data format commonly used in meteorology to store historical and forecast weather data. It is standardized by the World Meteorological Organization's Commission for Basic Systems. Currently there are two versions of GRIB, the first edition (GRIB1) is used operationally world-wide by all meteorological centers for Numerical Weather Prediction(NWP) output. A newer version was introduced, known as GRIB second edition (GRIB2), but it is used only by a few centers and in many cases not for operational broadcast yet. GRIB is an efficient format for transmitting large volumes of gridded data over the Internet.

2.4.2 Standards

The geospatial community uses GIS tools for data analysis and visualization, and has adopted OGC standards extensively. The geoscience community (e.g., atmosphere, ocean, and modeling science) typically uses three main client/server protocols for remote data access: OPeNDAP, ADDE, and netCDF access via HTTP protocol. The following sections introduce the standards and protocols used in both Geospatial and Geoscience communities.

OGC is one of the major geospatial standard organizations. In its one and half decade history, OGC has developed GML (Geograhpical Markup Language), Web Map

Server (WMS), Web Feature Server (WFS), Web Coverage Server (WCS), and many other geospatial standards. These standards define the interfaces or content for exchanging services among GIS web services.

- WMS is an OGC standard that can return a map within a variety of formats based on a user request, including PNG, GIF, JPEG and other raster formats, as well as WEB CGM, or SVG vector forms. Through the HTTP network protocol, WMS supports many operations defined by the URL, such as GetCapabilities, GetMap, GetFeatureinfo, DescribeLayer, GetLegendGraphic, GetStyles, and SetSytles.
- WFS supports insert, update, delete, search and discover services for geographical elements. According to a HTTP client request, WFS returns GML data. Basic WFS interfaces include GetCapabilities, DescribeFeatureType, GetFeature, DescribeFeatureType returning element structure, transaction and other operations.
- WCS is another standard that can be used to dispatch datasets in a variety of formats via standard HTTP to client applications and used by a variety of user groups: the scientific digital library community, the GIS community, as well as the broader Earth science research and education community (Nativi et al, 2006). It provides raster layers that contain geographic information or spatial properties, rather than a static map of access like WMS. WCS has two important operations including GetCapabilities and GetCoverage.

- Web Processing Server (WPS) is collaboratively developed by the OGC and the Open Grid Forum (OGF) and adopted as a standard in 2008. It is a workflow methodology that processes raw data into more valuable information for decision support systems. Processing includes some fundamental GIS operations, such as Union and Intersect.
- OPeNDAP is a data transport architecture and protocol widely used by Earth science governmental agencies, such as NASA and NOAA, to serve satellite, weather and other observed Earth science data. OPeNDAP includes standards for encapsulating structured data, annotating the data with attributes and adding semantics that describe the data. Usually, an OPeNDAP client is a graphics program (like GrADS, Ferret or ncBrowse) or a web application (like DChart) that is linked to an OPeNDAP library (http://en.wikipedia.org/wiki/OPeNDAP). An OPeNDAP server can serve an arbitrarily large collection of data. Data on the server is often in HDF or NetCDF format, but can be in any format including a user-defined format. Compared to ordinary file transfer protocols (e.g., FTP), a major advantage using OPeNDAP is the ability to retrieve subsets of files, and also the ability to aggregate data from several files in one transfer operation.
- REST (Representational state transfer) is an approach for getting content from a Web site by reading a designated Web page that contains an XML (Extensible Markup Language) file that describes and includes the desired content. While the well-known SOAP (Simple Object Access Protocol) has proved to be a powerful

standard for web services, the REST model has seen rapid growth in adoption by web service developers.

2.4.3 Tools

These OGC standards are utilized in many basic GIS platforms and open source GIS software to serve a variety of datasets in different formats. Intergraph has launched a WFS server and provided interoperability development kits. ESRI supports WMS, WFS and other specifications by integrating related components in the ArcIMS. GeoServer and MapServer are popular open source software for providing mapping services. The map server can receive standard WMS and WFS requests, and responds with the requested geospatial data. OpenLayers, a web client supports the interoperable standards, can handle WMS requests, and provide map dragging, zooming, and vector data editing functions.

There are a variety of client/server protocols used by different data and metadata access systems. However, some client applications can access data via some protocols while others can only access data via other protocols. THREDDS (Thematic Real-time Environmental Distributed Data Services) catalogs provide virtual directories of available data and their associated metadata by supplying information about which datasets are available via which services/protocols (Domenico et al., 2002). Therefore, the data access capabilities are augmented and integrated with THREDDS catalog services, which provides inventory lists and metadata access. Thus client applications can find out first what is available on the site via the THREDDS interface, then access the datasets themselves via the OPeNDAP, ADDE, WCS, WMS, or NetCDF/HTTP protocols (Nativi1 et al., 2006).

2.4.4 Related work

For decades, Earth scientists have continually investigated techniques to facilitate the accessibility and sharing of massive amounts of GeoInformation in a transparent manner through data interoperability (Yang and Raskin, 2009). For example, Cao et al. (2009) proposed an interoperable framework to disseminate Earth science data to different application domains through WMS. Within the framework, different Earth science data products and raster snapshots over time can be managed and handled efficiently through the use of relevant metadata information. In addition, a variety of international organizations, such as FGDC, ISO/TC211 and OGC, are also working to advance interoperability.

Efforts have also been made to bridge the gap between different science communities through interoperability, e.g., geospatial communities and geoscience communities. Hu et al (2008) developed a catalog middleware to mediate client/server interactions to share data between OGC catalog clients developed and used by geospatial communities and THREDDS servers greatly used by geosciences communities. Nativi et al. (2006) presented a solution, which develops a couple of data model crosswalks and protocol mediation middleware for THREDDS data servers, to facilitate the interoperability between Geoscience and Earth Science communities.

As data interoperability problems are intensively studied and widely resolved, model interoperability problems are emerging among the Earth science communities. Currently, efforts are focusing on enabling interoperability in the IT field, which aims to advance the reuse and communication of software components. For example, Wiese and Huzita (2006) present IMART, an interoperability model that can be integrated into distributed software development environments, enabling cooperative work among developers and sharing of artifacts produced by different tools. By leveraging a generative programming approach, Damevski (2006) develops a programmable code generator that bridges heterogeneous component instances for component interoperability. Chen and Hogue (2008) presented an automatic method for enabling the interoperability of 3D models within different types of games.

In Earth science, a physical Earth system model typically consists of several model components, which are coupled through the exchange of data. Less research has been done to facilitate coupling models or to make models interoperable among different communities. Currently, the Earth System Modeling Framework (ESMF, http://www.esmf.ucar.edu/) software includes a superstructure for coupling and exchanging data between component models (e.g., atmosphere, ocean) and model subcomponents (e.g., physics, dynamics), and an infrastructure consisting of (1) data structures for representing grids and fields and (2) an optimized, portable set of low-level utilities (Zhou 2006). The CCA (Common Component Architecture) Forum (http://www.cca-forum.org/) was organized to define a minimal set of standard interfaces that a high-performance component framework has to provide to components in order to promote interoperability between components developed by different teams across different institutions. ESMF is designed with specific component interface methods that are common to Earth system model components while CCA provides a more generic component interface that is not specific to any type of application. Zhou et al. (2007) designed and developed an ESMF–CCA prototype to investigate how an Earth system model component that is ESMF compliant can be supported in CCA. Hu and Bian (2009) discussed the identification of equation functions for interoperable hydrological models and built a customized surface runoff model using three components extracted from several existing hydrological models to predict surface runoff of a watershed. Within this approach, strong background knowledge about the environmental modeling and hydrological equations is required.

Based on the analysis above, interoperability enabled research has concentrated on data dissemination, data processing and sharing among different communities and disciplines, while some scientists from both IT and Earth science have worked on component software interoperability. For Earth science models, multiple inputs with strict format are required to execute the models (Xie et al., 2010). Although the required datasets for a model are actually provided online directly, or indirectly, conversion and transformation processes are required. For non expert users, it is very difficult and time-consuming to obtain such datasets and greater effort has to be made on data processes before datasets can be assimilated by the models. However, no systemic study has been done on how to integrate widely distributed data resources to enable the executions of Earth Science models. Moreover, there are situations where different models must work together to tackle complex problems. These problems cannot be resolved efficiently and accurately by only one model without major modifications to the original models (Zhou et al., 2007). This occurs because it is highly possible that a model was only designed and tailored to resolve specific issues and therefore is not interoperable.

In my research, I will investigate interoperability technologies that are needed to enhance dust storm forecasting by facilitating data access services, model input integration, model coupling, and output utilization and dissemination. This research aims to reduce the execution time for both Eta-8bin and NMM-dust and to introduce end-users to model products tailored, in this case, to satisfy the needs of public health services by enabling the interoperable and coupling execution of Eta-8bin and NMM-dust models.

2.5 Nested Models

Nested models are able to well simulate and predict spatial features and resolve processes with small scales on subregions which are subgrids of the coarse grid model. Therefore, through producing the regional high-resolution simulation results without the enormous computational cost of a global model at the same resolution, nested coupling models are used to enable a variety of research and operational applications (Ramón et al., 2002). For example, nested models are greatly used in the research of chemical transport and decomposing process over the atmosphere, and regional air pollution, regional and global weather and climate pattern (Ramón et al., 2002, Wang et al., 2004).Examples of nested modeling atmospheric chemistry transport study are:

- Wang et al.(2004) utilize a global three-dimensional chemical transport model (GEOS-CHEM) to conduct the research of chemical transport over Asia, using

CO as an example. GEOS-CHEM permits the treatment of a limited spatial regime with resolution as 1 x 1 degree, which is much higher than that adopted for the global background (4 x 5 degree).

- A nested grid version of the Regional Acid Deposition Model (RADM) is successfully used to simulate wet deposition amounts of sulfate and nitrate (Pleim et al., 1991). The horizontal grid interval size on the nested area of RADM is three times smaller than that of the coarse area of RADM (80/3 km = 26.7 km).

An example of a fully dynamic grid nested for air quality models is:

- An adaptive resolution system for modeling regional air pollution is reported to be able to simulate the air pollutants with the solution in different subdomains being computed with different spatial resolutions (Constantinescu et al.,2008). The experiment results confirm that adaptive resolution, based on a well-chosen refinement criterion, leads to the decrease in spatial error with an acceptable increase in computational time.

Examples of nested models in weather and climate pattern study are:

- McGregor (1997) utilized components of large-scale atmospheric circulation nested with high-resolution regional dynamical models to predict regional climatic details that were indistinct or even erroneous in medium resolution models but could be more skillfully predicted by the high-resolution model.
- A limited-area high-resolution atmospheric model (80 km as spatial resolution) was nested in the COLA global general circulations model (GCM) (Sela 1980) with 1.88x2.88 degrees as the grid size in longitude and latitude (spatial

resolution) for seasonal climate prediction over North America (Fennessy and Shukla, 2000).

- The experimental results of 15 seasonal winter hindcasts and 15 seasonal summer hindcasts showed that the nested model reduced the systematic errors in seasonal precipitation compared to the global model alone (Fennessy and Shukla, 2000).
- A nested regional climate model was employed to generate a scenario of climate change over the MINK region (Missouri, Iowa, Nebraska, and Kansas) due to a doubling of carbon dioxide concentration (CO_2) in an agricultural impact assessment study (Giorgi et al., 1996).
- Dynamic nesting of models was adapted to forecast regional to global plant migration in response to climate change. In this approach, a global simulation is performed on coarse grids while highly aggregated plant functional types and regional, species based models on finer spatial grids(Nellson et al., 2005).
- Jasper et al. (2002) used numerical weather predictions (NWP) model output with grid cell sizes between 2 and 14km directly taken as input for the hydrological model with 500m×500m grid to advance flood forecasting. The NWP models provided hourly time series of the following meteorological parameter fields: total precipitation, air temperature, wind speed, air humidity, and surface short wave total incoming radiation or net radiation as the input for the grid-based hydrological catchment model WaSiM-ETH (Jasper et al., 2002).

Interoperable process-based models are without doubt the direction of a future generation of Earth science models that offer the flexibility and efficiency to enable a wide range of users to "plug and play" without knowing details of models and modifications of the models (Zhou et al., 2007). However,

- Currently, most models are developed and nested in a "tightly-coupled" approach (McGregor 1997;Sela 1980; Fennessy and Shukla 2000;Giorgi et al., 1996;Nellson et al., 2005). Within this "tightly-coupled" approach, a high resolution model could be one-way, two-way or even triple-way nested with a low resolution model and both are executed together. For instance, at DNMI, the non-hydrostatic MM5 model is nested with HIRLAM, where a domain with 3 km resolution has been set up for the Oslo region in which MM5is one-way nested with HIRLAM (Baklanov et al., 2002). Extensive modifications of both models are required to enable them to be nested because of inconsistent subroutine interfaces, definition of physical constants, data structures etc. (Michalakes et al., 1998).
- Nesting of finer grids into coarser grids requires a priori knowledge of where to place the high-resolution subgrids inside the modeling domain (Constantinescu et al., 2008). For a real-time dust storm prediction system, users are not aware of where the dust storm will occur. Therefore, the traditionally static nesting approach is not suitable for real-time dust storm simulation.
- In addition, specific data and data format are used for both model input and output as described in chapter 1.2.1interoperable models. In this situation, non-expert users are not able to utilize the available distributed data resources to execute an

Earth science model and are not able to integrate multiple models to tackle a complex issue efficiently.

This research will investigate how to integrate widely distributed data resources to enable the execution of Earth science models, as well as how to integrate multi-resolution models to tackle complex problems. Usually, they cannot be resolve efficiently and accurately using only one single model without or with only slight modification from the original model via standard protocols and service interfaces. This would greatly promote the communication and integration of different models in a "loosely-coupled" manner and contribute to flexible/extensible global framework.

2.6 Spatial Cloud Computing

Cloud computing has become a key strategy for IT vendors, ISPs and telecom service providers and many cloud services are available. For example, the most famous and popular cloud services provider, Amazon, offers several kinds of cloud services from IaaS to PaaS. RESERVOIR is an IBM and European Union joint research initiative for cloud computing that will enable massive scale deployment and management of complex IT services across different administrative domains, IT platforms and geographic regions. Verizon's Computing as a Service (CaaS) allows customers to pay for data-center resources such as storage and application hosting dynamically based on the amount of resources they consume (http://www.verizonbusiness.com/products/itsolutions/caas/). Google published several research papers from 2003 to 2006, which outlined a type of PaaS cloud computing. The platform, which is called Google App Engine (GAE), was released to the public as a service in 2008. Increasingly more cloud computing providers are cooperating with others to construct more powerful cloud services. For example, NASA and RACKSPACE are joined by leaders from across technology industries like CITRIX, DELL, NTT DATA, RIGHTSCALE and others on the OpenStack project (http://openstack.org/), designed to create freely available code, badly needed standards, and common ground for the benefit of both cloud providers and cloud customers.

Up to now, several academic programs have put forth effort in investigating and developing technologies and infrastructure for cloud computing, for example, Nimbus (Nimbus, http://www.nimbusproject.org/), Stratus (http://www.acis.ufl.edu/vws/), OpenNebula (http://www.opennebula.org), and Virtual Workspaces (Keahey et al., 2005). Nimbus is an open source toolkit that allows the user to turn his/her cluster into an IaaS cloud (Nimbus, http://www.nimbusproject.org/). OpenNebula is an open-source toolkit used to easily build any type of cloud (private, public and hybrid).

A very important feature of cloud computing is the abstraction of the implementation, meaning that the client is unaware of application deployment details (e.g., where the hardware is located and how it is configured to run the application). This enables cloud computing to provide smart/broader discovery, enhanced access to data and services, on-the-fly integration of applications, and transparent platforms for model running process so that Earth scientists can focus on research without considering the underlying mechanism to implement the time-consuming computational task (Evangelinos and Hill, 2009).

Many studies have been conducted to explore the feasibility of utilizing cloud computing to support Earth science applications and to learn how to best adapt to this new computing paradigm. Huang et al.(2010) test the utilization of cloud computing for geosciences and shows that the EC2 cloud computing platform could provide geospatial applications with good a) elasticity and b) reliability, and c) reduce duplicated efforts among Geosciences communities.

Undoubtedly, in comparison to the current supports for Earth science research and applications, such as parallel computing technology or grid computing technology which only deliver computing power, Earth scientists could benefit more from cloud computing since computing power is only one of the capabilities of cloud computing. However, earlier investigations found that not only could cloud computing help geospatial sciences, but it can also be optimized with spatiotemporal principles to best utilize available distributed computing resources (Yang et al., 2011). Geospatial science problems have intensive spatiotemporal constraints and principles and are best enabled by systematically considering the general spatiotemporal rules for geospatial domains (De Smith 2007; Goodchild 1990; Goodchild et al., 2007; Yang et al., 2011b):

1) Physical phenomena are continuous and digital representations are discrete for both space and time;

2) Physical phenomena are heterogeneous in space, time, and space-time scales;

3) Physical phenomena are semi-independent across localized geographic domains and can, therefore, be divided and conquered;

4) Geospatial science and application problems include the spatiotemporal locations of the data storage, computing/processing resources, the physical phenomena, and the users. All four locations interact to complicate the spatial distributions of intensities;

5) Spatiotemporal phenomena that are closer are more related (Tobler' first law of geography).

Instead of constraining and reengineering the application architecture (Calstroka and Waston 2010), a cloud computing platform supporting geospatial sciences should leverage those spatiotemporal principles and constraints to better optimize and utilize cloud computing in a spatiotemporal fashion.

"Spatial Cloud Computing refers to the computing paradigm that is driven by geospatial sciences, and optimized by spatiotemporal principles for enabling geospatial and other science discoveries within distributed computing environment"(Yang et al., 2011).

In this research, the Amazon EC2 cloud cluster instances will be optimized through the spatiotemporal patterns and principles to enable the computability of dust storm forecasting over a large area with high resolution.

CHAPTER 3
OBJECTIVES

As discussed in chapter 2, many challenges exist for enabling the computability of dust storm forecasting with high spatiotemporal scale requirement, and effective sharing and disseminating of model results, and communicating among different models. This dissertation will address the performance and computing issues of dust storm simulation through the following aspects:

1) Improve the performance and optimize the parallel execution of dust storm simulation by spatiotemporal patterns and principles of phenomena, models and computing resources.

2) Solve the problem of data and configuration interoperability for model interoperability. Currently, most models are developed and tailored for specific applications and professional data and data format are used for both input and output. In this situation, non-expert users are not able to utilize the available distributed data resources to execute an Earth science model and are not able to integrate multi-data resources and multiple models to tackle a complex issue efficiently.

This research will investigate how to integrate widely distributed data resources to enable the execution of Earth Science models, as well as how to integrate multiple models to tackle complex problems which cannot be solved efficiently and accurately using only one single model without or with only slight modification from the original model via standard protocols and service interfaces. This would greatly promote the communication and integration of different models and contribute to a flexible/extensible global framework.

3) Enable large area forecasting through dynamic loosely-coupled nested models. The nested model strategy is recommended to support the requirement for high resolution and larger geographical coverage forecasting while completing the model within the time restrictions (2-hour for one day forecasting). Currently, the research for nested models to produce high resolution results for specific area is focused on 1) coupling the high resolution modules into coarse models (Baklanov et al., 2002), and 2) enabling model execution with different scales at different subgrids(Constantinescu et al., 2008). The former approach require extensive modifications of the original models to enable the tightly coupling of two different models with different data structures (Michalakes et al., 1998), while much modifications efforts are required for the later approach to enable a model to support multi-scale running. In addition, both approaches require knowing where the high resolution subregions are (Constantinescu et al., 2008). As dust storm has the features of different spatial distribution and evolutions, it is hard to predict the exact subregions where it will occur before analyzing the results. Therefore, such a static tightly coupled method is not suitable for dust storm forecasting.

This research will investigate a dynamic loosely-coupled method to enable real-time dust storm predictions providing high resolution results for those areas having dust storm over a large geographic area. In this approach, two independent models are used. The coarse model identify the subregions with the requirement of high resolutions results and the high resolution model then runs those subregions in parallel.

4) Enable the interoperable nesting models through spatial cloud computing. Currently both Eta-8bin and NMM-dust are actively involved in the forecast of PM2.5 for the continental United States. The former can produce low resolution results for a large area while the latter can produce results with high resolution results. However, high resolution results for a larger area and online real-time support for massive users are still at their infancy. Through nesting, the execution of the two models supported with spatial cloud computing optimized via spatiotemporal patterns and constraints, dust storm simulation and prediction can be delivered as a service with high resolution results and fast response in real-time fashion.

Loosely-coupled nested models require a large computing pool to run various hotspots identified by coarse model in parallel to achieve the best performance. This will cause computing spike requirements that can be best handled by elastic and on demand computing platform. Could computing provides a potential solution with a large, virtualized pool of computational resources (Armbrust et al., 2010). Cloud computing technologies, e.g., virtualization, now become more and more mature, and cloud infrastructure become more and more powerful. For example, Amazon EC2 offers cluster instances with 10 Gbps network connection. Each instance has 23Gbytes memory, a clock speed of 2.93 GHz, and two quad-core processors. Such hardware and network configuration are far much better than grid computing environments with heterogeneous computing resources and World Wide network (WAN) connection, and even better than most of private homogenous HPC cluster configurations. This makes cloud computing a new, advantageous computing paradigm to resolve scientific problems traditionally requiring leverage a special high-performance cluster (Rehr et al., 1020). The traditional cluster environment requires typically several months and usually hundreds of hours of labor to procure, install, configure new infrastructure. Traditional processes will involve multiple personnel with different technical backgrounds to ensure the success of the configuration. Often, greater efforts and costs are required to maintain the infrastructure while the utilizing of the infrastructure was used only 10%-15% of its full capacity (Marston et al. 2011). Therefore, the cloud parallel system would have a great impact on parallel scientific HPC applications in the near future by facilitating the deployment process while increasing the rate of utilization of computing resources (Yang et al., 2011b).

In addition, real-time dust storm simulation is a data intensive application, high memory and computing power are required when a large number of users are involved and each user may forecast different areas and harvest thousands of records. There are different access amounts at different periods within one day, and different computing resources should be leased at different time slots during the day. This will cause computing spike requirements that can be best handled by elastic and on demand computing platform. Cloud computing provides a potential solution for pay as you go with a pooled large computing resource (Armbrust et al., 2010). This dissertation will test the feasibility of cloud computing to support the nested execution of two dust storm models on the Amazon EC2 cloud computing platform. Within Amazon EC2, computing resources can be leased on demand, which can satisfy the computational requirements of dust storm forecasting at different times. The dissertation will introduce how to deploy

Earth science models, using NMM-dust as an example, on Amazon EC2 to provide guidance for utilizing cloud computing to support Geosciences applications for Earth scientists. The feasibility of Amazon EC2 to support the parallel execution of a large amount of small regions with scalable cloud resources will be tested.

CHAPTER 4
SPATIOTEMPORAL THINKING & COMPUTING

This chapter will present and discuss the parallelization of NMM-dust model, and experiments how to utilize spatiotemporal patterns and thinking to improve the performance and enable the computability of dust storm simulation. The spatiotemporal patterns enlighten the direction of improving the HPC performance through the way of 1) parallelizing the model(Section 4.1), 2) selecting and arranging the computing resources(Section 4.2), 3) nesting models(Section 4.3), 4) building a spatial cloud computing platform(Section 4.4) and 5) finally, enabling the computability of dust storm models by an optimized solution (Section 4.5).

4.1 Experiment Design

To enable the computability of dust storm forecasting (Koh et al., 2005) for higher resolutions, larger geographic scope, and longer time periods, we designed seven sets of experiments to understand and utilize various aspects of the spatiotemporal patterns of dust storm simulation: 1) *Parallelization degree is* the extent to which the system can effectively utilize an increasing number of processors (Natarajan et al., 1993). We conducted a benchmark for the scalability of the clusters used to support the NMM-dust model for the geographic scope of Southeast U.S. The parallel system used for the experiments is hosted in two geographically distributed facilities. 2) *Parallelization method* is used to test how to properly decompose the same domain under the same parallelization degree to achieve the best performance by reducing the communication overhead while keeping spatial consistency when leveraging more CPU cores. 3)

Subdomain & processor mapping is used to test the performance difference of different methods of dispatching subdomains to processors. 4) *Temporal scope* is designed to exploit the capability of HPC to support long-term dust storm prediction. 3) *Geographic scope* is to explore the HPC capability to support simulations of a large geographic scope. 5) *Spatial resolution* is to analyze the relationship between the spatial resolution and the number of processes and computing resources needed in the prediction. 6) *Network* is used to test the impact of network delay on the performance and to better configure the network connection and assign the computing resources so that we can minimize the network bottleneck and maximize performance. 7) *Storage* to analyze the impact of the file system and storage on the performance. 8) *Dust spatiotemporal pattern* is to utilize the dust spatial distribution and evolution patterns to obtain higher resolution results for areas with dust storms.

To investigate the improvement of performance and the increasing of the computing time as the parameter changes, such as spatial domain and resolution, we calculate the performance improvement factor (or computing time increasing factor) S (Equation 4.1):

$$s=\Delta t/T \qquad (4.1)$$

Where Δtm is the decrease in the computing time of dust-storm simulation, and T is the original computing time before parameter changes or parallel systems are optimized.

4.2 Experimental Environment

Two facilities (A & B) are used for this research: Facility A has 25 computing nodes and all nodes are connected through fast local area networks (LANs with 1Gbps). Each node has 24 Gbytes memory and two quad-core processors (8 physical cores) with a clock frequency of 2.33 GHz, a peak performance of 7.6 Gflops/core and a sustained performance of 1 Gflop/core. Facility B has a 10 Gbps network and 14 computing nodes, and each node has 96 Gbytes memory and dual 6-core processors (12 physical cores) with a clock frequency of 2.8 GHz. Most experiments are conducted on both facilities and we expect to obtain better results from facility B given its better configuration. Facilities A and B are connected through a Wide Area Network (WAN).

4.3 Parallelization

In this section, we present the NMM-dust model parallelization and how to utilize spatiotemporal patterns to optimize the parallelization.

4.3.1 Parallelization of the Model

Similar to other atmospheric model parallelization, a data decomposition approach is used for parallelization by decomposing the domain into multiple subdomains and distributing the computing load of each subdomain onto one processor as a process.

Figure 4.1 shows parallelizing a (4.5 x 7.1 degree) domain with a spatial resolution 0.02083 degree to 24 subdomains for one vertical layer. Based on the boundary size and spatial resolution, the grid cells of one vertical layer would be 215 x 345 while there would be 54 x 57 grid cells for each subdomain except for suddomains on the border. The processors processing the subdomains will need to communicate with their neighbor processors for local computation and synchronization. The processors being responsible for processing the subdomain within the inner of the entire domain such as subdoamin 5 will require communication among four neighbor processors. During the computation, the state and intermediate data representing the subdomain are produced in the local memory of a processor. Other processors need to access the data through file transfer across the computer network. The cost of data transfer due to the communication among neighbor subdomains is a key efficiency issue because it adds significant overhead (Baillie et al. 1997).

Subdomain 20 (54 x 57)	Subdomain 21 (54 x 57)	Subdomain 22 (54 x 57)	Subdomain 23 (53 x 57)
Subdomain 16 (54 x 57)	Subdomain 17 (54 x 57)	Subdomain 18 (54 x 57)	Subdomain 19 (53 x 57)
Subdomain 12 (54 x 57)	Subdomain 13 (54 x 57)	Subdomain 14 (54 x 57)	Subdomain 15 (53 x 57)
Subdomain 8 (54 x 57)	Subdomain 9 (54 x 57)	Subdomain 10 (54 x 57)	Subdomain 11 (53 x 57)
Subdomain 4 (54 x 57)	Subdomain 5 (54 x 57)	Subdomain 6 (54 x 57)	Subdomain 7 (53 x 57)
Subdomain 0 (54 x 58)	Subdomain 1 (54 x 58)	Subdomain 2 (54 x 58)	Subdomain 3 (53 x 58)

Figure 4.1. Parallelizing a (4.56 x 7.12 degree) domain with 0.02083 degree spatial resolution (about 3 km, 215 x 343 grid cells in total) to 24 subdomains (4 x 6)

Figure 4.2 illustrates all the NMM-dust core subroutines in the computing sequence for each subdomain (the subroutines with blue color require communication and synchronization): 1) PDTE is the process to integrate mass flux divergence, compute vertical velocity and update the pressure field. This process requires the communication and synchronization of the hydrostatic surface pressure PD (Pa) among neighboring processors. 2) ADVE is used for the horizontal advection of the variables of temperature T(K), u wind components U(m/s), v wind components V (m/s), and coriolis effect and curvature terms are applied. The communication among U, V, and geopotential height z is required for the subroutine ADVE. 3) VADZ is the process of vertical advection of geopotentional height and no communication is required as it happens in the vertical layers. 4) HADZ is used for the horizontal advection of height and the vertical wind speed W (dz/dt) is updated and the synchronization of W is required before the subroutine. 5) EPS is used for both vertical and horizontal advection of dz/dt and the vertical wave treatment is added in the subroutine.

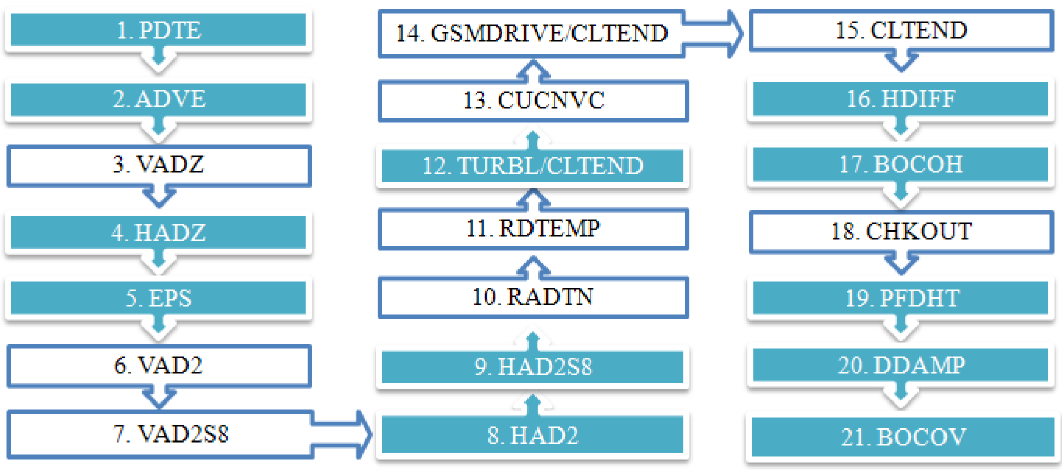


Figure 4.2. Computing subroutines and communication & synchronization for NMM-dust model

6) The following subroutines are vertical advection of the variables of specific humidity q, total cloud water condensate CWM (kg/kg), turbulent kinetic energy (m/s), 7) eight classes of dust particle load (s8). 8) The horizontal advection of the variables of turbulent kinetic energy Q, total cloud water condensate CWM(kg/kg), 2* turbulent

kinetic energy Q2 (m2/s2), and 9) eight classes of dust particle load (s8), local halo data of q, CWM, Q2 and s8 are communicated with neighboring processes. The following subroutines are 10) RADTN for radiation and 11) RADTEMP for applying temperature tendency due to radiation. 12) TURBL is to perform vertical turbulence and store original temperature array (CLTEND) and the communication of pd, T, q, CWM, s8, UZ0 and UZ0 is conducted. 13) CUCNVC is convective precipitation. 14) GSMDRIVER is to grid scale microphysics and then store the original temperature array (CLTEND). 15) Following is the subroutine to update temperature tendency due to cloud processes. 16) HDIFF is used for horizontal diffusion and the communication of T, q, U, V, Q2 and S8 needs to be exchanged. 17) BOCOH is to update boundary conditions for subdomains while the exchange of q, CWM, Q2, S8, pd, and T are required. 18) Every three hours in this case, the post profile data subroutine CHKOUT will be performed. 19) PFDHT is the subroutine to calculate pressure gradient force (PGF), update winds due to PGF, and compute divergence, and the variables pd, T, U, V, q, CWM(kg/kg), dw/dt, and non-hydrostatic pressure PINT (Pa) are exchanged. 20) DDAMP is used for divergence damping and the variable div would be exchanged. The final subroutine 21) BOCOV is used to update boundary conditions at the wind points and the communication of U and V as required.

4.3.2 Parallelization Degree

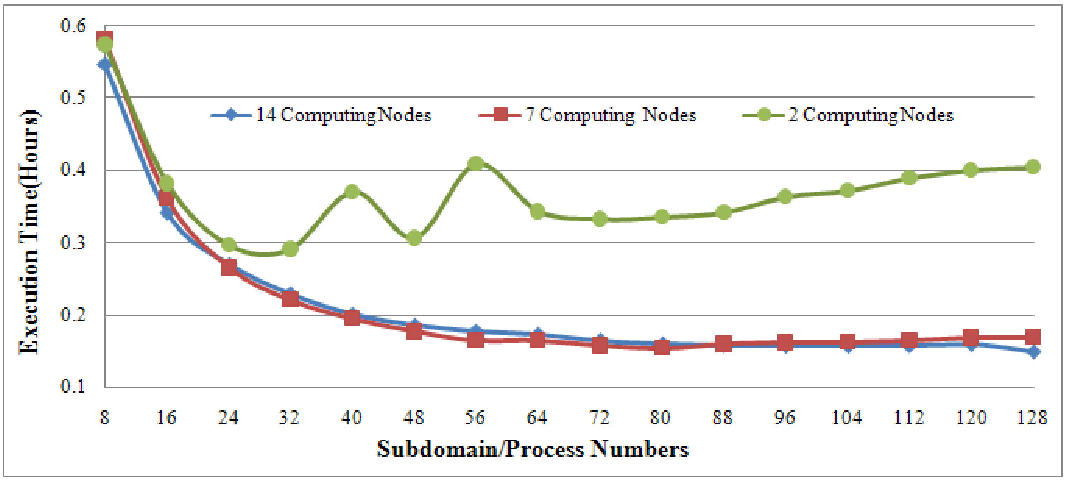


Figure 4.3. Scalability experiment with different computing nodes and different subdomain numbers to run the NMM-dust model over a rectangular area of 4.5 x 7.1 degree in the southwest US for 3 km resolution, and for 3-hour simulations

We parallelized the geographic scope of 4.5 * 7.1 degree along the longitude and latitude respectively into different sub-domains and utilized different CPU numbers and subdomain numbers to test the performance (Figure 4.3).The total execution time greatly decreases when increasing the process (subdomain) number from 8 to 16, and then to 24. After that, the execution time is still reduced but not significantly, especially when two computing nodes are used. The reason behind this, as shown in Figure 4.4, is the increase in communication and synchronization time that is observed to increase gradually until equal to the computing time when using 96 processes. The experiment result in Figure 4.3 also shows that the execution times of the model with different domain sizes converge to roughly the same values when the number of CPUs increases. The cases where 7 and 14 computing nodes are used yield similar performance. Especially, when more and more processes are utilized, seven computing nodes could have a little better performance than 14 computing nodes.

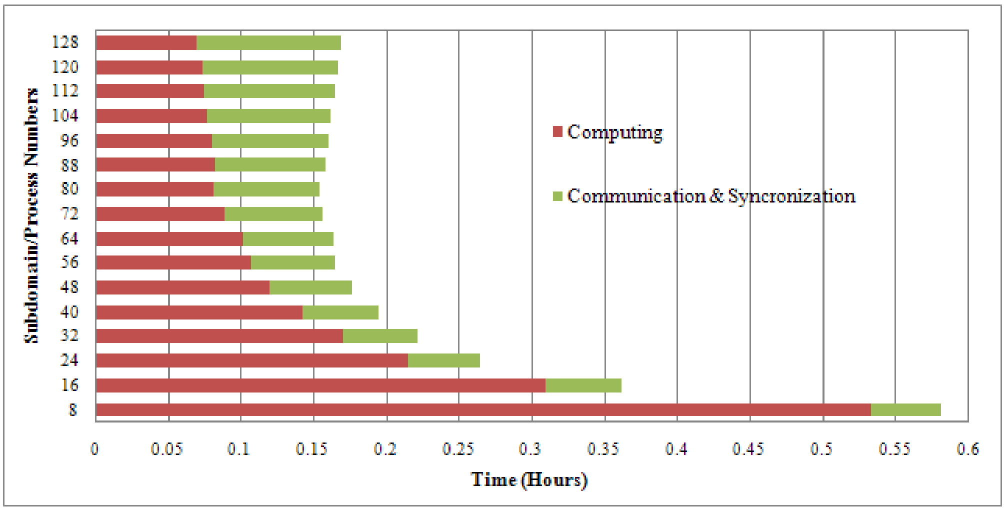


Figure 4.4. Comparison of the total computing time and communication & Synchronization time with different numbers of subdomains/processes involved using 7 computing nodes

In summary, this experiment demonstrates that the communication overhead could result in two scalability issues for the dust storm simulation: 1) no matter how many computing nodes are involved, there is always a peak performance point of the highest number of processes that can be leveraged for a specific problem size. The peak point is 128 processes for 14 computing nodes, 80 processes for 7 computing nodes, and 32 for 2 computing nodes; and 2) a suitable number of computing nodes should be used to complete the simulation.

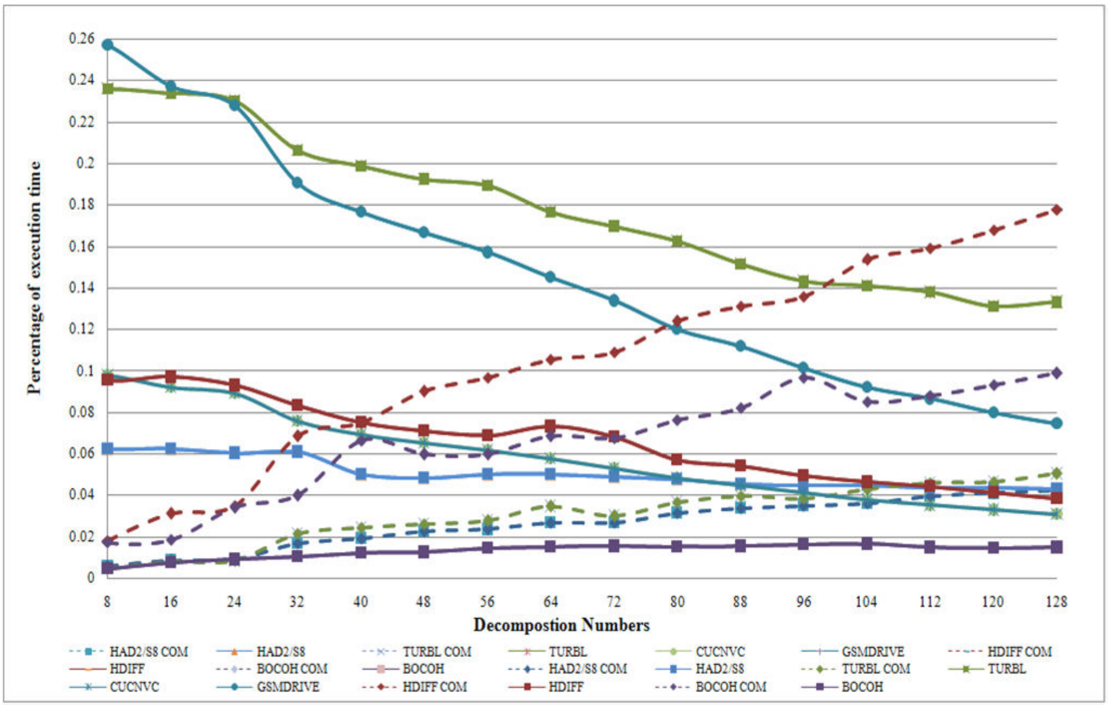


Figure 4.5. Percentage analysis for different subroutines and communication & synchronization

To further investigate how the communication overhead impacts scalability, we experimented and analyzed the time spent on each subroutine and their communication & synchronization. Figure 4.5 shows the percentage of the time spent executing each subroutine and the time spent on communication & synchronization: 1) Generally, the higher the number of subdomains, the smaller the percentage of computing time and the greater the communication & synchronization time required for updating each subdomain’s boundary conditions except BOCOH. 2) The subroutine TURBL is the most time consuming and more subdomains will decrease the time percentage of TURBL. 3) HDIFF will spend more time on communication & synchronization when increasing the subdomains up to 96. BOCOH's communication and synchronization time is also very high and becomes an important performance factor when increasing the number of subdomains. 4) The subroutines GSMDRIVE and CUCNVC's computing time decreases when increasing the number of subdomains and they do not require communication and synchronization.

4.3.3 Parallelization Method

Different parallelization along the longitude and latitude could result in different communication overhead. For example, Figures4.6a and 4.6b show two types of decomposition method. Method 4.6b requires 6grid cells' communication over the processors while method 4.6a requires only 4 grid cells' communication over the processors. Obviously, method 4.6b results in more data dependency and communication overhead.

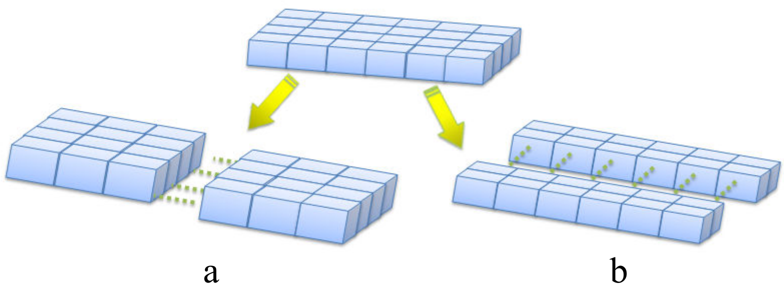


Figure 4.6.Diagrams of 2 X 1 and 1 X 2 at longitude and latitude decompositions for 4 x 6 grid cells

Experiment of parallelization method tests the parallel implementation through different decompositions of 24 subdomains along S-N and W-E directions for the same domain (Figure 4.7). It is observed that a one-dimensional decomposition in both longitude and latitude alone is a bad idea for parallel implementation as the 24*1 (24 columns at the longitude and only 1 column at the latitude) has the worst performance followed by the 1*24 (1 column at the longitude and 24 columns at the latitude). In addition, less decomposition along the longitude direction is preferred as 3, 4 and 2 decomposition along the longitude have higher performance. This is because the grid cells (215 x 345) along the longitude are less than the latitude. Therefore we can parallelize the whole region according to the length and width of the region to reduce the communication overhead and optimize the performance.

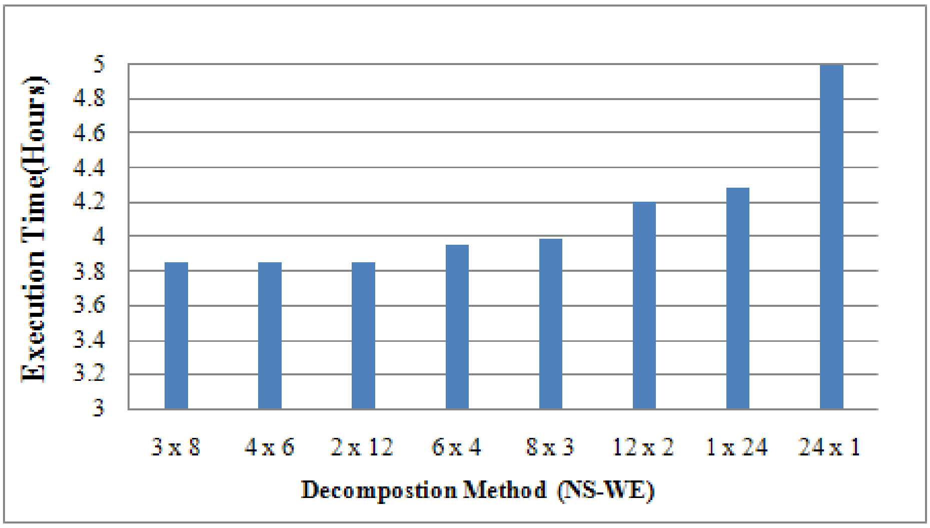


Figure 4.7. Domain Decomposition experiment using different decomposition methods along the longitude and latitude to parallelize the domain into 24 subdomains to run the NMM-dust model over a rectangular area 4.5 x 7.1 degree in the southwest US with 3 km as spatial resolution.

4.4 Computing Resources Arrangement

This section will introduce how to select and arrange computing resources based on the simulation parameters, including spatial domain size, spatial resolution, and temporal scope. In addition, we will discuss how to select and allocate computing resources, including computing nodes and storage based on network connection and topology.

4.4.1 Temporal Scope

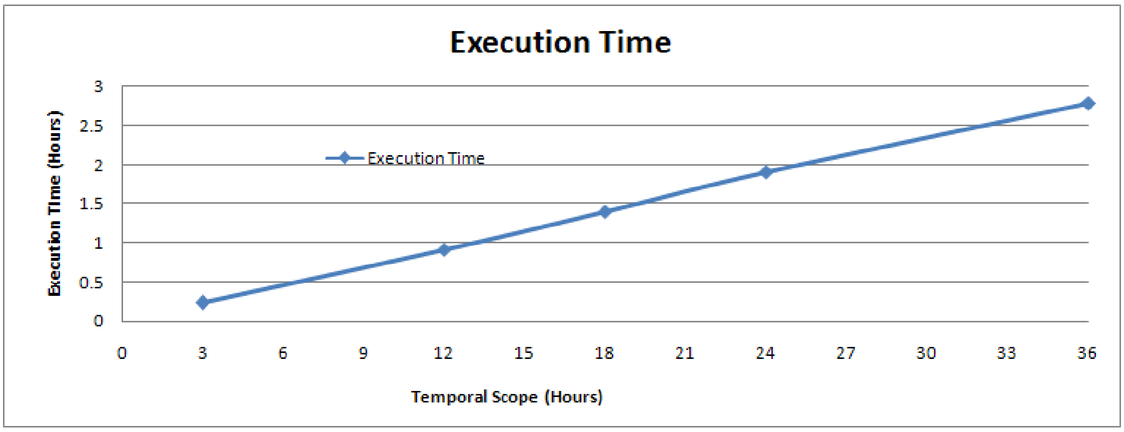
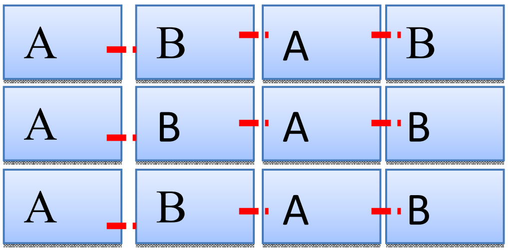


Figure 4.8. Execution time for running the NMM-dust storm model in facility A. 40 process numbers are used for domain 4.5 x 7.1 degree with 3 km resolution for 3, 12, 18, 24 and 36-hour simulations

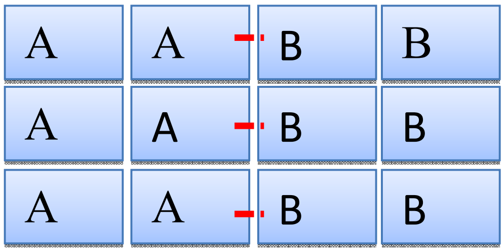
Figure 4.8 shows the total computing time needed for the predictions of 3, 12, 18, 24, and 36 hours executed on 24 computing nodes at facility A. The experiment results show that the computing time would increase linearly when increasing the temporal scope. This temporal pattern is very useful for designing and conducting experiments to test the computability of dust storm forecasting. This is because it enables us to estimate the computing time for a long-term forecasting through a short-term simulation. Usually, one needs to do a one-day simulation to know if the computation can be successfully completed within 2hours.Now, instead, we can do a 3-hour simulation to test if the one area forecasting is computable (Lenz, 2001). This means that we can test if a 3-hour simulation can be successfully completed in 0.25 hour (2 /8).

4.4.2 Subdomain & Computing Resources Mapping

Different subdomain & processor mapping methods could result in different communication overheads. Figures 4.9a and 4.9b show two mapping methods for dispatching 12 subdomains to two computing nodes A and B. Method 4.9b requires only six grid cells communications over two different computing nodes while method 4.9a requires 18 adjacent boundaries to exchange data over two computing nodes. Obviously, method 4.9b can reduce the communication overhead by making more communication occur within the same machine rather than over the Internet.



a. Non-neighbor mapping



b. Neighbor Mapping

Figure 4.9.Two mapping methods for dispatching 12 subdomains to two computing nodes A and B.

By default, the MPICH2 will dispatch the subdomains to the computing nodes sequentially. For instance, if we have two computing nodes and six subdomains, then the first, third and fifth subdomains will be dispatched to the first computing node and the second, fourth and sixth subdomains will be dispatched on the second computing node. Therefore, MPICH2 is the typical non-neighbor mapping method. In the subdomain &

computing resources experiment, two computing nodes are utilized and the first half subdomains are dispatched on the first computing node and the rest are dispatched on the other computing node, which is the neighbor mapping. The experiment results of different mapping methods (Figure 4.10) also support that if we map the neighbor subdomain to the neighbor processor, much higher performance can be obtained. Therefore, this pattern indicates that we should dispatch neighbor subdomains to the same computing node as much as possible to reduce the communication over long distance Internets.

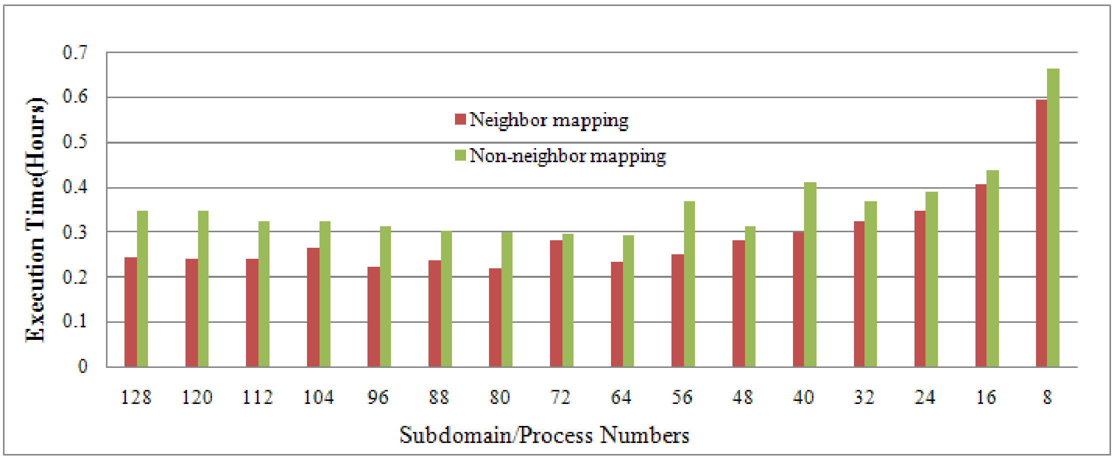


Figure 4.10. Execution time for running the NMM-dust storm model on 2 cloud clusters with mapping and non-mapping methods.

4.4.3 Spatial domain

Figure 4.11 shows the geographic scope experiment results using different process numbers initiated by 24 nodes from facility A to run the NMM-dust storm model over geographical scope 2.3 x 3.5, 4.5 x 7.1, and 9 x 14.2degree with 3 km spatial resolution for 3-hour simulations. It is observed that increasing the geographic scope greatly increases the computing time.

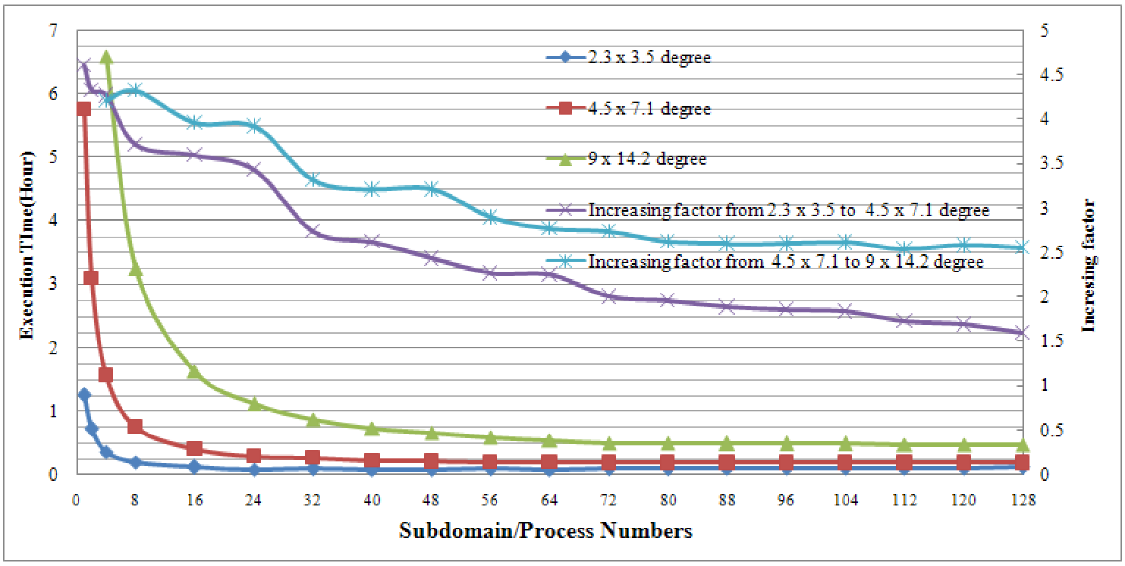


Figure 4.11. Geographic scope experiment using different process numbers with facility A nodes to run the NMM-dust storm model over domain 2.3 x 3.6, 4.5 x 7.1, and 9 x 14.2 degree with 3 km spatial resolution for 3-hour simulations.

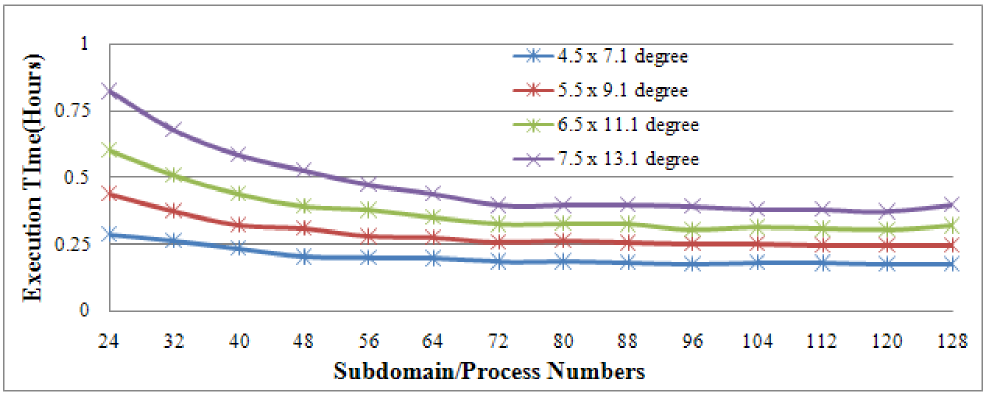


Figure 4.12. Geographic scope limitation experiment for facility A

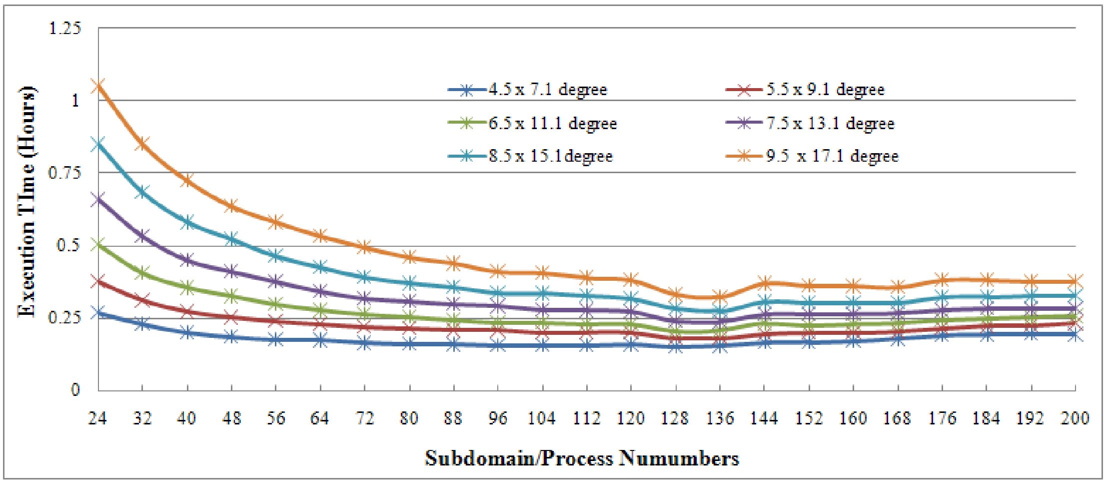


Figure 4.13. Geographic scope limitation experiment for facility B

Table 4.1. Geographic scope experiment over 2.3 x 3.5, 4.5 x 7.1, and 9 x 14.2 degree in facility A with 1, 2 and 4 processes

Spatial domain	2.3 x 3.5 degree	4.5 x 7.1 degree	9 x 14.2 degree
Grid numbers	IRow = 107 , IColumn = 177	IRow = 215 , IColumn = 343	IRow = 431 , IColumn = 683
1 process	1.25 (Hours)	5.77(Hours)	N/A (Model is not able to run)
2 Processes	0.717(Hours)	3.1(Hours)	N/A (Model is not able to run)
4 Processes	0.367(Hours)	1.57 (Hours)	6.6 (Hours)
8 Processes	0.202 (Hours)	0.757(Hours)	3.72(Hours)

Figure 4.11 also shows the increasing factor after doubling the geographic scope from 2.3 x 3.5 to 4.5 x 7.1 and then to 9 x 14.2 degree with different process numbers from 1, 2, 4 , 8 , and to 128. It is observed that: 1) doubling the geographic scope from 2.5 x 3.5 to 4.5 x 7.1 would result in approximately a four-fold increase in the model execution time compared to the cases when using one and two processes. Approximately, the model execution time increases 4.2 times when doubling the geographic coverage from 2.5 x 3.5 to 4.5 x 7.1 and from 4.5 x 7.1 to 9 x14.2 when using 4 processes; 2) Increasing the process number from 1 to 128 will reduce the increasing factor from 4.5 to 1.5; and 3) after increasing the geographic coverage to 9 x 14.2, one or two processes are not able to run the model because of memory constraints. In addition, it is observed that the system requires at least 0.48 hours, which is more than the 0.25 hours needed to complete a 3-hour simulation for a region of 9 x 14.2 degree. The other two experiments are conducted to find that facility A can support a maximum geographic scope of 5.5 x 9.1 degree (Figure 4.12) and facility B can support up to a geographic scope of 8.5 x 14.1 degree (Figure 4.13).

4.4.4 Spatial Resolution

Figure 4.14 illustrates the computing time for 3 km and 2km resolution executed by 8 to 128 processes using facility A. In this experiment, the simulation area is 4.5 x 7.1 degree and 37 vertical levels for the southwest U.S. with a 3-hour temporal scope and 3 km and 1.5 km as spatial resolution. The time step is determined by using 2.25 x (x is grid spacing in km) or about 330 x (angular grid spacing) to obtain an integer number as time step with second as the unit (WRF-NMM, 2011). Therefore, the time steps are set to 6 s and 3 s respectively for the 3 km and 1.5 km resolutions.

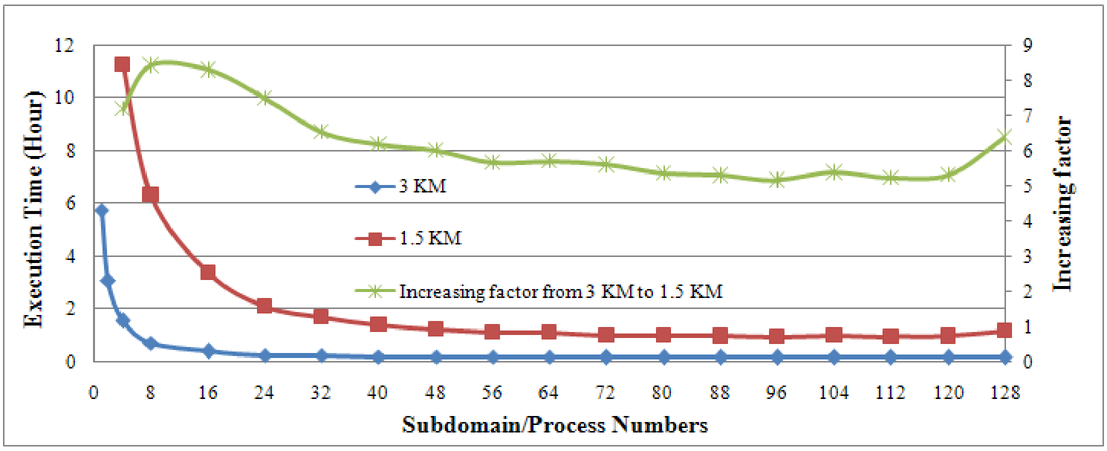


Figure 4.14. Spatial resolution experiment uses a simulation area of 4.5 x 7.1 degree x 37 levels in the southwest US with grid points spaced with 3 km and 1.5 km on 8, 16,…, 128 processes. The time steps are set as 6s and 3s respectively for the two resolutions.

Based on the result of the experiments, increasing the spatial resolution of the NMM-dust model, the computation time increases significantly. In addition, doubling the spatial resolution from 3 km to 1.5 km would result in about an 8-fold increase in the computing time. The experiment also indicates that, with an increase in the resolution to 1.5 km in each dimension, it is not possible to complete the NMM-dust model simulation in the distributed system for the domain of 4.5 x 7.1 degree within the time constraint of completing a 3-hour simulation within 0.25 hours. Therefore, although the NMM-dust model is able to provide high resolution results, it is not practical for this model to conduct real-time predictions with such high resolutions. In addition, with one or two processes, the model cannot even be started (see Figure 4.14 and table 4.2), and the facility A can only support a resolution of up to 3 km with the forecasting time constraint

(Figure 4.15). It would require redesigning the existing algorithms, code, and data structures of NMM-dust core modules.

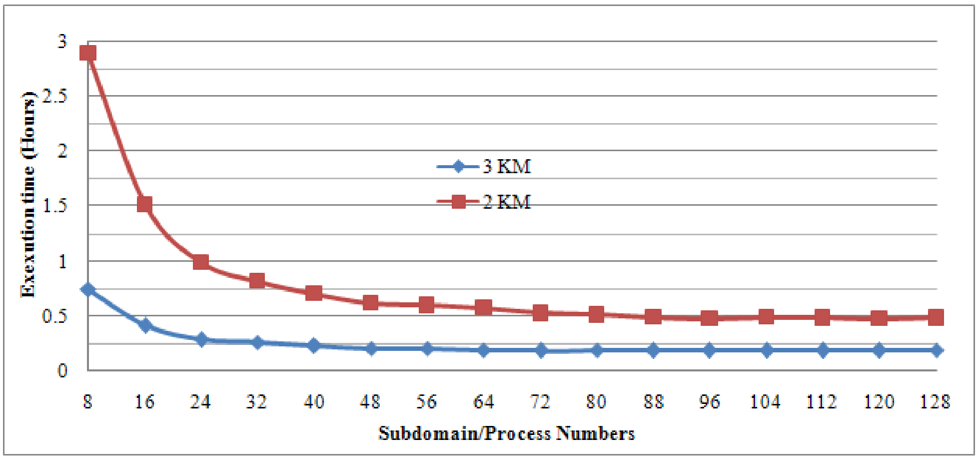


Figure 4.15. Spatial resolution limitation experiment for facility A

Table 4.2. Spatial resolution experiment over 4.5 x 3.5 degree with 1.5 km and 3 km in facility A with 1, 2 and 4 processes

Spatial Resolution	1.5 km	3 km
Time step	3s	6s
Grid numbers	IRow =433, ICcolumn =680	IRow = 215, IColumn = 343
1 process	N/A (Model is not able to run)	5.766667(Hours)
2 Processes	N/A (Model is not able to run)	3.1(Hours)
4 Processes	11.267 (Hours)	1.57 (Hours)
8 Processes	6.34(Hours)	0.751(Hours)

4.4.5 Network

To analyze the network impact, the execution time analysis is performed based on the result of a 3-hour dust storm simulation executed on four groups of different computing node numbers with three different network speeds at facility A and facility B. Figure 4.16 illustrates the execution time of different process numbers from these two computing nodes with an access rate to the local network of 10 Gbps, 1 Gbps and 100Mbps over the WAN. Compared to the CPU and memory factors, the network connection is more important. The performance when using 2 nodes located at two different facilities A and B is much worse than when using 2 nodes both located in the same facility A.

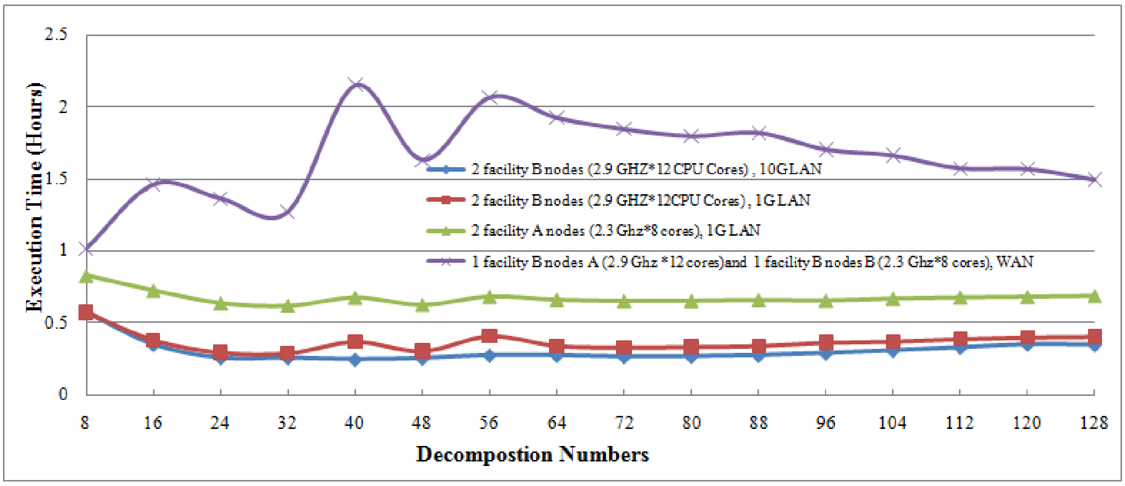


Figure 4.16. Compared to the CPU and memory factors, the network connection is more important. The performance when using 2 nodes located at two different data centers is much worse than when using 2 nodes both located in the same data center.

The result (see Figure 4.17) shows that the performance improvement from a 1Gpbs LAN connection to a 10 Gpbs connection using 2 facility B nodes is usually more than 10% and could reach 30% when using 40 and 56 processes. Although it may not be important for long-term, fine resolution predictions that may take longer time, it can be very significant for real-time, short-term dust storm simulations when results are desired in less than one hour (Yang et al. 2011a). In addition, the improvement from using WAN to using LAN is even more significant with almost 60% gain in average based on the performance comparison analysis, which is conducted between nodes from both facilities connected with WAN and both two nodes from facility A connected with a 1 Gpbs network.

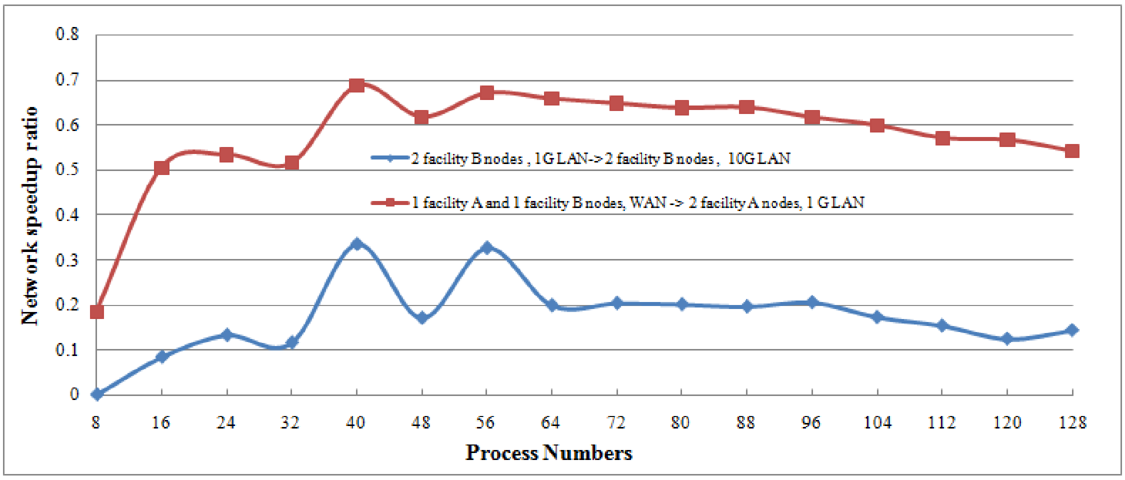


Figure 4.17. Network speedup ratio

4.4.6 Storage

During the simulation, each process will produce temporary files for its subdomains to integrate after simulation. The experiment results (Figure 4.18) demonstrate that it is possible to get much better performance if using local storage to store the temporary files and then transfer results to the master node after finishing the simulation than when using NFS to share remote storage. This is because all NFS processes will access the same remote storage and transfer data to master node in real-time. The patterns indicate that a good storage strategy is helpful in reducing the I/O cost and the network bottleneck for data intensive applications.

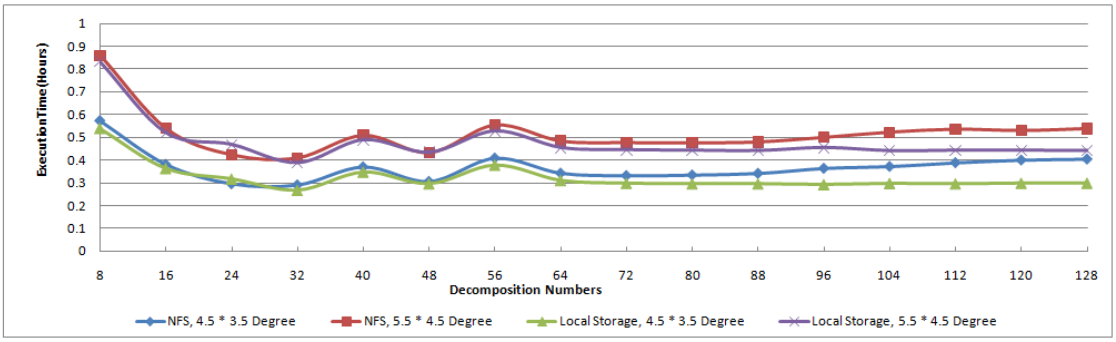


Figure 4.18. File System experiment using different storage model to run the NMM-dust storm model over domain 4.5*5.5, 5.5* 4.5, and 6.5 *5.5 degree in the southwest US with 3 km resolution, for 3-hour simulations and predictions.

4.4.7 Computing Capability

As the problem size increases, such as the increase of the spatial domain, spatial resolution, and temporal scope, more and more computing resources are required based on the experiment results of temporal scope, spatial domain, and spatial resolution. However, for a certain number of computing resources, the best number of processes that should be started is predictable. To demonstrate this, we use different numbers of computing resources arrangement to perform a series of simulations with different domain sizes. Figures4.19, 4.20, 4.21, and 4.22showthe simulation time for different domain sizes running on facility A and facility B using different numbers of computing nodes. The results indicate that 80 domain decompositions for different domain sizes yield the best performance in facility A. Figure 4.19 shows facility B supports different domain simulations using 14 computing nodes. Around 128 processes yield the best performance.

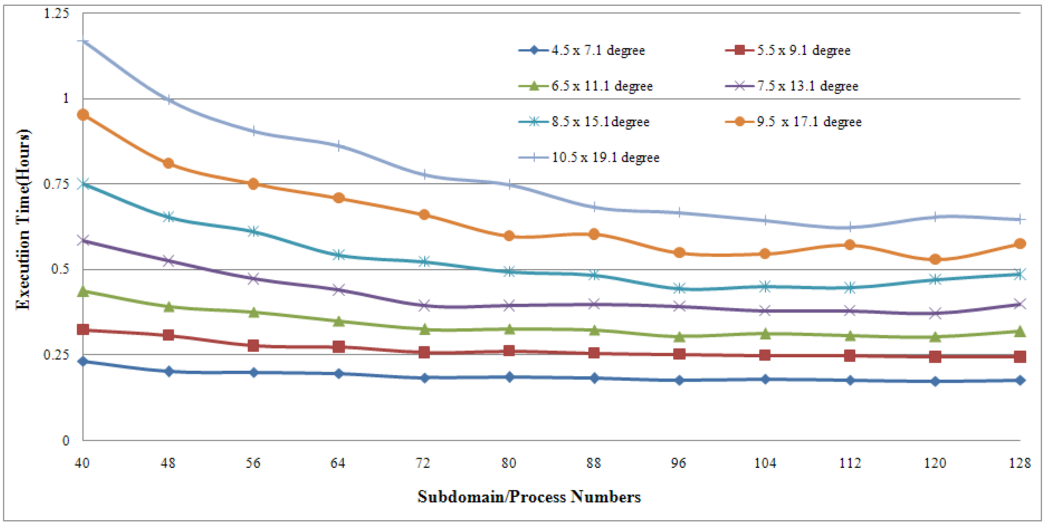


Figure 4.19. Facility A supports different domain simulations using 24 computing nodes. Around 96 processes result in the best performance.

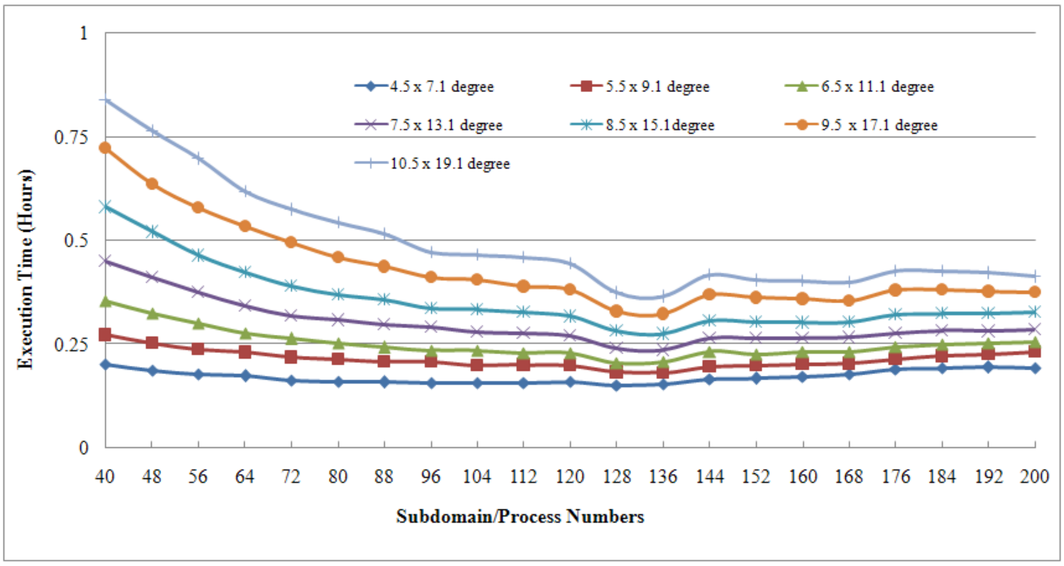


Figure 4.20. Facility B supports different domain simulations using 14 computing nodes. Around 128 processes result in the best performance.

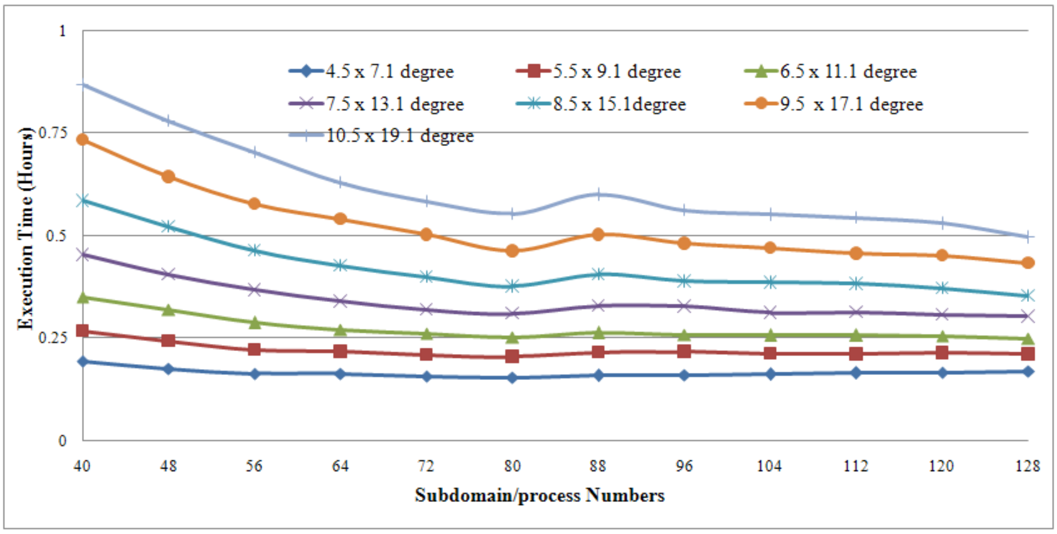


Figure 4.21. Facility B supports different domain simulations using 7 computing nodes. Around 80 processes result in the best performance.

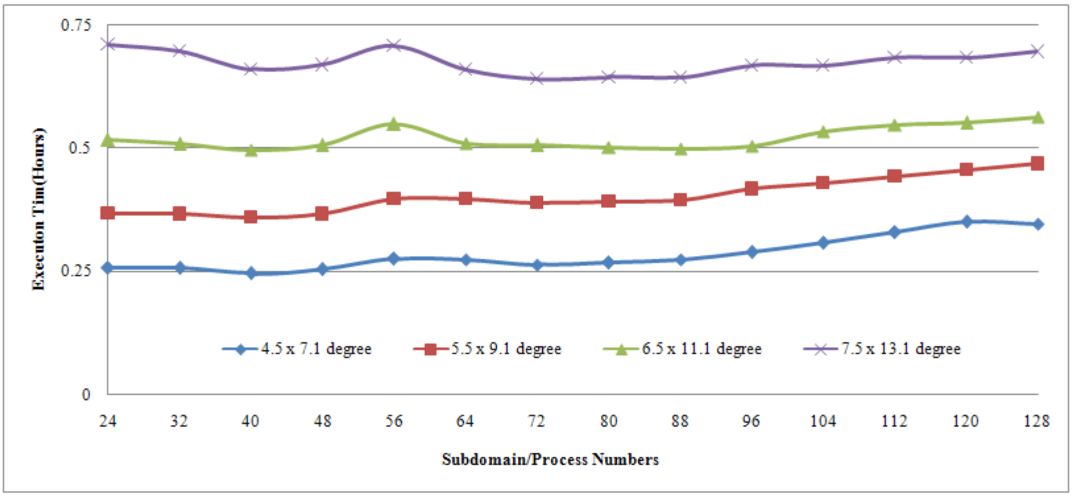


Figure 4.22. Facility B supports different domain simulations using 2 computing nodes. Around 40 processes result in the best performance.

Different computing resources with different configurations have different computing capability. For example, facility A can support a maximum geographic scope of 5.5 x 9.1 degree (Figure4.12) and facility B can support up to a geographic scope of 8.5 x 14.1 degree (Figure 4.13). Different numbers of computing nodes are used for the same task to demonstrate the computing capability of different amounts of computing resources.

From the previous results, we can successfully assign suitable computing resources and process numbers to complete the simulation within the time criteria and achieving the best performance.

4.5 Dust Spatiotemporal Pattern

The existence of dust spatiotemporal patterns is the reason that enables the implementation of loosely coupled nested models, which will be introduced in details in the chapter 5. In this approach, spatiotemporal correlation analysis is performed on low resolution results of dust storm model to identify high concentration but much smaller regions. The spatiotemporal pattern of dust storms indicates which and how many regions require computationally expensive simulations while reducing the computing time and satisfying the requirement for high resolution results.

4.6 Experimental Result Analysis

Table 4.3. Experimental results for better leveraging HPC with spatiotemporal patterns

Experiment		**How to leverage distributed HPC systems**	**Reason**
Parallelization	1.Parallelization degree	A proper number of computing nodes and processes should be planned for higher performance	As more processes participate, more communication is incurred, resulting into more synchronization and exchanges of boundary

			information. More computing nodes cause more communication in the network.
	2.Parallelization method	Proper decomposition method should be used to obtain higher performance	For the same degree of parallelization, different decomposition can result in different communication overhead
Computing resource arrangement	3. Subdomain & processor mapping	Dispatching subdomains to computing resources based on the spatial relationships of subdomains, as well as computing resources	Much higher performance can be obtained through mapping the neighbor subdomains to the neighbor processors
	4.Temporal scope	More computing resources, faster CPUs, and better network connections are required when increasing the temporal scope to enable the simulations to be completed within a reasonable time period.	A **linear increase in the** simulation period would result in a linear increase in the computing time.
	5.Geographic scope	An increasing of the computing resources, faster CPUs and better network connections are required for larger domain simulations	Around a **four-fold** increase of the computing time could be caused by doubling the geographic domain.

	6.Spatial resolution	An increasing of the computing resources, faster CPUs and better network connections are required for higher resolution simulations	Around an **eight-fold** increase of the computing time could be caused by doubling the spatial resolution.
	7. Network	Compared to CPU speed, the network bandwidth between the computing nodes and the geographical locations of the computing resources could be more important.	The network is a bottleneck for data intensive applications.
	8. Storage	Temporary files produced while the model is running should be stored on local storage rather than transferred to the master node in real-time.	Keeping the temporary files on local storage helps in reducing the I/O and network bottlenecks.
Dust storm spatiotemporal pattern	9. Dust storm spatiotemporal pattern	Divide and run hotspots area, which is identified based on the spatial pattern and temporal evolution of dust storm, in parallel	Dividing a large geographic scope into multiple small geographic coverage areas based on the spatiotemporal pattern of dust storm produces small areas whose simulation takes much less time to complete.

The eight sets of experiment results reveal spatiotemporal patterns and provide guidance for configuring and scheduling the HPC facilities for better performance. Table

4.3 shows how to integrate the experiment results into practical applications: 1) Experiment 1 indicates that the scheduler should incorporate the ability of automatic calculation of how many nodes and processes should be utilized whenever dispatching the tasks to a distributed computing pool to increase both the CPU utilization and performance.2) The experiment result of parallelization (Experiment 2) demonstrates that the overhead due to communication and synchronization among sub domains can be reduced through proper decomposition of the same domain under the same parallelization degree. 3) Experiment 3 results also support that if we mapping the neighbor subdomain to the neighbor processors, much higher performance can be obtained. The neighbor domain assign to the same machine to reduce the communication overhead by making the communication occurring in the same machine without going through the network, therefore increasing the parallelization degree.4) Based on the results of experiment 4, 5 and 6, the system should leverage more computing resources for different problem sizes with different temporal scopes, geographic coverage and spatial resolutions and. 5) Experiment 7 indicates that the spatial relationship of computing resources and network connections between computing resources is very significant to the performance, and its impact would be higher than CPU core numbers and speed. 6) Experiment 8 suggests that a good temporary file storage strategy would greatly improve the performance.7) Finally, Experiment9 indicates that the spatiotemporal pattern of a phenomenon can be utilized to enable the computability of a large geographic coverage simulation.

CHAPTER 5

NESTED DUST-STORM MODELS AND INTEROPERABILITY

Scientific research and societal benefit applications in the 21st century require better understanding of the past and prediction of the future for better decision support by integrating model simulations from different domains for different regions. This urgent need poses grand challenge in model integration to meet the 1) scientific accuracy and 2) computability of integrated models for high resolution results. This chapter tries to address the 2nd problem and reports the investigations on using interoperability technologies and nested model approach to enable the computability of data intensive Earth Science applications, using dust storm forecasting as a case study.

The approach adopting loosely-coupled interoperable nested models 1) enhances the model interoperability by enabling the utilization of multiples data resources as the input for the two models, Eta-8bin and NMM-dust models, communication between the two heterogeneous models, and coupling runs of the two dust models, 2) reduces the execution time for both Eta-8bin and NMM-dust dust storm models to reasonable timeframe, and 3) produces acceptable resolution in a timely manner by considering the size and movement of dust storm to serve public health applications and the public for quick response and preparation for the severe dust storm events. The research provides potential solution for both achieving higher resolution and covering large geographic regions.

In this chapter, the framework to enable the loosely-coupled interoperable models will be introduced, and case study and results achieved through this approach will be presented and discussed. Further research on spatial cloud computing needed to utilize the interoperability and nested modeling approach to their full extent, will be introduced in the next chapter (Chapter 6).

5.1 Framework

One model interoperability and loosely-coupled approach is to use a low spatial resolution model to identify regions of high predicted dust concentration and a computationally more expensive high-resolution model for only the previously identified high concentration areas (Benedict et al., 2011). This approach was employed in the research, in which two versions of the DREAM were used to perform low- and high-resolution dust forecasts. This required access to both models, and entailed execution of the high resolution NMM-dust model based upon the output of the coarse resolution ETA-8bin model.

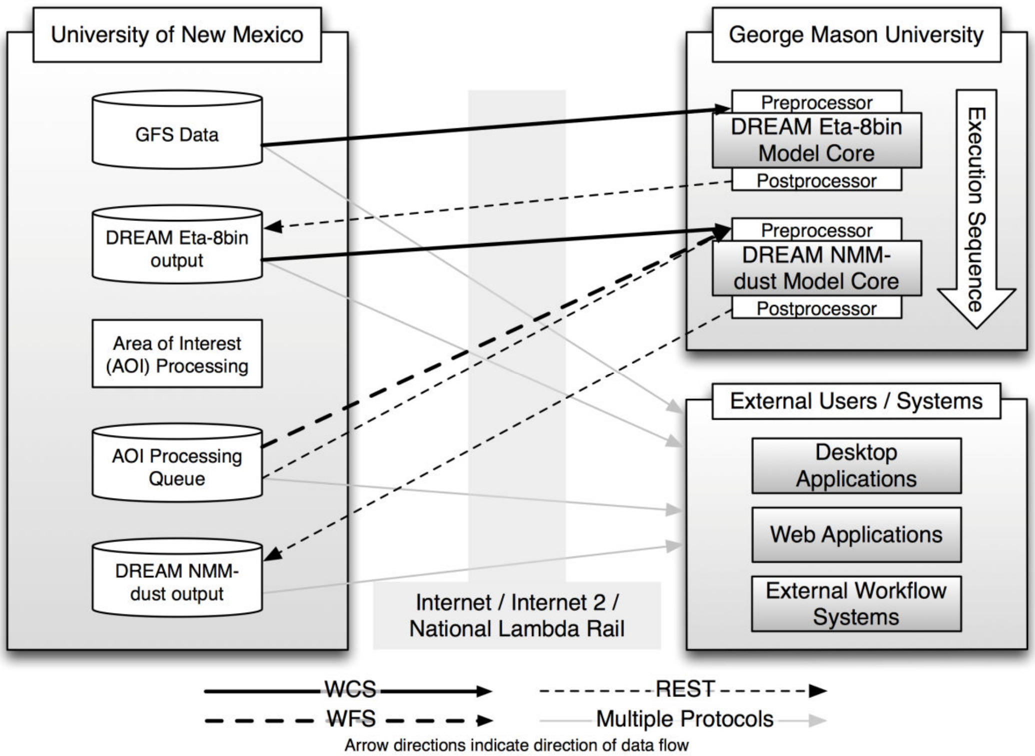


Figure5.1 Framework of the loosely-coupled nested dust storm forecasting

Figure 5.1 shows the framework of the loosely-coupled nested dust storm simulation and the interaction or interoperability between the two models. Within this framework, standards-based service interactions between the partner Universities take place over available high speed network connections (i.e. National Lambda Rail, Internet 2, or the standard Internet) in the sequence of:

1) Initialization of the regional ETA-8bin(Janjic, 2003; Xie et al., 2010) model located at GMU through an OGC WCS call to the Thredds data server (Nativi et al., 2004) hosted at the University of New Mexico's (UNM) Earth Data Analysis Center (EDAC),

2) Initialization of the regional ETA-8binthrough an OGC WCS call to servers at EDAC that are acquiring and publishing current MOD12 land cover products via WCS from the LP DAAC,

3) Execution of the ETA-8binmodel, and delivery of the model results to the servers at EDAC via a simple REST (Fielding, 2000) service interface that processes and republishes the outputs via WCS,

4) Processing (at EDAC) of the delivered ETA-8binmodel outputs to identify regions (Areas Of Interest – AOIs) within the model domain that have dust concentration values that exceed a defined threshold,

5) Retrieval of AOIs (at GMU) for a specific model run date, and initiation of a sub-regional NMM-dust model runs for each geographic area and time period via a REST service request to the server at EDAC,

6) Execution of the NMM-dust model for each AOI at GMU,

7) Delivery of NMM-dust model outputs for each AOI to the servers at EDAC, and

8) Publication of the delivered NMM-dust model outputs via WCS and WMS by the data server hosted at EDAC.

5.2 Nested models

5.2.1 Loosely-coupled nesting strategy

Currently, several numerical dust models have been proposed and developed. The Dust Regional Atmospheric Model (DREAM, Nickovic et al., 2001), designed to simulate dust entrainment and transport on a regional scale, is widely used for dust cycle modeling system. DREAM can be easily configured and incorporated to other atmospheric models. For instance, it has been successfully coupled with NCEP/eta as both the Eta-4bin (4 particle size classes) and the ETA-8bin (8 particle size classes) dust forecast models to simulate the dust cycle in the atmosphere. The performance of the system has been tested for a variety of dust storm episodes, in a variety of locations and resolutions. In conjunction with several previous NASA-funded projects (Morain and Sprigg, 2008, 2009; Yang et al., 2008; Zie et al., 2010; and the ENPHASyS project [http://enphasys.unm.edu/]) the Eta-4bin version was routinely run on servers at the Earth Data Analysis Center, generating an hourly collection starting in January 2006, and continuing through July of 2010. As part of an interoperability improvement and high-performance computing project (Yang et al., 2008), the ETA-4bin andETA-8bin model cores used in this project were developed to run on the servers at George Mason University.

The ETA-8bin model has shown considerable skills in forecasting severe storms, but its spatial resolution is too coarse for many potential applications (Xie et al., 2010). The horizontal grid spacing of ETA-8bin is 1/3 of a degree. With current horizontal resolutions, models used for numerical weather prediction (NWP) are approaching limits of validity for the hydrostatic approximation. The Eta model was replaced in the US National Weather Service (NWS) operations by a Non-hydrostatic Mesoscale Model

(NMM), which has a higher resolution and greater computational efficiency (Janjic, 2003). The coupling of DREAM dust forecasting algorithm and the NMM meteorological module (NMM-dust) forms a much higher resolution model, which enables an increase in the simulation horizontal grid spacing to the zip code level, or about 3KM by 3KM resolution. NMM-dust can produce higher resolution results for weather forecasting and is executable in parallel mode on distributed systems. Parallel processing is supported through the Message Passing Interface (MPI) programming model.

It would be ideal if the NMM-dust model could be run a large forecasting region (i.e. the western 2/3 of the continental United States), but NMM-dust is very computationally intensive and forecast run-times increase n^3 with resolution increases n times in 3D space and n^4 if increase n times in both space and time dimension (Baillie et al., 1997). Therefore, it is not feasible to run the high resolution NMM-dust model for the entire continent or the world. Instead, the course resolution ETA-8bin may be run for a large geographic region, while the NMM-dust model may be run at a higher resolution for specific sub-regions based on the identification of high dust concentrations for those sub-regions. In this way, high-resolution model results for specific sub-regions of interest (with the areas of interest being defined in terms of high dust concentrations) may be obtained more rapidly than would be possible given the execution of a high-resolution model over the entire domain.

In the context of this dissertation, model nesting and interoperability is achieved through the use of the low-resolution ETA-8bin model to identify regions of high predicted dust concentrations and run the higher-resolution NMM-dust model on only those subregions with much small area in parallel. This approach requires access to both models, and entails the execution of the high resolution NMM-dust model based upon the output of the coarse resolution ETA-8bin model. By this way, two “loosely coupled” nested models are integrated to provide a large area dust storm forecasting with high resolution results on the high dust concentration areas, which cannot be performed by only one model, ETA-8bin or NMM-dust.

5.2.2 Case Study

Specially, the dust event on July 1st, 2007 was used to test the feasibility of the loosely-coupled nested dust storm framework. When we utilize NMM-dust model only to simulate the entire domain (36 x 27 degree, Figure 5.2), NMM-dust model and computing power cannot support such a large domain size running with 3KM as spatial resolution due to high computing and memory consumption (Huang et al., 2011). To support the runs, we should either a) redesign the existing algorithms, codes, and data structures, or b) increase the speed of the CPU and the network connection.

Even though the model after reengineering the code can suppose such large area, the forecasting can not successfully complete within two hours for one-day forecasting. Forecasting an area with domain size at 10 x 10 degree would take about 12.7 hours with 8 CPU cores. If the domain size is doubled, it will need around 4 times more computing time (Section 4.4.3). Therefore, the entire domain size forecasting by NMM-dust model would be expected to finish more than 101 hours, which make the results to be invalid as it takes more than 3 days to complete one-day forecasting. Therefore, the proposed approach nested models should be adopted to enable the forecasting for such a large area forecasting.

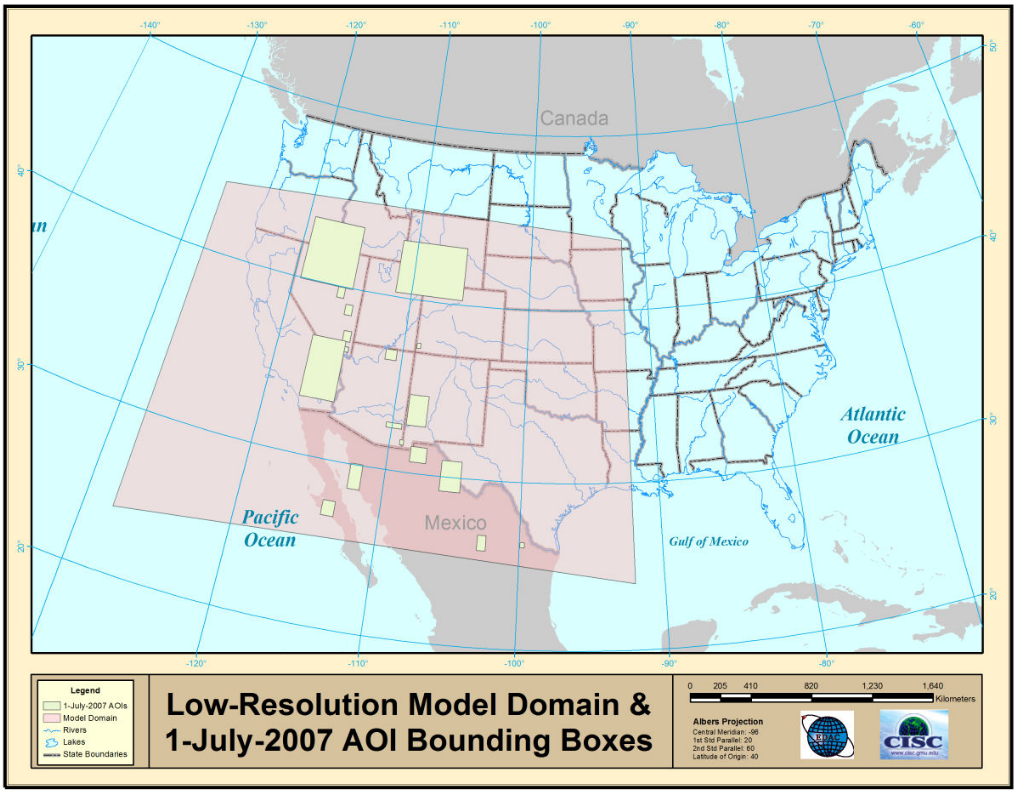


Figure 5.2 Low-resolution model domain area and sub-regions (AOIs) identified for high-resolution model execution

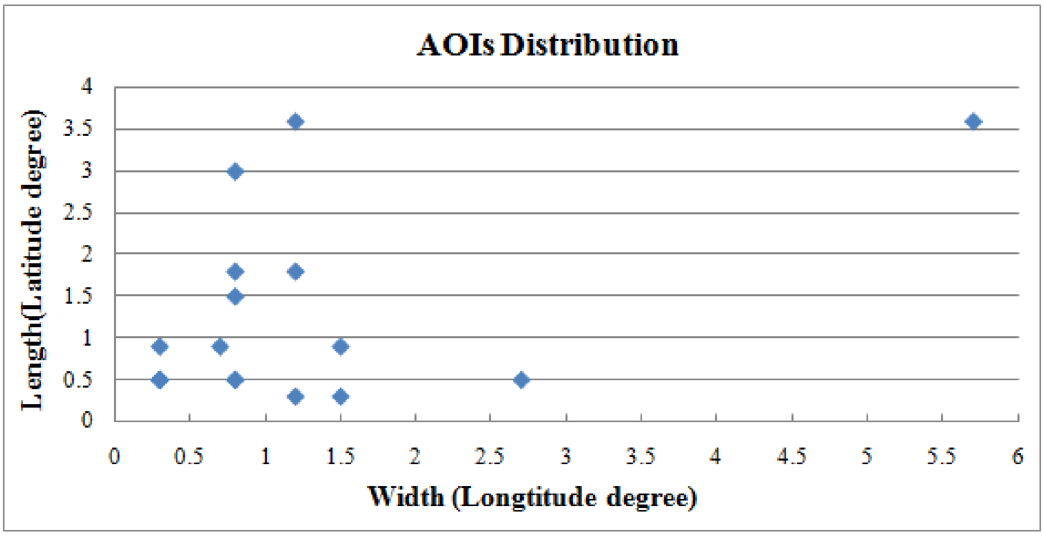


Figure 5.3. AOIs width and length distribution

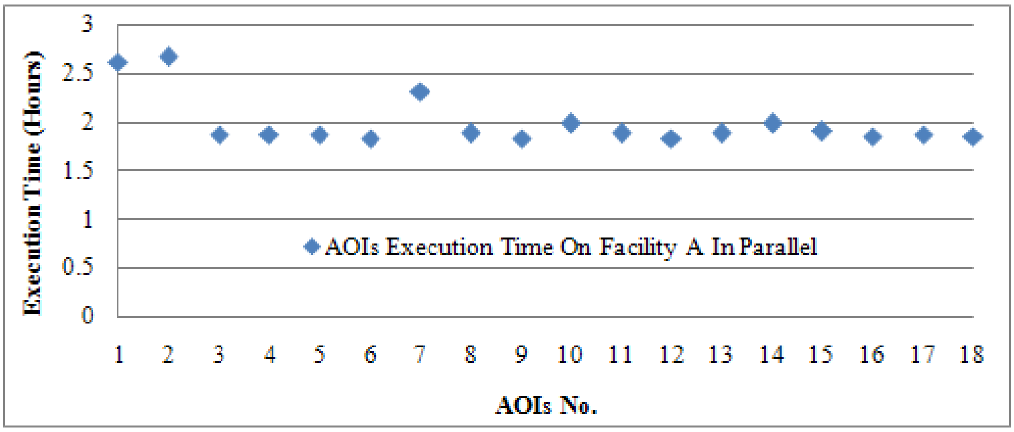


Figure 5.4.NMM-dust execution time for each AOI on facility A in parallel

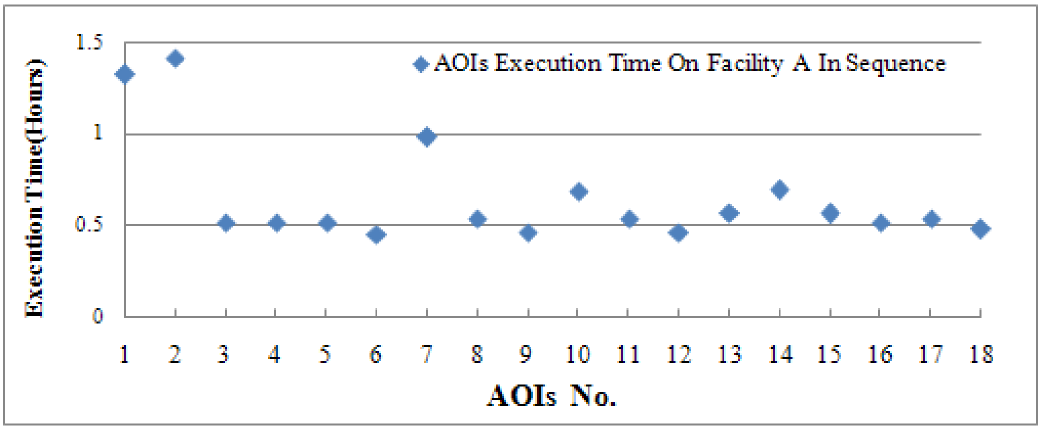


Figure 5.5. NMM-dust execution time for each AOI on facility A in sequence

Figure 5.4 shows the execution time required for different AOIs when the facility A handling all AOIs in parallel. Therefore, it is expected to finish the entire AOIs between 2.7 hours if all of the AOIs are simulated by the NMM-dust in parallel on the facility A. Figure 5.5 shows the execution time for each AOI if the facility A handle the AOIs in sequence. The results show that the AOI with the largest domain size (5.7 x 3.5degree) can be completed within 1.4 hours. This means that with enough groups of computing resources (e.g., 18 times of computing resources in Facility A) to handle all of the AOIs in parallel, it would make the AOI with the largest domain size determine the total execution time (1.4 hours in this case).

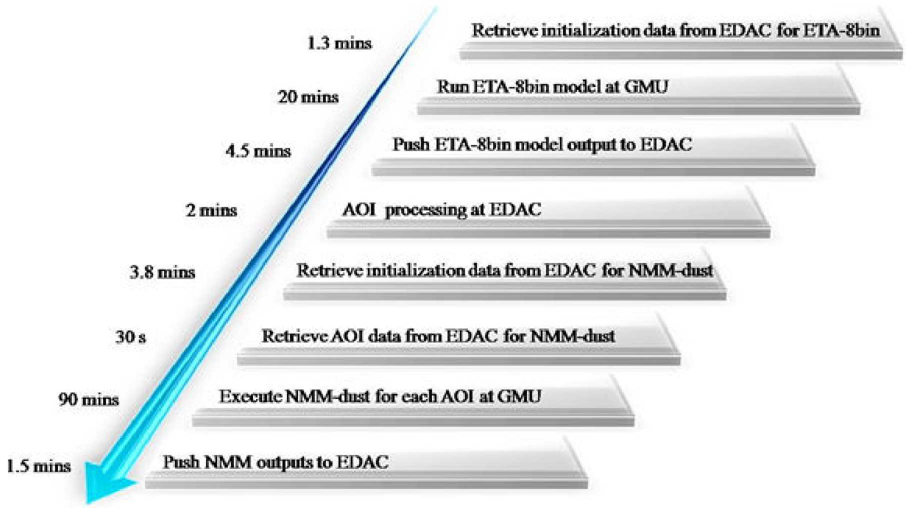


Figure 5.6. Execution steps for nested running of ETA-8bin and NMM-dust

During the feasibility study, the comparison between the nested running of ETA-8bin and NMM-dust and only utilizing NMM-dust model to obtain the results for entire domain size (36*27 degree) have been conducted. Figure 5.6 shows all required steps and time for each step to implement the proposed framework and enable the nested running of ETA-8bin and NMM-dust. To calculate the possibility of forecasting based on this approach, the time for the data transferring between the EDAC and GMU and the running of ETA-8binhave also been sampled and calculated as Figure 5.6.

Therefore, the total time for forecasting including data transferring and execution time for both ETA-8bin and NMM-dust is about 2 hours with enough computing resources to handle all AOIs concurrently. The simulation is done on 1Gbps and Quard Core, 16Gbytes machines in facility A. With the latest machines with 10Gbps connections and six core 64 Gbytes HPC computing nodes in facility B, the simulation can be done under one hour, therefore, the study found that the nested-model simulation is feasible to conduct dust forecasting for the Southwest U.S.

Figure 5.7 shows one time-frame result at 03 AM, July 02, 2007, produced by ETA-8bin and NMM-dust models in different spatial resolution. The model results show similar pattern for dust storm area and NMM-dust with 3KM picked up much more detailed information about the dust concentrations.

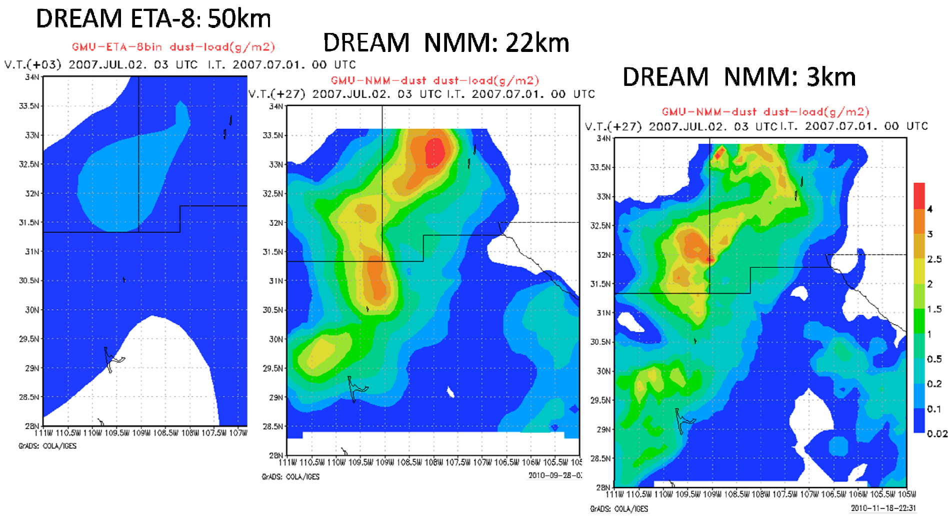


Figure 5.7 Comparison of the simulation results by ETA-8binand NMM-dust on AOI 10, 11, 12 and 13 at 03 AM, July 02, 2007

5.3 Model Interoperability

This framework (Figure 5.1) makes extensive uses of open standards, including the OGC's WMS(Beaujardiere, 2006) and WCS(Evans, 2006), and the World Wide Web Consortium's (W3C) Simple Objects Access Protocol (SOAP, W3C, 2007). The use of OGC standard service interfaces between models provides for future evolution of the system without having to work around model inter-dependencies that would exist in a tightly integrated sequential model execution scheme. As long as the service interfaces remain unchanged, the systems that operate those interfaces can be modified without any required changes to the components that interact with those systems. The expectation is that there will be little, if any, modification of the existing ETA-8bin and NMM-dusts model cores, with the above described service interconnections being enabled through a modification of the model pre- and post-processor components associated with the models.

The system interoperability activities for this research have been related to the development of interoperable interfaces for dust storm models to acquire distributed data resources as model input and in the development of enhanced interoperable services for the delivery of products and data to users.

5.3.1 Distributed data acquisition

OGC open standards are used to access and integrate data from different organizations, locations, and with different type of data, such as NetCDF, GRIB, HDF-EOS, through WMS, WCS, and CSW, as input for initialization of ETA-8bin model and support the running of nested models. The input of ETA-8bin model includes both static and dynamic meteorological and geospatial data sets. The static data sets consist of soil type, geographical information of the surface, e.g. digital elevation of the surface etc. The dynamic meteorological datasets for model input of ETA-8bin requires 5 time-varying meteorological fields which are the prognostic variables during the model simulations, including surface pressure p, geopotential height, specific humidity q, temperature T, and two horizontal wind components u and v (Wolters et al., 1995).

During the feasibility test, the real-time meteorological data is accessed and obtained from EDAC data center, which hosts an OGC Web Coverage Service (WCS) for the continuously growing collection of NOAA Global Forecast System (GFS) meteorological forecast products. EDAC has been acquiring GFS produces since 2006 in support of the multiple dust modeling related projects illustrated in Figure 5.8.

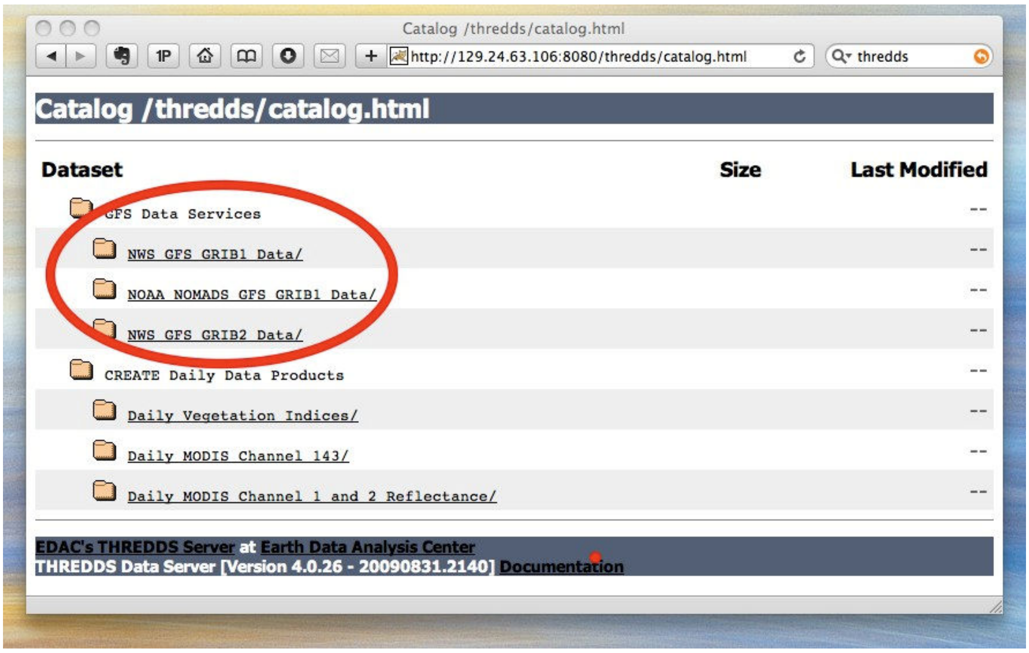


Figure 5.8. GFS THREDDS Catalog Hosted by EDAC (GFS data collections circled in the figure)

Traditionally, restricted data formats are required for the NWP models to extract these dynamic parameters and specific data centers are providing those data, such as NWP models should usually download the GRIB1 or GRIB2 data from NCAR websites. GRIB data, which are usually the data format for NWP models and can be produced from NCEP's NAM (http://stu-in-flag.net/nam.php) or GFS(http://www.emc.ncep.noaa.gov/gmb/moorthi/gam.html) model, typically contain more fields than needed to initialize ETA-8bin. However, the five meteorological parameters to initialize ETA-8bin model are provided by many data centers via different models or tools. Therefore, as long as these five meteorological parameters can be integrated, the ETA-8bin model may be executed using meteorological parameters from a wide variety of sources, including the GFS model output. In this way, model interoperability is facilitated in the aspects of preparing model input by developing the potential of integrating widely distributed data resources to enable the executions of Earth Science models.

The system currently accepts DataFed (2007) /AirNow inputs EPA observations through the WCS as dynamic geospatial data sets. Another data source, e.g., NASA LP DAAC's soil moisture data, is available. The new LP DAAC service can be quickly into the existing workflow through WCS. These data can be accessed without developing a new interface, and therefore the return-on-investment-in-interoperability (Bambacus and Reichardt, 2006) is achieved.

5.3.3 Across-site and model communication

Coupling the Eta-8binoutput and NMM-dust models are implemented to enable the communication of two different models. Modifications to the model pre- and post-processor components associated with the Eta-8bin models are required to enable model interoperability. Specifically, the model pre-processor code has been modified to support the retrieval of initialization parameters from remote systems via WCS requests that return data files which are then subject to further processing to match the required input formats (Fortran binary grids) expected by each model core. The Post-Processor code requires the addition of functionality for automatically pushing the model output to the directory where a THREDDS Data Server hosts ETA-8bin model output at GMU to disseminate the model output in an interoperable way, as well as triggering the EDAC data center to harvest the model output through WCS.

Standard input and output file formats should be implemented and utilized for both the Eta-8bin and NMM-dust models so that the outputs of one could be used as the input into the other. Specifically, it was decided to use the NetCDF as the file format for the outputs of both models, allowing standard meteorological data processing tools to access and process these products. Both models after modification of pre- and post-processors are able to read NetCDF files for model initialization and boundary condition specifications. The use of a common, well supported, data format for both model initialization and output significantly enable the communication and streamlines the process of developing multi-model workflows. In this way, the output of one model can be used to initialize another, either for a model run for a subsequent time step, or for the execution of a higher resolution model for the same time period over which a low-resolution model has already been run.

5.3.3 Model output real-time dissemination

The model output is served through WMS and SOAP, which enables easy access for all decision support systems (DSSs) supporting OGC, OpenDAP standards. The THREDDS Data Server running at EDAC and CISC will be expanded in its data collection to include the directories that host the generated ETA-8bin and NMM-dust model outputs that will be pushed to the servers at EDAC. This expanded THREDDS catalog will provide four access models for all of the project products: HTTP for direct download; WCS for data delivery with options for spatial, temporal, layer extraction, and coordinate transformation; WMS for visualization of model outputs as map images; and OpenDAP for delivery of multi-dimensional data suitable for use in a wide variety of implementing analysis systems. Figure 5.9 and 5.10 shows the utilization of dust storm forecasting results of both ETA-8bin and NMM-dust models for 4D visualizations accessed through the THREDDS server hosted at EDAC .

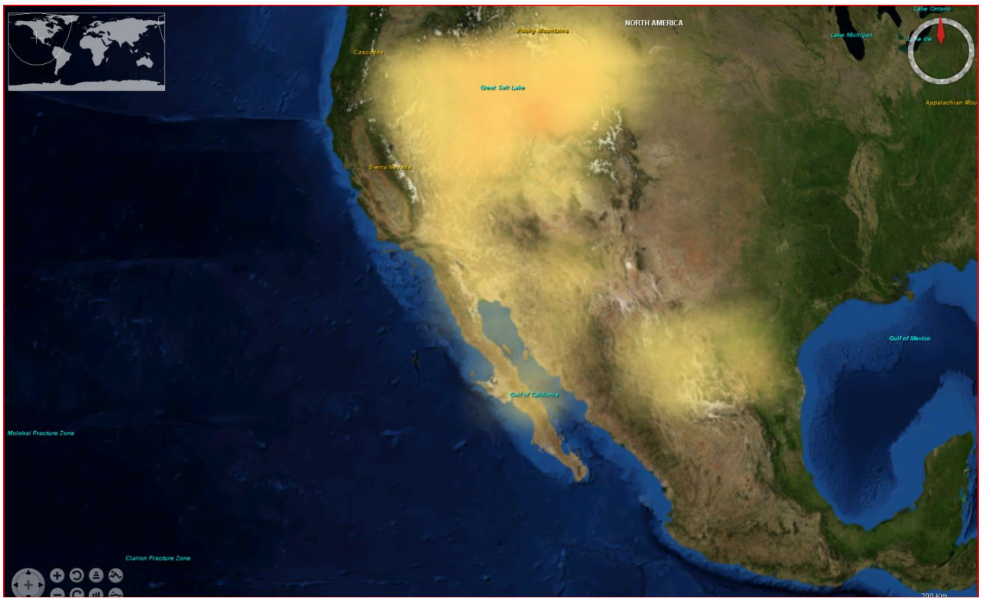


Figure 5.9 Utilize coarse resolution model ETA-8binoutputs

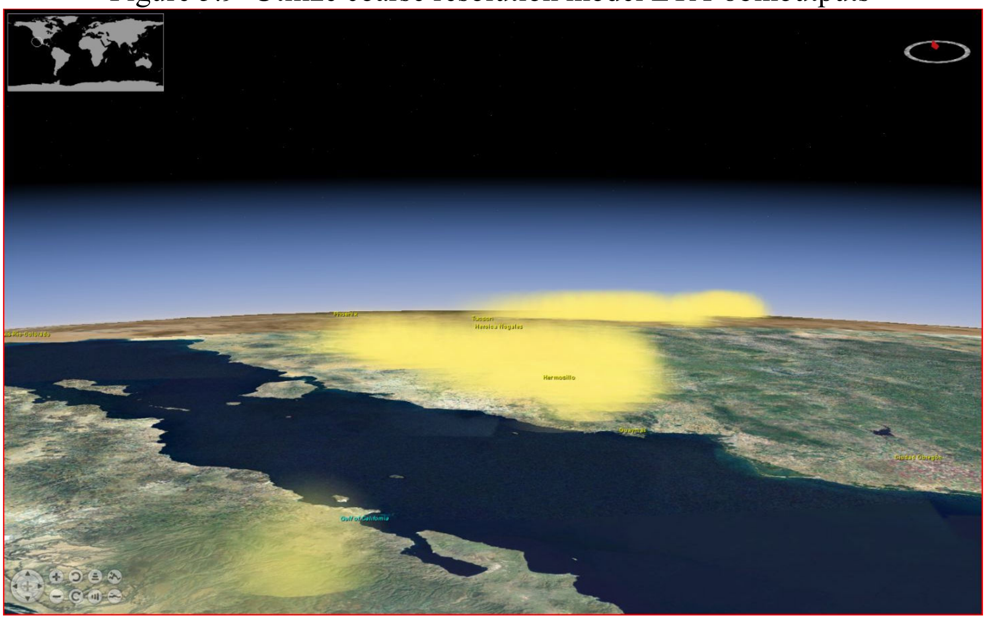


Figure 5.10 Utilize high resolution model NMM-dust outputs

CHAPTER 6

SPATIAL CLOUD COMPUTING

Loosely-coupled nested models require a large computing pool to run various hotpots identified by coarse model in parallel to achieve the best performance. This will cause computing spike requirements that can be best handled by elastic and on demand computing platform. Could computing provides a pay-as-you-go potential solution with a large, virtualized pool of computational resources (Armbrust et al., 2010). Cloud computing technologies, e.g., virtualization, now become more and more mature, and cloud infrastructure become more and more powerful. For example, Amazon EC2 offers cluster instances with 10 Gbps network connection. Each instance has 23Gbytes memory, a clock speed of 2.93 GHz, and two quad-core processors. Such hardware and network configuration are far much better than grid computing environments with heterogeneous computing resources and World Wide network (WAN) connection, and even better than most of private homogenous HPC cluster configurations. This makes cloud computing a new, advantageous computing paradigm to resolve scientific problems traditionally requiring leverage a special high-performance cluster (Rehr et al., 1020). The traditional cluster environment requires typically several months and usually hundreds of hours of labor to procure, install, configure new infrastructure. Traditional processes will involve multiple personnel with different technical backgrounds to ensure the success of the configuration. Often, greater efforts and costs are required to maintain the infrastructure while the utilizing of the infrastructure were used only 10%-15% of its full capacity (Marston et al. 2011). Therefore, the cloud parallel system would have a great impact on parallel scientific HPC applications in the near future by facilitating the deployment process while increasing the rate of utilization of computing resources (Yang et al., 2011b).

In addition, real-time dust storm simulation is a data intensive application, high memory and computing power are required when a large number of users are involved and each user may forecast different areas and harvest thousands of records. There are different access amounts at different periods within one day, and different computing resources should be leased at different time slots during the day. And, running high resolution models for a large geographic region will require a significant amount of computing resources and, often, the higher resolution will not be needed. This will cause computing spike requirements that can be best handled by elastic and on demand cloud computing platform.

This chapter will explore how to use spatial cloud computing to support dust storm simulation and predictions by enabling the nested running of dust storm models, which require spiking many computing instances to enable the AOIs to run in parallel. The Amazon EC2 cloud computing platform and the process to deploy the NMM-dust model on Amazon's EC2 cloud computing platform will be introduced. The AOIs identified by the coarse dust storm model will be run and tested in parallel with the scalable Amazon EC2 instances which are optimized through the spatiotemporal patterns and strategies proposed in Chapter 4.

6.1 Amazon Cloud Services

6.1.1 Amazon EC2

As a central part of Amazon's cloud services, Amazon EC2 allows users to deploy scalable resources on demand. Amazon EC2 is a typical IaaS. Based on Xen(Barham et al., 2003), EC2 enables users to boot an Amazon Machine Image (AMI) to create a virtual machine, which Amazon calls an "instance." AMI is a bootable virtual machine (VM) root image with various OS and any software desired to create a VM. At present, EC2 offers a number of different instance types to meet computing needs with each instance providing a predictable amount of dedicated compute capacity (CPU power, memory, disk etc.). Amazon classifies these EC2 instances into five categories including Standard instances, Micro instances, High-Memory instances, High-CPU Linux instances, Cluster Compute instances and Cluster GPU instances, and different categories are suitable for different types of applications.

The Standard Linux instance has memory to CPU ratios suitable for most general purpose applications, and it has three types of instances with different virtual hardware configurations. For example, the small instance has only one EC2 computing unit (1 virtual core with one EC2 computing unit). The computing speed of one EC2 computing unit is approximately 1.0-1.2 GHZ. As the computing power and memory increase, the cost for the instances also increases. High-Memory Linux instance offers larger memory sizes for high throughput applications, including database and memory caching applications. High memory extra large, double extra large and Quadruple extra large instances have 2, 4, and 8 virtual cores respectively with each virtual core having 3.25 EC2 compute unites. Instances of this category are relatively expensive, at least $0.5 per hour due to high computing power and memory hardware configuration. High-CPU Linux instance has proportionally more CPU resources than memory (RAM) and is well suited for computing intensive applications.

Cluster compute instances provide proportionally high CPU with increased network performance and are well suited for High Performance Compute (HPC) applications and other demanding network-bound applications. Cluster Compute Quadruple Extra Large instance has 23 GB memory, 33.5 EC2 Compute Units, 1690 GB of local instance storage, 64-bit platform, 10 Gigabit Ethernet. Unique to Cluster Compute and Cluster GPU instances is the ability to group them into clusters of instances with the high speed network connection for use with HPC applications. This is particularly valuable for those applications that rely on protocols like Message Passing Interface (MPI) for tightly coupled inter-node communication (http://aws.amazon.com/ec2/hpc-applications/).

6.1.2 Deploying Applications Onto The Cloud

Through the AWS (Amazon Web Service) Management Console (http://aws.amazon.com), or Amazon EC2 AMI Tools (http://aws.amazon.com/ec2/), users can request to launch an instance based on a specified AMI. If the request is authorized, a VM is deployed. The VM image could already be available on a cloud, or created by users. Figure 6.1 shows the process of launching an Amazon EC2 instance. Amazon has two types of storage services, including EBS (Elastic Block Store) and Simple Storage Service (S3). EBS is a type of storage that enables you to create volumes that can be mounted as devices by Amazon EC2 instances. Amazon EBS volumes behave like raw unformatted external block devices. S3 provides a simple web services interface that can be used to store and retrieve any amount of data over the Internet. Both storage types can store AMI volume and be used as the virtual storage device for an Amazon EC2 instance.

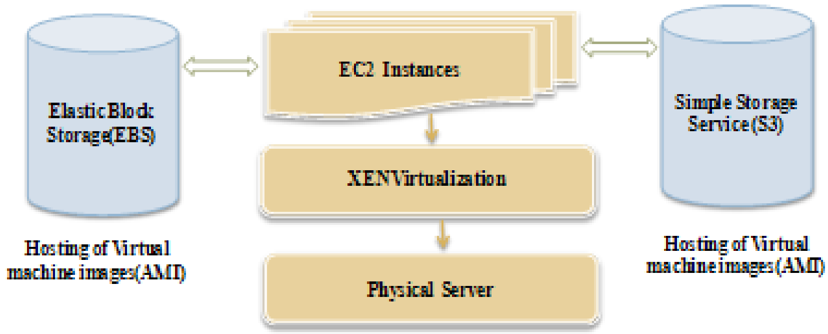


Figure 6.1 Launch an Amazon EC2 instance (http://aws.amazon.com)

In order to deploy applications on Amazon, an AMI should be prepared. In Linux, there are two common ways to prepare an AMI that offers a combination of user friendliness and detailed customization levels: 1) The easiest method involves starting from an existing public AMI and modifying it according to your requirements. This is applicable for both Amazon EBS-backed and Amazon S3-backed AMIs; 2) Another approach is to build a fresh installation either on a stand-alone machine or on an empty file system mounted by loopback. This is only applicable for AMIs backed by Amazon S3 and entails building an operating system installation from scratch.

The role of virtualization technology in the clouds is emphasized by identifying it as a key component as it provides cloud computing with the necessary level of abstraction to represent a unique usage mode and a narrow interface with well-defined purposes (Buyya et al., 2008). With the technology of virtualization, a cloud computing infrastructure, such as Amazon's Elastic Computing Cloud (EC2), provides computing capacity to end users on-demand from remote locations in the Internet (Liu and Orban, 2008).

A wide range of virtualization solutions have been proposed and three leading approaches are full virtualization, para-virtualization and hardware virtualization. Full virtualization is based on the host/guest paradigm and each guest runs on a virtual imitation of the hardware layer. Para-virtualization presents each virtual machine with an abstraction of the hardware that is similar, but not identical, to the underlying physical hardware. Para-virtualization attempts to provide most services directly from the underlying hardware instead of abstracting it. Hardware virtualization is when the hypervisor is embedded in the circuits of a hardware component instead of being called from a third-party software application.

6.2. Deploying Dust Storm Model onto the Cloud

Figure 6.2 demonstrates how to deploy dust storm models onto Amazon EC2 high performance platform. In this case, the first method of preparing the cluster instance AMI from an existing public AMI and modifying it is used. Amazon provides for users to launch cluster instance from a special EBS-backed Amazon Machine Image (AMI) using Hardware Virtual Machine (HVM) virtualization. 1) One or more cluster instances can be launched based on the dust storm forecasting requirement for the computing resources. After the instances that begin booting up with the AMI, users are able to interact with the cluster instances. 2) User can create an EBS volume to attach to the head node instance for keeping the dust storm model and required package suits to run the model, such as MPICH2 software, to schedule the dust storm simulation. Such a separate volume has two benefits: a) one is to restore the dust storm simulation system from the volume in case the current head node instance crashes, and b) another is that the volume could be any size from 1 GB to 1 TB in size. As the dust storm simulation is a data intensive application, hundreds or thousands of data could be produced. Therefore, such an EBS volume would be perfect to resolve the storage capability problems.

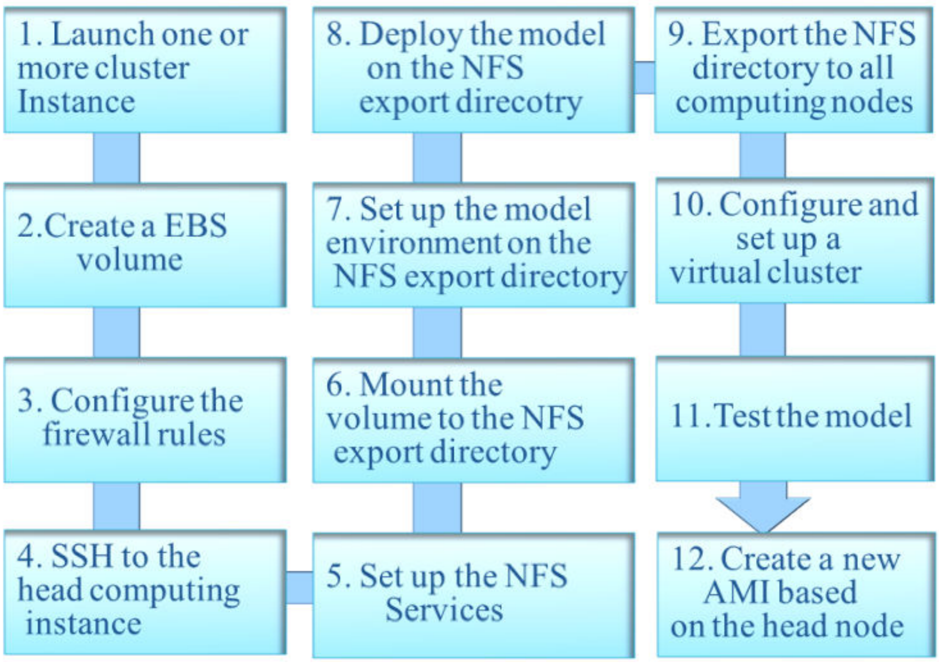


Figure 6.2. The process of deploying dust storm models onto Amazon EC2

3) The next step is to set up the firewall for the group of instances and the system administrator should have full control over the instances and the instances should be able

to communicate with each other over a range of ports while performing the simulation to exchange the data. This can be set up through Amazon’s command line tools or AWS Management Console. 4) Therefore, users can login to the instance to get full root access through secure remote access SSH (Secure Shell) after authorizing the network access performed in the previous step.

5) After logged in, the user can explore and play with the system however he/she likes: setup the required environment to host a web application or running a model. In this case, the database and the NFS service should be set up to enable all other computing instances to share the package suits. In this way, only the head node should set up the software environments and configure the model while other computing instances can share this environment without installing and configuring. After 6) mounting the EBS volume to the NFS export directory, 7) the package suits required to enable the model to run, including python and MPICH2 in this case, can be installed, and 8) NMM-dust dust storm model can be deployed on the NFS export directory.

9) After successfully setting up the head node environment, the computing nodes can share this environment by mounting the NFS export directory to the local directory like using a local storage. 10) As long as the firewall between the head node and computing nodes are properly set up, and the middleware MPICH2 is properly configured, the group cluster instances should be able to communicate, and 11) the model NMM-dust should be able to run successfully.

10) Finally, a new AMI can be created based on the running instance. In this way, the system can be restored in a very fast fashion if the head node crashes by launching an

Amazon EC2 cluster instance from this new AMI. In addition, geoscience communities can build their own dust storm forecasting system based on the current available system created from the new AMI. This would greatly reduce the duplication of efforts among organizations as other organizations can run the model based on the work without putting forth great efforts into compiling, installing and configuring the NMM-dust on their own efforts, which is a very complex process taking several days or even several weeks for a new user of such a system. Traditionally, various researchers and scientists have used the available Earth science data to build a large number of complex algorithms, models and applications tailored to their specific studies (Votava et al., 2002). The goals achieved by these algorithms, models and applications are not always completely different despite the differences in research objectives. If we build the functions or applications as cloud services to share with others, duplications can be eliminated. Hence, cloud computing would greatly facilitate Geosciences by fostering the reuse and sharing of functionalities and knowledge across Geosciences communities.

Amazon EC2 offers a highly reliable environment for dust storm models because the service runs within Amazon’s proven network infrastructure and datacenters. The Amazon EC2 Service Level Agreement (SLA) guarantees 99.95% availability for all Amazon EC2 regions, including US Standard, EU (Ireland), US West (Northern California) and Asia Pacific (Singapore). An additional EBS volume to keep dust storm models and running environment has been configured for added reliability. Since the EBS data volume can be attached to another instance, the dust storm simulation system is therefore protected from instance termination or failure. Therefore, Amazon EC2 improves the reliability of geospatial applications.

6.3 Performance test

Finally, we integrate all of the strategies to test the computability of a large area with high resolution forecasting requirements on a cloud platform. Elastic Amazon EC2 cluster instances are utilized to enable the nested modeling approach to its full extent. During the loosely-coupled nested model test discussed in Section 5.2, a large domain is narrowed down to 18small regions, which are identified by the coarse model ETA-8bin. As Section 4.3.2 experiment results indicate, a certain number of processes usually achieve the best performance for any problem size supported by a pool of computing resources. Therefore, before utilizing the Amazon EC2 cluster instance, the process number that can achieve the best performance should be identified by a set of experiments. Such identifying experiments only require to be conducted once, and all of other simulation tasks can use the identified process numbers. Figure 6.3 shows that the best performance achieved by one Amazon cluster instance is through 40 processes. Therefore, 18 Amazon cluster instances are launched, and each instance with 40 subdomain/processes, to achieve better performance, is responsible to handle one subregion.

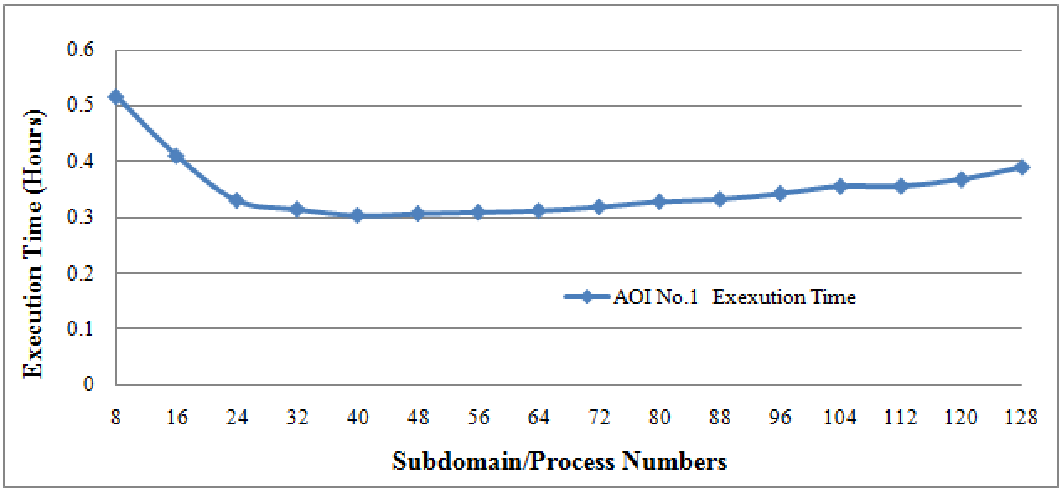


Figure 6.3. The scalability test of one Amazon cluster instance

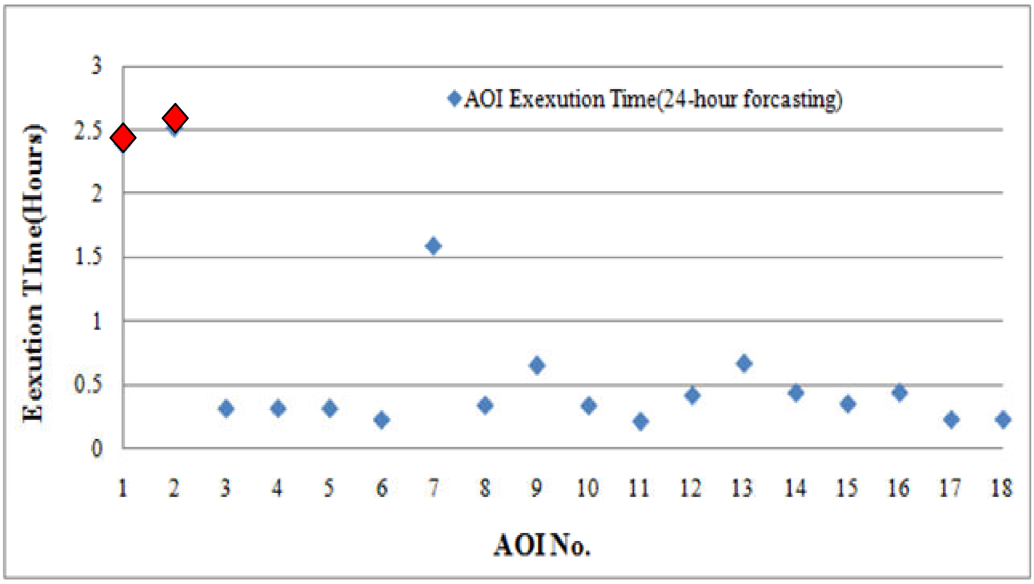


Figure 6.4. 18 Amazon cluster instances are launched to run 18 AOIs in parallel with each instance simulating one AOI region for 24-hour forecasting.

Figure 6.4 shows the execution time for each instance. The results reveal that most of AOIs can be successfully completed within 1 hour for 24-hour forecasting. However, two AOIs cannot be successfully completed within 2 hours. Therefore, more computing resources should be integrated to enable the computing to achieve the 2 hour for 24-hour simulation time constraint. Then, more computing resources should be integrated to improve the performance.

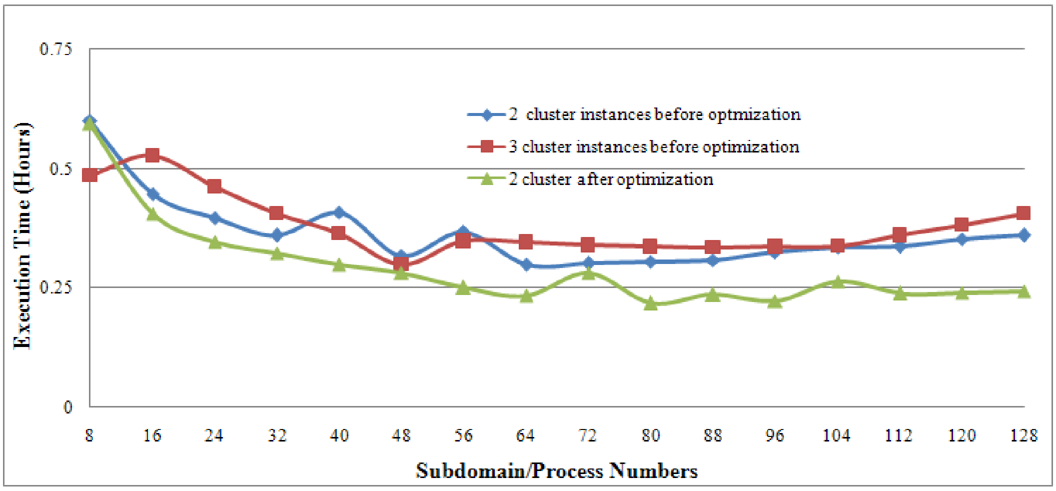


Figure 6.5 Performance comparisons of Amazon cluster instances before and after optimizations

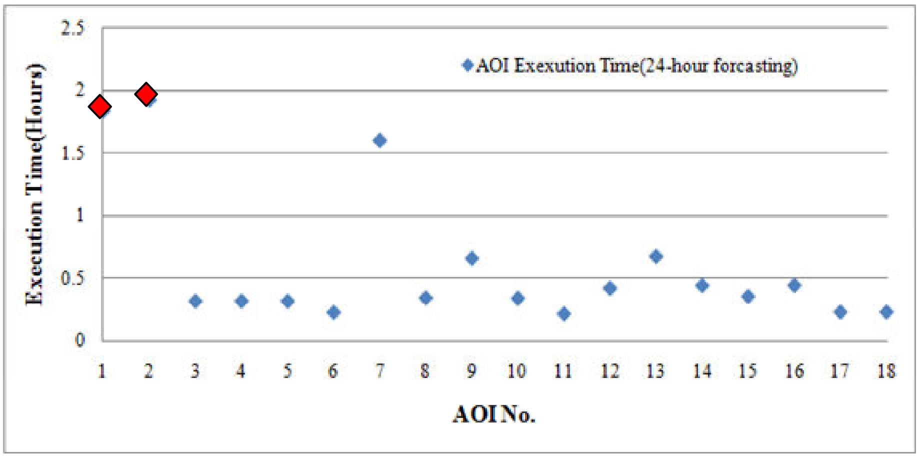


Figure 6.6. 18 Subregions run on Amazon EC2 cloud platform after optimization. Both AOI 1 and 2 utilize two optimized cluster instances.

Therefore, two and three cluster instances are used to test the computability of those two subregions. Figure 6.5 shows the performance of Amazon EC2 cloud platforms when utilizing two and three instances with and without optimizations for 3-hour forecasting over the first subregion. It is observed that the system cannot achieve the 0.25 hour time constraints even though three cluster instances are involved. Therefore, the two instances should be optimized through better parallelization and scheduling strategies, including neighbor-mapping and local storage strategies. Figure 6.5 shows that using more than 64 processes, the optimized two instances can successfully complete the forecasting within 0.25 hours. Finally, with this two instances and starting 64 processes, it is observed that the two AOIs can be successfully completed within two hours (Figure 6.6).

CHAPTER 7
CONCLUSIONS AND FUTURE WORK

This thesis reviewed relevant problems and suggested new approaches to improve the performance of dust storm simulation through exploring and utilizing spatiotemporal patterns. Model interoperability is achieved through integrating multiple distributed data resources as real-time model input, enabling two independent models to communicate and finally disseminating model results in a real-time fashion. Loosely-coupled nested models are adopted in this study to enable the computability of dust storm forecasting with high resolution and large geographic coverage requirement, and nested models are supported through spatial cloud computing.

7.1 Conclusion

My dissertation reports improving the performance of parallel systems through understanding, discovering and utilizing spatiotemporal patterns. A dust storm simulation model NMM-dust was used as an example to illustrate several aspects of spatiotemporal patterns that can apply to better leverage HPC. A series of experiments are designed and conducted to research spatiotemporal characteristics and constraints of dust storm phenomena, computing resources and dust simulation models. The experimental results show that faster CPU speeds, suitable computing resources, better connections, and a good storage strategy will speed up the simulation and enable prediction. The results also give insight to divide the problems and arrange computing resources based on the prediction requirements, including required geographic coverage, spatial resolution, and temporal scope.

Based on our analysis of the results of the experiments, we conclude that spatiotemporal principles are critical in their ability to optimize computing infrastructures by helping arrange, select, and utilize high end computing for compute intensive problems.

The research also improves model integration using nested modeling and interoperability approaches. Both the model results and performance comparison demonstrate that utilizing interoperable loosely-coupled nested models is capable to enhance dust storm forecasting by facilitating model integration, data discovery, data access, and data utilization for a) integrating widely distributed large amounts of geospatial datasets as model input, b) reducing the computing time, and c) increasing spatial resolution, domain size and lengthening the period of forecast.

Finally, this dissertation explored the feasibility of utilizing cloud computing to support geospatial science application with computing intensity, data intensity, spatiotemporal intensity and user concurrent intensity. With the capability of scaling computing resources on demand, cloud computing platforms offer an efficient solution to enable real-time dust storm forecasting systems.

7.2 Future Work

This dissertation addresses the computing demands of dust storm forecasting through utilizing spatiotemporal patterns and principles to improve the model performance. This dissertation also adopts loosely-coupled interoperable nested models and cloud computing to support the nested models and enable the computability of dust storm forecasting with high resolution and large domain size requirements. The research results demonstrate a great potential to solve computing problems that require conducting clustered high resolution phenomena prediction for a large geographic domain. More research would be required to greatly enhance the applicability of the methodologies used in this dissertation in the next decade including the following research aspects:

Domain integration: Model integration is eventually trying to interoperate multiple domains by sharing knowledge to best utilize data and information for the greatest societal impact. This domain interoperation will require scientists with different backgrounds to collaboratively address a complex problem such as dust storm impact to public health, by contributing their own domain knowledge and revise the knowledge structure to accept or output results to other domains. To achieve this objective, social studies to better capture and share domain knowledge would be required.

Automatic model integration: once we have proper metadata and model configurations for existing domain models which have been running separately for different domains. A workflow chaining process is needed to recruit the needed models for a specific scientific application or task on the fly, ideally, in an automatic fashion (Granell, Diaz, and Gould 2010).

Broad bandwidth and CPU speed: After parallelization, the speed of the model's execution depends on the numerical methods used and the implementation as well as the data involved. Thus, data communication speed and how fast a sequential computer process would become a generic computing science challenge (Yang et al., 2011). The latest advances from 2D CPU to 3D CPU will help improve the processing speed.

Together with the fast tens or hundreds of Gbps communication networks, faster CPUs will help enable many of such model integration efforts.

Middleware: In practice, the simulation of dust storms is very dynamic in spatiotemporal scales and, therefore, demands the dynamic allocation of computing resources. A middleware considering these spatiotemporal patterns would enable the allocation and use of computing resources for geospatial applications effectively and efficiently (Huang and Yang, 2010). In the future, it would be necessary to develop a middleware that can schedule the tasks in a way that improves the scalability and performance of networked computing nodes by fully considering the spatiotemporal patterns. Such an effort would also help to construct a better geospatial cyberinfrastructure (Yang et al. 2010a) and a spatial cloud computing platform (Yang et al. 2011b).

Spatial Cloud Computing: SCC is expected to be the next generation platform to support Earth science applications, such as dust storm simulation (Yang et al., 2011).The success of spatial cloud computing depends on many factors, such as the outreach of spatial cloud computing to geospatial scientists who can employ cloud solutions and to computing scientists and engineers to adapt spatiotemporal principles in designing, constructing, and deploying cloud platforms.

REFERENCES

1. Anthes, R., 1983. Regional models of the atmosphere in middle latitudes. *Monthly weather review*., 111, 1306-1335.
2. Anthes, R.A. and Warner, T. T., 1978. Development of hydrodynamic models suitable for air pollution and other meso meteorological studies. *Monthly Weather Review*, 106, 1045-1078.
3. Argent, R. M. 2004. An Overview of Model Integration for Environmental Applications – Components, Frameworks and Semantics. *Environmental modeling and software*, 19 (2004): 219-234.
4. Armbrust, M. et al. 2010. A view of cloud computing. *Communications of the ACM*, 53(4): 50-58.
5. Armbrust, M., Fox, A. and Griffith, R. et al., 2009. Above the Clouds: A Berkeley View of Cloud Computing, University of California, Berkeley, Berkeley, CA, 2009. *http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html (accessed March 12, 2010).*
6. Armstrong M.P. and Marciano R., 1996. Local interpolation using a distributed parallel supercomputer. *International Journal of Remote Sensing*, 10 (6):713–729.
7. Armstrong, M. P., Cowles, M. and Wang, S., 2005. Using a computational grid for geographic information analysis. *Professional Geographer*, 57 (3), 365–375.
8. Baer, F. and Zhang, B., 1998. Optimizing computations in weather and climate prediction models*. Meteorology and Atmospheric Physics*, 67(1-4), 153-168.
9. Baillie C, Michalakes J, Skilin R.1997. Regional weather modeling on parallel computers. Parallel Computing, 23(14):2135-2142.
10. Baillie, C. F., MacDonald, A. E. and Sun, S., 1995. QNH: A Portable, Massively Parallel Multi-Scale Meteorological Model. *Proceedings of the Fourth Int'l Conference on the Applications of High Performance Computers in Engineering*, Milan, Italy.
11. Baillie, C., Michalakes, J. and Skilin. R., 1997. Regional weather modeling on parallel computers. *Parallel Computing*, 23(14), 2135-2142.
12. Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., and Warfield, A. 2003.Xen and the art of virtualization. In Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles (Bolton Landing, NY, USA, October 19 - 22, 2003).SOSP '03. ACM, New York, NY, 164-177. DOI= http://doi.acm.org/10.1145/945445.945462
13. Barnum, B. H., Winstead, N. S. Wesely, J. Hakola, A. Colarco, P. R. Toon, O. B. Ginoux, P. Brooks, G. Hasselbarth, L. Toth, B., 2004. Forecasting dust storms using the CARMA-dust model and MM5 weather data*. Environmental Modeling & Software*, 19(2), 129-140.
14. Basart, S., Pérez, C. Cuevas, E., Baldasano, J.M. and Gobbi, G. P., 2009. Aerosol characterization in Northern Africa, Northeastern Atlantic, Mediterranean Basin and Middle East from direct-sun AERONET observations. *Atmos. Chem. Phys.*, 9(8), 265-8, 282.
15. Benedict, K., Huang, Q., Yang, C., 2011. Utilizing Model Interoperability and Nested Models to Enable Data and Computing Intensive Environmental Model Simulations-An example with Dust Storm forecasting for the Southeastern U.S. *Environmental Modeling & Software*. (In review)

16. Blythe, J., Jain, S., Deelman, E., Gil, Y., Vahi, K., Mandal, A., and Kennedy, K. 2005. Task scheduling strategies for workflow-based applications in grids. In *Proceedings of the Fifth IEEE international Symposium on Cluster Computing and the Grid (Ccgrid'05) - Volume 2 - Volume 02* (May 09 - 12, 2005). CCGRID. IEEE Computer Society, Washington, DC, 759-767.
17. Branger F., Braud I., Debionne S., Viallet P., Dehotin J., Henine H., Nedelec Y., Nedelic Y., Anquetin S. 2010. Towards multi-scale integrated hydrological models using the LIQUID framework. Overview of the concepts and first application examples. Environmental Modelling & Software 25(2010):1672-1682.
18. Brodeur, J., Bédard, Y., Edwards,G., Moulin, B., 2003. Revisiting the concept of geospatial data interoperability within the scope of human communication Processes. *Transactions in GIS*. 7(2):243–265.
19. Buyya, R., Abramson, D. and Giddy, J., 2000. Nimrod/G: An Architecture for a Resource Management and Scheduling System in a Global Computational Grid. *Proceedings of The Fourth International Conference on High-Performance Computing in the Asia-Pacific Region*, Beijing, China, IEEE Computer Society Press, vol. 1, pp. 283--289 , May 2000.
20. Cao, Y., Yang, C. and Wong, D. 2009. An interoperable spatiotemporal weather radar data disseminating system. *International Journal of Remote Sensing*, 30:1313-1326.
21. CCA, 2010. Common Component Architecture. http://www.cca-forum.org/ (accessed 4 February 2010).
22. CCSM, 2010. Community Climate System Model. http://www.ccsm.ucar.edu/ (accessed 4 February 2010).
23. Chen, J. S. and Hogue, A., 2008. Towards 3D model interoperability in games. *In Proceedings of the 2008 Conference on Future Play: Research, Play, Share*. (Toronto, Ontario, Canada, November 03 - 05, 2008). Future Play '08. ACM, New York, NY, pp.232-235. DOI= http://doi.acm.org/10.1145/1496984.1497032
24. Christensen, J.H., 1997. The Danish Eulerian hemispheric model – a three-dimensional air pollution model used for the Arctic, *Atmospheric Environment*, 31 (24), 4169–4191.
25. Clematis, A., Mineter, M., Marciano, R. 2003. High performance computing with geographical data. *Parallel Computing*, 29(10):1275-1279.
26. Coddington, P. D., Hawick, K. A., and James, H. A., 1999. Web-Based Access to Distributed High-Performance Geographic Information Systems for Decision Support. In *Proceedings of the Thirty-Second Annual Hawaii international Conference on System Sciences-Volume 6 - Volume 6* (January 05 - 08, 1999). HICSS. IEEE Computer Society, Washington, DC, 6015.
27. Constantinescu, M., Adrian Sandu, and Gregory R. Carmichael (2008), Modeling atmospheric chemistry and transport with dynamic adaptive resolution. Computational Geosciences, 12(2), 133-151.
28. Cotton, W. R., Pielke, R. A., Walko, R. L., Liston, G. E., Tremback, C. J., Jiang, H., McAnelly, R. L., Harrington, Y., Nicholls, J. M. E. and Carrio, G. G., 2003. RAMS 2001: Current status and future directions. *Meteorology and Atmospheric Physics*, 82(1-4), 5-29.
29. Damevski, K. 2006. Component Model Interoperability for Scientific Computing. *Doctoral Thesis*. UMI Order Number: AAI3235631.University of Utah, Salt Lake City, UT, USA .
30. Davis, C., Warner, T., Astling, E. and Bowers, J., 1999. Development and application of an operational, relocatable, meso-gammascale weather analysis and forecasting system. *Tellus*, 51A, 710–727.
31. de la Beaujardiere, J. (ed.), 2006. OpenGIS Web Map Server Implementation Specification, Version 1.3.0. OGC® 06-042. Open Geospatial Consortium. 85 pp.

32. Domenico, B., Caron, J., Davis, E., Kambic, R., and Nativi, S., 2002. Thematic Real-time Environmental Distributed Data Services (THREDDS): Incorporating Interactive Analysis Tools into NSDL, *Journal of Digital Information*, Vol. 2, No. 4. http://jodi.ecs.soton.ac.uk/Articles/v02/i04/Domenico/(accessed 4 February 2010).
33. Dulac, F., Moulin, C. And Lambert, C.E., 1996. Quantitative remote sensing of African dust transport to the Mediterranean. In: S. Guerzoni, R. Chester,eds. *The impact of desert dust across the Mediterranean*. Netherlands: Klwer Academic Publisher, 25-49.
34. Duncan, D., X. Chu, C.Vecchiola, and R. Buyya. 2009. The Structure of the New IT Frontier: Cloud Computing. *Strategic Facilities Magazine*, 9: 67-72.
35. Ekanayake, J., Fox, G., 2009. High Performance Parallel Computing with Clouds and Cloud Technologies. In: Jaatun, M.G., Zhao, G., Rong, C. (eds.) *Cloud Computing*. 2009. LNCS, vol. 5931, Springer, Heidelberg (2009), http://grids.ucs.indiana.edu/ptliupages/publications/cloudcomp_camera_ready.pdf(accessed 4 February 2010).
36. Erl T. 2005. Service-Oriented Architecture: Concept, Technology, and Design. Prentice Hall. Upper Saddle River, New Jersey.
37. Erl T. 2009. SOA Design Patterns. Prentice Hall. Upper Saddle River, New Jersey.
38. ESMF, 2010. Earth System Modeling Framework. http://www.esmf.ucar.edu/ (4 February 2010).
39. Etminani, K.and Naghibzadeh M., 2007. A Min-Min Max-Min selective algorithm for grid task scheduling. In *Proceedings of the 3rd IEEE/IFIP International Conference in Central Asia*, 26 Oct, 2007. DOI: 10.1109/CANET.2007.4401694.
40. Evangelinos, C. and Hill, C., 2008. Cloud Computing for Parallel Scientific HPC Applications: Feasibility of Running Coupled Atmosphere-Ocean Climate Models on Amazon's EC2. In *Proceedings of Cloud Computing and Its Applications,* 2008. http://www.cca08.orgpapers.php(accessed 4 February 2010).
41. Evans, J., 2003. (Ed.): OpenGIS® Web Coverage Service (WCS) Implementation Specification, OpenGIS Implementation Specification,OGC document number 03-065r6, October 2003. Fonseca, F.T., Egenhofer, M.J., Agouris, P., and Camara, G. 2002. Using ontologies for integrated geographic information systems. *Transactions in GIS*, 6 (3): 231-257.
42. Expertcore , 2009. What is Cloud Computing. http://www.expertcore.org/viewtopic.php?f=52&t=1441 (accessed Dec 12, 2009).
43. Fennessy, M. J., Shukla, J., 2000. Seasonal Prediction over North America with a Regional Model Nested in a Global Model. J. Climate, 13, 2605–2627.
44. Fielding R.T., 2000. Architectural styles and the design of network-based software architectures. PhD Thesis, University of California, Irvine, 2000.
45. Fletcher M., Pournelle J. R., Ramage D., Porter D. E., Shervette V., Kelsey R. H. 2009. A Southeast Regional Testbed for Integrating Complex Coastal and Ocean Information Systems. Paper presented at OCEANS 2009, MTS/IEEE Biloxi – "Marine Technology for Our Future: Global and Local Challenges". October 26-29, 2009
46. Foster I. and Kesselman C., 1998. The Grid: Blueprint for a New Computing Infrastructure. San Francisco, CA: Morgan Kaufmann Publishers Inc.
47. Garey, M. R. and Johnson D. S., 1979. Computers and Intractability: A Guide to the Theory of NP-Completeness. New York: W. H. Freeman & Co.
48. Giorgi, F., Mearns, L. O., Shields, C., McDaniel, L., 1996. Regional Nested Model Simulations of Present Day and 2 × CO_2 Climate over the Central Plains of the U.S. Climatic Change Volume 40, Numbers 3-4, 457-493, DOI: 10.1023/A:1005384803949

49. Gong, S. L., Zhang, X. Y., Zhao, T. L., McKendry, I. G., Jaffe, D. A. and Lu, N. M., 2003. Characterization of soil dust aerosol in China and its transport and distribution during 2001 ACE-Asia: 2. Model simulation and validation. *Journal of Geophysical Research*, 108(D9), 4262. doi:10.1029/2002JD002633.
50. Goodall, J. L., Horsburgh, J.S., Whiteaker, T.L., Maidment, D.R., Zaslavsky, I., 2008. A first approach to web services for the National Water Information System. Environmental Modelling & Software 23(2008): 404-411.
51. Goodall, J. L., Robinson, B. F., Catronova, A. M., 2011. Modeling water resource systems using a service-oriented computing paradigm, in Environmental Modelling & Software 26(2011): 573-582
52. Goudie, A. S.and Middleton, N. J.,1992. The changing frequency of dust storms through time. *Earth and Environmental Science* , 20(3), 197-225
53. Granell C., Diaz L, Gould M., 2010. Service-oriented applications for environmental models: Reusable geospatial services, in Environmental Modelling & Software 25(2010): 182-198.
54. Gray, K.,2009. Jim Gray on eScience: A Transformed Scientific Method, in The Fourth Paradigm, Data-Intensive Scientific Discovery, T. Hey, S. Tansley, and K. Tolle, Editors. 2009, Microsoft Research: Redmond, Washington. p. xvii-xxxi.
55. Guest, M. F., Apra, E., Früchtl, H. A., Harrison, R. J., Kendall, R. A., Kutteh, R. A., Long, X., Nicholas, J. B., Nichols, J. A., Taylor, H. L., Wong, A. T., Fann, G. I., Littlefield, R. J., and Nieplocha, J., 1999. High-performance computing in chemistry: NW Chem. *Future Generation Computer Systems*. 12(4): 273-289.
56. Han, Z., Ueda, H., Matsuda, K., Zhang, R., Arao, K., Kanai, Y. and Hasome, H., 2004. Model study on particle size segregation and deposition during Asian dust events in March 2002. *Journal of Geophysical Research*, 109, D19205, doi:10.1029/2004JD004920.
57. Harzallah Y., Michel V., Liu Q., Wainer G., 2008. Distributed Simulation and Web map mash-Up for Forest Fire Spread, paper presented at 2008 IEEE Congress on Services 2008 – Part 1. pp 176-183
58. Haug, E., Dubois J., Clinckemaillie J., Vlachoutsis S., and Lonsdale G., 1994. Transport vehicle crash, safety and manufacturing simulation in the perspective of high performance computing and networking, Future Generation Computer Systems, 10(2-3): 173-181.
59. Henderson, T., Baillie, C., Carr, G. Hart, L., Marroquin, A., and Rodriguez, B., 1994. Parallelizing the Eta Weather Forecast Model: Initial Results, *Proceedings of High Performance Computing '94. Society for Computer Simulation*, La Joya, California.
60. Hoff, M.L., Tucker, C.J., Lawrence, W.T., Stutzer, D.C. 2000. The use of multisource satellite and geospatial data to study the effect of urbanization on primary productivity in the United States .Geoscience and Remote Sensing, . 38(6): 2549 – 2556.
61. Horsburgh, J.S., Tarboton, D.G., Piasecki, M., Maidment, D.R., Zaslavsky, I., Valentine, D., Whitenack, T., 2009. *Environmental Modelling & Software* 24(2009):879-888.
62. Horzallah, Y., Michel, V., Liu, Q., Wainer, G., 2008. Distributed Simulation and Web Map Mash-Up for Forest Fire Spread. 2008 IEEE Congress on Services – Part I. 176-183.
63. HowStuffWorks, 2009. How Cloud Computing Works. http://communication.howstuffworks.com/cloud-computing.htm (accessed Dec 12, 2009).
64. Hu S. and Bian, L., 2009. Interoperability of functions in environmental models: a case study in hydrological modeling. *International Journal of Geographic Information Science*, 23(5): 657-684.
65. Hu, C., Di, L., Yang, W., Wei, Y., Bai, Y., Lynnes, C., Enloe, Y., Domenico, B., Ruteldge, G., 2008. Interoperability middleware between geoscience and geospatial catalog protocols.

In: Proceedings of IEEE International Geoscience and Remote Sensing Symposium (IGARSS) 2008, July 6-11, 2008. Boston, MA, USA.Vol. 5, No. 5, pp.89-92.

66. Huang Q., C.Yang, H.Wu, W.Li, J.Xie, and C.Yang. 2010. GeoInformation Computing Platforms. In *Advanced GeoInformation Science*, eds. C. Yang, D. Wong, Q. Miao and R. Yang, Taylor & Francis.
67. Huang, Q. and Yang, C., 2010. Optimizing grid computing configuration and scheduling for geospatial analysis -- An example with interpolating DEM , *Computers & Geosciences*, 37(2), 165-176.
68. Ibarra, O.H. and Kim,C.E., 1977. Heuristic Algorithms for Scheduling Independent Tasks on Nonidentical Processors. Journal of the ACM, 24(2): 280-289.
69. Irannejad, P. and Shao, Y., 1998. Description and validation of the atmosphere–land–surface interaction scheme (ALSIS) with HAPEX and Cabauw data. *Global and Planetary Change*, 19(1-4), 87-114
70. Jakeman, A.J., Letcher, R.A., Norton, J.P., 2006. Ten Iterative Steps in Development and Evaluation of Environmental Models. Environmental Modelling & Software 21(2006): 602-614.
71. Janjic, Z. I., 2003. A Nonhydrostatic Model Based on a New Approach. *Meteorology and Atmospheric Physics*, 82, 271-285.
72. Janjic, Z. I., J. P. Gerrity, Jr. and S. Nickovic, 2001. An Alternative Approach to Nonhydrostatic Modeling. *Monthly Weather Review*, 129, 1164-1178.
73. Janjic, ZI., 2003. A nonhydrostatic model based on a new approach. Meteorology and Atmospheric Physics, 82:271-285.
74. Jasper, K., Gurtz, J., Lang, H., 2002. Advanced flood forecasting in Alpine watersheds by coupling meteorological observations and forecasts with a distributed hydrological model. Journal of Hydrology, 267(1-2), 40-52
75. Jin H., Jost,G., Yan J., Ayguade,E., Gonzalez,M., and Martorell, X. 2003. Automatic multilevel parallelization using OpenMP. *Sci. Program*. 11(2):177-190. Jin H., Frumkin M., and Yan J. 2000. Automatic Generation of OpenMP Directives and Its Application to Computational Fluid Dynamics Codes .*High Performance ComputingLecture Notes in Computer Science*, 1940(2000):440-456.DOI: 10.1007/3-540-39999-2_42
76. Jin, H., Jost, G., Yan, J., Ayguade, E., Gonzalez, M. and Martorell, X., 2003. Automatic multilevel parallelization using OpenMP. *Sci. Program*, 11(2), 177-190.
77. Josuttis N. 2007. SOA in Practice. O'Reilly. Sebastopol, CA.
78. Kallos, G., et al, 1997. The Regional Weather Forecasting System SKIRON: An overview. In proceedings of the symposium on regional weather prodiction on parallel computer environments, pp. 109-122, University of Athen, Greece.
79. Koh, M., Peng, L. and See, S., 2005. Integration of Parallel MM5 with Distributed Resource Manager and Performance Evaluation. *Proceedings of the Eighth International Conference on High-Performance Computing in Asia-Pacific Region (HPCASIA'05)*, IEEE Computer Society, Washington, DC, pp.289.
80. Koziar, C., Reilein, R. and Runger, G., 2005. Load imbalance aspects in atmosphere simulations. *International Journal of Computational Science and Engineering*, 1(2-4), 215 – 225.
81. Kuligowski, R. J., and A. P. Barros (1999), High-resolution short-term quantitative precipitation forecasting in mountainous regions using a nested model, *J. Geophys. Res.*, 104(D24), 31,553–31,564, doi:10.1029/1999JD900938.

82. Lenz, C.J., Majewski, D. and Wetterdienst, D., 2002. Meteo-GRID: World-wide Local Weather Forecasts by GRID Computing. *Proceedings of TERENA Networking Conference,* Elsevier Science, Limerick, Ireland.
83. Leslie, L.M. and Wightwick, G.R., 1995. A new limited-area numerical weather prediction model for operations and research: formulation and assessment. *Monthly Weather Review*, 123, 1759-1775.
84. Li,J., Wu, H., Yang, C., Wong, D.W. and Xie, J., 2011. Visualizing dynamic geosciences phenomena using an octree-based view-dependent LOD strategy within virtual globes. *Computers &Geosciences*, doi:10.1016/j.cageo.2011.04.003 .
85. Liu, H., Orban, D.,2008. GridBatch: Cloud Computing for Large-Scale Data-Intensive Batch Applications, *2008 Eighth IEEE International Symposium on Cluster Computing and the Grid (CCGRID)*, pp.295-305.
86. Liu, Y., Gupta, H., Springer, E., Wagener, T., 2008. Linking Science with Environmental Decision Making: Experiences from an Integrated Modeling Approach to Supporting Sustainable Water Resources Management. Environmental Modelling & Software 23(2008): 846-858.
87. Maheswaran,M., Ali, S., Siegel,H.J., Hensgen, D. and Freund,R.F., 1999. Dynamic mapping of a class of independent tasks onto heterogeneous computing systems. Journal of Parallel and Distributed Computing, 59(2): 107-131.
88. Maidment, D., 2008. Bringing Water Data Together. Journal of Water Resources Planning and Management March/April 2008: 95-96.
89. Marta-Almeida, M., Ruiz-Villarreal, M., Otero, P., Cobas, M, Peliz, A., Nolasco, R., Cirano, M., Pereira, J., 2011. OOF_{ε}: A Python engine for automating regional and coastal ocean forecasts. Environmental Modelling & Software 26(2011): 680-682.
90. McGregor, J. L., 1997. Regional Climate Modelling. Meteorol. Atmos. Phys. 63, 105-117.
91. Michalakes, J.G., 2000. Rsl: A Parallel Runtime System Library For Regional Atmospheric Models With Nesting. Institute for Mathematics and Its Applications, Vol. 117, p.59
92. Mineter, M.J., Jarvis, C.H., Dowers, S., 2003. From stand-alone programs towards grid-aware services and components: a case study in agricultural modeling with interpolated climate data. Environmental Modelling & Software 18(2003): 379-391.
93. Morain, S.A., Sprigg, W.A., 2008. Public Health Applications in Remote Sensing – Final Benchmark Report. Manuscript submitted to NASA on September 30, 2008. NASA Agreement Number: NNSO4AA19A. Digital copy available: http://phairs.unm.edu/publ/PHAiRS%20Benchmark%20Addendum%202-09.pdf
94. Nanjundiah, R.S., 1998. Strategies for parallel implementation of a global spectral atmospheric general circulation model. *High Performance Computing, 1998. HIPC '98. 5th International Conference on 17-20, Dec. 1998*, pp. 452 – 458, IEEE ComputerSociety, Chennai, Madras, India.
95. Natarajan, C., Iyer, R. and Sharma, S., 1993. Experimental Evaluation of Performance and Scalability of a Multiprogrammed Shared Multiprocessor. *In Proc. of the 5th IEEE Symposium on Parallel and Distributed Processing*, pp. 11–18, December 1993, IEEE Computer Society Press, Dallas, TX.
96. Nativi, S., Blumenthal, B., Habermann,T., Hertzmann, D., Raskin, R., Caron, J. , Domenico, B., Ho, Y., and Weber, J., 2004. Differences among the data models used by the Geographic Information Systems and Atmospheric Science communities. In *Proceedings of American Meteorological Society – 20th Interactive Image Processing Systems Conference*, Seattle (WA), Jan 2004. Nickovic, S., Kallos, G., Papadopoulos, A. and Kakaliagou. O.,

2001. A model for prediction of desert dust cycle in the atmosphere. *Journal of Geophysical Research*. 106(D16): 18113-18130.

97. Nativi, S., Blumenthal, B., Habermann,T., Hertzmann, D., Raskin, R., Caron, J. , Domenico, B., Ho, Y., and Weber, J., 2004. Differences among the data models used by the Geographic Information Systems and Atmospheric Science communities. In Proceedings of American Meteorological Society – 20th Interactive Image Processing Systems Conference, Seattle (WA), Jan 2004.
98. Nativi, S., Domenico, B., Caron, J., Davis, E., and Bigagli, L., 2006. Extending THREDDS middleware to serve OGC community. Advanced Geosciences, 8: 57–62.
99. Neilson, R.P, Pitelka, L.F., Solomon, A.M., Nathan, R., Midgley, G.F., Fragoso, J.M.V., Lischke, H., Thompson, K., 2005. Forecasting Regional to Global Plant Migration in Response to Climate Change. American Institute of Biological Sciences. 55(9):749-759.
100. Nickling, W. G. and Gillies, J. A., 1993. Dust emission and transport in Mali, West Africa. *Sedimentology*, 40(5), 859–868.
101. Nickovic, S., Kallos, G., Kakaliagou, O., Jovic, D., 1997. Aerosol production/transport/deposition processes in the ETA model: desert dust cycle simulations. *Proceedings of the Symposium on Regional Weather Prediction on Parallel Computer Environments*, University of Athens, Greece, pp. 109–122.
102. Nickovic, S., Kallos, G., Papadopoulos, A. and Kakaliagou. O., 2001. A model for prediction of desert dust cycle in the atmosphere. Journal of Geophysical Research. 106(D16): 18113-18130.
103. Nickovic, S., Kallos, G., Papadopoulos, A., Kakaliagou, O., 2001. A model for prediction of desert dust cycle in the atmosphere : Quantifying the radiative impacts of mineral dust (DUST). *Journal of geophysical research*, 106(D16), 18113-18129
104. OASIS. 2006. Reference Model for Service Oriented Architecture 1.0, Official OASIS Standard (Normative PDF), Oct. 12, 2006. http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf. Accessed on 4/27/2011
105. OASIS. 2009. OASIS Reference Architecture Foundation for Service Oriented Architecture 1.0, Committee Draft 2 (Authoritative PDF). Oct. 14, 2009.http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/soa-ra-cd-02.pdf, Accessed on 4/27/2011.
106. Ohlman, B., Eriksson, A., Rembarz, R.,2009. What Networking of Information Can Do for Cloud Computing, *Enabling Technologies: Infrastructures for Collaborative Enterprises, 2009. WETICE '09. 18th IEEE International Workshops on* , pp.78-83.
107. Oram, A., 2001. Peer-to-Peer: Harnessing the Power of Disruptive Technologies.*O'Reilly Media, Inc.*
108. Overpeck, J., Rind, D., Lacis, A. and Healy, R., 1996. Possible role of dust-induced regional warming in abrupt climate change during the last glacial period. *Nature*, 384, 447 - 449
109. Oxley T., McIntosh B. S., Winder N., Mulligan M., G. Engelen, 2004. Integrated Modelling and Decision-Support Tools: a Mediterranean Example, in Environmental Modelling & Software 19(2004): 999-1010.
110. Papazoglou M. P., Traverso P., Dustdar S., Leymann F. 2007. Service-Oriented Computing: State of the Art and Research Challenges, in Computer November 2007: 38-45.
111. Platform Computing Inc., Platform LSF. http://www.platform.com/Products/Platform.LSF.Family/(accessed 4 February 2010).
112. Pleim, J. E., J. S. Chang, and K. Zhang (1991), A Nested Grid Mesoscale Atmospheric Chemistry Model, *J. Geophys. Res.*, 96(D2), 3065–3084, doi:10.1029/90JD02026.
113. Purohit, S., Kaginalkar, A., Jindani, I., Ratnam, J. V. and Dash, S. K.,1999. Development of parallel climate/forecast models on 100 GFlops PARAM computing systems. *Proceedings*

of the eight ECMWF workshop on the use of parallel processors in meteorology. World Scientific: Reading, UK.

114. Ramón, D.E., Laprise, R., Denis, B., 2002. Forecasting Skill Limits of Nested, Limited-Area Models: A Perfect-Model Approach. Mon. Wea. Rev., 130, 2006–2023. doi: 10.1175/1520-0493(2002)130<2006:FSLONL>2.0.CO;2.
115. Reed, D. 2008. High-Performance Computing: Enabling Climate Change Analysis, http://www.microsoft.com/environment/our_commitment/articles/high_performance_computing.aspx accessed 24 Dec, 2009).
116. Richardson, L. and Ruby, S., 2007. RESTful Web Services. *O'Reilly Media, Inc.*
117. Risien C.M., Allan J.C., Blair R., Jaramillo A.V., Jones D., Kosro P.M., Martin D., Mayorga E., Newton J.A., Tanner T., Uczekaj S.A., 2009. Paper presented at OCEANS 2009, MTS/IEEE Biloxi – "Marine Technology for Our Future: Global and Local Challenges". October 26-29, 2009.
118. Rodriguez, B., Hart, L. and Henderson, T., 1995. Comparing scalable programming techniques for weather prediction. *In Proceedings of the conference on Programming Models for Massively Parallel Computers (PMMP '95)*, pp.111 - 120, IEEE Computer Society: Washington, DC, USA.
119. Rodriguez, B., Hart, L. and Henderson, T., 1996. Parallelizing Operational Weather Forecast Models for Portable and Fast Execution. *Journal of Parallel and Distributed Computing*, 37(2),159-170.
120. Rosmond, T.E., Teixeira, J., Peng, M.,2002. Navy Operational Global Atmospheric Prediction System (NOGAPS). *Oceanography*, 15(1), 99-108.
121. Sanbonmatsu, K.Y., and Tung, C.S., 2007. High performance computing in biology: Multimillion atom simulations of nanoscale systems. *Journal of Structural Biology*, 157(3): 470-480.
122. Schut, P., 2007. OpengGIS® Web Processing Service. Open Geospatial Consortium. 87 pp.
123. Sela, J. G., 1980: Spectral modeling at the National Meteorological Center. Mon. Wea. Rev., 108, 1279–1292.
124. Shao, Y. and Dong, C.H., 2006. A review on East Asian dust storm climate, modelling and monitoring. *Global and Planetary Change,* 52(1-4), 1-22.
125. Shao, Y., Leys, J. F., McTainsh, G. H. and Tews, K., 2007. Numerical simulation of the October 2002 dust event in Australia. *Journal of Geophysical Research*, 112, D08207, doi:10.1029/2006JD007767.
126. Sikder I. U., 2008. Geospatial Web Services in Environmental Planning. Paper presented at the 11th International Conference on Computer and Information Technology (ICCIT 2008), 25-27 December, 2008, Khulna, Bangladesh. Pp. 424-429.
127. Sokolik, I. N., and Toon, O. B., 1996. Direct radiative forcing by anthropogenic airborne mineral aerosols. *Nature*, 381, 681 – 683.
128. Sterling, T., Savarese, D., Becker, D.J., Fryxell, B. and Olson, K., 1995. Communication overhead for space science applications on the Beowulf parallel workstation. *Fourth IEEE International Symposium on High Performance Distributed Computing (HPDC-4 '95),* 2-4 Aug, 1995, Washington, D.C., IEEE Computer Society, pp.23.
129. Sun Microsystems. 2010. Sun Microsystems, Inc., Grid Engine. http://gridengine.sunsource.net/(accessed 4 February 2010).
130. Sun White Paper, 2009. Introduction to Cloud Computing Architecture, 1 edition, June, 2009. http://webobjects.cdw.com/webobjects/media/pdf/Sun_CloudComputing.pdf (accessed Dec 12, 2009).

131. Toon , O.B., Turco, R.P., Westphal, D. L., Malone, R. and Liu, M.S., 1988. A multidimensional model of the physics and chemistry of aerosols and gases: Description of computational analogs. *Journal of the Atmospheric Sciences,* 45, 2097-2117.
132. Uno, I., Amano, H., Emori, S., Kinoshita, K., Matsui, I. and Sugimoto, N., 2001. Trans-Pacific yellow sand transport observed in April 1998: A numerical simulation. *Journal of Geophysical Research*, 106(18), 331–18,344.
133. Vaquero,L.M., Rodero-Merino, L., Caceres, J., Lindner, M., 2009. A break in the clouds: towards a cloud definition, *ACM SIGCOMM Computer Communication Review*, 39(1): 50-55.
134. Vecchiola,C., Duncan,D. and Rajkumar Buyya, 2010. The Structure of the New IT Frontier: Market Oriented Computing. *Strategic Facilities Magazine*, Issue 10: 59-66.
135. Voinov, A. A., DeLuca C., Hood R. R., Peckham S., Sherwood C. R., 2010. A Community Approach to Earth Systems Modeling, in EOS 91(13):117-118.
136. Votava, P., Nemani, R., Bowker, C., Michaelis, A. and Coughlan, J., 2002. Distributed application framework for Earth Science data processing. In *Proceeding of IEEE International of Geosciences and Remote SensingSymposium*(IGARSS'02) 2, pp. 24-28.
137. Vretanos, P.A., (ed.) 2005. Web Feature Service Implementation Specification, Version 1.1.0. OGC 04-094. 2005, Open Geospatial Consortium. 117 pp.
138. Wang S. and Armstrong M. P., 2003. A quadtree approach to domain decomposition for spatial interpolation in grid computing environments. *Parallel Computing*, 29 (10): 1481–1504.
139. Wang S., Armstrong M. P., and Bennett D. A., 2002. Conceptual Basics of Middleware Design to Support Grid Computing of Geographic Information. In *Proceedings of 2nd International Conference on Geographic Information Science*, September 25-28, 2002, Boulder, CO, USA, pp.197-200.
140. Wang, S., and Liu, Y. 2009. TeraGrid GIScience Gateway: Bridging Cyberinfrastructure and GIScience. *International Journal of Geographical Information Science*, 23 (5): 631 – 656.
141. Wang, Y. X., M. B. McElroy, D. J. Jacob, and R. M. Yantosca (2004), A nested grid formulation for chemical transport over Asia: Applications to CO, *J. Geophys. Res.*, 109, D22307, doi:10.1029/2004JD005237.
142. Westphal, D. L., Toon, O. B. and Carlson, T. N., 1988. A Case Study of Mobilization and Transport of Saharan Dust. *Journal of the Atmospheric Sciences*, 45(15), 2145–2175.
143. White, B. G., Paegle, J. , Steerburgh, W. J. , Worel, J. D. , Swanson, R. T., Cook, L. K., Onton, D. J., Myles, J. G., 1999: Short-term forecast validation of six models. Wea. Forecasting, 14, 84–108.
144. Whiteside, A., Evans, J.D. (eds.), 2006. Web Coverage Service (WCS) Implementation Specification, Version 1.1.0. 06-083r8. 2006, Open Geospatial Consortium. 129 pp.
145. Wieczorek, M. , Hoheisel, A. , and Prodan, R. 2009. Towards a general model of the multi-criteria workflow scheduling on the grid. *Future Generation Computer Systems*, 5(2009): 237-256.
146. Wiese, I. S. and Huzita, E. H. 2006. IMART: An Interoperability Model for Artifacts of Distributed Software Development Environments. In *Proceedings of the IEEE international Conference on Global Software Engineering*(October 16 - 19, 2006). ICGSE. IEEE Computer Society, Washington, DC, pp.255-256.
147. Wiki, 2009. Wiki Cloud Computing. http://en.wikipedia.org/wiki/Cloud_computing(accessed April 29, 2009)
148. Wolters, L., Cats, G. and Gustafsson, N., 1995. Data-parallel numerical weather forecasting. *Sci. Program*, 4(3), 141-153.

149. World Wide Web Consortium (W3C), 2007. SOAP Version 1.2 Part 1: Messaging Framework (Second Edition). Available from: http://www.w3.org/TR/soap12-part1/. Accessed 2008-06-27
150. WRF-NMM, 2011. WRF-NMM User Guide Version 3[online]. National Center for Atmospheric Research. http://www.dtcenter.org/wrf-nmm/users/docs/users_guide/V3/users_guide_nmm_chap1-7.pdf [Access 4 April 2011]
151. Wright, D.J., O'Dea, E., Cushing, J.B., Cuny, J.E., and Toomey, D.R., 2003. Why Web GIS may not be enough: A case study with the Virtual Research Vessel. *Marine Geodesy*, 26 (1-2):73-86.
152. Wu, M. and Sun, X., 2009. Memory conscious task partition and scheduling in Grid nvironments. Proceedings of the Fifth IEEE/ACM International Workshop on Grid Computing (GRID'04).
153. Xie, J., Yang C., Zhou B., Huang Q., 2010. High performance computing for the simulation of dust storms. *Computers, Environment, and Urban Systems*. 34(4):278-290.
154. Xue, M., Wang, D., Gao, J., Brewster, K. and Droegemeier, K.K., 2003. The Advanced Regional Prediction System (ARPS), storm-scale numerical weather prediction and data assimilation. *Meteorology and Atmospheric Physics*, 82(1-4), 139-170.
155. Yang, C., and Raskin, R., 2009. Introduction to Distributed Geographic Information Processing. *International Journal of Geographic Information Science*, 23(5):1-8.
156. Yang, C., D. Wong, B. Li., 2005. Introduction to computing & computational issues of Distributed GIS. *Geographic Information Sciences*, 11(1): 1-3.
157. Yang, C., Li, W., Xie, J. and Zhou, B., 2008. Distributed geospatial information processing: sharing earth science information to support Digital Earth, *International Journal of Digital Earth*, 1(3), 259-278.
158. Yang, C., Wong, D., Kafatos, M. and Yang, R., 2006. Implementing computing techniques to accelerate network GIS. In Proc. SPIE Vol. 6418, 64181C, Geoinformatics 2006: GNSS and Integrated Geospatial Applications (2006), ed. Li, D., Xia, L.
159. Yang, C., Wu, H., Huang, Q., Li, Z. and Li, J., 2011a. Using spatial principles to optimize distributed computing for enabling the physical science discoveries. *Proceeding of National Academy of Sciences of USA*, 108 (14), 5498-5503.
160. Yang, C., Goodchild, M., Huang, Q., Nebert, D., Raskin, R., Xu, Y., Bambacus, M., Fay, D., 2011b. Spatial Cloud Computing: How could geospatial sciences use and help to shape cloud computing. *International Journal on Digital Earth*. 4(4), 305-329.
161. Zeng, Y., Wu, B., Li, G., 2010. A Study of on-Demand Service for Spatial Data System. Paper presented at the 2nd Conference on Environmental Science and Information Application Technology. pp 98-101
162. Zhou, S., 2006. Coupling climate models with the Earth System Modeling Framework and the Common Component Architecture. *Concurrency and Computation: Practice and Experience*, 18(2): 203 – 213.
163. Zhou, S., Balaji, V., Cruz, C., da Silva, A., Hill, C., Kluzek, E., Smithline, S., Trayanov, A., and Yang, W., 2007. Cross-organization interoperability experiments of weather and climate models with the Earth System Modeling Framework: Research Articles. *Concurrency and Computation: Practice and Experience*. 19(5):583-5.

CURRICULUM VITAE

ACADEMIC EXPERIENCE

Ph.D. candidate, Department of Geography and Geoinformation Science, George Mason University, August 2007-Present

Dissertation: "Utilizing Model Interoperability and Spatial Cloud Computing to Enable the Computability of Dust Storm Forecasting"

M.S. Degree, Cartography and Geographical Information Systems. Peking University, September 2004-June 2007

B.E. Degree, Survey and mapping Engineer, Central South University, Changsha, China, September 2000-June 2004

RESEARCH INTERESTS

Cloud Computing, High Performance and Distributed Computing; Geographic Information Science and Systems; Data/model/services interoperability and integration; Spatial web services/portal; Geospatial Sciences

PUBLICATIONS – Peer Reviewed Manuscripts

1. Yang, C., Wu, H., Huang, Q., Li, Z. and Li, J., 2011a. Using spatial principles to optimize distributed computing for enabling the physical science discoveries. *Proceeding of National Academy of Sciences of USA*, 108 (14), 5498-5503.
2. Yang, C., Goodchild, M., Huang, Q., Nebert, D., Raskin, R., Xu, Y., Bambacus, M., Fay, D., 2011b. Spatial Cloud Computing: How could geospatial sciences use and help to shape cloud computing. *International Journal on Digital Earth*. 4(4), 305-329.
3. Huang Q., Yang C., Nebert D., Liu K., Wu H. Cloud Computing for Geosciences: Deployment of GEOSS Clearinghouse on Amazon's EC2. In *proceedings of ACM SIGSPATIAL International Workshop on High Performance and Distributed Geographic Information Systems (HPDGIS)*, November 2rd , 2010, San Jose, CA.
4. Huang Q. and Yang C. Optimizing grid computing configuration and scheduling for geospatial analysis -- An example with interpolating DEM ,*Computers & Geosciences*,37(2):165-176.
5. Huang Q., C.Yang, H.Wu, W.Li, J.Xie, C.Yang. 2010. GeoInformation Computing Platforms. In Advanced GeoInformation Science, eds. C. Yang, D. Wong, Q. Miao

and R. Yang, Taylor & Francis, pp. 79-126.

6. Yang, C., Wu, H., Huang, Q., Li, Z., Li, J., Li, W., Miao, L., and Sun, M., 2011. WebGIS Performance, *ISPRS book on Web and Wireless GIS*, Boca Ration: CRC Press.
7. Nebert D., Huang Q, Ling J. and Sun M. 2010. NSDI/GOS/FEA. In *Advanced GeoInformation Science*, eds. C. Yang, D. Wong, Q. Miao and R. Yang, Taylor & Francis, pp. 256-267.
8. Xie J., C. Yang, B. Zhou, and Huang Q. 2009. High performance computing for the simulation of dust storms. In *Computers, Environment, and Urban Systems,* doi: DOI: 10.1016/j.compenvurbsys.2009.08.002.
9. Huang Q., Mao S., Jiang Y., Ru B., Li M., Dong P. 2009. Utilizing Particle System to Simulate airflow of laneway in underground mine environment. Geoinformatics. 2009*. 17th International Conference on 12-14 Aug. 2009,* pp.1-6*,* doi: 10.1109/GEOINFORMATICS.2009.5293434.
10. Li J., Wu H., Yang C., Xie J., and Huang Q., 2009. Using progressive transmission of 3D/4D geospatial information over the Internet to facilitate geovisualization in World Wind, *Geoinformatics 2009*. 30:1313-1326.
11. Xie J., Yang C., Huang Q, Cao Y,Kafatos M. 2008. Utilizing grid computing to support near real-time geospatial applications, Geoscience and Remote Sensing Symposium, 2008. IGARSS 2008. IEEE International, pp.1290-1293.
12. Huang Q., Mao S., Li M., Ru P. 2006. The safety assurance system of coalmine based on the three-tiers B/S architecture. Coal Engineering, 2006(11): 10-14.

PRESENTATIONS AT NATIONAL AND INTERNATIONAL CONFERENCES

1. Huang Q. , Yang C. and Benedict K., 2011. Using spatiotemporal patterns of geospatial sciences to optimize high end computing for enabling dust storm forecasting and prediction. UCGIS Summer Assembly, June 21-23, 2011, Boulder, CO.
2. Huang Q., 2011. Deployment of GEOSS Clearinghouse on Amazon EC2 and Azure. AAG 2010 Annual Meeting, April 12th , Seattle, Washington.

3. Huang Q., Yang C., Nebert D., Liu K., Wu H. 2011. Cloud computing for geosciences: Deployment of GEOSS Clearinghouse on Amazon's EC2. Jan 4th , 2011, Washington DC, USA.
4. Huang Q., Yang C., Nebert D., Liu K., Wu H.2010. Cloud computing for geosciences: Deployment of GEOSS Clearinghouse on Amazon's EC2. November 2rd, 2010, San Jose, CA.
5. Huang Q., Yang C, Wu H, Liu K, Li J.2010. Cloud Computing for Earth Science. July 20-23, 2010, ESIP Federation Meeting, Knoxville, TN.
6. Huang Q., Yang C, Wu H, Xie J, Li J, Li Z, Sun M. 2010.Cloud Computing for Earth Science –Parallelize and schedule spatial computing for WRF-NMM model , AAG 2010 Annual Meeting, , April 15th , Washington, DC (Best student presentation, 2nd place).
7. Huang Q., Yang C, Xie J, Wu H, Li J. Utilizing Model Interoperability and High Performance Computing to Enhance Dust Storm Simulation , AGU 2009 Fall Meeting, Dec 15th , 2009, San Francisco, CA.
8. Huang Q., Mao S., Jiang Y., Ru B., Li M., Dong P. 2009. Utilizing Particle System to Simulate Airflow of Laneway in Underground Mine Environment. 17th International Conference on 12-14 Aug, 2009, Fairfax, VA, USA.
9. Huang Q., Yang C., Wu H., A near real-time Earth Observation Data dissemination service, AAG 2009, Las Vegas, USA.

HONORS AND AWARDS

2011.06	UCGIS Summer Assembly Student Travel Award
2010.04	AAG 2010 Best Student Competition Paper Awards (Cloud Computing for Earth Science)
2007- 2010	George Mason University Presidential Scholarship
2004. 06	Award for best undergraduate thesis in Department of Physical Engineering Information (June, 2004)

PROFESSIONAL EXPERIENCE and POSITIONS

2008	Teaching Assistant
	GEOG 563 (Advanced Geographic Information System), Spring Semester George Mason University
2007-	Research Assistant

Research and Development: High Performance Computing/Cloud Computing for data and computing intensive applications , Joint Center for Spatial Intelligent Computing, George Mason University

2006-2007 Research Assistant

Research and Development: 3D Ventilation System, Institute of Geographic Information System and Remote Sensing, Peking University,

2005-2006 Software Engineer

Research and Development: Information management system for Coal Mine, Longruan Technology Company, Beijing, P. R. China

2004-2005 Research Assistant

Research and Development: Geographic Information Systems for Coal Mine Industry, Institute of Geographic Information System and Remote Sensing, Peking University, Beijing, P. R. China

RESEARCH PROJECTS

2011.02- Cloud Enable GEOSS Clearinghouse

- Sponsor: Microsoft
- Objective: Explore the feasibility of using Microsoft Azure platform to facilitate data intensive applications.
- Responsibility: Migrating GEOSS Clearinghouse to the Microsoft Azure cloud platform, and optimizing the performance of GEOSS Clearinghouse under Azure cloud platform to enable massive concurrent users requests globally located.

2010.05- GeoCloud Sanbox Initiative

- Sponsor: U.S. Federal Geographic Data Committee (FGDC)
- Objective: Explore and document migration and management strategies of web applications into cloud, monitor usage and costing of Cloud services, and pursue shared system security profiles (certification and accreditation) for such cloud solutions
- Responsibility: Server as a pioneer to test and explore the cloud platforms, such as Amazon EC2, Enstratus, to support geospatial applications, and migrate GEOSS clearinghouse to Amazon EC2 cloud platform.

2010. 04- Spatial Cloud Computing

- Sponsor: NASA Goddard

- Objective: Facilitate the earth science research and applications through cloud computing.
- Responsibility: Implement Platform as a service (PaaS) for Earth Science, Geospatial middleware design and development.

2009.09 –2011.01 Multi-resolution Nested Dust Forecast System Feasibility Study

- Sponsor: NASA Public health
- Objective: Utilized model interoperability and cloud computing to enhance the dust storm simulation to support public health decisions
- Responsibility: Modify model pre- and post-processors of ETA-8, NMM-dust to support OGC standards and REST data transfer; evaluate performance characteristics of the nested model system; Build up DB, system design and implementation with JAVA, JSP, JS, XML, HTML, high performance computing technology, Linux Shell programming.

2008.09 - 2008.12 DEM Interpolation with the support of Grid computing

- Objective: Utilize the technology of grid computing which can provide a considerable amount of computational power to support the DEM interpolation.
- Responsibility: Algorithm design, system design and implementation with JAVA, grid computing technology, Linux Shell programming.

2008.05 - System administrator & Web master

- Responsibility: Managing the cluster and servers of CISC, and developing and maintaining several web sites: CISC (http://cisc.gmu.edu), ESG (http://esg.gmu.edu), EIE (http://eie.cos.gmu.edu), CPGIS (Chinese Professional in Geographic Information Systems, http://cpgis.org), COAA (Chinese-American Oceanic and Atmospheric Association, http://coaaweb.org), WMS (http://wms.gmu.edu).

2008.04 - 2008.08 WECHO: A Water and Energy Cycle EOS House Web Portal

- Objective: Use EOS Clearing House (ECHO) APIs to provide access and discovery of data resources related to the NASA Global Water and Energy Cycle Focus Area (WECFA).
- Responsibility: Implement the function of ordering data from LP DAAC, and user account management with Java Servlet, Java Script, XML, and HTML.
- Website: http://eie.cos.gmu.edu/web/guest/WECHO

2007.09- 2008.05 Real-time Routing

- Objective: Design and implement a reasonable path-finding algorithm applicable for real-time routing; based on the grid computing technology, to resolve the problem of getting the real-time traffic data from traffic simulation system when massive data and large area involved.
- Responsibility: Algorithm design, system design and implementation with JAVA, grid computing technology, Linux Shell programming.

PROFESSIONAL SERVICE

2010	Session Organizer Computational Geography Session on 2011 American Association of Geographers annual meeting
2009	Session Organizer High Performance Computing for Geographic Sciences Session on 2010 American Association of Geographers annual meeting
2009-2010	Web Master Assistant Chinese Professional in Geographic Information Systems(CPGIS, http://cpgis.org)
2009-	Web Master Assistant Chinese-American Oceanic and Atmospheric Association (COAA, http://coaaweb.org)

PROFESSIONAL MEMBERSHIPS

2007-	American Association of Geographers (AAG)
2008-	American Geography Union (AAG)
2010-	Association for Computing Machinery (ACM)

GRANT

2010 ESIP student funding (Geospatial Cloud Processing), $3000

2011-2014 Cloud Enable GEOSS Clearinghouse, Microsoft, $100k, (Co-I), PI: C. Yang

2009-2011 GeoCloud Sanbox Initiative, FGDC, $50k, (Co-I), PI: C. Yang