

UTILIZING VOLUNTEER COMPUTING AND VIRTUALIZATION TECHNOLOGY  
FOR CLIMATE SIMULATION

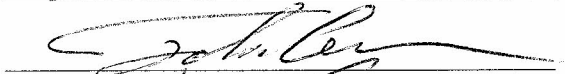
by

Kai Liu  
A Thesis  
Submitted to the  
Graduate Faculty  
of  
George Mason University  
in Partial Fulfillment of  
The Requirements for the Degree  
of  
Master of Science  
Geographic And Cartographic Sciences

Committee:



Dr. Chaowei Yang, Thesis Director



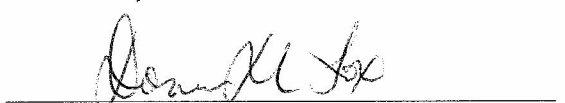
Dr. John Qu, Committee Member



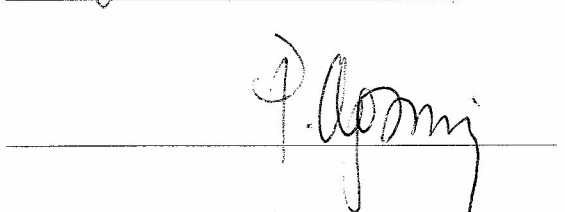
Dr. Ruixin Yang, Committee Member



Dr. Anthony Stefanidis, Department  
Chairperson



Dr. Donna M. Fox, Associate Dean, Office  
of Student Affairs & Special Programs,  
College of Science



Dr. Peggy Agouris, Interim Dean, College  
of Science

Date: 04/29/2014

Spring Semester 2014  
George Mason University  
Fairfax, VA

Utilizing Volunteer Computing and Virtualization Technology for Climate  
Simulation

A thesis submitted in partial fulfillment of the requirements for the degree of Master of  
Science at George Mason University

By

Kai Liu  
Bachelor of Science  
Wuhan University, 2006

Director: Chaowei Yang, Professor  
Department of Geography and Geoinformation Science

Spring Semester 2014  
George Mason University  
Fairfax, VA

Copyright: 2014 Kai Liu  
All Rights Reserved

## **DEDICATION**

This thesis is dedicated to my wonderful family and young brother Mao who always gave me positive and optimistic advice to overcome the difficulties.

This thesis is dedicated to my lovely wife Huifen Wang who has been a great source of motivate and inspiration.

## **ACKNOWLEDGEMENTS**

I would like to thank my advisor, Dr. Chaowei Yang for his mentorship during my graduate studies at George Mason University.

I would like to thank my committee members, Drs. John Qu and Ruixin Yang for their guidance on the dissertation.

I would like to thank the members of the Center for Intelligent Spatial Computing for Water/Energy Science (CISC) at George Mason University.

Finally, and most importantly, I would like to thank my parents and my wife. They have been always supportive of me in my life.

## TABLE OF CONTENTS

LIST OF TABLES .....	VI
LIST OF FIGURES .....	VII
LIST OF ABBREVIATIONS .....	VIII
ABSTRACT .....	IX
CHAPTER 1 INTRODUCTION .....	1
CHAPTER 2 RELATED WORK .....	5
2.1 Supercomputer .....	5
2.2 Volunteer Computing .....	6
2.3 Cloud Computing .....	7
2.4 Virtual Technology .....	8
CHAPTER 3 ARCHITECTURE AND WORKFLOW .....	11
3.1 Architecture .....	11
3.2 Workflow .....	12
CHAPTER 4 METHODOLOGIES .....	14
4.1 Virtual Image Prepare .....	14
4.2 Periodically Upload Mechanism .....	16
4.3 New Credit System (NCS) .....	18
4.4 Climate Simulation Visualization .....	19
CHAPTER 5 EXPERIMENT & RESULT .....	20
5.1 Hosts .....	20
5.2 Running Status .....	22
5.3 Visualization .....	23
CHAPTER 6 CONCLUSION .....	26
REFERENCES .....	30
CURRICULUM VITAE .....	34

## LIST OF TABLES

Table	Page
Table 1 Comparison of super computer, volunteer computing and cloud computing.....	10
Table 2 Statistics of volunteer hosts .....	22
Table 3 Top ten countries with credits in Climate@Home .....	23

## LIST OF FIGURES

Figure	Page
Figure 1 Volunteer computing infrastructure for climate simulation .....	11
Figure 2 Volunteer computing workflow for climate simulation .....	13
Figure 3 VDI file size history .....	16
Figure 4 Periodically Upload Mechanism .....	17
Figure 5 Global distribution of hosts .....	21
Figure 6 Example of NCS.....	22
Figure 7 Dynamic analysis page of the Climate@Home.....	24
Figure 8 2D visualziation in four windows.....	25



## LIST OF ABBREVIATIONS

Atmospheric Model Intercomparison Project .....	AMIP
Area of Interest .....	AOI
Berkeley Open Infrastructure for Network Computing .....	BOINC
Control Data Corporation .....	CDC
Data as a Service .....	DaaS
Floating-point Operations Per Second .....	FLOPS
GNU Compiler Collection .....	GCC
Global Earth Observation System of Systems .....	GEOSS
Goddard Institute for Space Studies .....	GISS
Graphic User Interface .....	GUI
Infrastructure as a Service .....	IaaS
New Credit System .....	NCS
Operating System .....	OS
Platform as a Service .....	PaaS
Periodically Upload Mechanism .....	PUM
Software as a Service .....	SaaS
UK Met Office Unified Model .....	UM
Virtual Disk Image .....	VDI
Virtual Machine .....	VM
Virtual Machine Monitor .....	VMM

## **ABSTRACT**

### **UTILIZING VOLUNTEER COMPUTING AND VIRTUALIZATION TECHNOLOGY FOR CLIMATE SIMULATION**

Kai Liu, M.S.

George Mason University, 2014

Thesis Director: Dr. Chaowei Yang

The climatological community relies increasingly on computing intensive models and applications to study atmospheric chemistry, aerosols, carbon cycle and other tracer gases. These models and applications are becoming increasingly complex and bring geospatial computing challenges for scientists as follows: 1) enormous computational power is required for running these models and applications to produce results in a reasonable timeframe; 2) climate models are always sensitive and require special computing environments; 3) these models are challenging to provide convenient and fast solution to transfer the big data outputs from climate simulations. Presently, volunteer computing is getting more powerful and provides a potential solution for these problems by obtaining super computational resources from global volunteers. Meanwhile, the virtualization technology based on hardware or platforms allows researchers to run sensitive models in a predefined virtual machine. This thesis reports on research to

integrate and optimize volunteer computing and virtualization technology for climate simulation based on the following: 1) utilizing volunteer computing resources so that the heterogeneous home computers can support climate applications; 2) utilizing virtualization technology to make the climate application run on different platforms; 3) optimizing the output collection mechanism to periodically upload climate model output; and 4) optimizing the credit system to grant credits periodically for long time climate simulation tasks. The research is based on NASA and George Mason University's collaborative project Climate@Home, which is the first volunteer computing project using virtualization technology in the climate domain.

## **CHAPTER 1 INTRODUCTION**

In climate domain, there has been a variety of climate models to simulate the interactions of the atmosphere, oceans, land surface, and ice. Energy Balance Models (EBMs) calculates a balance between the radiation arriving at the earth and the energy leaving the earth. The energy balance plays a key role as climate driver. Radiative Convective Models (RCMs) are based on the radiative and convective energy transport up to the atmosphere. General Circulation Models (GCMs) are most complex models and always three-dimensional. GCMs include the physics of the atmosphere, land, sea and ice(Sellers et al. 1986, Liang et al. 1994). Different climate models have been created and used in different projects. For example, climateprediction.net (Stainforth et al. 2002, Stainforth et al. 2005) is a climate prediction project using versions of the Hadley Centre Climate Model (HadCM3(Gordon et al. 2000), HadSM3 (Williams et al. 2001)) and the global set-up of the UK Met Office Unified Model (UM (Cullen 1993)).

NASA Goddard Institute for Space Studies (GISS) has developed a series of GCMs to simulate global climate (Hansen et al. 1983, DelGenio and Yao 1993). The initial model, Model I, used for 60 climate sensitivity experiments with integration times from 3 months to 5 years (Hansen et al. 1983). Based on the Model I, several modifications were incorporated on Model II. Before the current version ModelE (also called as Model III) was developed, some improvements and modification were

incorporated based on the previous models. Atmospheric Model Intercomparison Project (AMIP) used an updated version to determine the systematic climate errors of atmospheric models under realistic conditions and class for the simulation of the climate of the decade 1979-1988 (Gates 1992). Liu et al. (2003) used GISS model to investigate the sensitivity of sea ice to different parameterizations. ModelE version incorporates numerous improvements in basic physics, the stratospheric circulation, and forcing fields (Schmidt et al. 2006). It provides the ability to simulate many different configurations of Earth System Models. These configurations include interactive atmospheric chemistry, aerosols, carbon cycle and other trace gases as well as the standard atmosphere, ocean, sea ice and land surface components. Many projects have been conducted using the GISS ModelE (Hansen et al. 2007, Koch and Hansen 2005, Shindell et al. 2006). Climate@Home is a project using ModelE (Sun et al. 2012) and is used as the experimental project in this thesis.

Recently, climate models and their applications have become increasingly comprehensive (Palmer et al. 2005). Hundreds of variables are used in the model calculation and vast computing resources and enormous run times are required to implement the models and applications. For example, the ModelE contains about 300 variables and it needs 5 to 7 days to finish a 10-year simulation task on a single processor home PC (i.e. a Pentium IV/2GHz machine). The climateprediction.net's full resolution ocean work unit and 45-year simulations in HadSM3 require 4-6 weeks for a single processor home PC to finish (Christensen et al. 2005). Normally, scientists need to run hundreds of tasks for an experiment, which may have different variable values or

different time periods. Accordingly, multiple years are needed to complete the experiment in a single machine. Hence, the computational power is a big challenge for scientists.

Climate models are normally sensitive and critical for computational environments because most of the models are UNIX based and can not be transferred to run on other platforms directly and various libraries, programming language, and compilers are required to install and run these models, For example, Network Common Data Form (NetCDF) 4 libraries (Rew and Davis 1990), Fortran 95 (Metcalf et al. 2004) and GNU Compiler Collection (GCC) 4.4 (Griffith 2002) compilers are required to compile and run ModelE. Additionally, the models are very sensitive and cannot continue to run once the machines have crashed.

Many of the models and applications create enormous data outputs and get bigger with the time frame of the simulation. Taking ModelE as an example, the outputs of a 10-year monthly simulations would be 10 Gigabytes of which 9 are temporary and 1 is monthly simulation results to be uploaded or returned for the further analysis. If scientists need to run 300 ModelE tasks, 300 Gigabytes results are collected.

In this thesis, volunteer computing and virtualization technology are used to solve the above challenges for climate simulation. Volunteer computing can collect and use computing resources from volunteers around the world. It is free and different from other supercomputer. Virtualization technology provides a more convenient way to transfer the climate models to different platforms. New output collection mechanism is created to periodically upload climate model output using File Transfer Protocol via SSL (FTPS).

Credits are always used to represent how many computational resources volunteers have donated to volunteer computing projects. In this thesis, a new credit system is created and used to grant the credits periodically for long time climate simulation tasks. To demonstrate the usages of the volunteer computing and virtualization in real climate applications, the methods in Clamte@Home project are utilized as an experimental case in this thesis.

## **CHAPTER 2 RELATED WORK**

### **2.1 Supercomputer**

A supercomputer (Hayes et al. 1986) is one that has highest processing capacity, which is far higher than a normal computer. Supercomputers were first introduced in the 1960's primarily by Seymour Cray at Control Data Corporation (CDC). The first generation of supercomputers used only a few processors. Since then, more and more supercomputers have been created with increasing computing power. In 2013, the world's fastest supercomputer, Tianhe-2, was deployed at the National Supercomputer Center in Guangzhou, China. It provided a performance of 33.86 petaFLOPS (Floating-point Operations Per Second) on the Linpack benchmark.

In 2004 NASA built the Columbia supercomputer for increasing NASA's high-end computing capability ten-fold for missions in aeronautics, space exploration, and earth and space sciences. Columbia is a constellation comprised of 20 nodes, each containing 512 Intel Itanium2 processors and running on the Linux operating system (Brooks et al. 2005). Many scientists use Columbia to solve problems across many scientific and engineering disciplines. For example, Menemenlis et al. (2005) used Columbia to estimate ocean circulation constrained by in situ and remotely sensed observations; and Mavriplis et al. (2007) investigated the high resolution aerospace applications.



The supercomputer provides fast computing for complex applications and simulations. However, supercomputers have the disadvantages of being extremely expensive to develop and use. With thousands of processors, it consumes large amounts of electrical power and generates too much heat that needs to be cooled. The cost to power and cool the supercomputer can be significant. Moreover, supercomputers require special programming and hardware skills to write the code for applications.

## **2.2 Volunteer Computing**

Volunteer computing uses internet-connected computers, volunteered by their owners, as a source of computing power and storage (Anderson and Fedak 2006). Since 1996 in which the first volunteer computing project "Great Internet Mersenne Prime Search" was launched, volunteer computing has been used in a wide range of scientific projects such as SETI@Home (Anderson et al. 2002) and Folding@Home (Larson et al. 2002). In the early days of the volunteer computing, developers created their applications to combine scientific computation and distributed computing infrastructure. More recently, most of the volunteer computing projects utilize middleware systems to deploy applications such as Berkeley Open Infrastructure for Network Computing (BOINC) (Anderson 2004), Xtremweb (Fedak et al. 2001) and xgrid.

Climateprediction.net is a successful volunteer computing project in climate research. It aims to harness the spare CPU cycles of a million individual users' PCs to run a massive ensemble of climate simulations using an up-to-date, full resolution, three dimensional atmosphere-ocean climate model (Stainforth et al. 2002 ). The BOINC is

used as the volunteer computing middleware in climateprediction.net. In Climateprediction.net, most volunteers' computers run Microsoft Windows operational system, however, the models are UNIX based. To solve the problem, each model version is done under a Linux implementation which includes various pre-processing modules to create the desired FORTRAN code. They are then transferred to Windows version. The main difficulties encountered in porting the code from Linux to Windows were the identification of suitable compiler options and changes in the way environment variables are used (Stainforth et al. 2002).

### **2.3 Cloud Computing**

Cloud computing is a model for enabling ubiquitous, 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). Cloud computing can provide 'Infrastructure as a Service' (IaaS), 'Platform as a Service' (PaaS), 'Software as a Service' (SaaS), and 'Data as a Service' (DaaS) for end users in a 'pay-as-you-go' mode.

Cloud computing has been widely used in geospatial sciences. Huang et al. (2010) used Amazon EC2 cloud to support Global Earth Observation System of Systems (GEOSS) Clearinghouse (Liu et al. 2011) deployment. Using the load balance and auto scalability provided by Amazon EC2, they launched more virtual machines to solve the computation and traffic requirements at peak times and shutting down virtual machines to

save cost when the number of the accesses decreases. Their experiment reveals that the EC2 cloud computing platform facilitates geospatial applications the aspects of scalability, reliability, and reducing duplicated efforts among the geosciences communities. They concluded that different applications are justified and optimized when deploying onto the EC2 platform for a better balance of cost and performance.

The term “Spatial Cloud Computing” was coined by Yang et al. (2011). It refers to the cloud computing paradigm that is driven by the geospatial sciences and optimized by spatiotemporal principles for enabling geospatial science discoveries and cloud computing within distributed computing environment. Utilizing spatial cloud computing can support some climate studies. In this approach, cloud consumers decide how many computing resources are needed for their applications and deploy their application onto commercial or private cloud. In the deployment, they can set scalability and load balancing rules to optimize the computing resource usage.

Cloud computing is advantageous for spatial cloud computing. However, it still has some disadvantages in climate simulations. Given cloud computing ‘pay as you go’ mode, the more computing resources consumed, the more one needs to pay. The climate simulation tasks are different from other applications since they are time consuming. The long time running time will require large budgets to run on cloud computing platforms.

## **2.4 Virtualization Technology**

A virtualized system includes a new layer of software, the Virtual Machine Monitor (VMM). The VMM's principal role is to arbitrate accesses to the underlying

physical host platform's resources so that multiple operating systems (which are guests of the VMM) share them. Numerous systems have been designed to subdivide the ample resources of a modern computer (Barham2003, Liu and Abali 2009). Using virtualization technologies, Virtual Machines (VMs) could be created from the underlying hardware resources and act like a real computer with an Operating System (OS). Despite the underlying OS, VMs could be launched with different OS. For example, a computer that is running Microsoft Windows may host a virtual machine that looks like a computer with the Ubuntu Linux operating system; Ubuntu-based software can be run on the virtual machine (Turban et al. 2008).

There are advantages for volunteer computing to use virtualization technology. First, it packages libraries, programming language, compilers and operating systems to a black box and makes it easier for scientists to develop applications. Second, it provides increased security for volunteers since VMs are a very strong security barrier since a program running in a virtual machine has no access to the files on the "host" operating system. Third, VM apps are automatically "restartable". The contents of the VM are written to disk every few minutes, and if your computer is turned off the application can restart close to where it left off. Recognizing the extreme benefits of using VMs, several volunteer computing projects have been using VMs, including Test4Theory (Lombrana, 2012), Beauty@LHC and RNAworld.

Test4Theory uses volunteer computing and virtualization technology allowing users to participate in running simulations of high-energy particle physics with their home computers. However, it is different from climate simulation tasks. Compared to the

output for Test4Theory, which is about 3-4 Megabytes per 24 hours, climate models and applications require a convenient and fast way to upload big outputs.

Comparing the three computing type: supercomputer, volunteer computing and cloud computing. Table 1 displays the differences in computing, programming and price, illustrating different advantages and disadvantages.

Table 1 Comparison of super computer, volunteer computing and cloud computing

Computing type	Computing Efficiency	Programming	Price
Supercomputing	Data moved between processors rapidly	Costly and difficult to write programs	Expensive
Volunteer Computing	Less efficiency	Stand alone programming	Free to use volunteers' computational resources
Cloud Computing	Less efficiency	Stand alone programming	Pay as you go

In light of the advantages and disadvantages, volunteer computing has been selected as the computing type for the Climate@Home project since it is free and can provides virtual super computer resources. In addition, it attracts public interested in the climate change. Virtualization technology is used in this thesis to package the climate model to a virtual machine for all platforms.

## CHAPTER 3 ARCHITECTURE AND WORKFLOW

### 3.1 Architecture

The infrastructure used in the thesis is “three-tier” (Figure 1).

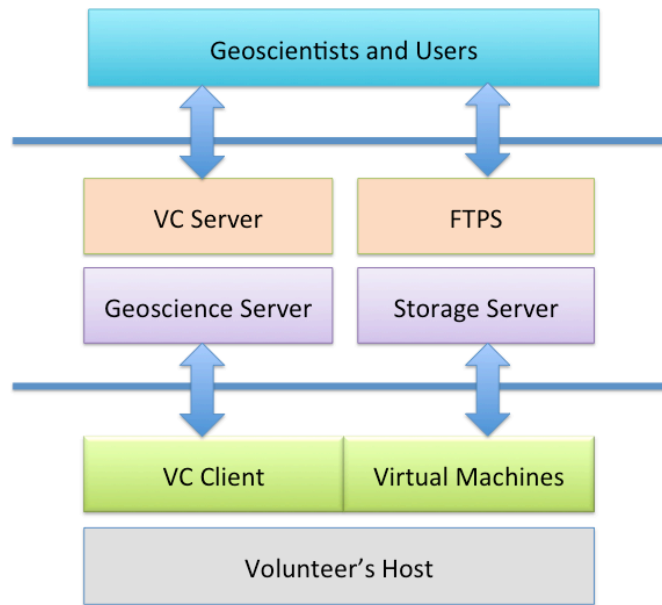


Figure 1 Volunteer computing infrastructure for climate simulation

The bottom illustrates volunteer tier. Herein, volunteer computing client middleware and virtualization software are installed on volunteer's machine. In Ciamte@Home project, BOINC is used as volunteer computing client middleware, and

VirtualBox is used as virtualization software. The BOINC client is responsible for communication with the BOINC server, science application and data downloading, and configuration of donated computational resources such as (e.g., how much CPU, RAM and hard disk to be donated). The climate model runs in the virtual machine launched by the virtualization software.

The middle tier is the server, containing volunteer computing server middleware and FTPS storage. The BOINC server middleware is installed in Climate@Home project and distributes Virtual Machine Images (i.e., geosciences models, applications and necessary environments), assigns climate simulation tasks, gets feedback from the volunteer's hosts, and grants credits to the volunteers who have already run the climate simulation. The FTPS stores monthly results from volunteers' virtual machines. A necessary mechanism to support SSL validation and provides secure way to receive the results.

The top tier is client, which present the project status for scientists. Meanwhile, A Graphic User Interface (GUI) is deployed to visualize the climate simulation results.

### **3.2 Workflow**

A typical workflow for volunteer computing and virtualization for Climate@Home (Figure 2) highlights several features of the research. First, the scientists prepare and forward model to the deployer. Second, the deployer configures the model, data with the operation system, required libraries and tools into a virtual image and updates the Climate@Home application on BOINC server. Third, the BOINC server will

assigns simulation tasks to different volunteers who run the climate simulation tasks. Fourth, the results are uploaded to FTPS server periodically and the running status is returned to the BOINC server. And finally, scientists download the results from FTPS server, and GUI presents the simulation tasks. In this workflow, scientists don't need to know how to do parallel programming to write their code and don't need to understand how to distribute their applications onto volunteer computing platform. The deployer is only responsible for creating the virtual images and deploying them onto the volunteer computing platform.

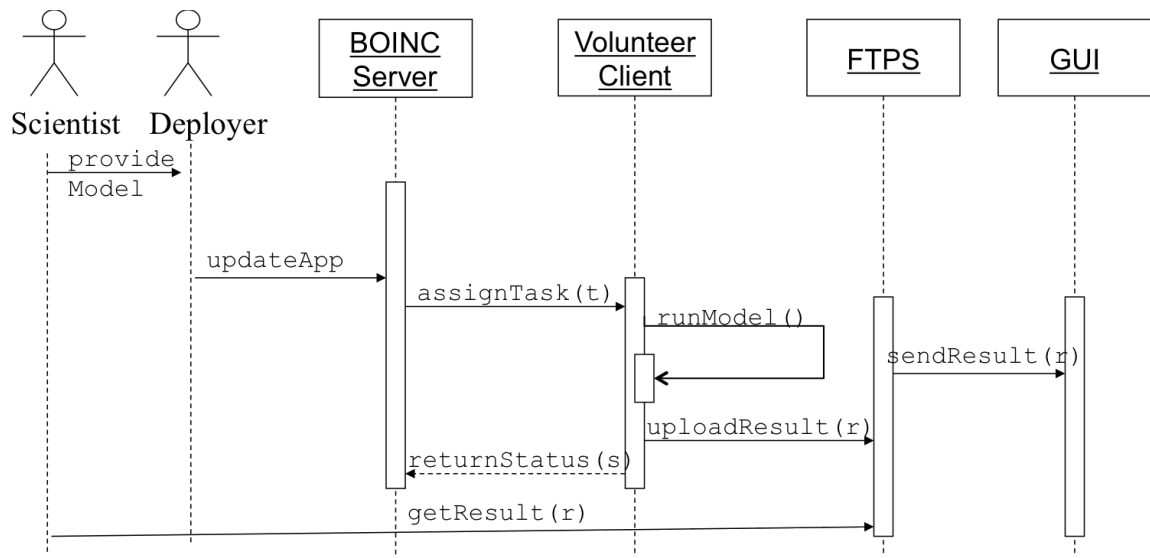


Figure 2 Volunteer computing workflow for climate simulation



## CHAPTER 4 METHODOLOGIES

### 4.1 Virtual Image Preparation

The Virtual Disk Image (VDI) file of VirtualBox is used in the Climate@Home project to store ModelE and data. In addition, the VDI file should also contain an operation system and required libraries and tools. Since ModelE is a Linux based model, Ubuntu 12.04 is used as the basic operation system in the project. After the Ubuntu 12.04 has been installed in the VDI, GCC 4.4, GFortran, OpenJDK 1.7 and NetCDF are installed before ModelE's installation. At last, ModelE is installed and model data are transferred to the VDI. In addition, a periodically upload application (see Section 4.2) is also installed in the VDI.

The VDI file is uploaded to BOINC server middleware, and Volunteers download it automatically to run ModelE after they added Climate@Home project in their BOINC client. To save download time for the virtual image and volunteers' network cost, the following steps minimize the size of the VDI file:

1. Compress the virtual image before sending to the volunteers. GZIP (Deutsch 1996) is used, which reduces the size of the VDI file by about 50%.
2. Shrink the virtual image. The VDI file uses "Dynamically Expanding Storage" option to allocate disk storage and save the ModelE outputs. The

option expands the VDI size on preparation of the original VDI by expanding the disk when install new applications (GCC 4.4, GFortran, OpenJDK 1.7) in the VM. However, the VDI size does not shrink or return to its previous size when the installation is finished. In this project, zerofree patch (Boutcher and Chandra, 2008) frees the expanded spaces. Then, clonehd (Dash 2013) creates a small copy of the VDI file. The small copy operates same as the original VDI file and is uploaded to the BOINC middleware.

3. In the installation of GCC, GFortran, and OpenJDK, it will create the cache of package files. By default, the Ubuntu keeps all the packages it has downloaded in case they are needed in the future. This makes the Virtual Image getting bigger. Since the virtual machine is only used to run ModelE and volunteers don't need to modify or upload it, the cache of packages files are cleaned to make the virtual machine image smaller.

Since the first version of Climate@Home released in November 2012, 16 versions have been released through April 23, 2014. The first version of VDI (Figure 3) was about 2.4 Gigabytes, and declined to 1.2 Gigabytes using GZIP to compress in version 2.0. Subsequently, the size dropped to about 850 Megabytes after shrinking in version 4.0. The current size is less than 700 Megabytes after all the cache of package files are cleaned.

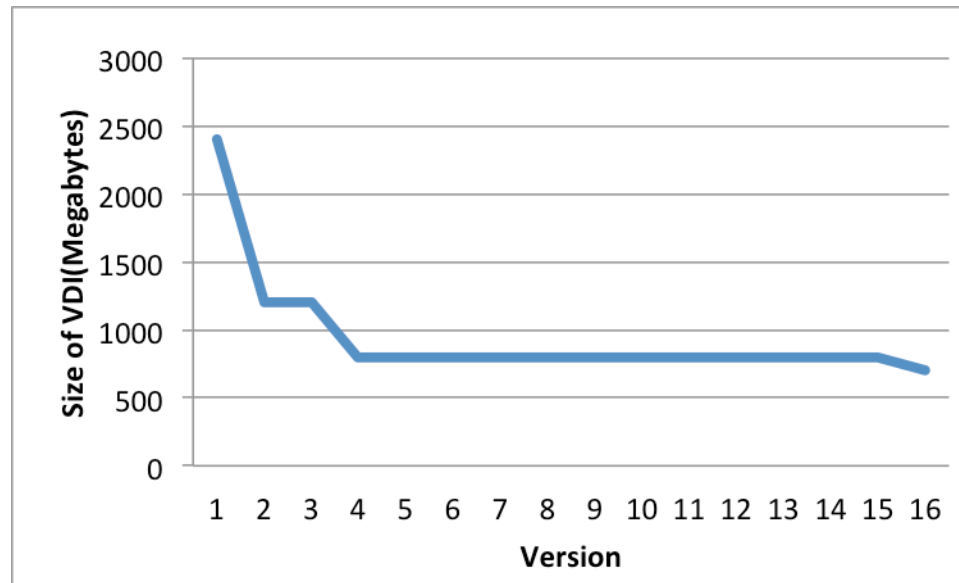


Figure 3 VDI file size history

#### 4.2 Periodically Upload Mechanism

In BOINC's traditional upload mechanism, the output from volunteers machine are uploaded after application finishes. This presents two problems for climate simulation tasks. First, the climate simulation task always creates large output files which are difficult to upload incrementally. Second, the output for climate simulation tasks is always organized in time series with uniform time intervals (e.g., hourly, daily, monthly, yearly). To recognize the specialty of the climate simulation outputs, a Periodically Upload Mechanism (PUM) has been used in the Climate@Home project (Figure 4).

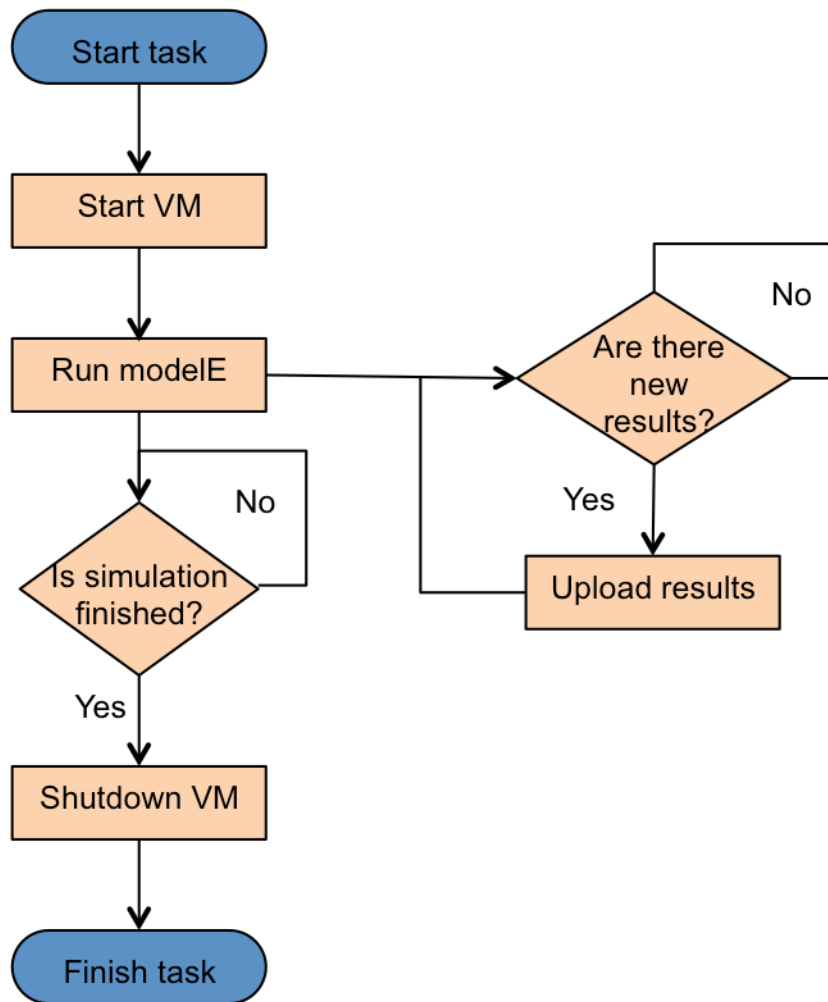


Figure 4 Periodically Upload Mechanism

Two checkers are added in the PUM. The first checker determines if a new result is created. The second checker determines if the task is completed. The PUM starts to run once the VM is launched. The two checkers run in set time intervals (25 minutes is used in Climate@Home project). If the results checker finds that a new new result is created,

the result is uploaded to FTPS server. If the task checker finds that the simulation task is finished, the VM is shutdown.

### **4.3 New Credit System (NCS)**

Credits are used to present how much work a computer, a user, or a team has contributed to the Volunteer Computing project. It is very critical for volunteer retention since increasing credit gives individual volunteers confirmation that their continuing contribution and credit provide a basis for competition among users and teams (Anderson and Koren 2004). Thus, it is one of the most important incentives for participation in the Volunteer Computing project. The BOINC provides two credit systems: 1) the first credit system which grants the credits based on the CPU runtime; 2) the second credit system which grants the credits based on the FLOPS actually performed by the application. They are awarded in small increments, according to the type and length of time to run a climate model.

However, these are not suitable for Climate Simulation projects since they only grant credits for completed tasks and climate projects are always take a long time to run. Thus, it is not acceptable for some users to wait a long time to accumulate the credits. A New Credit System (NCS) was created to address these problems and increase users incentives.

In the NCS, credits are granted in the two steps: check at intervals the FTPS server to get the number of monthly results and granting the host based on the hourly, daily, monthly or yearly results.

The NCS grants credits to those successful tasks using Equation 1,

$$C = n * CM \quad (1)$$

where  $C$  is the granted credits,  $n$  is the number of finished sub results (e.g., hourly, daily, monthly or yearly) and  $CM$  is granted credits for every month. The value for  $CM$  in the Climate@Home project is 30.

#### **4.4 Climate Simulation Visualization**

Climate simulation visualization is critical for both scientists and volunteers. However, interactive geovisual analytics over the Internet to facilitate collaborative climate research are immature (Sun et al. 2012). In order to analyze the ModelE output, a web-based climate visualization system in the Climate@Home project was developed. The final output of ModelE is NetCDF which is not convenient for rendering in web browsers. Thus, the NetCDF file is transformed into image files and then rendered in web browsers.

## **CHAPTER 5 EXPERIMENT & RESULT**

Using volunteer computing and virtualization technology, the first version of Climate@Home was launched on the website <http://climateathome.org/climateathome> on November 2012. And since then 16 versions have been released, the latest version is version 16.0. This thesis reflects the status as of April 23, 2014, including only activated hosts.

### **5.1 Hosts**

Since the BOINC does not provide the location information about the hosts. That information is derived from external IP of hosts from the BOINC database, and linking that to the web IP locator API to get the location of the host's location (Figure 5). From these data, the volunteers are from 40 different countries with most located in United States and Europe (Figure 5).

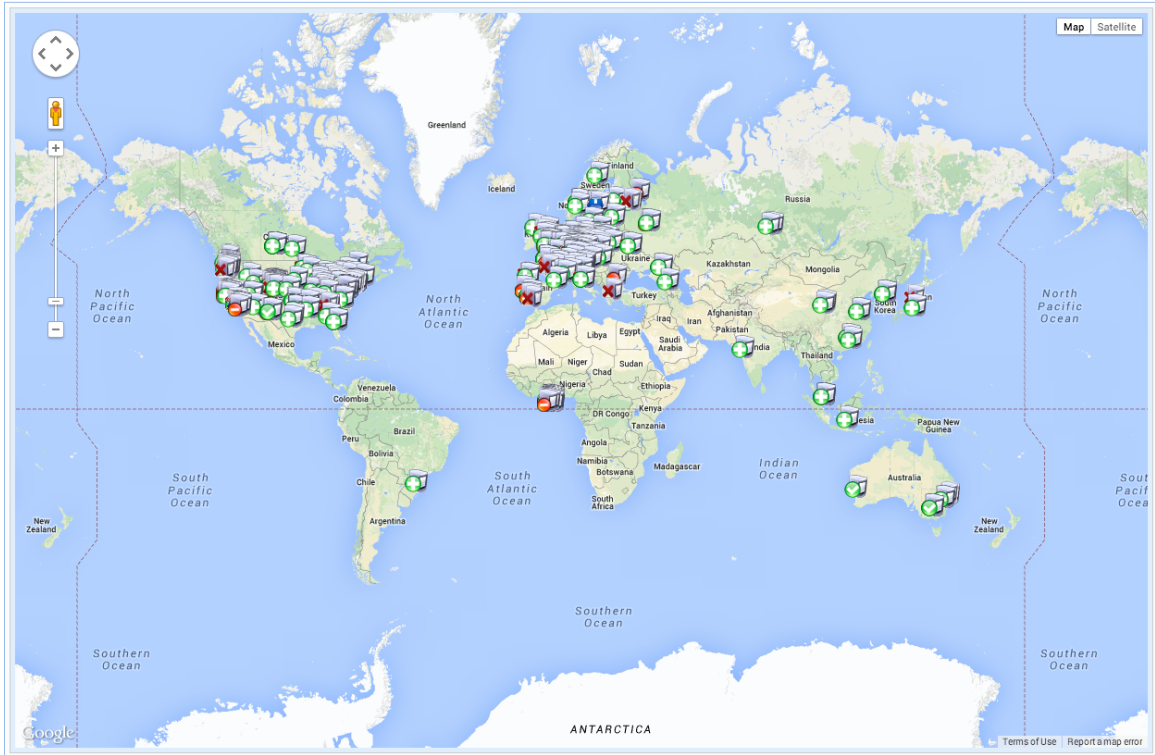


Figure 5 Global distribution of hosts

Of the activated hosts, windows machines account for 92%, Linux machines for 3.7% and Apple machines for 4.3%. Windows machines have a total of 3399 cores, which account for 92.1% FLOPS. Linux machines have a total of 218 cores, which account for 3.3% FLOPS. And Apple machines have 137 core, which account for 4.6% FLOPS (Table 2).



Table 2 Statistics of volunteer hosts

OS	Number	CPU Core	GFLOPS
Windows	544	3399	1543.5
Linux	22	218	56
Apple	25	137	76.4
Total	591	3754	1675.9

## 5.2 Running Status

Using NCS, the Climeate@Home grant the credits based on how many results the hosts uploaded and this is illustrated in Figure 6. In this example, all four tasks are in progress and each of them has created nine monthly results. Based on the NCS, 270 credits have been granted to each.

Task click for details Show names	Work unit click for details	Sent	Time reported or deadline explain	Status	Run time (sec)	CPU time (sec)	Credit
19319	8498	28 Apr 2014   2:50:25 UTC	8 May 2014   21:43:45 UTC	In progress	---	---	270.00
19286	8465	28 Apr 2014   2:50:26 UTC	8 May 2014   21:43:46 UTC	In progress	---	---	270.00
19215	8394	28 Apr 2014   2:50:26 UTC	8 May 2014   21:43:46 UTC	In progress	---	---	270.00
19214	8393	28 Apr 2014   2:50:26 UTC	8 May 2014   21:43:46 UTC	In progress	---	---	270.00

Figure 6 Example of NCS

Through April 23, 2014, the Climate@Home has created 1.2 Terabytes data, and the project has granted 3,938,718 credits to volunteers. Of the top ten countries, United

States account for 39.5% (Table 3), and the top ten countries account for almost 83% of the total credits.

Table 3 Top ten countries with credits in Climate@Home

United States	1580559
Germany	560007
Canada	542301
France	229927
Poland	115750
Norway	98363
China	90958
Czech Republic	76366
Australia	75157
Netherlands	70496
United Kingdom	67294

### 5.3 Visualization

Using the web-based visualization tool, clients can visualize the ModelE outputs. As example, Figure 7 shows the net thermal radiation change in two different Areas of Interest (AOIs) from the task E4M20a\_000029. Users can select different AOIs, variables or tasks for on-demand dynamic visual analysis.

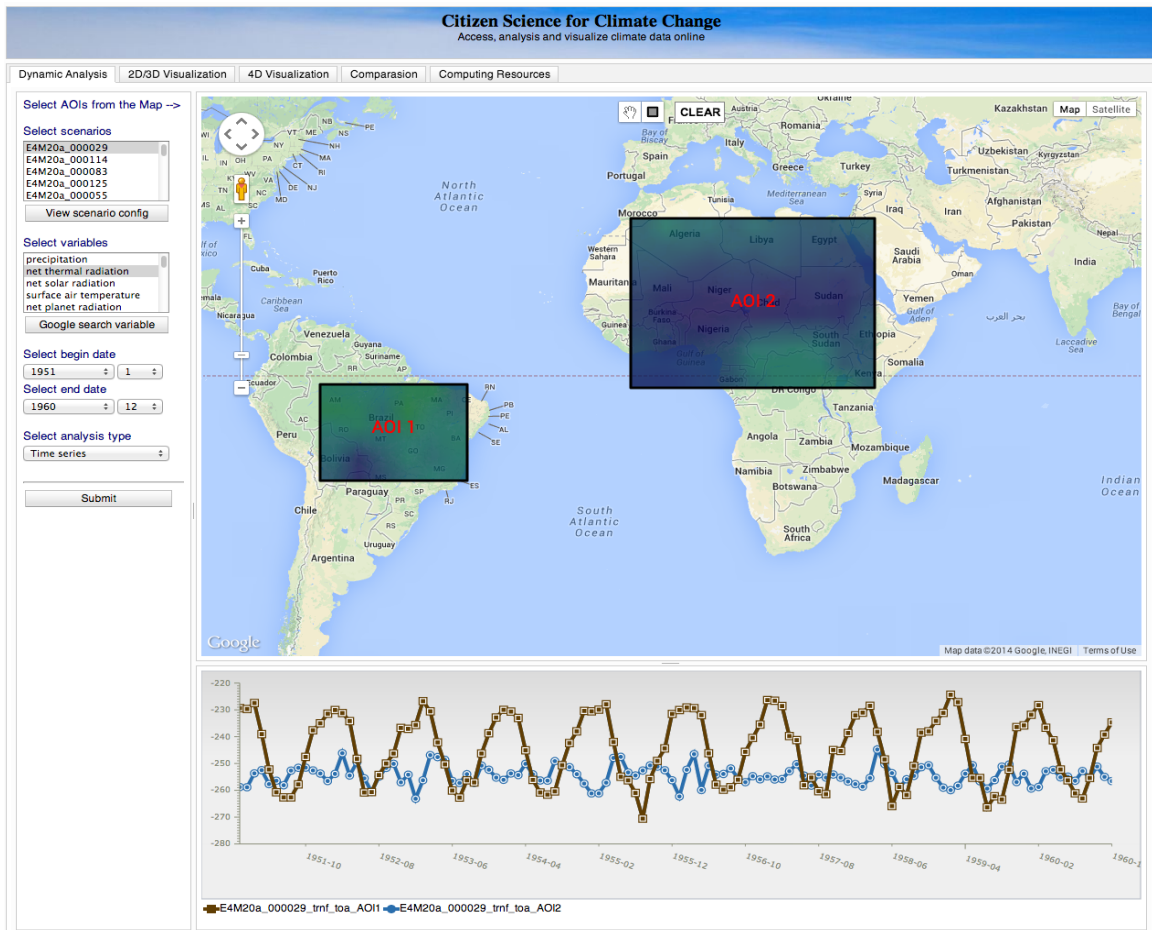


Figure 7 Dynamic analysis page of the Climate@Home

The 2D visualization is shown in four windows (Figure 8). Users select different tasks and variables to display data in different window, using the 2D visualization in Microsoft Bing Map.

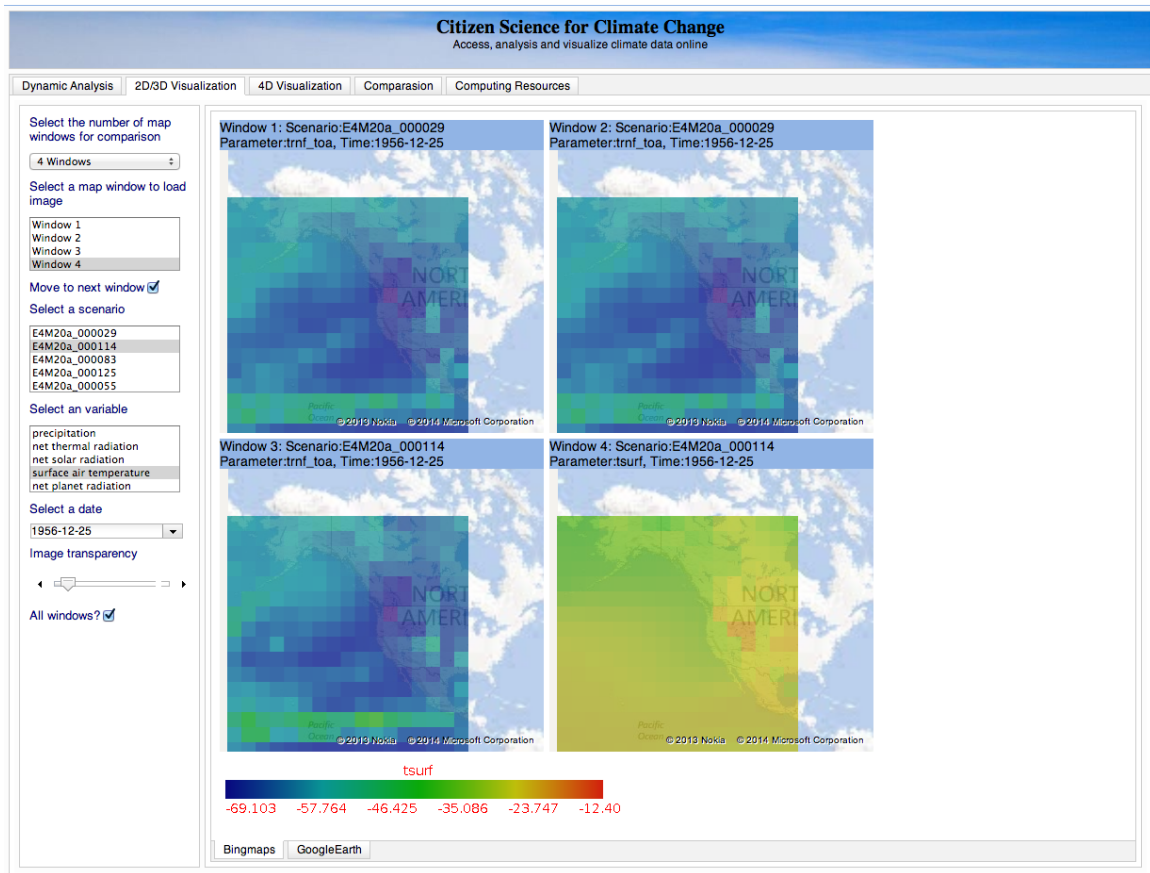


Figure 8 2D visualziation in four windows

## CHAPTER 6 CONCLUSION

Climate@Home is the first volunteer computing project using virtualization technology in climate domain. This thesis reports the research to integrate and optimize volunteer computing and virtualization technology for climate simulation. The most salient finding from this research are as follows:

- Compared to expensive supercomputer and cloud computing, the volunteer computing could provide enormous computational resources with essentially no cost. In addition, more volunteers and hosts are registered in the Climate@Home project, and consequentially computational resources are getting more powerful.
- Virtualization technology could improve the development of climate application in volunteer computing project. Generally, 32 bit virtual machines can run on both 32 bit and 64 bit machines. By packaging climate models, data, operation system, libraries and tools into a 32 bit virtual image, scientists only need to develop one version of their model rather than transferring their code to different platforms. In addition, deployers only need to update one virtual image if there is new version of model and then use the new virtual image to upload for different platforms. Comparing the traditional method which needs to update codes for different platforms, it saves extreme time for the new version release. In addition, the virtual image saves the running status to a snapshot and it enables climate models to run again from the breakpoint once they have crashed.

These advantages of virtual technologies bring better convenience to develop and test climate models.

- PUM can upload the climate model outputs periodically. On one hand, it saves time to upload outputs since it upload output since uploading occurs when models are running rather than uploading them after the model finishes. Conversely, it reduces the network stress by uploading small monthly outputs incrementally.

- NCS improves the volunteers' incentives to participate. By using the new credit system, it is fairer to grant credits based on number of completed monthly tasks and the faster machine will get more credits as they complete more tasks.

In spite of these advantages, this infrastructure has some shortcomings and is not suitable for all the climate applications. The principal shortcomings are as follows:

- There is no guarantee that all simulation tasks will be finished on time since volunteers may have a variety of site-specific issues such as hardware error, download error or abortion of the tasks. Although the virtualization technology enable to run climate models on virtual hardware, the models still can not continue once the physical hardware problem occurs. Most download errors occur because of the network problem. Due to the long running time of the climate simulation tasks, volunteers may abort some tasks to release computing resources for other projects. Although volunteer computing could provide a replication to assign each job to some different hosts to increase the reliability to run climate projects, it may waste volunteers' computing resources since only those unsuccessful tasks need to be assigned again.

- The infrastructure can not support real-time climate tasks. These tasks needs that the computers can get the real-time data rapidly and execute the climate tasks as fast as possible. However, the virtual image and data transfer in the volunteer computing relies on network and volunteer's machine. The network speed is far lower than transfer data in the supercomputer. In addition, the machine hardware differs widely between different hosts and it causes those hosts finish tasks in different time.

- The infrastructure can not support applications which need parallel processing, such as Message Passing Interface (MPI) applications. Parallel processing emphasizes the feasible exploitation of available concurrency in a computational process. By the decomposition, parallel processing partitions the large-scale computational problems to small sub-domains. Parallel processing requires to change data rapidly and very often between sub-domains.

More research and advancements in relevant fields could optimize the volunteer computing and virtualization for climate domain and these are listed below.

- A better task scheduler to improve the reliability of the climate simulation tasks from volunteers. The scheduler should be able to get the success rate of each host to run the simulation tasks and should use some optimization algorithm to assign the tasks to different hosts based on the success rate to improve the reliability. It will assign the unsuccessful tasks to different hosts rather than assign each task to different hosts.

- A hybrid computing environment using volunteer computing and cloud computing should be created to improve the current infrastructure. The climate simulation tasks always need to create large VDI file (700 Megebytes in Climate@Home

project). Volunteers' hosts are located in different places and they may have different network speeds to download the file from BOINC middleware. It is time-consuming for volunteers to download the VDI file, especially those who don't have fast connection speeds. Cloud computing companies provide different virtual machines in different zones, this advantage of cloud computing could be used to set up a volunteer computing server in each zone. Each server is responsible for only the hosts in that zone. Using this distributed server infrastructure, it can speed-up the download of the large VDI file.



## REFERENCES

- Anderson, D., et al., 2002. SETI@ home: an experiment in public-resource computing. *Communications of the ACM*, 45(11), 56-61.
- Anderson, D., 2004. Boinc: A system for public-resource computing and storage. *In: Grid Computing, 2004. Proceedings. Fifth IEEE/ACM International Workshop on* (pp. 4-10). IEEE.
- Anderson, D., and Fedak, G., 2006. The computational and storage potential of volunteer computing. *In: Cluster Computing and the Grid, 2006. CCGRID 06. Sixth IEEE International Symposium on* (Vol. 1, pp. 73-80). IEEE.
- Arakawa, A., & Lamb, V. R., 1977. Computational design of the basic dynamical processes of the UCLA general circulation model. *Methods in computational physics*, 17, 173-265.
- Barham, P., et al., 2003. Xen and the art of virtualization. *ACM SIGOPS Operating Systems Review*, 37(5), 164-177.
- Bicknell, B., et al., 1997. *Hydrological simulation program--Fortran: User's manual for version 11*. Athens, GA: US Environmental Protection Agency, National Exposure Research Laboratory.
- Boutcher, D., and Chandra, A., 2008. Practical techniques for purging deleted data using liveness information. *ACM SIGOPS Operating Systems Review*, 42(5), 85-94.
- Brooks, W., et al., 2005. Impact of the Columbia supercomputer on NASA science and engineering applications. *In Distributed Computing--IWDC 2005* (pp. 293-305). Springer Berlin Heidelberg.
- Caupp, C., Brock, J., and Runke, H., 1991. *Application of the dynamic stream simulation and assessment model (DSSAM III) to the Truckee River below Reno, Nevada: Model formulation and program description*. Raipd Creek Water Works.

Christensen, C., Aina T., and Stainforth, D., 2005. The challenge of volunteer computing with lengthy climate model simulations. *In e-Science and Grid Computing, 2005. First International Conference on* (pp. 8-pp). IEEE.

Cullen, M., 1993. The unified forecast/climate model. *Meteorological Magazine*, 122(1449), 81-94.

Dash, P., 2013. *Getting Started with Oracle VM VirtualBox*. Packt Publishing Ltd.

Del Genio, D., and Yao, S., 1993. Efficient cumulus parameterization for long-term climate studies: The GISS scheme. *The Representation of Cumulus Convection in Numerical Models, Meteor. Monogr*, 46, 181-184.

Deutsch, L. P., 1996. *GZIP file format specification version 4.3*.

Fedak, G., Germain, C., Neri, V., and Cappello, F., 2001. Xtremweb: A generic global computing system. *In Cluster Computing and the Grid, 2001. Proceedings. First IEEE/ACM International Symposium on* (pp. 582-587). IEEE.

Gates, W. L., 1992. AMIP: The atmospheric model intercomparison project. *Bulletin of the American Meteorological Society*, 73(12), 1962-1970.

Gordon, C., et al., 2000. The simulation of SST, sea ice extents and ocean heat transports in a version of the Hadley Centre coupled model without flux adjustments. *Climate Dynamics*, 16(2-3), 147-168.

Griffith, A., 2002. *GCC: the complete reference*. McGraw-Hill, Inc.

Hansen, J., et al., 1983. Efficient three-dimensional global models for climate studies: Models I and II. *Monthly Weather Review*, 111(4), 609-662.

Hansen, J., et al., 2007. Climate simulations for 1880–2003 with GISS modelE. *Climate Dynamics*, 29(7-8), 661-696.

Hayes, J. P., et al., 1986. Architecture of a Hypercube Supercomputer. *In ICPP* (pp. 653-660).

Havnø, K., et al., 1995. MIKE 11-a generalized river modelling package. *Computer models of watershed hydrology*, 733-782.

Huang, Q., et al., 2010, November. Cloud computing for geosciences: deployment of GEOSS clearinghouse on Amazon's EC2. *In Proceedings of the ACM SIGSPATIAL*

*International Workshop on High Performance and Distributed Geographic Information Systems* (pp. 35-38). ACM.

Koch, D., & Hansen, J. (2005). Distant origins of Arctic black carbon: a Goddard Institute for Space Studies ModelE experiment. *Journal of Geophysical Research: Atmospheres* (1984–2012), 110(D4).

Larson, S., Snow, C., and Shirts, M., 2002. Folding@ Home and Genome@ Home: Using distributed computing to tackle previously intractable problems in computational biology. *arXiv preprint arXiv:0901.0866*.

Liang, X., Lettenmaier, D. P., Wood, E. F., and Burges, S. J., 1994. A simple hydrologically based model of land surface water and energy fluxes for general circulation models. *Journal of Geophysical Research: Atmospheres* (1984–2012), 99(D7), 14415-14428.

Liu, J., and Abali, B., 2009. Virtualization polling engine (VPE): using dedicated CPU cores to accelerate I/O virtualization. In *Proceedings of the 23rd international conference on Supercomputing* (pp. 225-234). ACM.

Liu, J., et al., 2003. Sensitivity of sea ice to physical parameterizations in the GISS global climate model. *Journal of Geophysical Research: Oceans* (1978–2012), 108(C2).

Liu, K., et al., 2011. The GEOSS Clearinghouse high performance search engine. In *Geoinformatics, 2011 19th International Conference on* (pp. 1-4). IEEE.

Lombrana Gonzalez, D., et al., 2012. Virtual Machines & Volunteer Computing: Experience from LHC@ Home: Test4Theory project. In *Proceedings of The International Symposium on Grids and Clouds (ICGC 2012)*. 26 February-2 March. Taipei, Taiwan. Published online at <http://pos.sissa.it/cgi-bin/reader/conf.cgi?confid=153>, id. 36 (Vol. 1, p. 36).

Mell, P., and Grance, T., 2009. The NIST definition of cloud computing. *National Institute of Standards and Technology*, 53(6), 50.

Menemenlis, D., et al., 2005. NASA supercomputer improves prospects for ocean climate research. *Eos, Transactions American Geophysical Union*, 86(9), 89-96.

Metcalf, M., Reid, J. K., and Cohen, M., 2004. *Fortran 95/2003 Explained* (Vol. 416). Oxford: Oxford University Press.

Palmer, T., et al., 2005. Representing model uncertainty in weather and climate prediction. *Annu. Rev. Earth Planet. Sci.*, 33, 163-193.

- Rew, R., and Davis, G., 1990. NetCDF: an interface for scientific data access. *Computer Graphics and Applications*, 10(4), 76-82.
- Sellers, P. J., Mintz, Y. C. S. Y., Sud, Y. E. A., and Dalcher, A., 1986. A simple biosphere model (SiB) for use within general circulation models. *Journal of the Atmospheric Sciences*, 43(6), 505-531.
- Shindell, D. T., et al., 2006. Simulations of preindustrial, present-day, and 2100 conditions in the NASA GISS composition and climate model G-PUCCINI. *Atmospheric Chemistry and Physics*, 6(12), 4427-4459.
- Stainforth, D., et al., 2002. Climateprediction. net: Design Principles for Publicresource Modeling Research. In *IASTED PDCS* (pp. 32-38).
- Stainforth, D. A., et al., 2005. Uncertainty in predictions of the climate response to rising levels of greenhouse gases. *Nature*, 433(7024), 403-406.
- Sun, M., et al., 2012. A Web-Based Geovisual Analytical System for Climate Studies. *Future Internet*, 4(4), 1069-1085.
- Turban, E., King D., Lee, J., and Viehland, D., 2008. "Chapter 19: Building E-Commerce Applications and Infrastructure". *Electronic Commerce A Managerial Perspective*. Prentice-Hall. p. 27.
- Williams, K., Senior, C., and Mitchell J., 2001. Transient climate change in the Hadley Centre models: The role of physical processes. *Journal of Climate*, 14(12), 2659-2674.
- Yang, C., Goodchild, M., Huang, Q., Nebert, D., Raskin, R., Xu, Y., ... & Fay, D., 2011. Spatial cloud computing: how can the geospatial sciences use and help shape cloud computing?. *International Journal of Digital Earth*, 4(4), 305-329.

## **CURRICULUM VITAE**

Kai Liu is a graduated student in the Department of Geography and GeoInformation Sciences in the College of Science at George Mason University. Previously he was a visiting scholar at the Center of Intelligent Spatial Computing for Water/Energy Science, and worked for 4 years at Heilongjiang Bureau of Surveying and mapping in China. His formal education was acquired at Wuhan University, China, BA Geographic Information Science. His research focuses on Geospatial semantics, volunteer computing and spatial cloud computing.