OPTIMIZING ACCESS TO BIG EARTH OBSERVATION DATA WITH SPATIOTEMPORAL PATTERNS -- AN EXAMPLE WITH THE GEOSS CLEARINGHOUSE

by

Jizhe Xia
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

_______________________________ Dr. George Taylor, Committee Member

_______________________________ Dr. Ruixin Yang, Committee Member

_______________________________ Dr. Songqing Chen, Committee Member

_______________________________ Dr. Paul Houser, 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, Dean, College of Science

Date: _________________________ Spring Semester 2015
George Mason University Fairfax, VA

Optimizing Access to Big Earth Observation Data with Spatiotemporal Patterns -- An Example with the GEOSS Clearinghouse

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

By

Jizhe Xia
Master of Science
George Mason University, 2015

Director: Chaowei Yang, Professor
Geography and Geoinformation Science

Spring Semester 2015
George Mason University
Fairfax, VA

DEDICATION

This thesis is dedicated to my parents. For their endless love, support and encouragement.

ACKNOWLEDGEMENTS

As my dissertation supervisor, Dr. Chaowei Yang provided wise advice, insightful criticisms, and patient encouragement for me. He supported my Ph.D. studies with research funding and networking. He also mentors me to conduct the research and teaches me how to think critically and creatively, and form, organize, and prove my ideas. I would also like to thank my committee members, Dr. George Talyor, Dr. Ruixin Yang, Dr. Paul Houser and Dr. Songqing Chen, who were more than generous with their expertise and precious time in commenting, reading, encouraging, and being patience throughout the entire process. Finally, I would like to thank my dear friends in the U.S./ GMU (Qunying, Yunfeng, Zhipeng, Jame, Jing, Peter, Min, Winne, Kai, Zhenglong, Diwu, Xiaoyun, Minshao, Han, Manzhu, Haiyu, Weirong, Fron, Tony and many other dear friends not listed), who have supported me throughout my graduated study; thank 33 and sisi for understanding my stubborn nature and making me peaceful when I need it most; thank my dear mentors and friends in the 06GIS/SCNU (Dr. Li Weihong, Dr. Yaolong Zhao,Wyman, Yingbin, Junjie, Peishen, Xiangxin, Xiaohu, Xiaojun, Yingzhi, Huakeng, Chenhuai, Haodan, Xiaolong, Yarong, Meiai and many other dear friends not listed), for sharing the most wonderful 4 years with me ; thank my dear friends/guiders (Dr. Zhigang Li, Dr. Qianjun Miao, Dr. Qingyun Du, Xuekan, Lifang, Miaomiao, Zhangchi, Madiao, Pengjie, Xiao and others not listed), who have guided my career and share our happiness since my childhood.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

OPTIMIZING ACCESS TO BIG EARTH OBSERVATION DATA WITH SPATIOTEMPORAL PATTERNS -- AN EXAMPLE WITH THE GEOSS CLEARINGHOUSE

Jizhe Xia, PhD

George Mason University, 2015

Dissertation Director: Dr. Chaowei Yang

Big Data becomes increasingly important in almost all scientific domains, especially in geographical studies where millions to billions of sensors are collecting data of the Earth continuously. Recognizing the importance of managing the Big Earth observation Data, Group on Earth Observations selected the Global Earth Observation System of Systems Clearinghouse (CLH) to harvest, manage and share Earth observation metadata. Building a CLH to support global operation is very challenging, because it is essential for CLH to effectively manage and index Big Earth observation Data, provide accurate data service evaluation, and execute these services using fast provision computing resources to different space and time locations to support dynamic global user access. Although various optimization mechanisms (e.g., index, workload balancing, service model, cache) have been proposed, few approaches optimize the Earth observation data access with the spatiotemporal patterns of the data utilization. This dissertation investigates a variety of

spatiotemporal optimizations to better support Big Earth observation Data access using the CLH as an example. Specifically, the objectives are the following: (1) develop a new indexing mechanism to accelerate Big Data access. The new indexing mechanism integrates the spatiotemporal user access patterns into traditional index structures. The experiment result showed that the new index yields 9-20% performance gain for the data access compared to a classic R*-tree index; (2) develop a new service performance model to improve the service evaluation accuracy. The new service model collects globally distributed service information with cloud services and volunteers, and integrates the spatiotemporal service characteristics to provide evaluation end users at different space-time locations. The proposed spatiotemporal service model yields 3-18% accuracy improvements gains, thereby helping end users better choose service for data access; and (3) develop a cloud computing adoption framework to better support global user access and spiking access. The cloud framework automatically provisions and delivers computing resources for different data access tasks with spatiotemporal computing workloads, and globally deploys system instances to different regions. The experiment result showed that the cloud framework helps the CLH achieve about 10 seconds' performance gains for global and spiking user access. The significance of this research is that it provides a potential solution for optimizing access to Big Earth observation data using spatiotemporal data utilization patterns, thereby better supporting various Big Data related studies with faster data access.

CHAPTER 1 INTRODUCTION

1.1 GEOSS Clearinghouse (CLH)

The rapid advancement of Earth observations (EO) systems dramatically increased our ability in acquiring geospatial data. A variety of EO systems are monitoring the Earth to provide petabytes of EO data for scientists and decision makers on a daily basis. This explosively growing data can also be referred as the data intensity or Big Data (Whitehouse 2012; Manyika et al., 2011; Howe et al., 2008). To store, manage and share the Big EO Data, a variety of Spatial Data Infrastructure (SDI) (Maguire and Longley, 2005; Masser, 2005; Bernard et al., 2005) components have been developed by numerous agencies, such as NASA Global Change Master Directory (GCMD) (Greene et al., 1999, Miled et al., 2001), Geospatial One Stop (GOS) (Goodchild et al., 2007), Geospatial Platform (FGDC, 2013) and Ocean Data Portal (ODP) (Reed et al., 2010). In order to integrate and coordinate these widely distributed EO data and SDIs, the intergovernmental Group on Earth Observation (GEO) (GEO, 2013) proposed the framework of Global Earth Observation System of Systems (GEOSS) (GEOSS, 2013). This framework aims at promoting scientific connections between the EO systems and the need for effective geospatial information accessing in various domains such as Health, Energy, Climate and Agriculture (Pearlman et al., 2011). The GEOSS Common

Infrastructure is an implementation of GEOSS that allows global users to access, search and utilize geospatial resource through the GEOSS. The infrastructure consists of four main elements:

- The GEO Portal provides a web interface for end users to access GEOSS information and services.
- The GEOSS Components and Services Registry is a library catalogue for GEOSS members to provide essential details (e.g. name, topic) of their data contribution.
- The GEOSS Standards and Interoperability Registry enables the sharing and interoperating of information between GEOSS members through system configuration and standards.
- The GEOSS Clearinghouse (CLH), serving as the engine for GEOSS, drives the entire infrastructure by providing access to the Big EO metadata for other GEOSS components (Liu et al., 2011). The CLH also provide protocols/APIs (e.g. OGC Catalogues Service for the Web, CSW) (Nebert, Whiteside and Vretanos 2007) for third party Data Brokers (e.g. GEO Portal, EuroGEOSS discovery broker, Pearlman et al., 2013) to access the metadata hosted in the CLH database.

The CLH harvests metadata from data providers with various metadata standards such as FGDC CSDGM, ebRim Component, ebRim Service and WxS (WMS, WCS, WFS, WPS). All metadata are transformed into ISO: 19139 standard and stored in the PostgresSQL database (PostGIS 2013). By April 2th 2015, 105 catalogues and 125 thousand metadata records for higher granule resources had been registered into the CLH and shared among over 140 countries.

1.2 CLH Challenges

Since the operation from Dec 22th 2010, the CLH have been severing the GEOSS global operations for various GEOSS users. As a global operational system, it is critical for the CLH to (1) effectively manage and index Big EO Data, (2) provide accurate evaluations of EO data services, and (3) dynamically provision computing resource to support global users accesses. To achieve such objectives, CLH has been optimized with various mechanisms and technologies. For example, CLH duplications were deployed to balance the workloads (Bunt et al., 1999; Huang et al., 2011). Server side cache mechanisms can be adopted to reduce data retrieval time by pre-fetching frequently used data into memory (García Martín et al., 2013; Li et al., 2012; Kang, Kim and Kim 2001). A Lucene index was used to provide fast retrieval of text-based information (e.g. metadata title, keywords, abstract). A spatial index was utilized to improves the performance of Big EO Metadata retrieval by leveraging spatial relationships among features (Güting 1994). However, building an effective CLH is still very challenging due to the following issues:

- Big EO Data index and fast response

In the internet Era, end users expect to receive responses from web applications in 4-6 seconds, otherwise they will feel frustrated (Nah 2004, Yang et al., 2007). However, providing fast response to Big EO Data is very challenging in that: (1) data query is time consuming for big datasets (e.g. millions or even billions of records), especially for geospatial data that has complex spatiotemporal data structure and large volume. For each response to an end user, the CLH usually has to process large amount of data and complex algorithms; (2) the data size of the CLH is continuously increasing because significant amount of EO data/metadata are produced and registered into the CLH on a daily basis; (3) responding to a large number of concurrent access is time consuming. With the fast growth of the needs of EO data, popular EO data portal (e.g. GOS, CLH) may receive thousands of user accesses in a few seconds, which results in very slow response with limited computing resources. Therefore, a spatiotemporal indexing mechanism that can effectively index and manage Big EO Data is needed.

- Accurate Results

The service performance is a critical benchmark for EO data access, distributed geo-processing and decision making (e.g., public health and hazard emergency response). However, it is very challenge for the CLH to guarantee the service accuracy because the service performance is dynamically changing in space and time (Yang et al., 2011b). Exiting service performance evaluation systems are based on the sever location, which may deliver misleading service performance information for global users. Therefore, a spatiotemporal service model that can monitor and evaluate performance for globally distributed end users is needed.

- Spatiotemporal Dynamics

The CLH end users, EO data/service and computing workloads are widely distributed at different regions at different times. These spatiotemporal dynamics require the CLH to dynamically adapt computing resources in both space and time so that it could provide good user experience to global user. For example, the CLH need to fast provision computing resource in hours/minutes to handle spiking access (a sudden increase of concurrent access, Bodk et al., 2010). Dynamic computing resource provision in different regions is needed for the CLH to support global user access. However, dynamic computing resource provision is very inefficiency with traditional computing infrastructure. An on-demand, scalable, and globally distributed computing framework is needed.

1.3 Spatiotemporal Optimizations

Social and physical phenomena evolve in a four-dimensional world with specific spatiotemporal patterns. Studies have been conducted to investigate and utilize spatiotemporal patterns and principles for better decision making in different domains such as computing resource management, transportation and meteorology (Mountrakis and Gunson 2009; Huang et al., 2013; Sedda et al., 2011). Spatiotemporal principles and patterns govern the interactions between scientific parameters by providing connections and constraints for the evolvement of phenomena. And the principles are summarized as (Yang et al., 2011b): "

- Physical phenomena are continuous and digital representations (scientific parameters) are discrete for both space and time
  a. Closer things are more related than those farther away:
     1) Correlations exist among parameters, time, and space
     2) Neighboring discrete representation cells need communication across time

3) Duplication along domain borders is needed for analyzing/ simulating phenomena

b. Multiscalar

1) Phenomena are represented at global, regional, and local scales
2) Human thinking and digital representation of phenomena are hierarchical and fractal
3) Information frequency determines the hierarchical structure

- Physical phenomena are heterogeneous in space and time
  a. Higher resolution will include more information
  b. Phenomena evolve at different speeds (the faster a dynamic process, the quicker an exchange occurs among neighbors)
  c. The longer a dynamic process persists and the larger its spatial scale, the more exchanges are needed among neighbors ”

An integrative understanding and utilization of spatiotemporal patterns will help us address many 21st century challenges by better simulating geographic phenomena, managing EO data, and developing enabling applications. The spatiotemporal distribution of end users, computing infrastructure, EO data/services, and CLH workload result in a number of CLH challenges (Section 1.2). However, the spatiotemporal patterns within these phenomena provide a great potential for the optimization of CLH. For example, a better indexing mechanism can be designed by integrating user data demand patterns. However, the utilizations of spatiotemporal principles and patterns are still missing in the CLH indexing, computing resource arrangement, load balancing, deployment, and EO service evaluation mechanisms.

1.4 Cloud Computing

Cloud computing is an emerging computing paradigm for global users to utilize virtualized computing resources through the Internet. Cloud computing is often characterized by on-demand self-service, scalability, measured cost and availability (Mell and Grance 2009), which provides several features such as on-demand resources pooling, rapid elasticity, broad network, reliability and economy (Yang et al., 2011a; Buyya et al., 2009; Mell and Grance, 2011). It provides a “pay-as-you-go” service mode to release cloud users from spending labor and money to purchase, host, configure and manage traditional computing infrastructure, which is a long-held dream for distributed computing (Armbrust et al., 2010). For the past few years, increasing number of organizations, institutions and companies have been migrating the traditional computing facilities to embrace the cloud, such as Ubisoft, Netflix, Ebay.com, Adobe Systems and University of Washington (Amazon, 2013; Azure, 2013). On the other hand, an increasing number of computing facility magnates has been transferring their traditional infrastructure to the cloud paradigm. The cloud paradigm provides an effective and systematic manner for computing resources management and it is also proven to be cost and energy efficient (Lee and Chen, 2010; Marston et al., 2011). The Amazon Elastic Compute Cloud (EC2) and Windows Azure (Azure) are two of the most popular cloud services that deliver powerful cloud function such as elastic storage, cloud-based database management system (DBMS), workload balancing and Virtual Machines (VM). Commercial cloud services are soon followed by various open-source cloud solutions such as Eucalyptus (Eucalyptus, 2013), Cloudstack (Cloudstack, 2013), and Nebula (Nebula, 2013). With the advancement and maturity of cloud services, the cloud computing has becoming a powerful and affordable computing solution for scientific studies, especially for geoscience studies that requires massive, scalable and reliable computing resources. Various studies have been carried out to utilize cloud services for geoscience data sharing, data processing, modelling and other geoscience studies (Bürger et al., 2012; Lee and Liang, 2011; Evangelinos and Hill, 2008; Huang et al., 2013; Guan et al., 2013; Li et al., 2011).

Cloud computing provides great potential for us to utilize spatiotemporal patterns and address aforementioned CLH challenges (Section 1.2): (1) the on-demand computing resources provision enables the CLH to fast adapt to the change of workloads in space and time; (2) many cloud services provide globally distributed computing resources through data centers in different geo-locations. The accessibility of distributed computing resources is critical for the CLH to better address spatiotemporal issues by better locating CLH instances and providing more accurate results (e.g. location-specified service evaluation); (3) many cloud services are capable to provide powerful VMs to meet the computing resource requirement of CLH. For example, EC2 can provide a "high performance instance" with two quad-core processors, 23 Gbytes memory and 10 Gbps network connection.

1.5 Research Objective

This dissertation explores the feasibility of optimizing Big EO Data access in the CLH using spatiotemporal patterns of end user accesses, computing infrastructure usage, EO data/services, and system workload. Specifically, spatiotemporal patterns are captured in the following objectives: (1) to develop a spatiotemporal indexing mechanism to accelerate Big EO metadata access; (2) to develop a spatiotemporal service evaluation model to improve the EO data service evaluation accuracy; and (3)to develop a cloud computing-based adoption framework to better utilize on-demand and globally distributed cloud resource. Such a spatiotemporal optimized CLH could provide a potential solution for better facilitating the management, integration and access of Big EO Data.

CHAPTER 2 LITERATURE REVIEW

2.1 Addressing Big EO Data index issues

An index is an auxiliary structure specifically designed to speed up retrieval of records (Worboys and Duckham 2004). The index data structure consumes extra space in disk but can speed up data retrieval process by using an index field for ordering data and a pointer field to locate data from disk blocks. A spatial index significantly improves spatial Data/Metadata retrieval performance by leveraging spatial relationships among features (Güting 1994). As one of the most popular spatial indices, the R-Tree (Guttman 1984) extended the B-tree from one dimension to multi-dimension for spatial feature indexing. Spatial features and feature relationships are reorganized and stored in a tree structure so that these features can be fast retrieved by tracing the tree structure. An R-tree structure contains leaf nodes and non-leaf nodes. A leaf node is formed as (MBR, ID), where ID is the pointer to the data object and MBR is the minimum bounding rectangle of the data object. A non-leaf node is formed as (MBR, ptr), where ptr is the pointer to a lower level node (child node) of the tree and MBR is the MBR of the entire node.

The classic R-Tree data structure has to allow overlaps of MBR and multiple query paths in the Tree, which is a major issue for the query performance (Beckmann et al., 1990). Researchers have proposed several improvements to the R-Tree using different strategies.

For instance, Sellis et al. (1987) focused on local optimizations by avoiding the overlapping rectangles in non-leaf nodes. A corresponding data structure (the R+-tree) was proposed and tested with spatial datasets. Beckmann et al. (1990) considered both local and global optimizations of the index construction. They proposed a new local feature grouping algorithm to reduce the overlap and coverage of MBRs by introducing a penalty metrics ("dead zone", perimeter and overlaps). In terms of global optimizations, a forced reinsert algorithm was proposed to defer split processes and refine the overall structure of the Tree. Kamel and Faloutsos (1993) utilized the space-filling curves to better sort spatial features before the index construction. A better order of spatial feature can reduce overlaps of MBRs and also defer the splitting of the tree. Because of the performance advantage, the R-tree family is widely used in spatial DBMSs such as Oracle Spatial, PostgresSQL, Informix, and others (Informix 2003; Kothuri, Hanckel and Yalamanchi 2008; Kothuri, Ravada and Abugov 2002; PostGIS 2013).

However, the spatiotemporal patterns of user access are still missing in the current spatial index optimizations. Ignoring spatiotemporal query patterns and spatiotemporal distribution of end users, GEOSS systems and computing infrastructure could result in inefficient operational systems and poorly structured indices (Achakeev, Seeger and Widmayer 2012).

2.2 Addressing Accurate Result Issues

With the rapid growth of web services and available data, Service Oriented Architecture (SOA) has become a major framework for web-based applications by integrating distributed application components through standard interface (Erl 2005). As the abundance of service providers, the service consumers (e.g. developers, general end users) are facing the issue of choosing the "right" service. The service performance is an essential benchmark for service consumers to differentiate web services because the level of service performance has great impact on the service usability, utility and popularity of the service (Mani and Nagarajan 2005). For example, Zeng et al (2003) investigated an efficient mechanism to utilize multiple web services for constituting a single task based on the composition of service performance. Therefore, it is critical to provide accurate evaluation of service performance. Mani and Nagarajan (2005) summarized seven major factors of service quality including Availability, Accessibility, Integrity, Performance, Reliability, Regulatory, and Security. Curbera et al., (2005) and Roman et al., (2005) also introduced the concept of Interoperability and Scalability to service measurement to indicate of ability of a service to communicate and interoperate, and the ability of a service to consistently respond to service consumers despite the volume of concurrent requests. These measurements methods provide best practices for understanding the attributes of service. However, no concrete methodologies were followed to implement the evaluation of service performance.

Li et al., (2011) proposed a performance optimized framework to search OGC web services through CSW catalogues. A performance evaluation module was embedded in the system for end users to choose the “right” web services. However, this service module can only evaluate OGC web services (e.g. WMS, WFS and WCS) that are already in the system database, and it cannot interoperate with other systems. Federal Geographic Data Committee (FGDC) Service Status Checker (SSC) (FGDC Checker, 2013) provides a web service to validate, test and score geospatial web services. The FGDC SSC supports a wide range of geospatial services such as ArcGIS Map Server, ArcGIS Feature Server, SOS, Z39.50, Web Accessible Folder (WAF) and OGC web services. It can provide a set of summary and test diagnostic information for other systems through standard web services. Gui et al., (2013) proposed a service quality enhanced search engine to assists user decision-making by integrating a service evaluation system, a service visualization tools, and multiple data viewers. It supports service performance evaluation in different granularities (overall quality grade from long-time statistics and dataset levels performance in a certain period of time). This service performance evaluation module also considered service performance grading information from other systems such as FGDC SSC, MapMatters, and QWS Dataset. Moorsel (2001) extended service evaluation to the concepts of quality of experience (QoE) and quality of business (QoB) to address the increasingly importance of quantifying the user experience. User feedback module was added to a regular service evaluation system. These service evaluation systems provided rich information for end users to better utilize EO data services.

However, few research consider the dynamic of service performance across space and time. Yang et al., (2011b) evaluated a number of WMSs for users in different regions of the United States. Evaluation results showed that an EO service tends to have better performance when the service is close (network distance) to the location of the end user. Misleading service information could be delivered to users in different regions if services are monitored from a single location. Therefore, a spatiotemporal service performance evaluation mechanism is needed to provide accurate results to end users.

2.3 Addressing Spatiotemporal Dynamics

2.3.1 Workload balancing

The fast increase of traffic on the World Wide Web (WWW) causes large amount client requests to popular applications (e.g. a popular web site). Multiple computers have been utilized to support large amount of workload by partitioning and scheduling the entire task onto different processors/computers. Load balancing could be either static or dynamic. Traditional static workload balancing mechanisms usually predefined the workload to each processor/computer based on statistical partitioning that could approximately equal the workloads. For example, Li et al., (2009) proposed a line-scanning-based load balancing mechanism for large-scale urban simulation. The balancing algorithm pre-defined the workload scheduling rules based on equal simulation area or equal workload, so that waiting time between different computers during the parallel simulation can be reduced. However, some computing tasks are very dynamic and hard to be pre-partitioned before the load balancing process. To overcome this issue, various dynamic load balancing mechanisms have been proposed to distribute workloads during the task execution by monitoring processors/computers. Cardellini et al., (1999) reviewed various dynamic load balancing mechanisms for websites such as client-side proxies, Round Robin DNS, Packet Rewriting and HTTP Redirection. Zhang et al., (1997) compared static and dynamic workload balancing by using a simulated task in a heterogeneous distributed system. The results showed that dynamic load balancing had the performance advantage but static load balancing is more stable when communication cost is not negligibly small. The reason is because dynamic load balancing has to involve more complex algorithms and monitoring overheads. Load balancing mechanisms usually required a computer (master node) outside of the cluster to be run, so that it can independently monitor and to respond dynamically to achieve load balancing (Pacheco, 1997). For example, Jackson and Humphre (1995) developed a workload balancing extension for Parallel Virtual Machine (PVM) systems. A master node was required for gathering CPU load information and to provide a programming interface to system administrator. To address this limitation, Zeng and McMechan (2002) proposed a workload balancing mechanism requires no master nodes, nor priori knowledge for workload monitoring. Their mechanism was successfully applied to a ground-penetrating radar data processing in a heterogeneous cluster. Aforementioned load balancing mechanisms seldom utilize the spatial or temporal relationship between workload. Huang et al., (2011) proposed a load balancing mechanism to reduce the high network transmission load for Web-GISs. Spatial analysis was conducted during a spatial process (e.g. spatial query). However, this mechanism only balanced the workload between the server side and browser side. A spatiotemporal load balancing mechanism for scheduling workload to different instances is still missing.

2.3.2 Cloud computing adoptions

Addressing spatiotemporal dynamic issues (e.g. global user access, spiking access) with traditional computing paradigm is usually ineffective and inefficient. To adapt computing resources to the requirements in different regions at different times, system administrators usually have to spend long time and make great efforts to purchase the hardware, install and configure networking, OS and related software packages. Also, computing resource usually cannot be provisioned in time for spiking access (Lee and Chen, 2010). Compared to traditional computing paradigm with relatively static infrastructure, Cloud Computing has the characteristics of on-demand resource pooling, rapid elasticity, reliability and others. Cloud services not only support end users to elastically utilize computing resources based on temporal distribution of computing requirement, but also provide a number of cloud regions across the world so that computing resources can be provisioned at multiple geo-locations for global users. Because of the advantage of cloud service, various studies have been conducted to test the feasibility of cloud service in supporting scientific studies. Kim et al., (2009) discussed several issues that impede the adoption of cloud computing in scientific studies such as the risk of outage, security and the performance overheads. Ostermann et al., (2010) tested the performance of cloud services by analyzing the computation, I/O, memory hierarchy and reliability of five VM types provided by EC2. Iosup et al., (2011) utilized several benchmarks such as LMbench (Barham et al., 2003), Bonnie (Kowalski, 2013) and HPCC (Luszczek et al., 2006) to evaluate the performance of four commercial cloud services: EC2, GoGrid, ElasticHosts and Mosso. These cloud evaluation results show that the cloud provides great potentials for scientific research and application.

In geosciences, cloud services have been used to support scientific processing and analysis functions. For example, Bürger et al., (2012) reported a cloud-based hydrologic simulation platform by utilizing hydrology and visualization packages (e.g. VisIt, 2013) pre-installed in cloud services. Also, cloud service have been used to manage large-scale spatial data. For example, Lee and Liang (2011) proposed a cloud-based geo-location data service system: Geopot, for various location-based mobile applications. The Amazon Simple Storage Service (Garfinkel 2007) was used for big spatial data management. In addition, cloud services was used by various geo-science models. Evangelinos and Hill (2008) adopted EC2 to enable the computability of a coupled atmosphere-ocean climate model. Relatively cost-saving instances can be utilized to execute a climate model while the performance is comparable to low-cost cluster systems. Huang et al., (2013) utilized cloud computing to support the execution of dust model by leveraging a large group of computing resources in a few minutes to accelerate large-domain dust storm forecasting. Cloud services also enable high performance data processing for computing and data intensive applications. Guan et al., (2013) integrated the cloud service to a data processing framework, which provided performance gain for intensive data processing.

Li et al (2011) proposed a cloud-based framework for intensive floating car data (FCD) processing. EC2 and cloud techniques (e.g., Bigtable, Chang et al., 2008, and MapReduce, Dean and Ghemawat, 2008) are used to address the data intensity problem inside the FCD processing workflow.

These studies provide practices for utilizing cloud services to support geoscience studies. However, few research explore and analyze the feasibility of utilizing cloud service to support Big EO Data access. In addition, many major benefits of cloud service (e.g. elasticity, load balancing and distribution) have not been integrated or implemented in major SDIs (e.g., the CLH) for EO Data access.

2.4 Limitation of exiting research

The EO data are produced at different spatiotemporal scales and the spatiotemporal patterns of the phenomena have been utilized to better support geospatial science studies. For example, Mountrakis and Gunson (2009) analyzed moose-vehicles collision hotspots in space and time, which assists planners to better design transportation facilities. Huang et al., (2013) adopted spatiotemporal principles among computing resources and dust storm phenomenon to improve the computability of dust storm forecasting by about 20% performance improvement. Sedda et al., (2011) discovered that cork oak height was significantly correlated to the spatiotemporal distribution and intensity of wind. This spatiotemporal pattern helps better manage cork oak in the young phase to increase cork production. The LOST-Tree (Huang and Liang 2012) was a spatiotemporal tree structure considering spatiotemporally distributed sensor networks so that redundant transmission of data pre-fetching can be minimized.

Section 2.1, 2.2 and 2.3 reviewed a number of mechanisms that have been or could be adopted for CLH to address Big EO Data access challenges (Section 1.2). However, the spatiotemporal principle, as a key for understanding physical and social phenomena, is still missing. The spatiotemporal distribution of end users, computing infrastructure, EO data/services, workload and their patterns provide a great potential for us to better index and manage data, arrange computing resource, balance user workload, deploy CLH replications and evaluate EO services.

Therefore, this dissertation explore the feasibility of addressing Big EO Data access challenges by utilizing spatiotemporal patterns among end users, computing infrastructure, EO data/services, and workload. Specially, the spatiotemporal optimization approach includes:

- **A spatiotemporal index mechanism** that provides better data query performance by leveraging the (1) the spatiotemporal distribution of users and (2) the pattern of user queries.
- **A spatiotemporal service model** that provides more accurate service information to global users by utilizing the volunteer computing and cloud computing for multiple-location service monitoring.

- **A spatiotemporal cloud adoption framework** that utilize cloud services to (1) automatically adapt computing resource to dynamic workloads, (2) support large number of concurrent access by balancing workload to proper CLH instances, and (3) deploy CLH instance to different cloud regions to better support global users.

CHAPTER 3 METHODOLOGIES

3.1 Spatiotemporal indexing mechanism

This section introduces a spatiotemporal index mechanism that better supports the index and access of Big EO Data/Metadata. This mechanism includes two major components: (1) the Predefined Multiple Indices Mechanism (PMIM) that builds and utilizes different indices using Big Data user access patterns (2) Access Possibility R-Tree (APR-tree) that modifies a traditional R-Tree with feature access patterns for the PMIM operation.

3.1.1 User behavior pattern mining

Generally, user behavior are affected by spatiotemporal principles that govern the relationships of phenomena (Yang et al., 2011b), and user behaviors are expected to have the following patterns:

(1) Local interests: Users from neighbouring regions tend to query similar data. Users from a specific region tend to request data related to this region (Tobler 1970; Zheng, Xie and Ma 2010).

(2) Temporal interests: Active users in a specific time window tend to query similar data (Horvitz et al., 1998).

(3) Backgrounds interests: Users with similar backgrounds (e.g. occupation, hobbies, age and gender) tend to have similar query behaviors (Horvitz et al., 1998).

(4) Combined patterns: A comprehensive pattern that includes spatial, temporal and other factors of user behaviors, data, and applications.

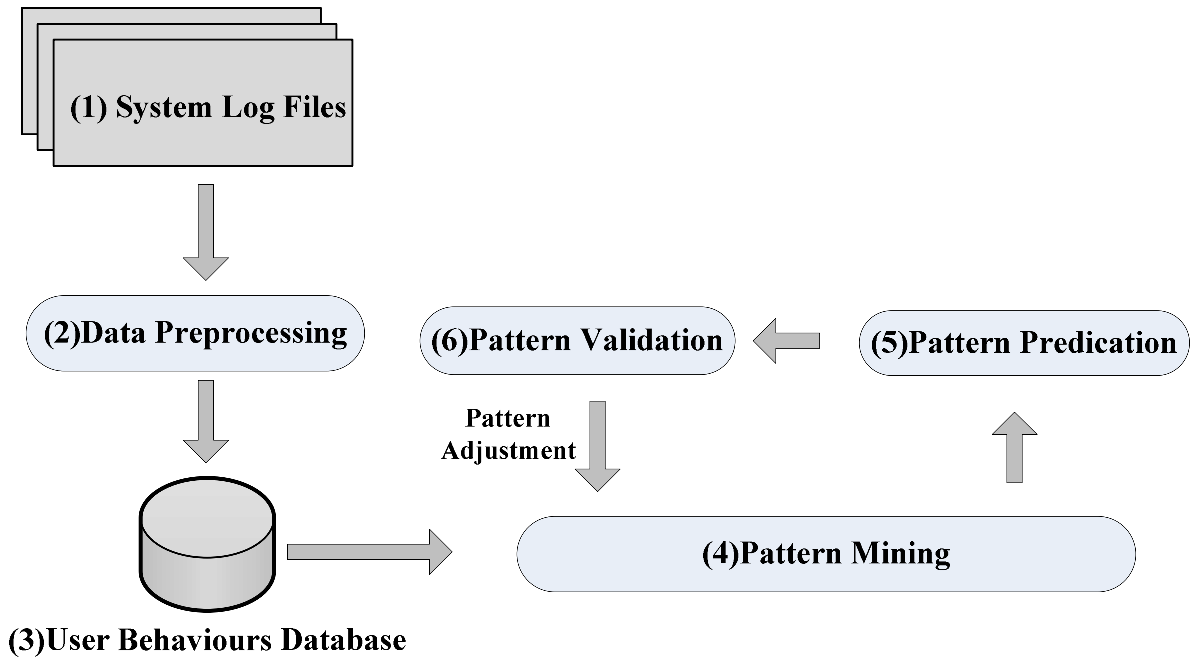


Figure 1 Automatic spatiotemporal patterns detection workflow

Figure 1In order to detect user patterns, a workflow for automatic spatiotemporal pattern detection is designed (Figure 1), which include the following steps:

(1) Data collection (System log file collection)

The CLH generates log files to record system operation information. These log files are stored as fat files and generated by CLH on a daily basis (

Figure 2). This step downloads and collects CLH log files from the CLH server (operation from Dec. 22th 2010 to Dec. 31st 2011) and Amazon EC2 Cloud server (operation from Jan. 1st 2012 to the present).

geonetwork.log.2011-05-28	5/29/2011 12:53 A...	2011-05-28 File	44,335 KB
geonetwork.log.2011-05-29	5/30/2011 12:54 A...	2011-05-29 File	214,536 KB
geonetwork.log.2011-05-30	6/1/2011 12:49 AM	2011-05-30 File	1,549 KB
geonetwork.log.2011-05-31	6/2/2011 12:38 AM	2011-05-31 File	6,734 KB

Figure 2 Log files of the GEOSS Clearinghouse are stored as flat files

(2) Data preprocessing

Data preprocessing extracts user behavior information from unstructured log files (Figure 3). The log files from Step 1 are imported to the data preprocessor to extracted the following user behavior information:

a) Access IP
b) Access Time
c) Access Channels (e.g. web portal and CSW (Nebert, Whiteside and Vretanos 2007))
d) Session ID
e) Operation Type (e.g. search, metadata access and others)
f) Search parameters
   f1) text parameters
   f2) spatial parameters

Abnormal accesses caused by users "robots" are excluded at pre-processing stage. For example, we eliminated a user "robot" from Spain who intensively accesses the entire metadata of CLH during Oct 2nd 2011 (about 150 thousand accesses).

```
2010-12-22 12:01:46,591 INFO  [jeeves.service] - Dispatching : xml.info
2010-12-22 12:01:46,592 INFO  [jeeves.request] - ==========================================================
2010-12-22 12:01:46,592 DEBUG [jeeves.service] -  -> parameters are :
<request>
  <type>categories</type>
</request>
2010-12-22 12:01:46,592 INFO  [jeeves.request] - HTML Request (from 127.0.0.1) : /geonetwork/srv/en/xml.harvesting.info
2010-12-22 12:01:46,592 DEBUG [jeeves.request] - Method       : POST
2010-12-22 12:01:46,592 DEBUG [jeeves.request] - Content type : application/xml; charset=UTF-8
2010-12-22 12:01:46,592 DEBUG [jeeves.request] - Accept       : text/javascript, text/html, application/xml, text/xml, */*
2010-12-22 12:01:46,593 DEBUG [jeeves.request] - Session id is 1hbsj3lytoaya
2010-12-22 12:01:46,593 INFO  [jeeves.request] - ==========================================================
2010-12-22 12:01:46,594 INFO  [jeeves.request] - HTML Request (from 127.0.0.1) : /geonetwork/srv/en/xml.harvesting.info
2010-12-22 12:01:46,594 DEBUG [jeeves.request] - Method       : POST
2010-12-22 12:01:46,594 DEBUG [jeeves.request] - Content type : application/xml; charset=UTF-8
2010-12-22 12:01:46,594 INFO  [jeeves.service] - Dispatching : xml.info
2010-12-22 12:01:46,594 DEBUG [jeeves.request] - Accept       : text/javascript, text/html, application/xml, text/xml, */*
2010-12-22 12:01:46,594 DEBUG [jeeves.service] -  -> parameters are :
<request>
  <type>categories</type>
</request>
```

Figure 3 User information in CLH log file is unstructured

(3) User behaviors data in store

This step imports user behaviors information into a DBMS for better data management and retrieval. The design of the database table is shown in Table 1.

Table 1 Table design for user behaviors information management

Field Name	Data Type	Description
Id	INT	Primary key
Ip	VARCHAR(50)	
Lon	FLOAT	Longitude of IP
Lat	FLOAT	Latitude of IP
City	VARCHAR(100)	City of IP
Country	VARCHAR(100)	Country of IP
Time	VARCHAR(50)	The time of access in a day
Date	VARCHAR(50)	The date of access
sessionID	VARCHAR(80)	Access session
accessType	VARCHAR(100)	Operation type
textWords	text	Text parameter of a query
Ifspatial	boolean	If or not a spatial query

spatialType	VARCHAR(80)	Spatial query type (e.g. intersect, within)
West	FLOAT	Query bounding box
East	FLOAT	Query bounding box
North	FLOAT	Query bounding box
South	FLOAT	Query bounding box

(4) Pattern Mining

Pattern mining analyzes user patterns based on user behaviors database (Step 3). Various data mining technologies are applied to discover spatiotemporal user pattern:

- **Clustering**

The clustering is the process of partitioning data into subsets/cluster so that data belonging to a cluster is more similar to data belonging to the same cluster than to data belonging to other clusters. It could be utilized to identify user groups, intensive access regions and intensive access hours. Simple K-mean (Kanungo and Tapas 2002), Hierarchical Clustering and DBSCAN (Ester and Martin 1996) are utilized for the user clustering.

- **Association Rules**

Association Rules learning (Agrawal et al., 1993) help discover the relations between variables by finding association rules that can predict the occurrence of an instance based on the occurrences of other instances. It could be applied to investigate associations of end users' data demand. For example, when a user is searching for Africa data, he or she may also want data in South America.

- **Classification**

Classification is the process of labeling data items according to pre-determined set of classes. It could help the CLH to predict user access intensity based on historical patterns. Decision tree classification methods (e.g. J48) could be used for the classification process.

- **Geographically Weighted Regression (GWR)**

Regression has been widely used to model the relationship between dependent variable and a set of independent variables. An assumption underlying the basic regression model is that the observations should be independent of one another. However, this is not always true for geo-science applications because Tobler's observation that "Everything is related to everything else, but near things are more related than distant things." (Tobler, 1970). The GWR (Brunsdon et al., 2002) considers the phenomena relationships in space. In GWR, the value of the dependent variable in one spatial unit is affected by the independent variables in nearby units. The GWR could be applied to help the CLH discover the relationship between access frequency and user distribution.

(5) Pattern predication

Based on available spatiotemporal patterns, predict user data demands and access workloads from different regions (e.g. from Asia, Europe or North America) at different times (next minutes, hour, or day).

(6) Pattern validation

Step 6 validates the predication by comparing the projections with the observation data.

(7) Pattern adjustment

With more user historical information analyzed and validated, we could adjust the parameters and methods in Step 4 for more accurate user behaviors prediction.

3.1.2 *Predefined Multiple Indices Mechanism (PMIM)*

The distribution and configuration of SDIs (e.g., CLH) should consider the distributions of data services, computing resources, application features, end users and their diversified preferences and requirements (Yang et al., 2011a). End users from different regions at different times have different demands for geospatial data thereby performing different queries. For example, European users may request European data more frequently than other data. Therefore, simply replicating and distributing a single spatial index cannot efficiently respond to heterogeneous queries.

To address this issue, I propose the PMIM.

Figure 4 illustrates the difference between the PMIM and the traditional indexing mechanism. Traditional indexing mechanism uses a single index to support all user queries (

Figure 4a). Users access different index replications with the same index structure. The optimization of this single index is limited because this index can't efficiently support all end users.

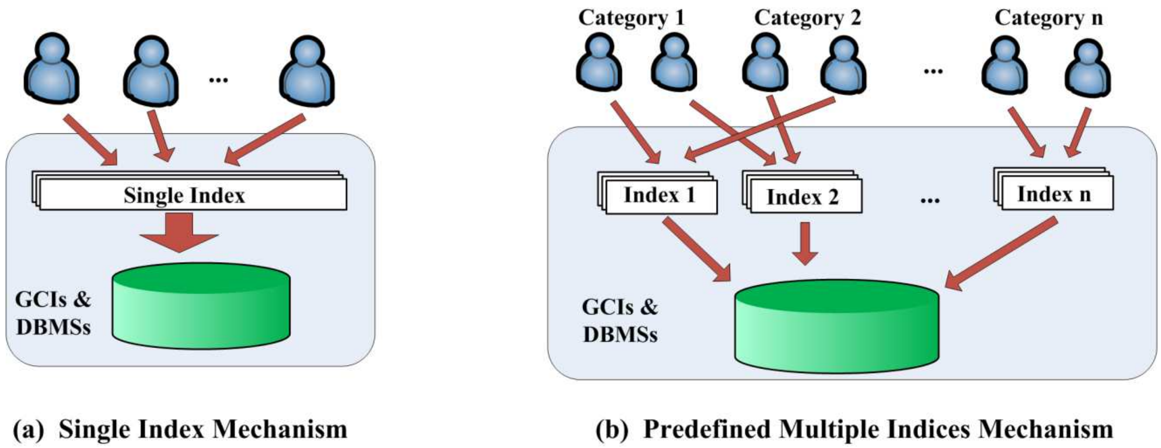


Figure 4 The PMIM utilizes multiple-indices to support global users while only one index is used in most traditional indexing mechanisms

In the PMIM (

Figure 4b), multiple-indices are leveraged to support different categories of users. End users are categorized into various categories according to user distributions and query behaviors. Users belonging to the same category have more similar query behavior than that of users belonging to other categories. Each user category corresponds to a specially optimized index. For example, most users in "Category 1" are from Western Europe and they intensively query Europe data in local morning hours. The "Index 1" is optimized based on this query pattern to better support "Category 1" users. To support various user categories, multiple indices are predefined and managed in the indexing store. The entire PMIM improves index performance by considering heterogeneous user queries over space and time.

3.1.3 Access Possibility R-Tree (APR-tree)

By adopting PMIM, each index only needs to handle user queries with certain pattern instead of all heterogeneous queries. Therefore, a new indexing rule needs to be defined. Figure 5 illustrates this modified indexing rule based on query patterns. In traditional index mechanism (

Figure 5a), Feature 1 (F1), F2 and F3 are indexed into Node 1 (N1) while F4, F5 and F6 and indexed into Node (N2) because of the spatial closeness. This tree structured index reduces disk access by tracing features along the tree structure instead of traversing all features. In the second scenario (

Figure 5b), a user query pattern is displayed by showing the access possibilities of features. F4 (46% access possibility) and F5 (51% access possibility) are indexed into N1 while F1, F2, F3 and F6 are indexed into N2. In the modified indexing structure, features with high access possibility are separated from features with low access features so that high access features/nodes can be effectively pre-fetched into computer cache. An effective managed cache significantly reduces the disk access and data retrieval time (García Martín et al. 2013).

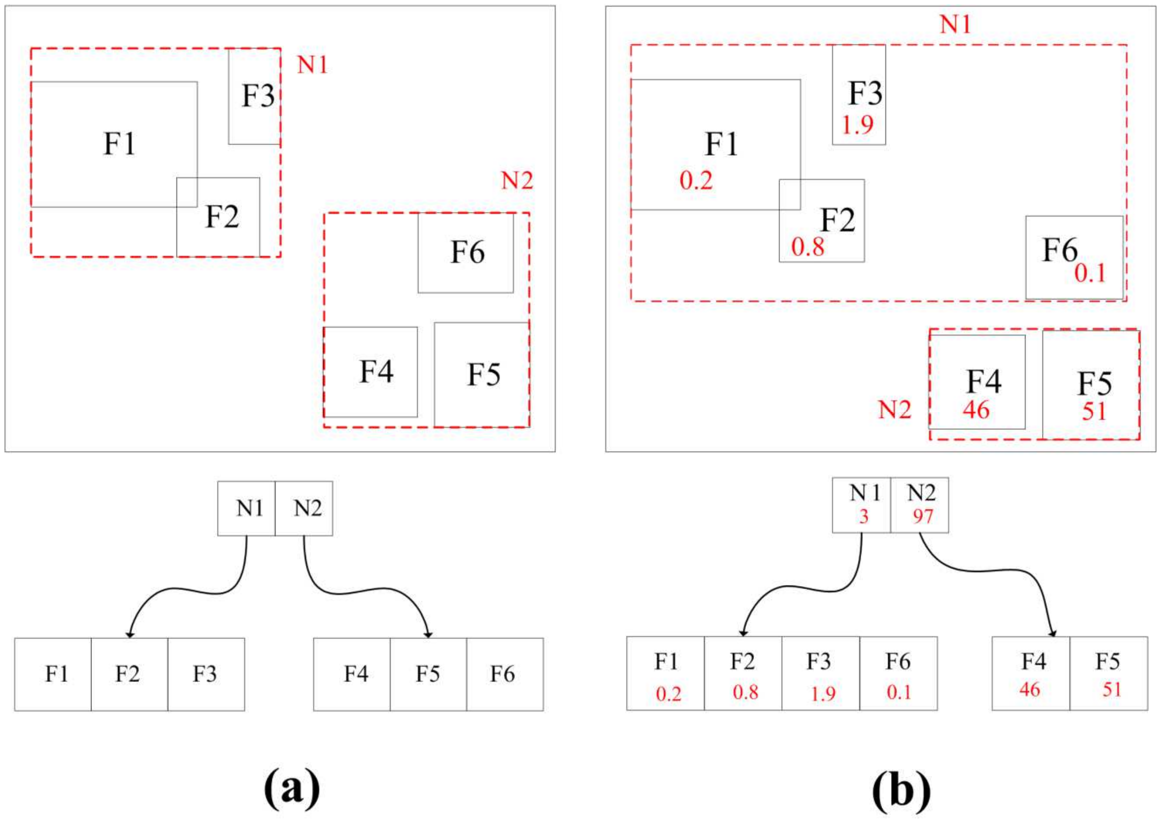


Figure 5. Indexing with a query pattern

Based on such indexing rules, I propose the APR-tree to support indexing with spatiotemporal query patterns. In the PMIM, multiple APR-trees are predefined to support various user categories. To integrate query patterns into a classic R-tree based data structure, the APR-tree records the access possibilities to each feature and node. This extra information is stored at APR-Tree leaf node entries and non-leaf node entries. An APR-Tree leaf node entry is formed as (MBR, Oid, AP), where the AP is the access possibility of the data object. A non-leaf node entry stores the access possibility of the entire entry by modifying the form to (MBR, *ptr*, AP) where AP is calculated based on all features belonging to this entry. This APR-tree data structure (

Figure 6) supports fast retrieval of both MBR and access possibility for feature.

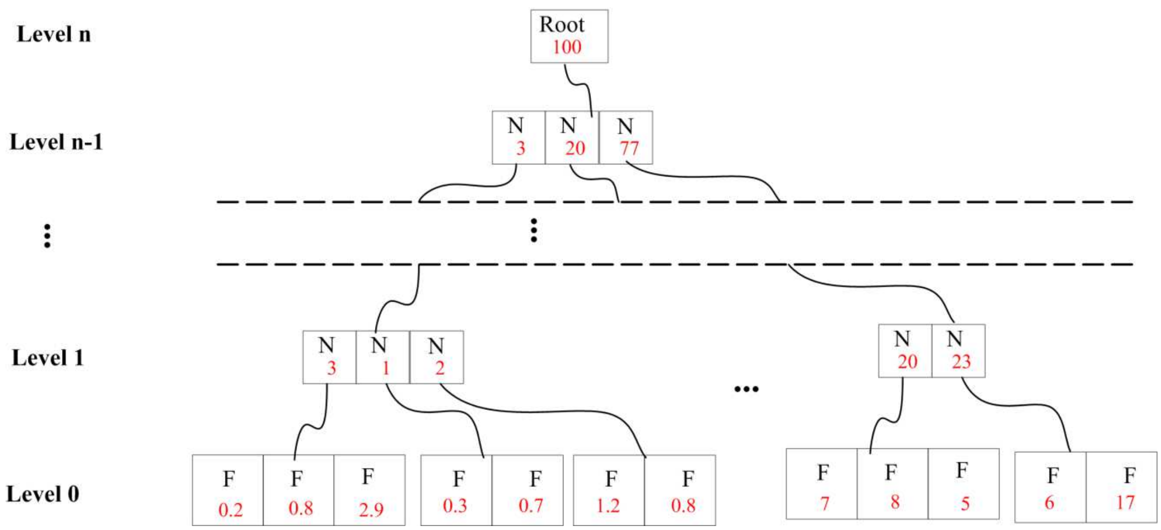


Figure 6. Data structure of an APR-tree

3.2 Spatiotemporal service performance evaluation

This section introduces a spatiotemporal service performance evaluation mechanism. This mechanism includes two major components: (1) the spatiotemporal service monitor collecting performance information of globally distributed GIServices using cloud computing and volunteer computing, and (2) the spatiotemporal performance model integrating spatiotemporal dynamics for better performance evaluation for users from different regions at different times.

3.2.1 Spatiotemporal service performance monitor

To collect globally distributed performance information, a new monitoring mechanism was designed by utilizing volunteer and cloud computing.

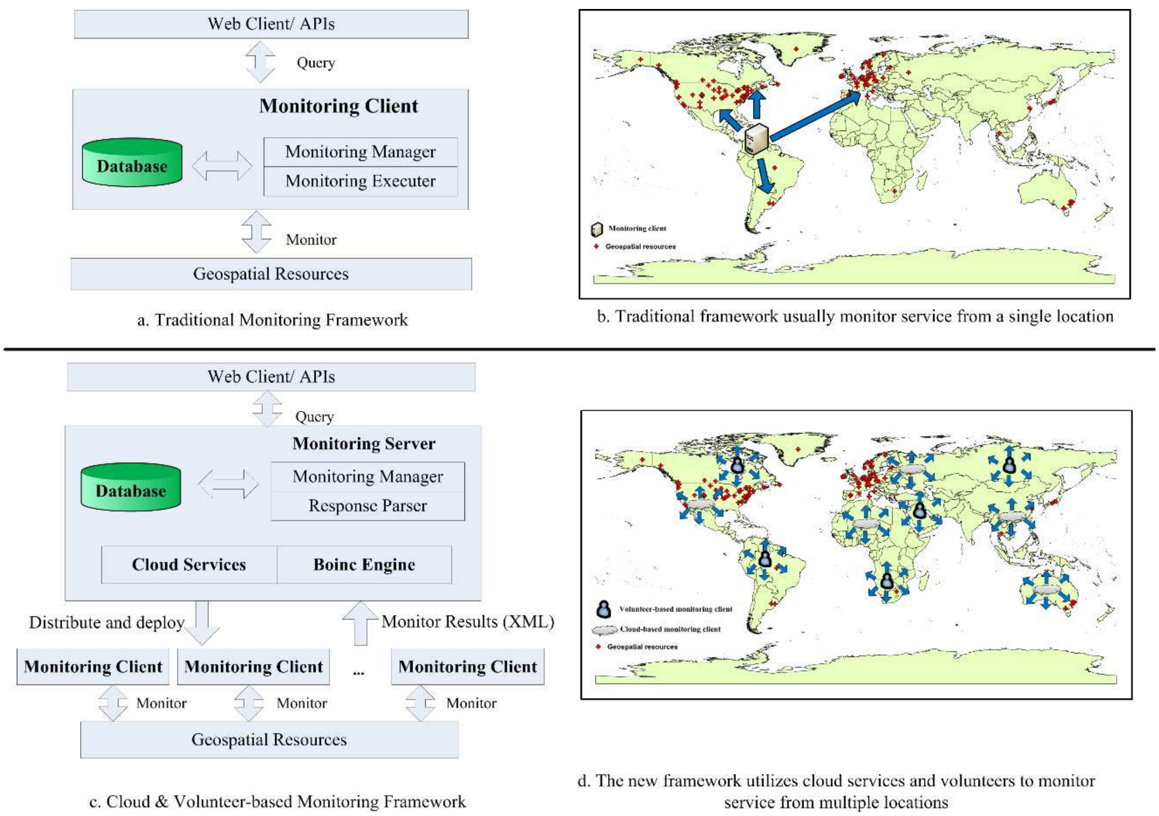


Figure 7 Cloud and volunteer-based monitoring mechanism (c and d) verses the traditional mechanism (a and b)

The traditional mechanism (Figure 7 a and b) uses a single or limited few monitoring sites to collect performance information. This monitoring client consists of three components: (1) the monitoring manager which configures and manages the monitoring process (e.g., monitoring schedule configuration); (2) the monitoring executer which collects performance information by periodically sending requests to services; and (3) the database which stores monitoring records. In addition, the monitoring client provides a web client /API for end users or other system to access collected information. However, such a single location-based monitoring mechanism may deliver misleading information for users far from the monitoring location.

A cloud and volunteer computing based monitoring mechanism is proposed to collect performance information from different geo-locations. The advancement of cloud and volunteer computing offers an effective and affordable global service monitoring solution by providing globally distributed computing resources (Yang et al., 2014; Korpela 2012). Based on cloud services, monitors can be expeditiously deployed to different continents to collect service performance information, whereas the volunteer computing engine arranges globally distributed computers from volunteers to perform distributed service monitoring with no or little cost. The proposed framework consists of the server and a large number of cloud and volunteer-based monitoring clients (Figure 7**Error! Reference source not found.**c). Similar to the traditional framework, monitoring activities are managed by the monitoring manager and information is stored in a database. The cloud service and Berkeley Open Infrastructure for Network Computing (BOINC) engine schedule, distribute, and deploy a number of monitoring clients at disparate geo-locations. Each monitoring client collects performance information independently from its geo-location, and the monitoring results (XML format) return to the server side for parsing (by the response parser). This framework utilizes widely distributed volunteer and cloud services and collects location-time-specific performance information.

3.2.2 Spatiotemporal service performance model

The proposed monitoring mechanism (Section 3.2.1) collects distributed service performance values and produces a discrete set of sampled performance points at different space-time locations. However, to provide performance evaluation, a spatiotemporal performance model is needed to provide value estimations at any un-sampled spatiotemporal point.

3.2.2.1 A unified spatiotemporal service performance model

The service performance was observed to be influenced by the access location of end user, access time, and location of the service provider. To integrate these spatiotemporal dynamics, a unified spatiotemporal performance model is proposed (eq. 1):

$$q = f\ (service, X_{server}, Y_{server}, X_{user}, Y_{user}, T) \qquad \text{(eq1)}$$

where *service* is the identification for a web service (service ID), *X-server* and *Y-service* are the location (latitude and longitude, respectively) of the service provider, *X-user* and *Y-user* are the user access location, and *T-* is the user access time. *q* is the performance.

3.2.2.2 Performance spatiotemporal neighbouring model

To estimate un-sampled performance values, a simple modeling approach is to retrieve a sampled performance value that is spatiotemporally closer to the prediction location. The formula for this neighbouring approach is as follows:

$$q\ (x, y, t) = q\ (x_s, y_{s,} t_s) \quad (eq2)$$

where $q\ (x, y, t)$ is the predicted service performance value at location (x, y) and time (t), $q\ (x_s, y_{s,} t_s)$ is a sampled value, and $(x_s, y_{s,} t_s)$ and (x, y, t) have the minimum distance (great-circle distance).

3.2.2.3 Performance spatiotemporal interpolation model

The performance spatiotemporal interpolation model processes sampled performance values to provide performance estimations at un-sampled spatiotemporal point (x, y, t). In order to interpolate spatiotemporal performance values, a traditional spatial interpolation algorithm (IDW) is extended by integrating reduction and extension metrics.

The reduction approach performs a traditional IDW calculation (eq 3) to find the nearest neighbours for each un-sampled performance point and calculates the weights based on the distance between two points. Subsequently, a linear temporal interpolation is applied on the spatial interpolation (eq 4). Finally, un-sampled performance data at location (x, y) and time (t) are calculated.

$$q\ (x, y, t) = \sum_{i=1}^{N} \lambda_i q_i(t), \qquad \lambda_i = \frac{\frac{1}{d_i}}{\sum_{k=1}^{N}(\frac{1}{d_k})} \quad ,(eq3)$$

$$q_i(t) = \frac{t_2 - t}{t_2 - t_1} q_i 1 + \frac{t - t_1}{t_2 - t_1} q_i 2, \qquad (eq4)$$

where $q\ (x, y, t)$ is the predicted performance value at location (x, y) and time (t); N is the number of nearest sampled points surrounding (x, y); $q_i(t)$ are the known performance values at time (t); and λ_i is the weight assigned to $q_i(t)$. The weights (λ_i) are calculated based on the great circle distance between sampled performance points and prediction point. The temporal interpolation algorithm that estimates $q_i(t)$ based on two sampled performance value $q_i 1$ and $q_i 2$ at t_1 and t_2, where $t_1 < t < t_2$ is shown in Eq. 4.

In the extension approach, time is the third dimension in the traditional IDW algorithm (Eq 5 and 6).

$$q\ (x, y, t) = \sum_{i=1}^{N} \lambda_i q_i, \qquad \lambda_i = \frac{\frac{1}{d_i}}{\sum_{k=1}^{N}(\frac{1}{d_k})} \quad ,(eq5)$$

$$d_i = \sqrt{(x_i - x)^2 + (y_i - y)^2 + (t_i - t)^2} \quad ,(eq6)$$

Unlike the reduction approach, q_i is the sampled performance value close (spatiotemporal distance) to the predication point (x, y, t); and d_i is the spatiotemporal distance between known performance (x_i, y_i, t_i) and prediction points (x, y, t). Proper scales and weights for both spatial and temporal dimensions are needed for the extension approach, which requires a large number of experiments to determinate.

3.3 Spatiotemporal cloud computing adoption

This section introduces a cloud-enabled framework for the CLH to utilize cloud services and spatiotemporal user patterns in the CLH workflow. The framework (Figure 8) includes five main components: (1) the Elasticity Manager (EM) supports automatic computing resource scaling; (2) the Cloud Workload Balancer (CWB) enables user workload balancing; (3) the Deployment Manager (DM) optimizes CLH deployment in a global cloud environment; (4) the Cloud Resource Pool (CRP) provides computing resources and global deployment environment for the entire framework; and (5) the Spatiotemporal Knowledge Store (SKS) manages and provides spatiotemporal patterns to DM and EM to better locate and scale computing resource. It also monitors the CRP and analyzes the system log files to provide cloud resource distribution of and user patterns to the entire framework.

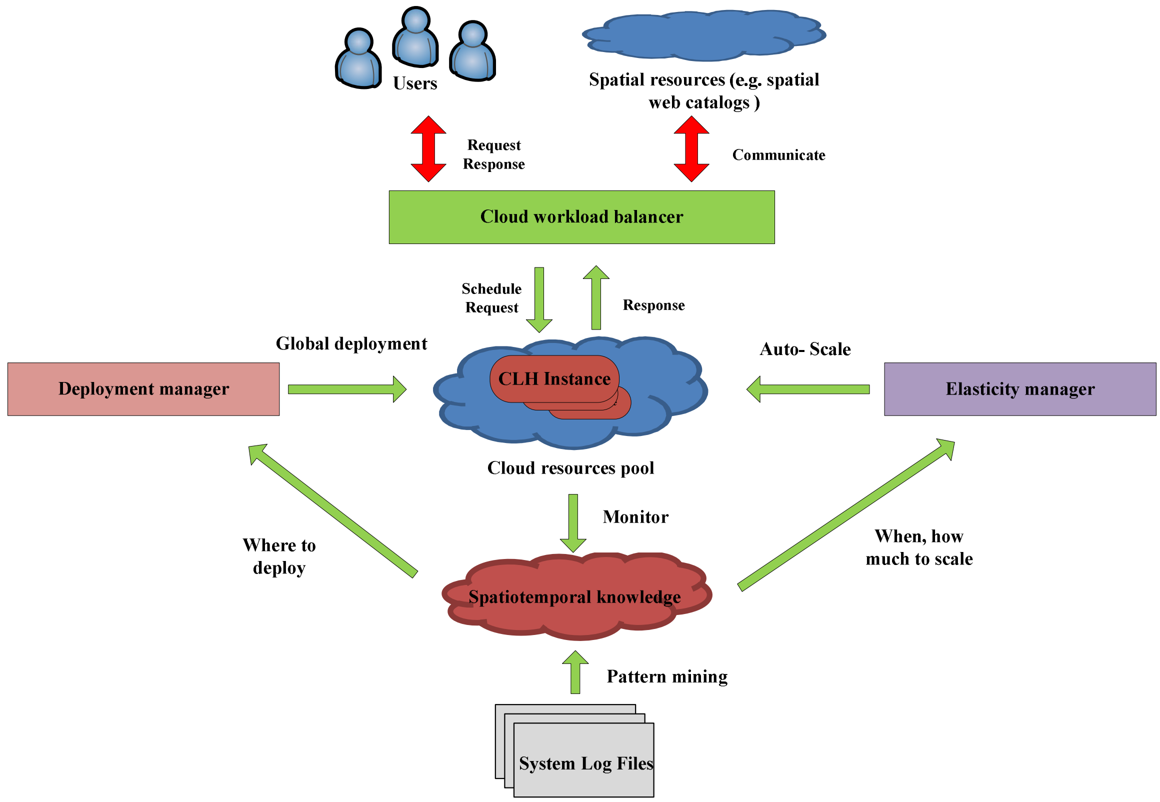


Figure 8 Spatiotemporal and cloud-based supporting framework

3.3.1 Elasticity Manager (EM)

The EM enables the automatic provision/release of computing, storage, and network resources in two directions: (1) vertical scaling changes CLH instance configurations (e.g. CPU cores number, memory size) on-the-fly to meet the computing demand; (2) horizontal scaling increases or reduces the number of CLH instance. Figure 9 shows the general framework for the EM. The Elasticity Executor provisions and releases cloud resource from CRP based on pre-defined auto-scaling rules. There are two categories of auto-scaling rules: (1) routine scaling rules and (2) dynamic scaling rules. The routine scaling rules automatically scale computing resource based on the routine patterns. For example, user access periodically increases and decreases with two rush hours (9:00 am to 10: am, and 7: 00 pm to 8: 00 pm) every day. Based on such an access pattern, the number of CLH instance could be scaled up by horizontal scaling right before rush hours. The dynamic scaling rules define auto-scaling mechanism for dynamic and unexpected accesses (e.g. spiking access). Dynamic scaling rules can be defined to help CLH provision sufficient computing resource after detecting unexpected accesses. For example, one CLH instance needs to be provisioned when existing CLH instances in a cloud service region cannot respond to users within certain predefined threshold (e.g., 6 seconds). Meanwhile, a CLH instance could be released if the utilization rate (CPU, memory or network) is low (e.g. lower than 10%). In order to determine when, where and how much computing resource needs to be provisioned/released, the dynamic scaling rules and routine scaling rules need to consider current user workload and identified spatiotemporal patterns from CWB and SKS.

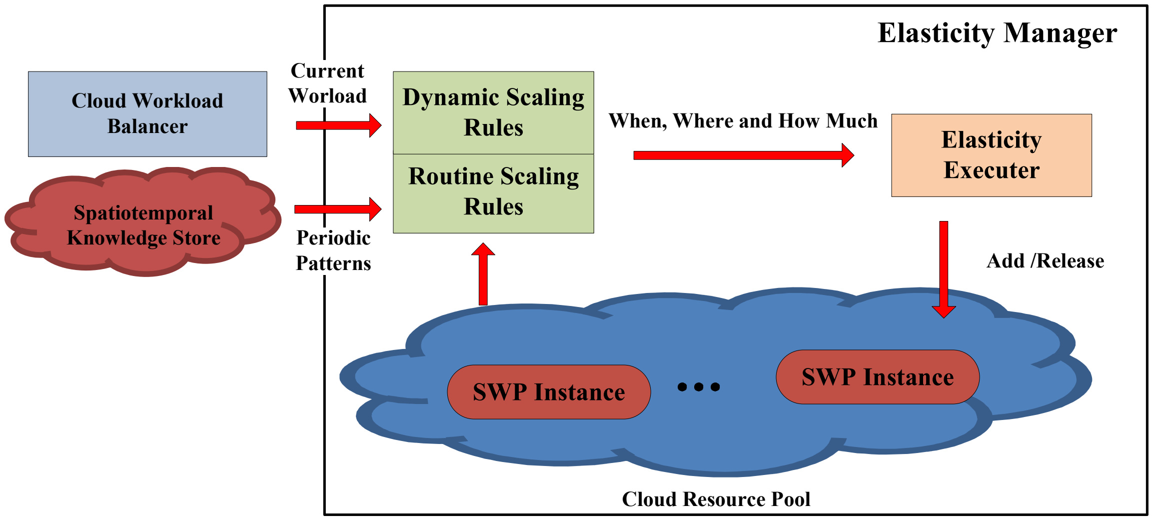


Figure 9 Elasticity Manager framework

3.3.2 Cloud Workload Balancer (CWB)

The CWB automatically distributes end user requests to multiple CLH instances to (1) reduce the response time, (2) handle larger number of concurrent requests by utilizing multiple CLH instances, and (3) achieve greater fault tolerance. Figure 10 shows the framework of the CWB. Within the framework, multiple CLH instances are provisioned from the CRP supported by EM and these instances are distributed in different cloud service regions by DM. These CLH instances are configured to provide a single CLH access end-point (domain name/URL). The Instance Detector periodically detects CLH instance status such as CPU, Memory and Disk utilization rate, instance location, the network traffic, and the average response time. The Workload Distributor forwards end user requests to different CLH instances based on current instance status. The workload forwarding rules is determined by (1) user request intensities and distributions, (2) CLH instance distributions, and (3) the current workload of existing CLH instances. Requests are forwarded to a CLH instance with less workload and close to the end users.

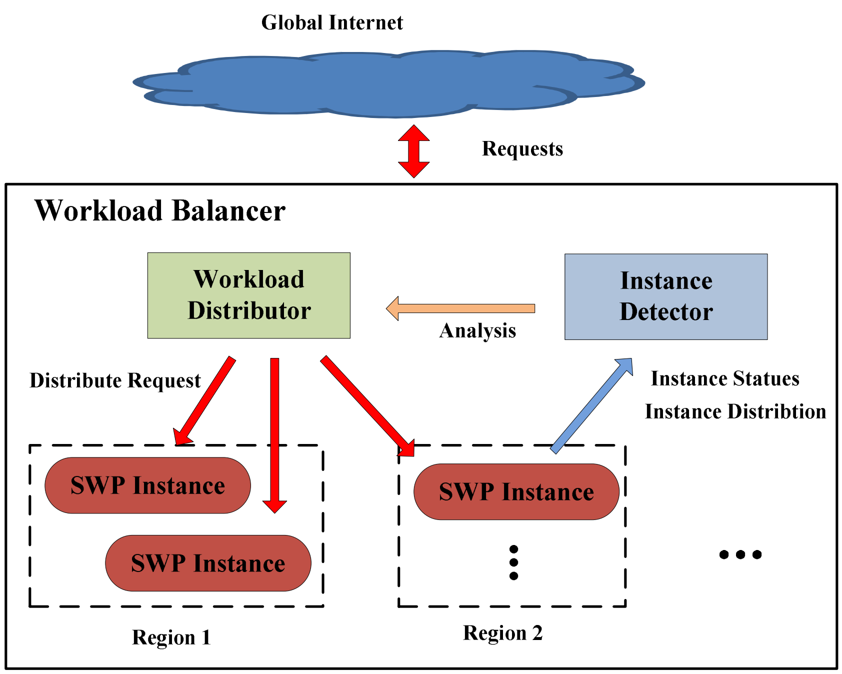


Figure 10 Cloud Workload Balancer framework

3.3.3 Deployment Manager (DM)

DM supports global deployment to optimize the distribution of CLH instance. Cloud service providers usually have multiple datacentres (cloud regions) across the world. By utilizing this cloud service feature as well as leveraging spatiotemporal principles (Yang et al., 2011b), the DM helps the CLH gain location advantages by allocating CLH instances and end users and geospatial resources. To integrate the cost for operation, a cost model can be included to improve the DM and EM by considering the cost of cloud instance in different regions. Figure 11 shows the distributions of the CLH, end users, geospatial resources and the system servers. The traditional CLH server located in the United States may have very slow responses to users in Europe and Asia. CLH and geospatial resource distribution also results in large volumes of data transfer over inter-continent networking. By utilizing global deployment, the CLH can be fast deployed at North America, South America, Europe, Africa, and Asia so that the communication cost between global users, spatial resource and CLH can be reduced.

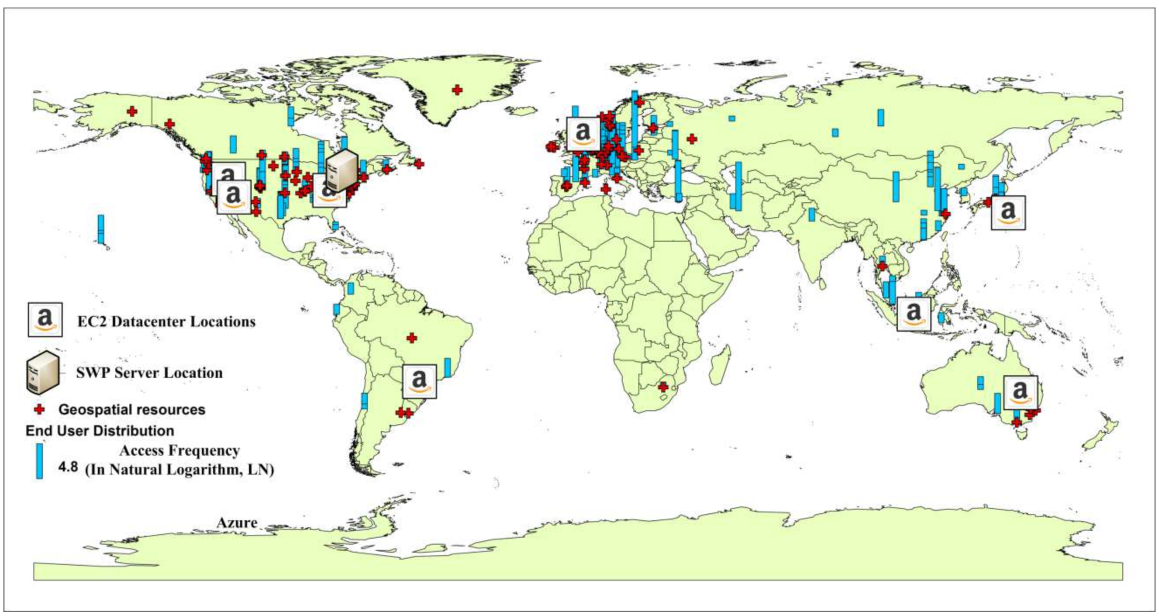


Figure 11 The distribution of geospatial resources, end users, traditional CLH server, and available EC2 datacenters

CHAPTER 4 EXPERIMENT & RESULTS

4.1 Spatiotemporal index experiment

4.1.1 Identified spatiotemporal user behavior patterns

Based on CLH log file, various spatiotemporal patterns of user behaviors are identified. From Dec 22th 2010 to present, the CLH recorded a total number of 2,202,660 user accesses. Identified spatiotemporal patterns include:

- **High Access Frequency Regions:**
- Figure 12a illustrates the global distribution of end users with access frequency.

The United States, Europe and East Asia are three high access frequency regions where more replication instances and indices are needed.

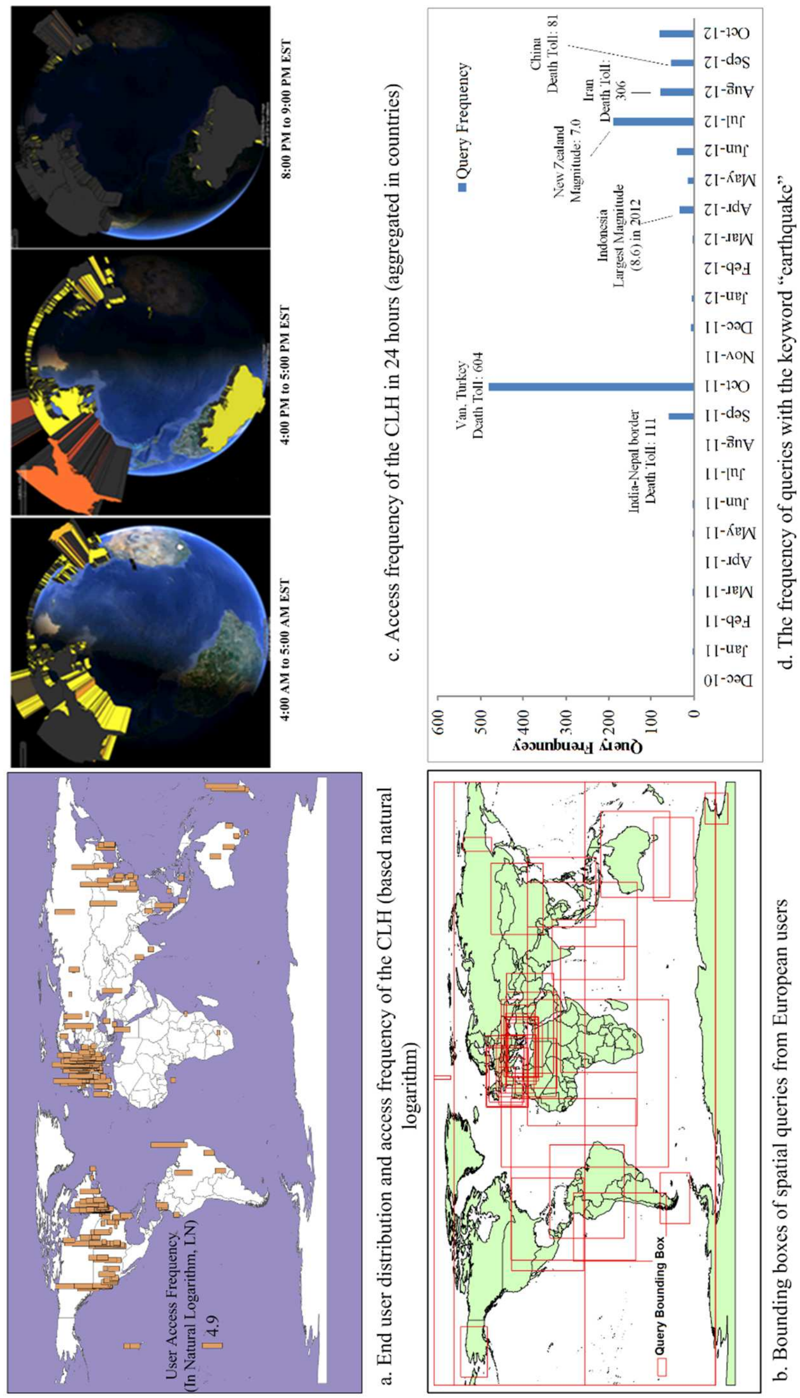


Figure 12. Identified spatiotemporal patterns of user behaviors in CLH

- **Local Interests: Error! Reference source not found.**b displays the bounding boxes of spatial queries from European users. Bounding boxes concentrate on European region, which indicates European users tend to request data closer to their location.

- **Periodical Access & rush hours:** Figure 12c captures three screenshots of an animation presenting the CLH access frequency (grouped into counties) in 24 hours. Afternoon and morning (local time) are two time windows with high access frequency. This pattern is very strong in some regions, such as Australia and New Zealand where about 80% of the total requests are generated in rush hours. More replication instances, cloud computing resource and indices are needed in rush hours. Index updating should be avoided during rush hours.

- **Spiking access:** Figure 12d shows the query frequency for a keyword "earthquake" from Jan 2011 to Oct 2012 and related earthquake events. Significant increase of query frequency can be observed in October, which may be resulted by the earthquake in Van, Turkey. This 7.1 magnitude earthquake costs more than 534 lives on October 23th (USGS 2011). Specially designed index, cache configurations and updating mechanism are needed for spike workloads.

4.1.2 Spatiotemporal index performance evaluation

3.2.2.1 Experiment data

In order to comprehensively evaluate index performance, a number of feature datasets were collected.

- **Observation dataset:**

The GEO-One-Stop (GOS) (http://geo.data.gov/geoportal/csw) and the CLH (http://clearinghouse.cisc.gmu.edu/geonetwork/srv/en/csw) are two well-known and widely used metadata catalogues for EO data. On Dec 26th 2012, all available metadata from the GOS and GEOSS CLH are subscribed through OGC: CSW. The total number of subscribed metadata is 479,477 (369,478 from GOS and 109, 999 from CLH). The bounding boxes that describe the spatial coverage of EO data were extracted from the metadata.

- **Simulation/derived dataset (uneven distribution)**

In order to demonstrate the performance of an index with a certain number of features, simulated bounding boxes are generated based on the frequency of observation dataset. For all spatial features, I classified and calculated the frequency of features with the same spatial coverage. For example, 0.98% of features have the same spatial coverage with (-180, -90, 180, 90). Based on the frequencies and proportions, simulated bounding box datasets with specific numbers of features (10k, 100k, 1000k and 10000k) were generated.

- **Simulation/derived dataset (even distribution)**

Features in read dataset are unevenly distributed. For example, about 80% of published data concentrated on United States and Europe regions (Figure 13a). The uneven distribution of features may affect index performances. In order to demonstrate index performances with unevenly distributed features, datasets with specific numbers (10k, 100k, 1000k and 10000k) of evenly distributed features are generated (Figure 13b).

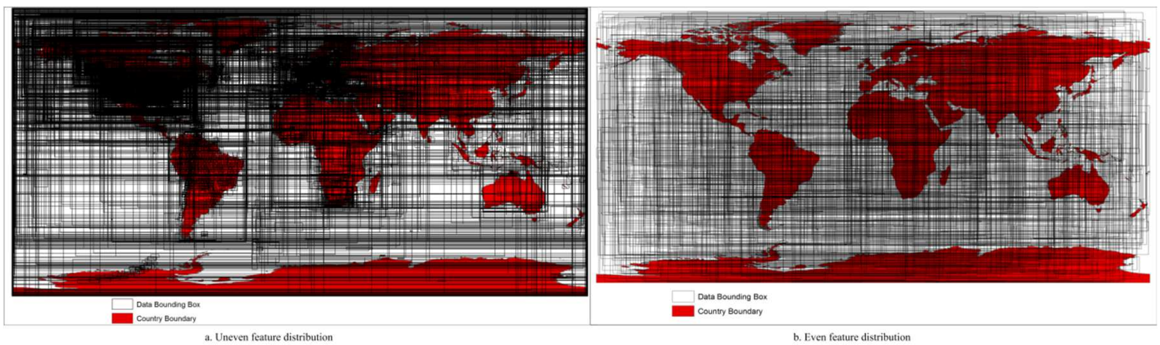


Figure 13 Features in observation data concentrated on United States and Europe, and evenly distributed features are simulated for performance comparison

3.2.2.2 Index performance evaluation

The first set of experiments compared the APR-Tree performance against the R*-Tree in four scenarios: (1) Uneven feature distribution, uneven query distribution; (2) Even feature distribution, even query distribution; (3) Uneven feature distribution, even query distribution; and (4) Even feature distribution, uneven query distribution. In each scenario, the performance of R*-Tree and APR-Tree was recorded using 0.2, 0.5 and 0.8 P values with 10k, 100k, 1000k and 10000k features. The performance was measured in terms of the query time (Figure 15) and the number of nodes needed to be accessed (Figure 14) to perform a collection of queries. The request/ response message passing time in the network is not included in the evaluation, because it is not affected by the index design. Performance gains are defined in $(eq5)$ and $(eq6)$.

Node Access Gain (NAG) = (R*-Tree Node Access - APR-Tree Node Access) / APR-Tree Node Access * 100% $(eq5)$

Query Time Gain (QAG) = (R*-Tree Query Time - APR-Tree Query Time) / APR-Tree Query Time * 100% $(eq6)$

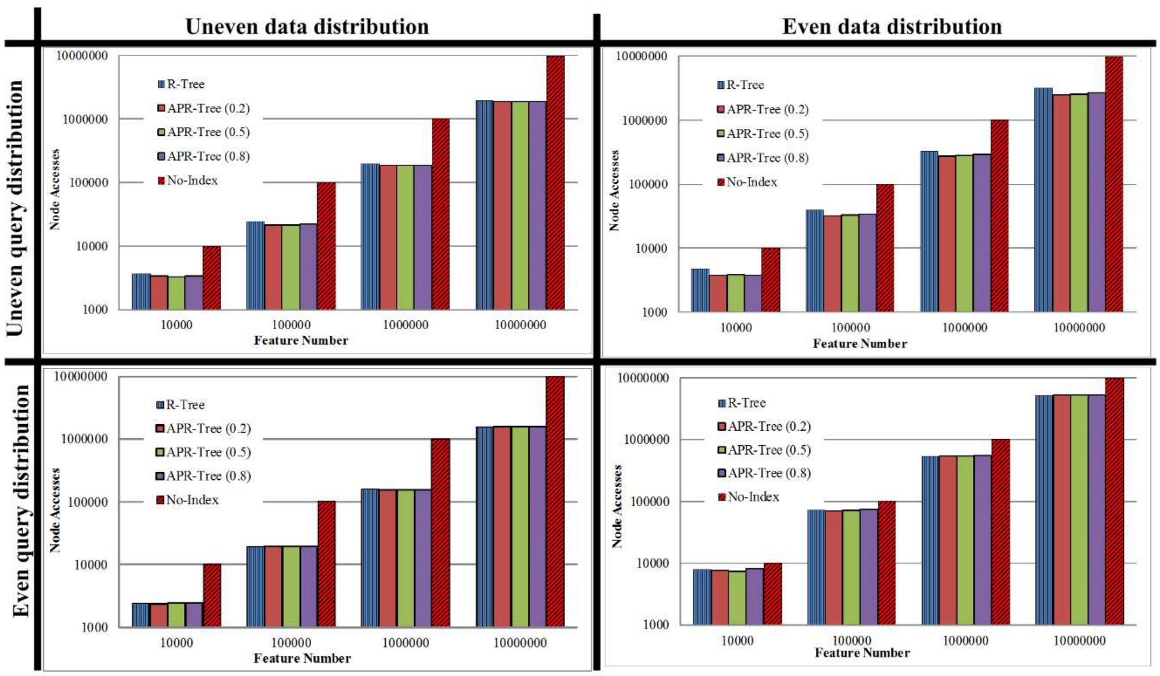


Figure 14. APR-Trees reduce node accesses (y-axis is in log scale) compared to R*-tree; 0.2, 0.5 and 0.8 are p values (the weights of access separability)

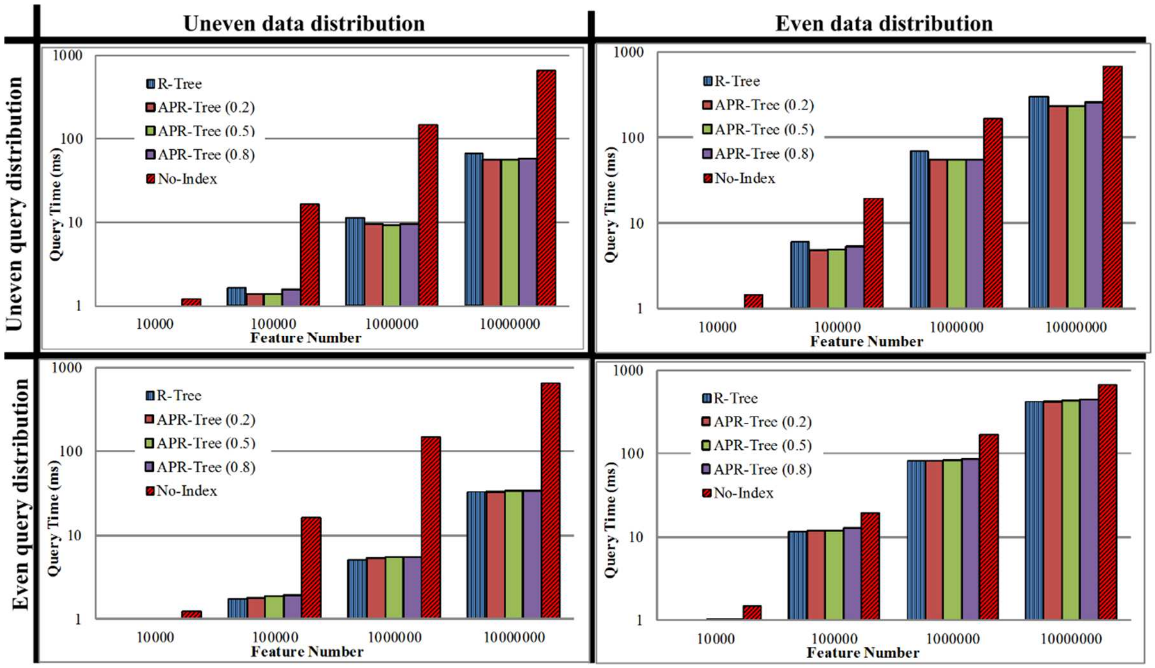


Figure 15. APR-Trees reduce query time (y-axis is in log scale) compared to R*-tree; 0.2, 0.5 and 0.8 are p values (the weights of access separability)

The following trends were observed from experimental results:

- An indexing mechanism significantly reduces node accesses (112% - 422% gain) and query time (127% - 891% gain), because features are reorganized into tree structures to avoid redundant looping processes.
- When query is unevenly distributed (real world pattern)
    - The APR-Tree consistently outperforms the regular R*-Tree with different feature distribution (even or uneven) and different feature number (10k, 100k, 1000k and 10000k). The performance gains are about 9% on node accesses and 18% on query time using unevenly distributed features; and about 19% on node accesses and 23% on query time with evenly

distributed features. The reason for this is that the APR-Tree considers users' query patterns, and balances the spatial discrimination and feature access separability in the index. In addition, evenly distributed features tends to results in APR-Tree performance gains, because evenly distributed features tends to have larger bounding box and overlap areas thereby weakening the advantage of the R*-Tree.

    - In general, APR-Trees with 0.2 (about 18% performance gain) and 0.5 P values (about 20% performance gain) outperform APR-Trees with 0.8 P value (about 13% performance gain). A good balance between spatial discrimination and feature access separability results in better performance of the APR-Tree and vice versa. An APR-Tree with 0.8 P value over-weighted the feature access separability by significantly reducing the spatial discrimination capability of the index.

- When query is evenly distributed
    - The regular R*-Tree outperforms the APR-Tree. The performance overheads of APR-Tree on node accesses and query time are about 4% and 9% respectively. An APR-Tree is optimized for a certain pattern of queries, which may also result in performance overhead if the spatial discrimination capability of the index is reduced but no improvement on feature access separability.
    - APR-Trees with 0.2 P value result in less performance overheads (about 2% on query time) than APR-Trees with 0.5 P value and 0.8 P value (about 6%

and 9% respectively on query time). This is because higher weight of feature access separability leads to worse spatial discrimination capability of the index.

APR-Trees in the first test set are optimized for United States users. In order to demonstrate the efficiency of the APR-Tree in other regions, three APR-Trees are built for three regions: United States, Europe and China by using 470K features with uneven distribution. Figure 16 illustrates the performance of APR-Trees and R*-Trees for three regions (uneven queries), where the performance gains are 7.7 %, 7.9% and 13.5% on node access and 22%, 23.8% and 27.6% on query time for United States, Europe and China respectively.

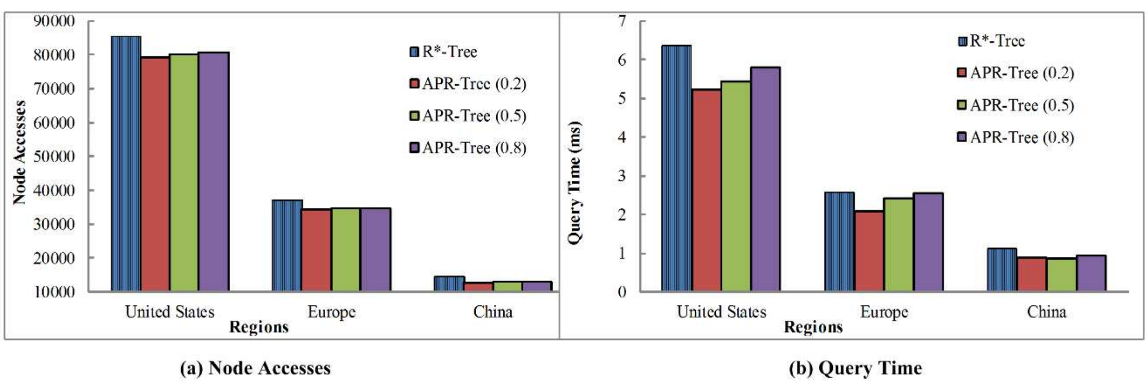


Figure 16. APR-Tree provides better performance than R*-Tree in three different regions

4.2 Spatiotemporal service evaluation experiment

4.2.1 Status of the global service monitoring system

Over 250 distributed service monitors (Figure 17) exist on a global scale with the following attributes: 247 volunteer-based monitors are distributed in North America, Europa, Asia and Oceania; six cloud-based monitors (supported by Window Azure) are located in North America, Europa, and Asia; two cloud-based monitors (supported by Amazon EC2) are located in South America and Oceania; and one traditional monitor is located in Virginia, United States. A number of the 3,557 GIServices (3,188 OGC:WMS, 328 OGC: WFS, and 41 OGC:WCS) are checked by these globally distributed monitors on a daily basis and whose status is summarized in Table 2.

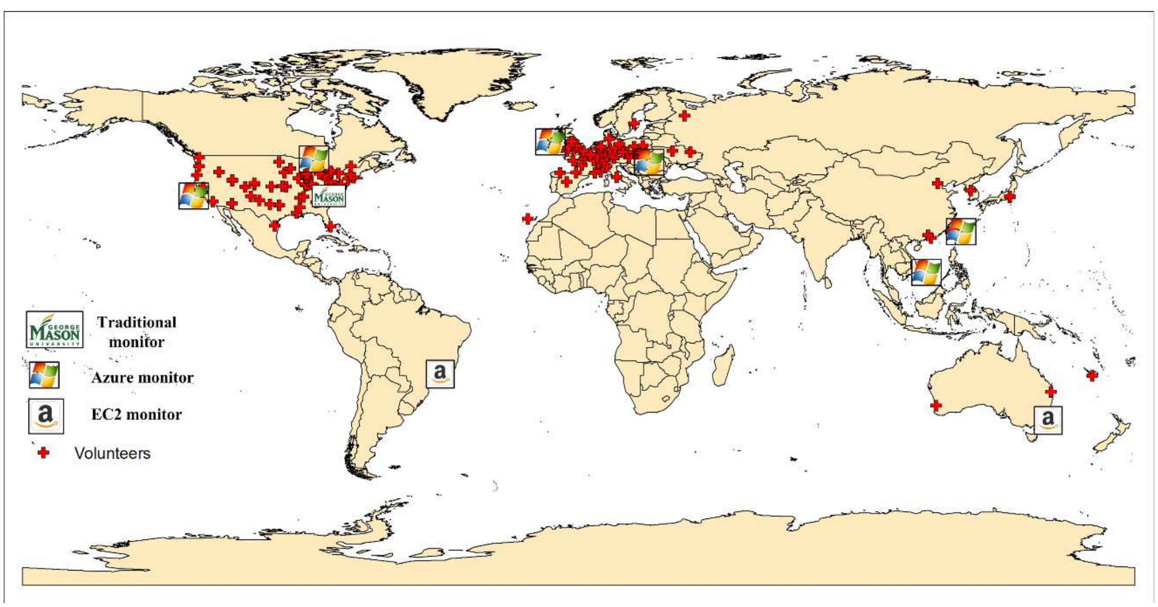


Figure 17 Distribution of service monitors

Table 2 Status of volunteer, cloud and traditional service monitors

	Traditional	Cloud	Volunteer
Period	Sep 2011 to Present	April 2014 to Present	April 2014 to Present
Number	1	8	225
Frequency	Once per day	Once per two hours	Twice per day
Performance records	328,607	456,000	2,513,621
Cost	N/C	$25 per day	N/C

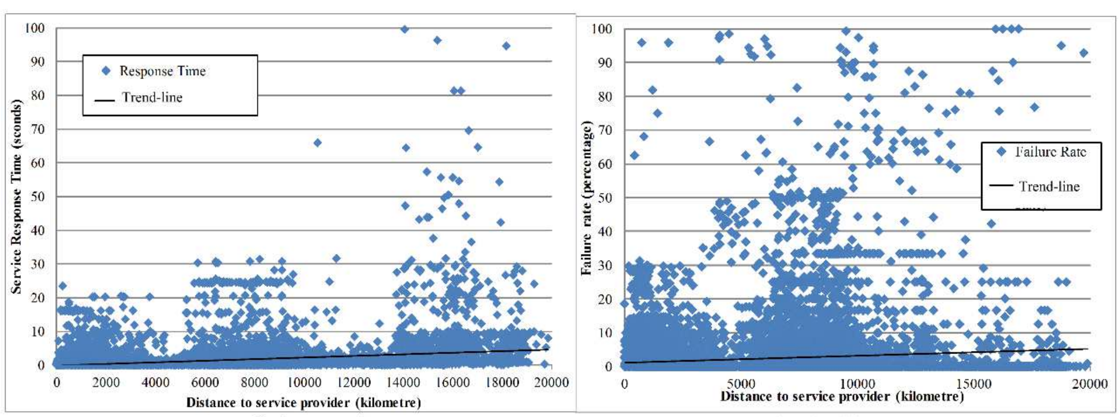
4.2.2 Identified service performance patterns

By crawling GIService information from a number of popular SDIs, we obtained 3583 (WMS), 322 (WFS), and 42 (WCS) addresses, which provide XML-based descriptions of web service interfaces. More than 80,000 spatial dataset (e.g., web map layers from WMS) were retrieved from these OGC: WxS. By leveraging global and hourly monitoring to these web services with the proposed monitoring mechanism, we collected performance information (2.7+ million records) such as response time, throughput and failure rate.

Collected monitoring information was archived in a spatiotemporal database with a two hour interval. Statistics of service performance distributions were generated according to different services, spatial regions, and the great circle distance between monitoring site and the service (user-server distance). Temporal characteristics were analyzed based on performance curve monthly, each day of a week, and each hour of a day. By analysing this performance information, spatiotemporal service performance patterns were identified.

- Spatial Patterns

End users from different regions receive disparate levels of service performance, with shorter commutation distances (end user to service provider) yielding better performance. The response time to request OGC: WxS GetCapabilities with different user-server distances is depicted (Figure 18a). About 41% of the records have <5,000 km user-server distance (23.46 s maximum response time, 0.71 s average response time); 40% of the records are within 5,000 to 10,000 km (31.3 s maximum, 1.4 s mean); and 20% of the records have >10,000 kilometre user-server distance (with 934 s maximum, 1.8 s mean). The trend-line for service response time generally increases with additional user-server distance. As shown in Figure 18b, service failure rate (percentage of failed service access) also increases with the increase of user-server distance.



a. Service response time   b. Service failure rate

Figure 18 Response time (2a) and failure rate (2b) as a function of distance from service provider

To demonstrate spatial patterns, the spatial distribution of service performance for two popular OGC:WMS is investigated (Figure 19). In general, the same WMS provided better performance to end users close to the service provider. For example, The European WMS (http://www.premis.cz/atlaszp/isapi.dll) located at Czech Republic provided shorter average response time to access a map layer (2000 pixel width and 1000 pixel high) for European users than users from other continents. A similar pattern was evident for the American WMS (http://mrdata.usgs.gov/cgi-bin/state/oh) in Herndon, United States. Two WMSs also provided lower service failure rate for end users closer to the service provider. The service failure rate of the European WMS for European, North America and Asian users was 0, 3.6, and 5.7 %, in order.

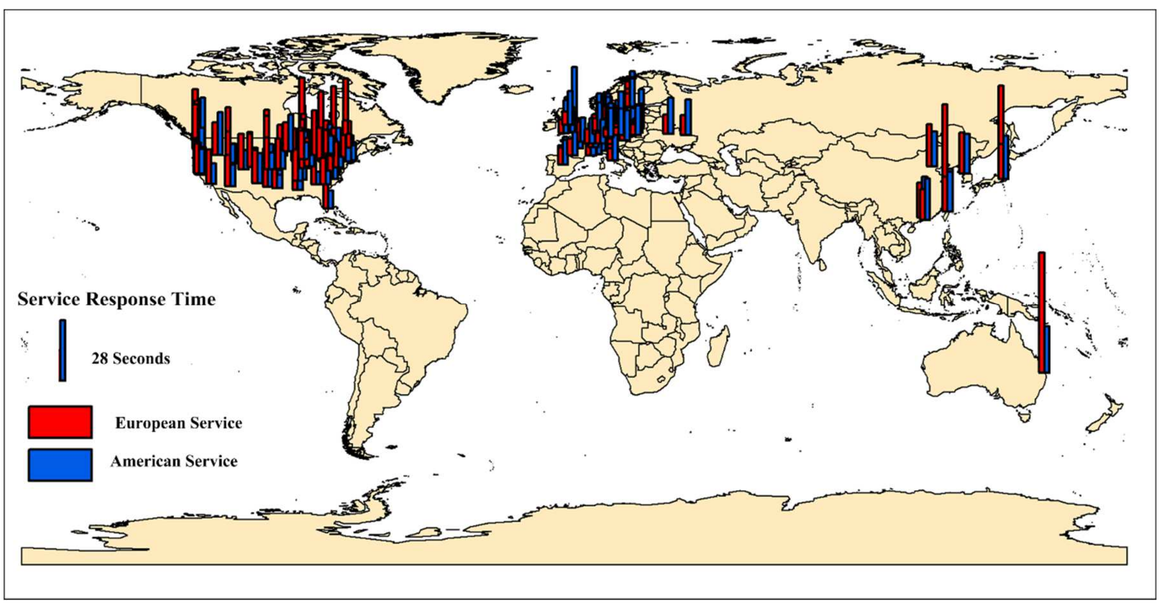


Figure 19 Global users service response time from the European service (red bar) and United States (blue bar)

- Temporal Patterns

End users received dissimilar levels of service performance at different times. Within the monitoring period between September 15 2011 and April 23 2014, three temporal patterns were observed.

First, service performance gradually decreased over a long time period. The average response time of all monitored services (Virginia, the United States) gradually increased from 0.48 (September 2011) to 0.61 s (April 2014) (
Figure 20a). The service failure rate increased from 1.5% (September 2011) to 2.2% (April 2014) (
Figure 20b). It proposed that this increasing failure rate resulted from the fast growth of end user accesses to these OGC: WxS, which was faster than the computing infrastructure upgrading speed for these GIServices.

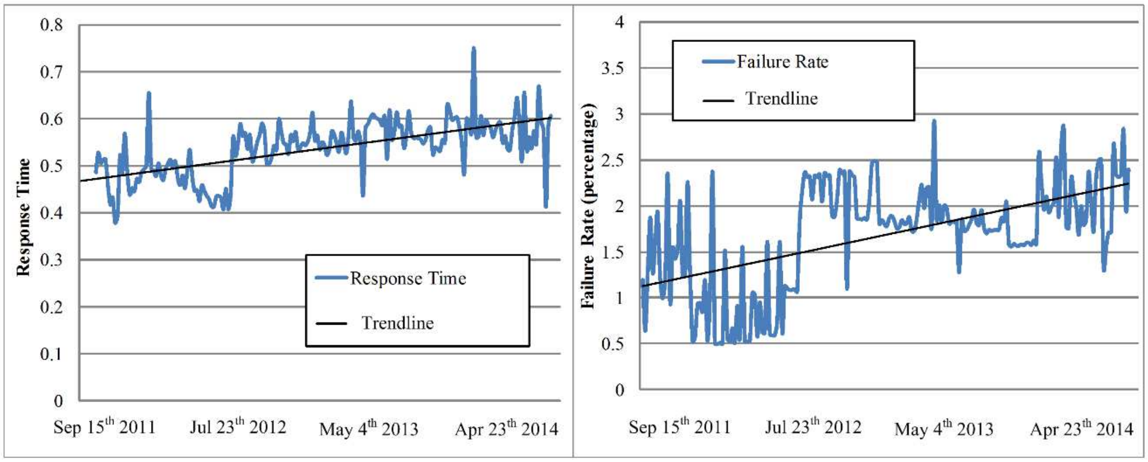


Figure 20 The OGC: WxS response time (left 4a) and failure rate (right 4b) over time from September 15th 2011 to April 23th 2014 as monitored from Virginia, United States)

The service performance fluctuates (Figure 21). End users received better performance on weekends compared to weekdays (weekly pattern). But no clear periodic pattern was observed for the 24 hour daily pattern. The weekly pattern resulted by the decreasing needs of geospatial resource during weekend. Most GIServices are accessed by global users, weakening the daily access pattern because these services have active end users in different time zones.

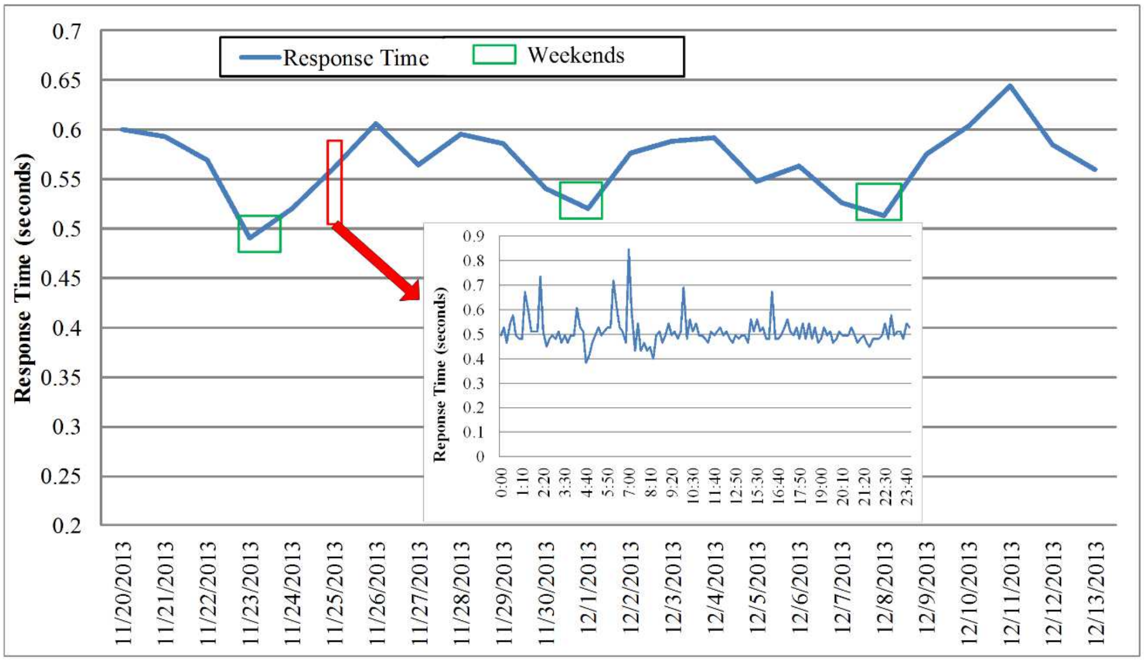


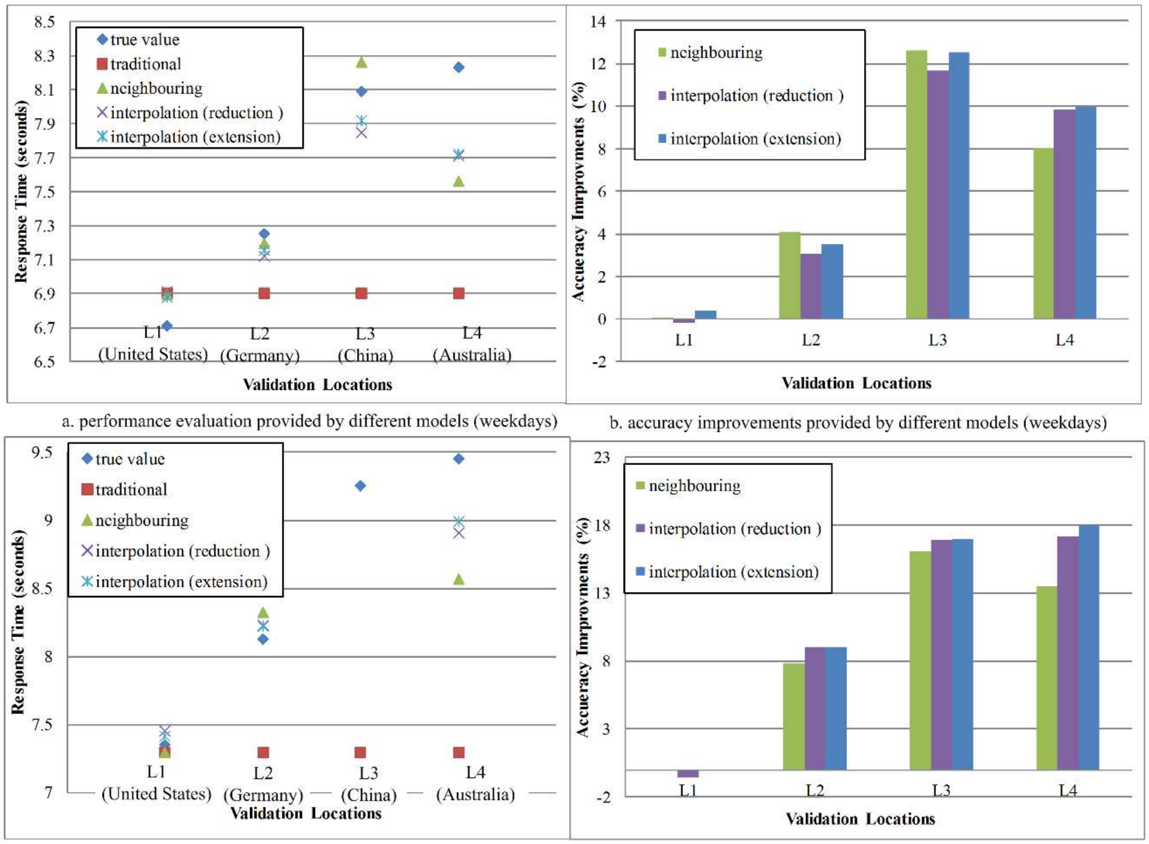
Figure 21 The response time as a function of weekends and weekdays

4.2.3 Spatiotemporal service model validation

4.2.3.1 Experiment data

The service performance data were separated into training and observation dataset. The training dataset was used as the model input for the service performance predication, whereas the observation dataset was used for the validation of predication accuracy. A number of 1,035 globally distributed performance values were selected as observation dataset. To ensure the validation accuracy, these observation data were spatially distributed on four continents and temporally distributed on both weekdays and weekends. The remainder of the collected data was used as training dataset.

4.2.3.2 Model validation



a. performance evaluation provided by different models (weekdays)

b. accuracy improvements provided by different models (weekdays)

c. performance evaluation provided by different models (weekends)

d. accuracy improvements provided by different models (weekends)

Figure 22 Validation results for different performance models

Figure 22a (weekdays) and Figure 22c (weekends) show the response time true value and response time evaluated by the traditional monitoring mechanism, neighboring model, interpolation (reduction) model and interpolation (extension) model. Figure 22b and Figure 22d shows the accuracy improvements provided by four models (calculated by Eq. 7).

$$Accuracy\ improvement = \frac{|Q_o - Q_t| - |Q_n - Q_t|}{Q_t} * 100\% \quad \text{(Eq. 7)}$$

where, Q_t is the true performance value, Q_o is the performance value provided by the traditional monitoring mechanism, and Q_n is the performance value predicated by the spatiotemporal performance models.

The results highlight five salient points and each is presented below.

a) First, at the validation location 1 (United States), the traditional mechanism provided accurate evaluations due to the closeness to the service provider. Therefore, the performance estimation improvements are very low. However, for the other three validation locations, spatiotemporal performance models provided 4%-20% performance estimation improvements over the traditional mechanism. End users in those regions could receive more accurate performance information from spatiotemporal performance evaluations.

b) Second, the neighboring provided accurate estimations (about 0.08 relative residual ($|Q_n$-$Q_t|/Q_t$)) when a monitoring site was proximal to the prediction location. The estimation accuracy decreased if monitoring sites were farther away (e.g. location 4 with 0.96 relative residual, Figure 22a, Figure 22b). The accuracy improvements were about 4%-16% (Figure 22c, Figure 22d).

c) Third, interpolation generally had good accuracy for regional predication with high density monitoring sites in the same region. For example, the relative residuals were about 0.02 and 0.06 for location 1 and location 2 (Figure 22a, Figure 22b). The interpolation extension approach was slightly better than the reduction approach in most validation locations. The interpolation method also provided similar or slightly better accuracy improvements than the neighboring approach.

4.3 Spatiotemporal cloud computing adoption

4.3.1 Experimental environment

The experiment is conducted on traditional servers, commercial cloud services and open-source cloud solutions. Figure 23 illustrates the geographic locations of the experiment facilities:

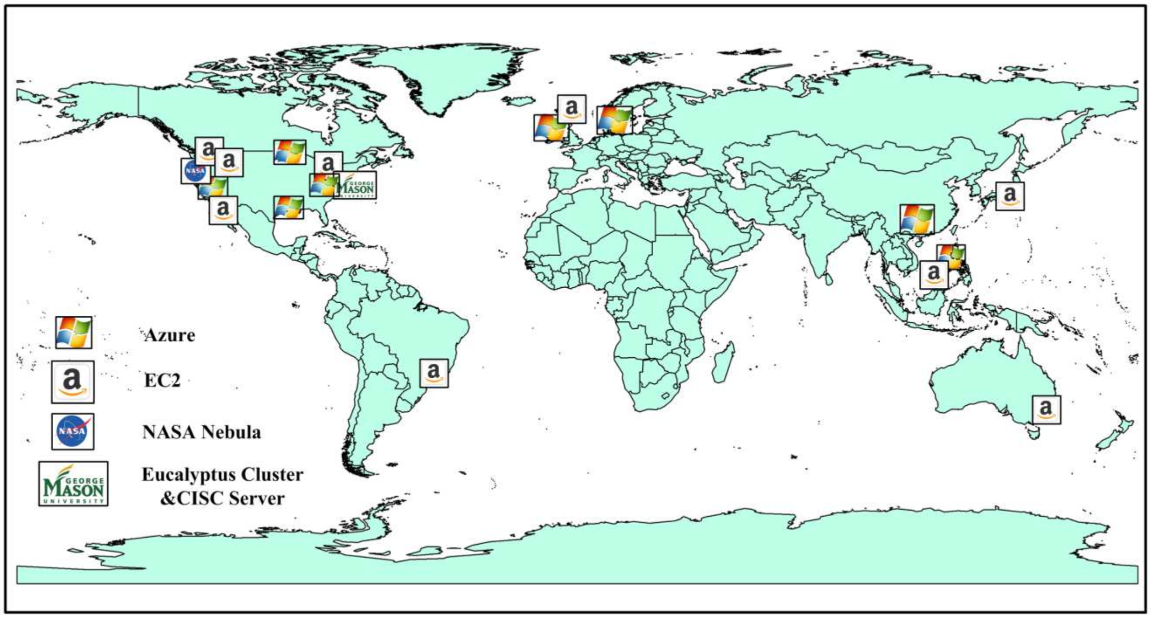


Figure 23 Globally distributed experiment facilities

(1) Azure has four data centers in United States: Illinois, Texas, California and Virginia; two data centers in Europe: Dublin, Ireland and Amsterdam, Netherlands; and data centers in Asia: Hong Kong, China and Singapore.

(2) EC2 has three data centers in United States: California, Virginia and Oregon; one data center in South America: São Paulo, Brazil; one data center in Europe: Ireland; two data centers in Asia: Singapore and Tokyo, Japan; and one data center in Oceania: Sydney, Australia.

(3) NASA Nebula locates at Ames California.

(4) The Open-Source Cloud clusters and CISC traditional server locate in Fairfax Virginia.

The experiment facilities in United States are connected through the National LamdaRail (National LamdaRail 2013) that provides high speed Internet connection while facilities in different continents are connected with global networks.

The CISC computing cluster is used to provide the computing and network environment for the Open-Source Cloud solutions. Each cloud platforms is configured with the same computer/network environment. CISC cluster has 25 computing nodes that are connected through local area networks (1Gbps bandwidth). Each node has 16GB memory and a dual quad-core CPU of 2.33 GHz.

4.3.2 Performance measurement across different cloud services

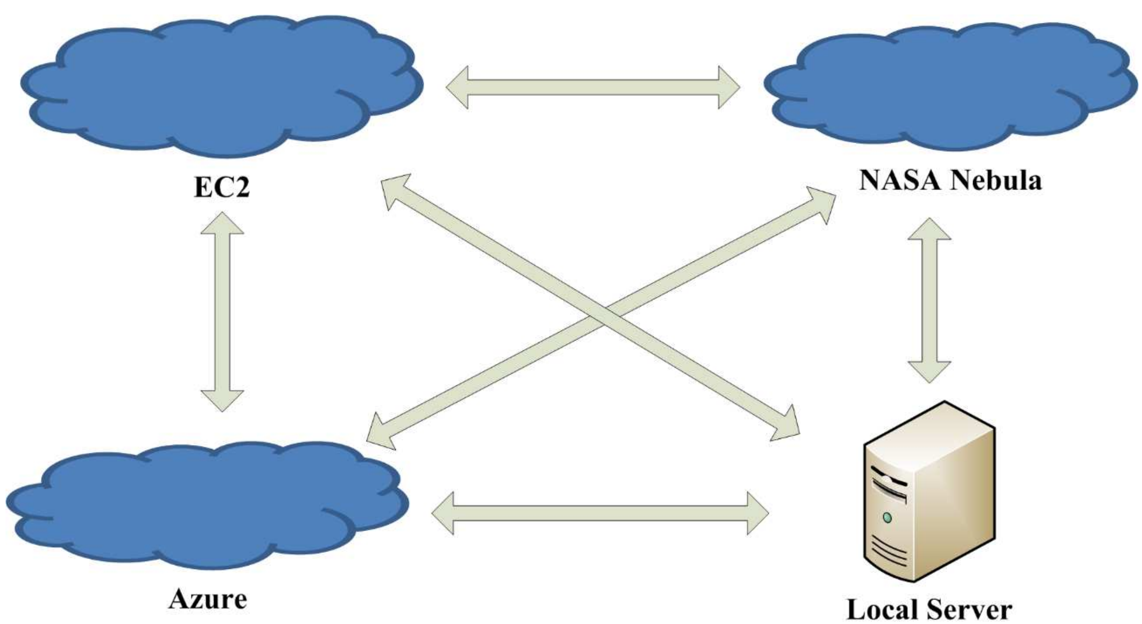


Figure 24 Test scenarios across cloud services

This set of experiments compare the performance of three cloud services to a traditional server in supporting CLH (Figure 24). CLH instances and clients are installed and configured on VMs in different cloud services and the traditional server. Different numbers of concurrent spatial query requests are sent from clients to CLH instances in different cloud services. In order to reduce the impact of noise and random errors in dynamic network environment, a repeat rule was adopted to repeat the following sets of experiments 50 times to calculate average performance results (e.g. average response time).

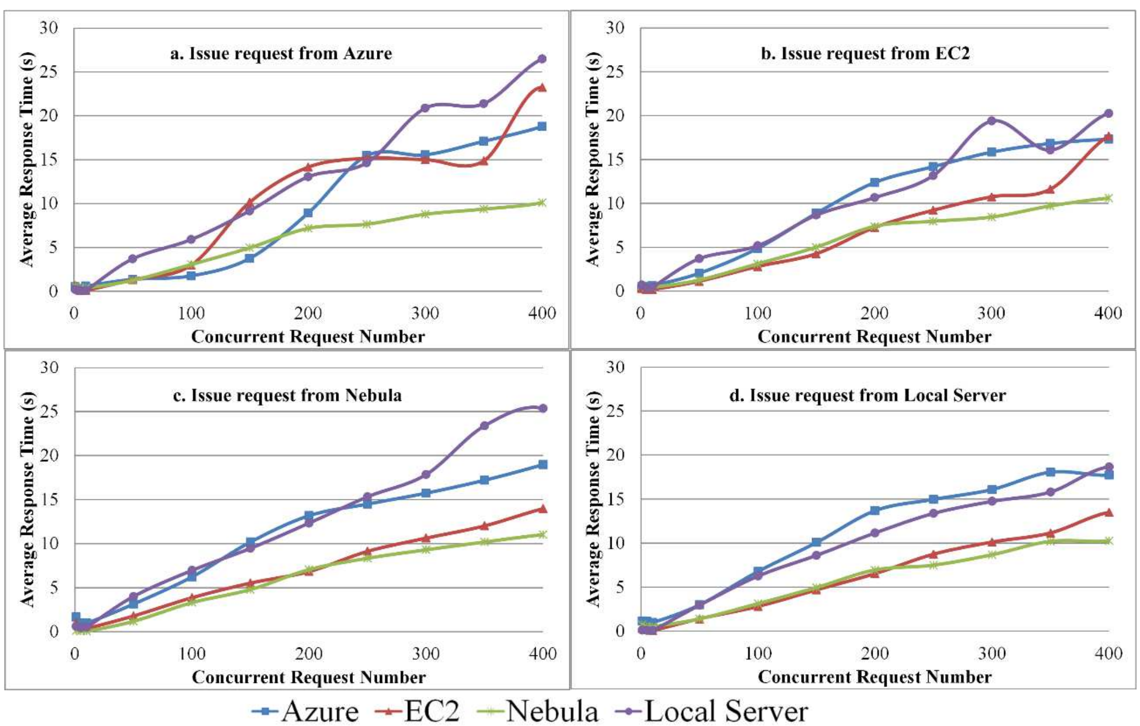


Figure 25 CLH average response time on cloud and traditional server

The performance results demonstrate (Figure 25): (1) Better hardware and network configuration usually yields a shorter response time. NASA Nebula provided the best performance because it had the best computing configuration and a stable network environment (private cloud). EC2 had the second best performance, followed by Azure and the traditional server had the worst performance; (2) In general, cloud services and the traditional server provided similar performances in supporting the CLH. Although the results were affected by different hardware configurations, no significant overheads were found from the virtualized computing resources in supporting the CLH; (3) the distribution of end user requests and cloud services impacts CLH performance. In general, closer distance (geographic and network between users and services) results in shorter response time. For example, it took about 24 seconds for a EC2-hosted CLH to respond to 400 concurrent requests (requested from Azure site) while it only took 18 seconds if requests were issued from EC2. In addition, passing request/response across long network distance usually results in performance fluctuation because of the dynamic network environment. For example, fluctuations are found by issuing requests from three cloud services to traditional server while fluctuation smoothens if issuing request from the traditional server to itself.

4.3.3 Dynamic cloud resource pooling

This set of experiments investigates the performance of cloud-enabled CLH in handling dynamic workloads. An auto-scaling rule is setted up for dynamic cloud resource pooling: (1) when the average response time is longer than 6 seconds, the CLH Elasticity Manager will automatically provision a new CLH instance; (2) In order to constrain the cost, we defined a maximum number of five instances that can be provisioned by Elasticity Manager.

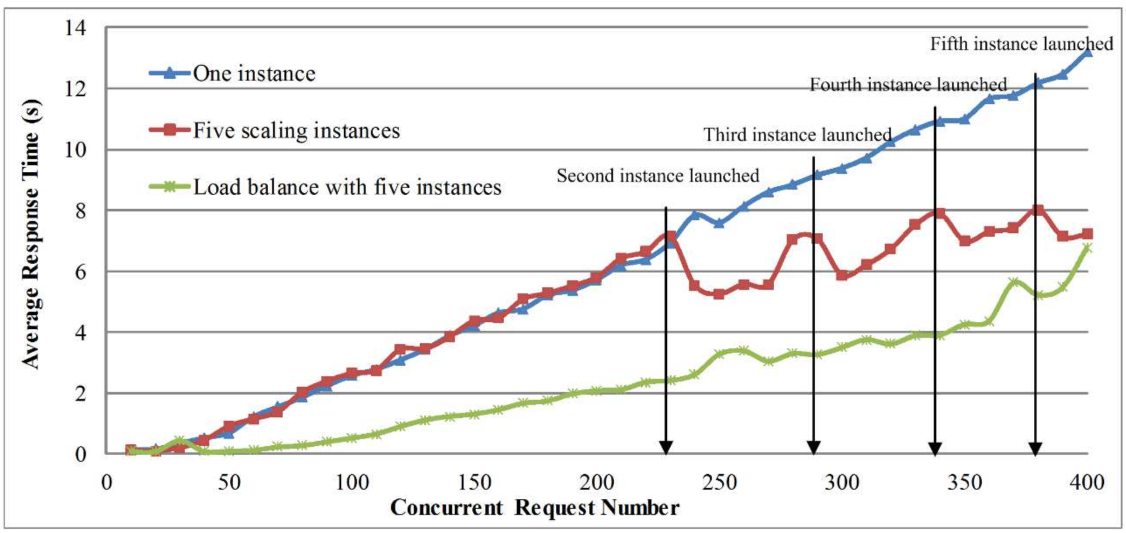


Figure 26 Elasticity Manager helps a CLH fast provision computing resources to handle dynamic workload

Figure 26 demonstrates the performance of auto-scaling and workload balancing enabled CLH in handling dynamic workload: (1) Elasticity Manager and auto-scaling mechanism help CLH quickly invoke more computing capabilities to the changing (increasing) workload. At the beginning of the test, only one CLH instance was utilized for concurrent access from 1 to 210 because CLH can respond to the end users within six seconds. When the concurrent access number was increased to 230, the Elasticity Manager utilized the second CLH instance because the response time exceeded the predefined threshold (6 seconds). Two CLH instances in the Cloud Workload Balancer worked together to balance the workload and it reduced the response time to about 5 seconds. Based on similar mechanism, the Elasticity Manager utilized the 3^{rd}, 4^{th} and 5^{th} CLH instance when the concurrent access number reached 280, 330, and 380. (2) Auto-scaling enables CLH to achieve a good balance between handling computing intensity and saving cloud service cost. With a number of 400 concurrent requests, the response time of auto-scaling enabled CLH was significantly reduced to 7 seconds compared with one-instance CLH (14 seconds). In addition, auto-scaling-enabled CLH was much cost-efficient compared to load balancing-enabled CLH that utilized redundant computing resource (five instances) even with light workload. Because of the efficiency of the auto-scaling mechanism, auto-scaling enabled CLH are usually used to handle dynamic or spiking access in a region. While load-balancing is usually used to better support routine requests from different geo-locations of the world.

4.3.4 Global cloud deployment for handling global access

This set of experiments investigates the performance of cloud enabled CLH in supporting global user access by adopting global deployment with the following workflow:

(1) Based on the spatial distributions of end users, launching four CLH instances in

three continents with high access frequency (one from North America, two from Europe, and one from Asia).

(2) To simulate global user access, load performance test software (Apache JMeter) was installed on VMs in different continents and concurrent spatial query requests were sent to CLH in different continents.

(3) To monitor geospatial resource quality, VMs in different continents continuously communicate and record quality information from seven popular geospatial web services that located in different continents.

(4) The repeat rules were used to calculate average response times and resource quality information.

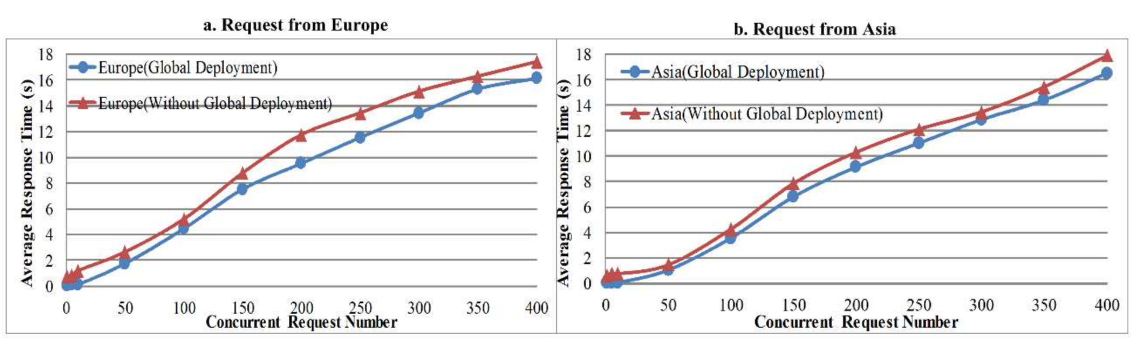


Figure 27 CLH average response time is reduced with the global deployment

Figure 27 demonstrates the performance of CLH response to global users with and without using cloud-enabled global deployment. With global deployment, the CLH achieved 0.7 to 2.2 seconds performance improvements for European users and 0.5 to 2.1

seconds improvements for Asian users with only one CLH instance. This improvement was contributed by the optimized distribution of CLH server based on the distribution of the end users. For example, a CLH instance was deployed on Azure North Europe data centre to handle European user requests. However, these requests were forwarded to the CLH server in the United States without using the global deployment mechanism. In addition, global-deployment-enabled CLH could achieve further performance improvement as more instances are utilized with the workload balancing or auto-scaling mechanism.

CHAPTER 5 DISCUSSION

5.1 Spatiotemporal index applicability

As demonstrated in the result section, the proposed indexing mechism requires certain spatotemporal pattern to achieve data access performance gain. To delimit the applicability of the APR-Tree, I investigate the distribution of the APR-Tree performance with different users' query patterns and P values. 470k features with uneven distribution are used to produce results (Figure 28). A query pattern is described by the combination of x and y values where the x-axis presents the percentage of data that is intensively accessed, and the y-axis presents the percentage of queries against the intensively accessed data. For example, when 50% of user queries intensively access 10% of the feature, the x value will be 10% and y value should be 50%; and when all users' query is evenly distributed, both x and y will be 100%.

Figure 28 shows that the APR-Tree is more effective when a large proportion of users' queries concentrate on a small proportion of features. With 470K features and uneven distribution, the APR-Tree (0.3 P value) brings up to 12.7% NAG when most user queries concentrate on 10% of the features. APR-Trees result in performance gain when P value increases from 0.1 to 0.4. When large proportion of user queries are more evenly distributed, the performance gain becomes less and more overheads will be added when p value increases from 0.4 to 0.9. This set of experiment demonstrates that a value of 0.2~ 0.4 is a good range for P value.

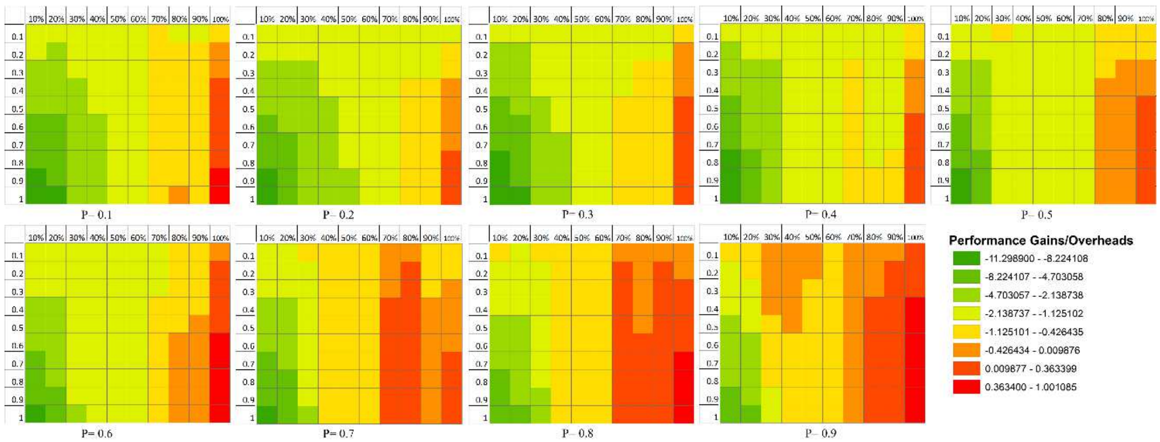


Figure 28. APR-Tree performance changes with different query patterns and P values

Experimental result shows that an APR-Tree brings about 9% to 22% performance gains over the R*-Tree depending on the query patterns, number of features and feature distribution under actual/uneven user query scenarios. This is a promising improvement considering that (1) the Hilbert R-tree, one of the most successful index in arranging spatial objects (Haverkort and Walderveen 2008), brings up to 14.28% performance gain on an R*-Tree for rectangle only dataset (Kamel and Faloutsos 1993); (2) the R*-Tree, one of the most popular R-tree variant (Kamel and Faloutsos 1993), brings about 16.4 % to 43.3% performance gains on a regular R-tree (quadratic algorithm) for intersection queries (Beckmann et al. 1990). However, the APR-Tree performance largely depends on user query patterns. The APR-Tree tends to achieve good performance with unevenly distributed queries while contributing to overheads with evenly distributed queries. The user behavior investigation shows that user queries are heterogeneous and unevenly distributed across space and time, which indicates that the APR-Tree tends to achieve better performances in actual scenarios. From GEOSS operation perspective, the PMIM helps the CLH achieve better query performance while the maintenance cost also increases. The net impact of the PMIM is highly related to the index usage and updates frequency, and should be evaluated in the index life cycle.

5.2 Spatiotemporal service performance model

5.2.1 Comparison of different monitoring mechanisms

The characteristics of traditional, cloud-based, and volunteer-based monitoring mechanisms are summarized in Table 3.

Table 3. Comparison of traditional, cloud-based, and volunteer-based monitoring mechanisms

	Traditional	Cloud	Volunteer
Distribution	Low	Medium	High
Controllability	High	Medium	Low
Frequency	High	High	Low
Heterogeneity	Low	Medium	High
Efficiency	Low	High	Low
Cost	High	Medium	Low

Distribution: distribution of the monitoring sites for traditional mechanism is limited to one location. Cloud services usually provide a number of data centers (cloud regions) globally. Volunteer computing provides wider distribution. For example, a cloud region (e.g., East Asia) may have one cloud data center but house hundreds of volunteers over different geographic locations.

Controllability: traditional mechanism provides full control of the monitoring process. The configurations (e.g., O/S, networking, server location) can be controlled with available VM type in cloud service, while most configurations cannot be controlled in volunteer-based performance monitoring.

Frequency: monitoring frequency for volunteer is low (currently 12 hours per monitoring cycle) because high monitoring frequency has a negative impact on the enthusiasm of volunteers. The frequency can be increased to < 2 hours with traditional mechanism and cloud service.

Efficiency: efficiency measures how fast the monitoring process can be started. The efficiency is slow for traditional server-based monitoring because it is time-consuming to purchase, host, and configure the traditional computing infrastructure. It is also a slow process for volunteer-based monitoring to recruit globally distributed volunteers. Conversely, cloud services support the computing resource provision (usually a VM) in minutes.

Cost: traditional and cloud-based monitoring mechanisms are expensive for purchasing servers and cloud services. Volunteer-based mechanism is more cost efficient because computing resources are provided by volunteers.

Since the three monitoring mechanisms have their own advantages and disadvantages, a hybrid monitoring mechanism is recommended with the following attributes: (1) volunteers as major monitors to collect distributed performance information with low cost; (2) cloud services for high frequency monitoring (e.g., once per hour) and used in regions where volunteers are minimal; and (3) traditional monitors for specific monitoring tasks (e.g., urgent earthquake-related service monitoring) since they have the highest controllability.

5.2.2 Comparison of different performance evaluation approaches

Table 4. Comparison of neighbouring and interpolation models

	Pro	Con
Neighbouring	Effective and accurate if a monitoring site is nearby	Inaccurate if no monitoring site nearby
Interpolation	Great accuracy with high density monitoring site nearby	Not suitable for global interpolation, not suitable for future prediction

Table 4 compares the neighbouring and interpolation models. The validation results showed that **the neighboring approach** provides decent evaluation accuracy (< 0.08 relative residual) if the prediction location is a nearby monitor. With increasing number of volunteers and cloud services in the global monitoring system, further accuracy improvement of this approach is expected in the near future. In addition, the evaluation process of the neighboring approach is efficient due to the fast retrieval of the performance values from the closest monitoring site, which only consumes a few computing resources. Therefore, a global SDIs (e.g. Geospatial-One-Stop, Geospatial platform, GEOSS Common Infrastructure, Digital Earth Systems) is proposed to adopt the neighboring approach for improving the evaluation accuracy. **The interpolation approach** provided significant accuracy improvements (about 13%) for evaluation in regions with a high density of monitoring sites. However, the evaluation accuracy decreased significantly for global and regional predictions with few monitoring sites.

5.3 Spatiotemporal cloud computing

Table 5 Key cloud features for addressing challenges

	Elastically Resource Pooling	Workload Balancing	Global Deployment
Computing intensity	X		
Concurrent intensity	X	X	
End user spatiotemporal distribution	X	X	X

The key features in the cloud-based framework for addressing Big EO Data discovery challenges are summarized in Table 5:

- **Computing Intensity Handling:**

To better handle computing intensity, the elastically resource pooling feature was implemented by the Elasticity Manager to enable the CLH (1) to acquire or release sufficient computing resources for intensive processing and (2) to adapt the computing resources to the dynamic workloads (e.g. a spiking workload).

- **Concurrent Intensity Handling:**

To better address massive concurrent user accesses, the elastically resource pooling feature first acquires sufficient number of CLH instances. Then massive concurrent workloads were dynamically scheduled to different CLH instances by the Cloud Workload Balancer. The Cloud-enabled workload balancing mechanism significantly reduces the response time: up to 16 seconds performance gain for 400 concurrent requests by using five CLH instances for workload balancing.

- **End User Spatiotemporal Distribution**

The cloud-enable global deployment approach helps CLH reduce physical and network distances between end users and CLH instances so that better system performance can be achieved. The CLH can achieve up to 2.2 seconds performance gain for Asian and European users using one CLH instance.

CHAPTER 6 CONSLUSION AND FUTURE RESEARCH

6.1 Conclusion

This dissertation reviewed relevant problems and suggested new approaches to optimize Big EO Data access through exploring and utilizing spatiotemporal patterns. Table 6 summaries the key methods for addressing the Big EO Data access challenges using the CLH as an example.

Table 6 Key methods for addressing Big EO Data access challenges

	Big EO Data Access	Accurate Results	Spatiotemporal Dynamics
Spatiotemporal Index	X		X
Spatiotemporal Service Model		X	X
Cloud Computing Adoption		X	X

- Big EO Data access

Various geographical studies and EO data discovery require intensive and timely accesses to Big Data. However, the volume and complexity of spatiotemporal nature of the data make it very difficult to provide fast access to the data. One of the most widely used approaches to speed up the access is to build good index. The proposed spatiotemporal index structure optimized existing index structure by integrating the spatiotemporal access patterns of global users. The experiment result showed that the new spatiotemporal index yields 9-20% performance gains for the data access compared to a classic R*-tree index.

- Accurate results

Providing accurate service performance result is critical for EO data discovery by helping user choose the "right" service for data access, distributed geo-processing and model composition. However, the service performance evaluation accuracy is limited with traditional single-location-based models because service performance is dynamically changing at different locations and times. The proposed spatiotemporal service model evaluates the performance based on end user space-time location and yields 3-18% accuracy improvements gains compared to that of a traditional model. The model was built based on the spatiotemporal characteristics of 3+ million service performance records using cloud computing based monitor and volunteer based monitor.

- Spatiotemporal dynamic

The EO data/service, end users, computing workloads and related problems are widely distributed at different regions and different times. However, the traditional computing infrastructure fails to effectively adapt computing resources in space and time, which resulting in poor user experience in some regions (e.g., high access density regions). The proposed spatiotemporal cloud computing framework explores the feasibility of adopting cloud computing for CLH hosting. The experimental result shows that the cloud-based framework can fast deliver computing resource to different regions and better handle global and spiking user accesses . In addition, the proposed spatiotemporal index optimizes global EO data access by customizing different indices for users in different regions. Finally, the proposed spatiotemporal service model improves global user experience with better service evaluation accuracy according the end user space-time locations.

6.2 Future research

This dissertation proposed a variety of spatiotemporal optimizations for Big EO Data discovery. The research results demonstrate a great potential of spatiotemporal optimizations in addressing various geographical challenges. More research would be required to further enhance the applicability of the methodologies used in this dissertation in the next decade including the following aspects:

- Distributed EO data indexing

A distributed indexing mechanism utilizes multiple computers to support data index process. The proposed spatiotemporal indexing strategy could be extended to R-Tree-based distributed indexing structures so that distributed cloud computing resource can be utilized. In addition, user spatiotemporal patterns could be further utilized in the allocation of tree-node to different computers and the distribution of computers to different regions.

- Proactive user behavior prediction

User behavior pattern is an essential element for the proposed spatiotemporal optimization methods. The current user behavior data are collected and analyzed from the CLH log file so that computing resource, index and service evaluation can be delivered according to users' need. However, the current reactive pattern analaysis approach may result in significant delay for the optimizations because the system have to react after user pattern is discovered. A proactive user behavior prediction approach is desired to predict and apply optimizations before the issues (e.g. spiking access).

- Spatiotemporal semantic search

The search accuracy is another important element for the CLH to deliver accucate result to end users. Currently, most EO data search engines adopt keyword-based search, which often results poor query accuracy. Keyword-based matching is hard to guarantee the recall and precision because the search content usually cannot be expressed with several keywords explicitly. A semantic embedded search engines is desired to improve the discovery recall and precision for end users from different disciplines. Similar to service performance, the human knowledge structure also changes in space and time. Therefore, to further improve GEOSS global operation, it is necessary to extend the exiting semantic search to spatiotemporal semantic search engine.

REFERENCES

REFERENCES

1. Achakeev, D., Seeger, B., & Widmayer, P. 2012. Sort-based query-adaptive loading of R-trees. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, New York, NY, USA, 2080–2084 doi:10.1145/2396761.2398577

2. Agrawal, Rakesh, Tomasz Imieliński, and Arun Swami. 1993. Mining association rules between sets of items in large databases. ACM SIGMOD Record. Vol. 22. No. 2.

3. Amazon, 2013. (http://aws.amazon.com/solutions/case-studies/)

4. Armbrust, M., et al., 2010. Above the clouds: a Berkeley view of cloud computing. Communications of the ACM, 53 (4), pp.5058.

5. Azure, 2013. (http://www.windowsazure.com/en-us/overview/case-studies).

6. Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, and Warfield, A., 2003. Xen and the art of virtualization. In: ACM SIGOPS Operating Systems Review, 37(5), pp. 164-177.

7. Beckmann, N., Kriegel, H.-P., Schneider, R., & Seeger, B. 1990. The R*-tree: an efficient and robust access method for points and rectangles. In Proceedings of the 1990 ACM SIGMOD international conference on Management of data, New York, NY, USA, 322–331. doi:10.1145/93597.98741

8. Bernard, L., Kanellopoulos, I., Annoni, A., & Smits, P. 2005. The European geoportal—one step towards the establishment of a European Spatial Data Infrastructure. Computers, environment and urban systems, 29(1), 15-31.

9. Brunsdon, Chris, Stewart Fotheringham, and Martin Charlton. 2002. Geographically weighted regression. Journal of the Royal Statistical Society: Series D (The Statistician) 47.3 : 431-443.

10. Bunt, R. B., Eager, D. L., Oster, G. M., & Williamson, C. L. 1999. Achieving load balance and effective caching in clustered Web servers. In Proceedings of the 4th International Web Caching Work, 159–169.

11. Bürger, C. M., Kollet, S., Schumacher, J., & Bösel, D., 2012. Introduction of a web service for cloud computing with the integrated hydrologic simulation platform ParFlow. Computers & Geosciences, 48, pp. 334-336.

12. Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., and Brandic, I., 2009. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation Computer Systems, 25(6), pp. 599–616. doi:10.1016/j.future.2008.12.001

13. Cardellini, V., Colajanni, M., & Yu, P. S. 1999. Dynamic load balancing on web-server systems. Internet Computing, IEEE, 3(3), 28-39.

14. Chang, F., Dean, J., Ghemawat, S., Hsieh, W. C., Wallach, D. A., Burrows, M., Chandra, T., Fikes, A., and Gruber, R. E. 2008. Bigtable: A distributed storage system for structured data. ACM Transactions on Computer Systems (TOCS), 26(2), pp. 4.

15. Curbera, F., Leymann, F., Storey, T., Ferguson, D., & Weerawarana, S. 2005. Web services platform architecture: SOAP, WSDL, WS-policy, WS-addressing, WS-BPEL, WS-reliable messaging and more. Englewood Cliffs: Prentice Hall PTR.

16. De Smith MJ, Goodchild MF, Longley P. 2007. Geospatial Analysis: A Comprehensive Guide to Principles (Troubador Publishing Ltd., Leicester, UK).

17. Dean, J., and Ghemawat, S., 2008. MapReduce: simplified data processing on large clusters. Communications of the ACM, 51(1), pp. 107-113.

18. Erl, T. 2005. Service-oriented architecture (Vol. 8). New York: Prentice Hall.

19. Ester and Martin. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. Proceedings of the 2nd International Conference on Knowledge Discovery and Data mining.

20. Evangelinos, Constantinos, and C. Hill., 2008. Cloud Computing for parallel Scientific HPC Applications: Feasibility of running Coupled Atmosphere-Ocean Climate Models on Amazon's EC2. Ratio, 2.2.40: pp. 2-34.

21. FGDC Checker, 2013 http://registry.fgdc.gov/statuschecker/

22. FGDC, 2013 https://www.fgdc.gov/initiatives/geospatial-platform

23. García Martín, R., de Castro Fernández, J. P., Verdú Pérez, E., Verdú Pérez, M. J., & Regueras Santos, L. M. 2013. An OLS regression model for context-aware tile prefetching in a web map cache. International Journal of Geographical Information Science, 27(3), 614–632. doi:10.1080/13658816.2012.721555

24. Garfinkel, S., 2007. An evaluation of Amazons grid computing services: EC2, S3 and SQS. Harvard University, Technical Report TR-08-07, Harvard University.

25. GEO, 2013 http://www.earthobservations.org/index.shtml

26. GEOSS, 2013 http://www.earthobservations.org/geoss.shtml

27. Goodchild MF, Yuan M, Cova TJ. 2007. Towards a general theory of geographic representation in GIS. Int J Geogr Inf Sci 21:239–260.

28. Goodchild, M. F., Fu, P., & Rich, P. 2007. Sharing geographic information: an assessment of the Geospatial One-Stop. Annals of the Association of American Geographers, 97(2), 250-266.

29. Greene, S., Tanin, E., Plaisant, C., Shneiderman, B., Olsen, L., Major, G., & Johns, S. 1999. The end of zero-hit queries: Query previews for NASA's global change master directory. International Journal on Digital Libraries, 2(2-3), 79-90.

30. Guan, H., Li, J., Zhong, L., Yongtao, Y., & Chapman, M., 2013. Process virtualization of large-scale lidar data in a cloud computing environment. Computers & Geosciences. 60, pp. 109-116

31. Gui, Z., Yang, C., Xia, J., Liu, K., Xu, C., Li, J., & Lostritto, P. 2013. A performance, semantic and service quality-enhanced distributed search engine for improving geospatial resource discovery. International Journal of Geographical Information Science, 27(6), 1109-1132.

32. Güting, R. H. 1994. An introduction to spatial database systems. The VLDB Journal, 3(4), 357–399.

33. Guttman, A. 1984. R-trees: A Dynamic Index Structure for Spatial Searching. In Proceedings of the ACM SIGMOD international conference on Management of data, Boston, Massachusetts, USA, 47–57.

34. Haverkort, H., & Walderveen, F. V. 2008. Four-dimensional hilbert curves for R-trees. *Journal of Experimental Algorithmics (JEA)*, *16*(3.4). doi:10.1145/1963190.2025380

35. Horvitz, E., Breese, J., Heckerman, D., Hovel, D., and Rommelse, K. 1998, The Lumiere Project: Bayesian user modeling for inferring the goals and needs of software users. In Proceedings of the Fourteenth Conference on Uncertainty in artifiical Intelligence, Madison, WI, USA, 256-265.

36. Howe, D., Costanzo, M., Fey, P., Gojobori, T., Hannick, L., Hide, W.,& Rhee, S. Y. 2008. Big data: The future of biocuration. Nature, 455(7209), 47-50.

37. Huang, C.-Y., & Liang, S. H. L. 2012. LOST-Tree: a spatio-temporal structure for efficient sensor data loading in a sensor web browser. International Journal of Geographical Information Science, 27(6), 1190-1209. doi:10.1080/13658816.2012.743663

38. Huang, H., Li, Y., Gartner, G., & Wang, Y. 2011. An SVG-based method to support spatial analysis in XML/GML/SVG-based WebGIS. International Journal of Geographical Information Science, 25(10), 1561–1574. doi:10.1080/13658816.2010.532133

39. Huang, Q., Yang, C., Benedict, K., Rezgui, A., Xie, J., Xia, J., & Chen, X. 2013. Using adaptively coupled models and high-performance computing for enabling the computability of dust storm forecasting. International Journal of Geographical Information Science, 27(4), 765–784. doi:10.1080/13658816.2012.715650

40. IGRASS'08, 6–11 July 2008. Boston, MA, USA, II-1278–1281.

41. Informix. 2003. IBM Informix R-tree Index User's Guide. IBM. Retrieved from www.informix.com.ua/doc/9.40/ct1tana.pdf [Accessed 14 June 2013]

42. Iosup, A., Ostermann, S., Yigitbasi, M.N., Prodan, R., Fahringer, T., Epema, D.H.J., 2011. Performance analysis of cloud computing services for many-tasks scientific computing. In: IEEE Transaction on Parallel and Distributed Systems 22(6), pp. 931-945.

43. Jackson, D. J., & Humphres, C. W. 1997. A simple yet effective load balancing extension to the PVM software system. Parallel Computing, 22(12), 1647-1660.

44. Kamel, I., & Faloutsos, C. 1993. Hilbert R-Tree: An Improved R-Tree Using Fractals. In Proceedings of the 20th International Conference on Very Large Databases, Santiago de, Chile, 500–509.

45. Kang, Y.-K., Kim, K.-C., & Kim, Y.-S. 2001. Probability-Based Tile Pre-fetching and Cache Replacement Algorithms for Web Geographical Information Systems. In Proceedings of the 5th East European Conference on Advances in Databases and Information Systems, London, UK, 127–140.

46. Kanungo and Tapas. 2002. An efficient k-means clustering algorithm: Analysis and Implementation. Pattern Analysis and Machine Intelligence, IEEE transactions on 24.7, pages 881–892.

47. Kim, W., Kim, S. D., Lee, E., and Lee, S. 2009. Adoption issues for cloud computing. In: Proceedings of the 7th International Conference on Advances in Mobile Computing and Multimedia, pp. 2-5.

48. Klemm R. 2006. Principles of Space-Time Adaptive Processing (Institute of Engineering and Technology, London, UK).

49. Korpela, E. J. 2012. SETI@ home, BOINC, and volunteer distributed computing. Annual Review of Earth and Planetary Sciences, 40, 69-87.

50. Kothuri, R. K. V., Ravada, S., & Abugov, D. 2002. Quadtree and R-tree indexes in oracle spatial: a comparison using GIS data. In Proceedings of the 2002 ACM SIGMOD international conference on Management of data, New York, NY, USA, 546–557. doi:10.1145/564691.564755

51. Kothuri, R., Hanckel, R., & Yalamanchi, A. 2008. Using Oracle Extensibility Framework for Supporting Temporal and Spatio-Temporal Applications. In 15th International Symposium on Temporal Representation and Reasoning. Montréal, Canada.

52. Kowalski A, 2013. Bonnie file system benchmarks. Jefferson Lab Report[Online] Available: http://cc.jlab.org/docs/scicomp/benchmark/bonnie.html.

53. Lee, D., and Liang, S. H., 2011. Geopot: a Cloud-based geolocation data service for mobile applications. International Journal of Geographical Information Science, 25(8), pp. 1283-1301.

54. Lee, Y. T., & Chen, K. T., 2010. Is server consolidation beneficial to MMORPG? A case study of World of Warcraft. In: Proceedings of 3rd International Conference on Cloud Computing, pp. 435-442.

55. Li, Q., Zhang, T., and Yu, Y., 2011. Using cloud computing to process intensive floating car data for urban traffic surveillance. International Journal of Geographical Information Science, 25(8), pp. 1303-1322.

56. Li, R., Guo R., Xu Z., & Feng W. 2012 A Prefetching Model Based on Access Popularity for Geospatial Data in a Cluster-based Caching System. International Journal of Geographical Information Science 26(10), 1831–1844.

57. Li, X., Zhang, X., Yeh, A., & Liu, X. 2010. Parallel cellular automata for large-scale urban simulation using load-balancing techniques. International Journal of Geographical Information Science, 24(6), 803-820.

58. Li, Z., et al., 2011. An optimized framework for seamlessly integrating OGC Web services to support geospatial sciences. International Journal of Geographical Information Science, 25 (4), 595–613.

59. Li,W. and Yang, C., 2008. A semantic search engine for spatial web portals. In: Proceedings of IEEE

60. Liu, K., Yang, C., Li, W., Li, Z., Wu, H., Rezgui, A., & Xia, J. 2011. The GEOSS clearinghouse high performance search engine. In Proceedings of 2011 19th International Conference on Geoinformatics, 1–4.

61. Longley P, Goodchild MF, Maguire DJ, Rhind DW. 2005. Geographical Information Systems and Science (Wiley, San Francisco, CA) p 497

62. Lucene, 2013 http://lucene.apache.org/

63. Luszczek, P. R., Bailey, D. H., Dongarra, J. J., Kepner, J., Lucas, R. F., Rabenseifner, R., and Takahashi, D., 2006. The HPC Challenge (HPCC) benchmark suite. In: Proceedings of the ACM/IEEE conference on Supercomputing, pp. 213.

64. Maguire, D. J., and Longley, P. A., 2005. The emergence of geoportals and their role in spatial data infrastructures. Computers, environment and urban systems, 29(1), pp. 3-14.

65. Mani, A., & Nagarajan, A. 2005. Understanding quality of service for Web services.

66. Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., & Byers, A. H. 2011. Big data: The next frontier for innovation, competition, and productivity.

67. Marston, S., Li, Z., Bandyopadhyay, S., Zhang, J., & Ghalsasi, A. 2011. Cloud computing—The business perspective. Decision Support Systems, 51(1), pp. 176-189.

68. Masser, I. 2005. GIS worlds: creating spatial data infrastructures (Vol. 338). Redlands: ESRI press.

69. Mell P. and Grance T. 2009. The NIST Definition of Cloud Computing, http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf, Last accessed Jul.15, 2013.

70. Miled, Z. B., Sikkupparbathyam, S., Bukhres, O., Nagendra, K., Lynch, E., Areal, M., and Savage, N. 2001. Global change master directory: object-oriented active asynchronous transaction management in a federated environment using data agents. In Proceedings of the 2001 ACM symposium on Applied computing (pp. 207-214). ACM.

71. Moorsel, A. 2001. Metrics for the internet age: Quality of experience and quality of business. In Fifth International Workshop on Performability Modeling of Computer and Communication Systems, Arbeitsberichte des Instituts für Informatik, Universität Erlangen-Nürnberg, Germany (Vol. 34, No. 13, pp. 26-31).

72. Mountrakis, G., & Gunson, K. 2009. Multi-scale spatiotemporal analyses of moose–vehicle collisions: a case study in northern Vermont. International Journal of Geographical Information Science, 23(11), 1389–1412. doi:10.1080/13658810802406132

73. Nah, F. F.-H. 2004. A study on tolerable waiting time: how long are Web users willing to wait? *Behaviour & Information Technology*, *23*(3), 153–163. doi:10.1080/01449290410001669914

74. NASA GCMD, 2013 (http://gcmd.nasa.gov/learn/index.html)

75. National LamdaRail, 2013 (http://www.nlr.net/)

76. Nebert, D., Whiteside, A., & Vretanos, P. 2007. OpenGIS catalog services specification, Version 2.0.2. OGC. Retrieved from https://portal.opengeospatial.org/modules/admin/license_agreement.php?SuppressHeaders=0&access_license_id=3&target=http://portal.opengeospatial.org/files/%3fartifact_id=20555 [Accessed 14 June 2013]

77. Olfat, H., Rajabifard, A., & Kalantari, M. 2010. A synchronisation approach to automate spatial metadata updating process. Coordinates Magazine, VI (3), 27-32.

78. Ostermann, S., Iosup, A., Yigitbasi, N., Prodan, R., Fahringer, T., and Epema, D, 2010. A performance analysis of EC2 cloud computing services for scientific computing. In: ICST International Conference on Cloud Computing. pp. 115-131.

79. Pacheco, P. S. 1997. Parallel programming with MPI. Morgan Kaufmann.

80. Pearlman, J., Craglia, M., Bertrand, F., Nativi, S., Gaigalas, G., Dubois, G., ... & Fritz, S. 2011. EuroGEOSS: an interdisciplinary approach to research and applications for forestry, biodiversity and drought. In Proceedings of the 34th international symposium on remote sensing of environment (pp. 1-4).

81. PostGIS. 2013. PostGIS 2.0.0 Manual. PostGIS. Retrieved from http://postgis.refractions.net/documentation/manual-2.0 [Accessed 14 June 2013]

82. Ramachandran, R., et al., 2006. Ontology-based semantic search tool for atmospheric science. In: 22nd International conference on interactive information processing systems for meteorology, oceanography, and hydrology

83. Ramsey, N., & Csirmaz, E. 2001. An algebraic approach to file synchronization. In ACM SIGSOFT Software Engineering Notes (Vol. 26, No. 5, pp. 175-185).

84. Raskin, R. G., & Pan, M. J. 2005. Knowledge representation in the semantic web for Earth and environmental terminology (SWEET). Computers & Geosciences, 31(9), 1119-1125.

85. Reed, G., Keeley, R., Belov, S., & Mikhailov, N., 2010. Ocean Data Portal: a standards approach to data access and dissemination. In: Proceedings of the OceanObs.

86. Roman, D., Keller, U., Lausen, H., de Bruijn, J., Lara, R., Stollberg, M., ... & Fensel, D. 2005. Web service modeling ontology. Applied ontology, 1(1), 77-106.

87. Sedda, L., Atkinson, P. M., Filigheddu, M. R., Cotzia, G., & Dettori, S. 2011. Spatio-temporal analysis of tree height in a young cork oak plantation. International Journal of Geographical Information Science, 25(7), 1083–1096. doi:10.1080/13658816.2010.517534

88. Sellis, T., Roussopoulos, N., & Faloutsos, C. 1987. In Proceedings of the 13th International Conference on Very Large Data Bases, Brighton, England, 507-518.

89. Şen Z. 2009. Spatial Modeling Principles in Earth Sciences (Springer, New York).

90. Tobler, W. R. 1970. A computer movie simulating urban growth in the Detroit region. Economic geography, 46, 234-240.

91. USGS, 2011. Earthquake Hazards Program [Online]. Available from: http://earthquake.usgs.gov/earthquakes/eqinthenews/2011/usb0006bqc [Accessed 14 June 2013].

92. VisIt, 2013. (https://wci.llnl.gov/codes/visit/).

93. Westbrooks, E. L. 2003. Distributing and synchronizing heterogeneous metadata for the management of geospatial information repositories. In International Conference on Dublin Core and Metadata Applications. pp-67.

94. Whitehouse. 2012. Obama administration unveils "BIG DATA" initiative: announces $200 million in new R&D investments. Whitehouse. Retrieved from http://www.whitehouse.gov/sites/default/files/microsites/ostp/big_data_press_release_final_2.pdf [Accessed 14 June 2013]

95. Worboys and Duckham. 2004. GIS: A Computing Perspective, Second Edition, CRC Press

96. Xia, J., Yang, C., Gui, Z., Liu, K., & Li, Z. 2014a. Optimizing an index with spatiotemporal patterns to support GEOSS Clearinghouse. International Journal of Geographical Information Science, 28(7), 1459-1481.

97. Xia, J., Yang, C., Liu, K., Gui, Z., Li, Z., Huang, Q., & Li, R. 2014b. Adopting cloud computing to optimize spatial web portals for better performance to support Digital Earth and other global geospatial initiatives. International Journal of Digital Earth. DOI:10.1080/17538947.2014.929750.

98. Xia, J., Yang, C., Liu, K., Li, Z., Sun, M., & Yu, M. 2015. Developing a spatiotemporal model to forecasting global performance of GISciences. International Journal of Geographical Information Science. DOI: 10.1080/13658816.2014.968783

99. Yang, C., Goodchild, M., Huang, Q., Nebert, D., Raskin, R., Xu, Y., & Fay, D. 2011a. Spatial cloud computing: how can the geospatial sciences use and help shape cloud computing? *International Journal of Digital Earth*, *4*(4), 305–329. doi:10.1080/17538947.2011.587547

100. Yang, C., Huang, Q., Li, Z., Xu, C., and Liu, K., 2014. Spatial Cloud Computing: A Practical Approach. Taylor & Francis. ISBN: 978-1-4665-9316-9.

101. Yang, C., Wu, H., Huang, Q., Li, Z., and Li, J., 2011b. Using spatial principles to optimize distributed computing for enabling the physical science discoveries. In: Proceedings of the National Academy of Sciences, 108(14), pp. 5498-5503.

102. Yang, P., Cao, Y., & Evans, J. 2007. Web map server performance and client design principles. GIScience & Remote Sensing, 44(4), 320-333.

103. Zeng, L., Benatallah, B., Dumas, M., Kalagnanam, J., & Sheng, Q. Z. 2003. Quality driven web services composition. In Proceedings of the 12th international conference on World Wide Web (pp. 411-421).

104. Zeng, X., & McMechan, G. A. 2002. Load balancing across a highly heterogeneous processor cluster using file status probes. Computers & geosciences, 28(8), 911-918.

105. Zhang, Q., Zhu, Q., Boutaba, R., 2011. Dynamic resource allocation for spot markets in cloud computing environments. In: Proceedings of Fourth IEEE International Conference on Utility and Cloud Computing, Victoria, NSW, 178-185.

106. Zheng, Y., Xie, X., & Ma, W. Y. 2010. GeoLife: A Collaborative Social Networking Service among User, Location and Trajectory. In *Proceedings of IEEE Data Engineering Bulletin*, 33(2), 32-39.

CURRICULUM VITAE

Jizhe Xia obtained his M.S. degree in Geographic and Cartographic Sciences from George Mason University in 2013 and B.S. degree in Geographic information system from the South China Nortaml University in 2010. In fall 2011, he was admitted as a full-time Ph.D. student in Earth System and Geoinformation Science, College of Science, George Mason University. His research is focused on spatiotemporal optimizations and spatial cloud computing for geosciences. He has published over 10 peer reviewed articles in various international journals and conferences.

Journal Articles

1. **Xia, J.**, Yang, C., Liu, K., Li, Z., Sun, M., & Yu, M. 2015. Developing a spatiotemporal model to forecasting global performance of GISciences. *International Journal of Geographical Information Science*. DOI: 10.1080/13658816.2014.968783
2. **Xia, J.**, Yang, C., Gui, Z., Liu, K., & Li, Z. 2014. Optimizing an index with spatiotemporal patterns to support GEOSS Clearinghouse. *International Journal of Geographical Information Science*, *28*(7), 1459-1481.
3. **Xia, J.**, Yang, C., Liu, K., Gui, Z., Li, Z., Huang, Q., & Li, R. 2014. Adopting cloud computing to optimize spatial web portals for better performance to support Digital Earth and other global geospatial initiatives. *International Journal of Digital Earth*. DOI:10.1080/17538947.2014.929750.
4. Li, Z., Yang, C., Huang, Q., Liu, K., Sun, M., & **Xia, J**. 2014. Building Model as a Service to support geosciences. *Computers, Environment and Urban Systems*. DOI: 10.1016/j.compenvurbsys.2014.06.004
5. Gui, Z., Yang, C., **Xia, J.**, Huang, Q., Liu, K., Li, Z., & Jin, B. 2014. A Service Brokering and Recommendation Mechanism for Better Selecting Cloud Services. *PLOS ONE*, *9*(8), e105297.
6. Li, Z., Yang, C., Liu, K., **Xia J.** 2014. Automatic Scaling Hadoop Cluster in the Cloud to Support Geoscience Data Processing. Geoinformatica (Submitted).
7. Gui, Z., Yu, M., Yang, C., Jiang, Y., Chen, S., **Xia, J.**, Huang, Q., Liu, K., & Li, Z. 2014. Developing New Scheduling Algorithms Based on Spatial Computing to Accelerate Dust Storm Simulation. *PLOS ONE*. (Submitted)
8. Huang Q., Yang C., Liu K., **Xia J**., Xu C., Li J., Gui Z., Sun M., Li Z.. 2013. Evaluating Open Source Cloud Computing Solutions for Geosciences, *Computers & Geosciences*, 59(9): 41-52.

9. Gui, Z., Yang, C., **Xia, J**., Li, J., Rezgui, A., Sun, M., Xu, Yang., & Fay, D. 2013. A visualization-enhanced graphical user interface for geospatial resource discovery. *Annals of GIS*, 19(2), 109-121.
10. Liu K., Yang C., Li W., Gui Z., Xu C., & **Xia J.** 2013. Using Semantic Search and Knowledge Reasoning to Improve the Discovery of Earth Science Records: An Example with the ESIP Semantic Testbed, *International Journal of Applied Geospatial Research*. DOI: 10.4018/ijagr.2014040104
11. Gui, Z., Yang, C., **Xia, J.**, Liu, K., Xu, C., Li, J., & Lostritto, P. 2013. A performance, semantic and service quality-enhanced distributed search engine for improving geospatial resource discovery. *International Journal of Geographical Information Science*, 27(6), 1109-1132.
12. Huang Q., Yang C., Benedict K., Rezgui A., Xie J., **Xia J.**, & Chen, S. 2013. Using Adaptively Coupled Models and High-performance Computing for Enabling the Computability of Dust Storm Forecasting, *International Journal of Geographic Information Science*, 27(4): 765-784.

Refereed Conference Proceedings

13. Rezgui, A., Malik, Z., **Xia, J.**, Liu, K., C, Yang. 2013. Data-intensive Spatial Indexing on the Clouds, *Procedia Computer Science*, 18: 2615-2618.
14. Yu, M., Yang, C., Huang, Q., Gui, Z., & **Xia, J**. 2013. Utilizing high spatiotemporal resolution soil moisture for dust storm modeling. *Second International Conference on Agro-Geoinformatics (Agro-Geoinformatics)*, pp. 176-181. IEEE.
15. Huang Q., **Xia J.**, Yang C., Liu K., Li J., Gui Z., Hassan M., Chen S., 2012. An Experimental Study of Open-Source Cloud Solutions for Dust Storm Forecasting, *ACM SIGSPATIAL GIS*. DOI: 10.1145/2424321.2424408.
16. Xu, C., Yang, C., Li, J., **Xia, J.**, Qu, X., Sun, M., Xu, Y., Fay, D., Bambacus, M., 2011. A service visualization tool for spatial web portal, *COM.Geo '11 Proceedings of the 2nd International Conference on Computing for Geospatial Research & Applications*, doi: 10.1145/1999320.1999351.
17. Liu, K., Yang, C., Li, W., Li, Z., Wu, H., Rezgui, A., **Xia, J.**, 2011. The GEOSS Clearinghouse High Performance Search Engine. *The 19th International Conference on Geoinformatics*, June 24-26, 2011, Shanghai, China.
18. Qu, X., Sun, M., Xu, C., Li, J., Liu, K., **Xia, J.**, & Fay, D.. 2011. A spatial web service client based on Microsoft Bing Maps. *The 19th International Conference on Geoinformatics*, June 24-26, 2011, Shanghai, China.