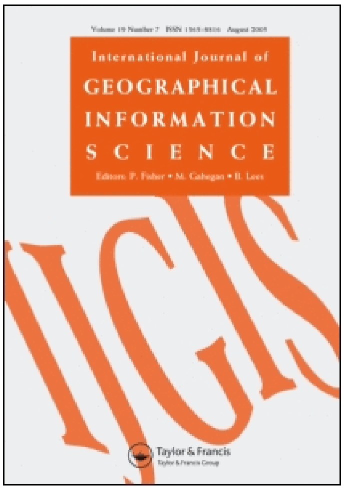
This article was downloaded by: *[San Diego State University]*
On: *27 July 2009*
Access details: *Access Details: [subscription number 906957039]*
Publisher *Taylor & Francis*
Informa Ltd Registered in England and Wales Registered Number: 1072954 Registered office: Mortimer House, 37-41 Mortimer Street, London W1T 3JH, UK



International Journal of Geographical Information Science
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713599799

Developing a grid-enabled spatial Web portal for Internet GIServices and geospatial cyberinfrastructure

Tong Zhang [a]; Ming-Hsiang Tsou [b]
[a] Transportation Research Center, State Key Laboratory of Information Engineering In Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan, Hubei, P.R. China [b] Department of Geography, San Diego State University, San Diego, CA, USA

Online Publication Date: 01 May 2009

To cite this Article Zhang, Tong and Tsou, Ming-Hsiang(2009)'Developing a grid-enabled spatial Web portal for Internet GIServices and geospatial cyberinfrastructure',International Journal of Geographical Information Science,23:5,605 — 630

To link to this Article: DOI: 10.1080/13658810802698571

URL: http://dx.doi.org/10.1080/13658810802698571

PLEASE SCROLL DOWN FOR ARTICLE

Full terms and conditions of use: http://www.informaworld.com/terms-and-conditions-of-access.pdf

This article may be used for research, teaching and private study purposes. Any substantial or systematic reproduction, re-distribution, re-selling, loan or sub-licensing, systematic supply or distribution in any form to anyone is expressly forbidden.

The publisher does not give any warranty express or implied or make any representation that the contents will be complete or accurate or up to date. The accuracy of any instructions, formulae and drug doses should be independently verified with primary sources. The publisher shall not be liable for any loss, actions, claims, proceedings, demand or costs or damages whatsoever or howsoever caused arising directly or indirectly in connection with or arising out of the use of this material.

International Journal of Geographical Information Science
Vol. 23, No. 5, May 2009, 605–630

Taylor & Francis
Taylor & Francis Group

Research Article

Developing a grid-enabled spatial Web portal for Internet GIServices and geospatial cyberinfrastructure

TONG ZHANG† and MING-HSIANG TSOU*‡
†Transportation Research Center, State Key Laboratory of Information Engineering In Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan, Hubei, P.R. China
‡Department of Geography, San Diego State University, San Diego, CA, USA

(*Received 27 December 2007; in final form 17 December 2008*)

Geospatial cyberinfrastructure integrates distributed geographic information processing (DGIP) technology, high-performance computing resources, interoperable Web services, and sharable geographic knowledge to facilitate the advancement of geographic information science (GIScience) research, geospatial technology, and geographic education. This article addresses three major development issues of geospatial cyberinfrastructure: the performance of grid-enabled DGIP services, the integration of Internet GIService resources, and the technical challenges of spatial Web portal implementation. A four-tier grid-enabled Internet GIService framework was designed for geospatial cyberinfrastructure. The advantages of the grid-enabled framework were demonstrated by a spatial Web portal. The spatial Web portal was implemented based on current available Internet technologies and utilizes multiple computing resources and high-performance systems, including local PC clusters and the TeraGrid. By comparing their performance testing results, we found that grid computing (TeraGrid) is more powerful and flexible than local PC clusters. However, job queuing time and relatively poor performance of cross-site computation are the major obstacles of grid computing for geospatial cyberinfrastructure. Detailed analysis of different computational settings and performance testing contributes to a deeper understanding of the improvements of DGIP services and geospatial cyberinfrastructure. This research demonstrates that resource/service integration and performance improvement can be accomplished by deploying the new four-tier grid-enabled Internet GIService framework. This article also identifies four research priorities for developing geospatial cyberinfrastructure: the design of GIS middleware, high-performance geovisualization methods, semantic GIService, and the integration of multiple GIS grid applications.

Keywords: Cyberinfrastructure; Internet GIServices; Grid computing; Web services; Web portals

1. Introduction: what is geospatial cyberinfrastructure?

By extending the original definition of cyberinfrastructure in the 2003 National Science Foundation (NSF) report *Revolutionizing Science and Engineering through Cyberinfrastructure* (NSF 2003), this article refers to 'geospatial cyberinfrastructure' as a combination of distributed geographic information processing (DGIP; Yang and

---

*Corresponding author. Email: mtsou@mail.sdsu.edu

International Journal of Geographical Information Science
ISSN 1365-8816 print/ISSN 1362-3087 online
http://www.tandf.co.uk/journals
DOI: 10.1080/13658810802698571

Raskin 2009) technology, high-performance computing (HPC) resources, interoperable Web services, and sharable geographic knowledge to facilitate the advancement of geographic information science (GIScience), geospatial technology, and geographic education. The development of geospatial cyberinfrastructure will enhance current Internet GIServices (Peng and Tsou 2003) and provide powerful and effective computing capabilities for advanced spatial analysis methods and GIS models.

This article introduces a four-tier grid-enabled Internet GIService framework to support geospatial cyberinfrastructure. A traditional three-tier client/server GIS architecture is not capable of supporting highly scalable, secure, powerful, yet easy-to-use GIServices, which are indispensable elements of geospatial cyberinfrastructure (Zhang and Tsou 2005). The three-tier client/server architecture is limited to delivering centralized Web resources, such as Web sites or database servers, and is susceptible to capacity limits. If a single server is requested by a large number of concurrent clients (users), the server might be overloaded or crashed. The traditional client/server architecture also has a dilemma of choosing thin-client or thick-client approaches (the problem of assigning the bulk of the workload to the client-side machine or the server-side machine). In addition, most three-tier client/server systems are vulnerable on the server side in disaster recovery and fault tolerant events. Therefore, a new framework is required for geospatial cyberinfrastructure to facilitate the sharing and exchanging of DGIP-related data/information, processing resources, and knowledge across networks (Figure 1).

Among many technical concerns of geospatial cyberinfrastructure, performance is one of the most salient issues. Some technical barriers have been identified in the literature. For example, Anderson and Moreno-Sanchez (2003) pointed out that even a simple DGIP task may produce a large amount of geography markup language documents and raise parsing and storage problems. Data transfer mechanisms also pose a major challenge of GIServices. As reported by Anselin *et al.* (2004) in their Java-based spatial data analysis tools, Java applets (clients) may achieve poor performance while downloading a large amount of data from Web servers. Recent geoprocessing tests also reveal that high overhead could be incurred while transmitting simple object access protocol (SOAP) messages by Web coverage services between clients and servers (Scholten *et al.* 2006).

The overall performance issues include not only computing performance but also network bandwidth, visualization processing speed, memory limits, and storage capacity. While using complicated spatial analysis methods, such as spatial

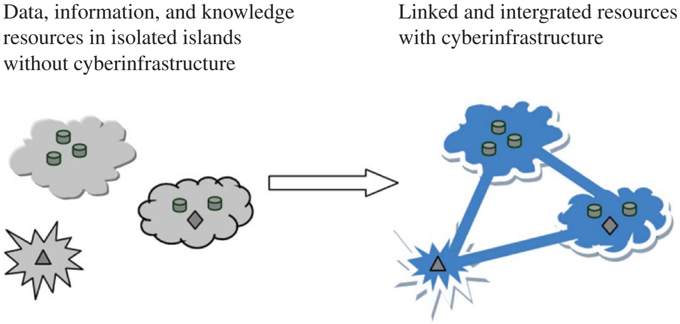


Figure 1. Cyberinfrastructure facilitates the integration of data, information, and knowledge resources.

autocorrelation algorithms, central processing unit (CPU) performance is critical for good overall performance. Massive GIS data sets, such as satellite images, may require very large random access memory (RAM) and disk storage on both clients and servers. Advanced geovisualization methods, such as 3D virtual globes, will rely on powerful graphics memory and graphics processing units. High-speed networks will be needed for communication-intensive geospatial applications, such as real-time collaborative Web mapping services.

The grid computing architecture, which has powerful computing capacity, large amounts of storage space, and high-speed networks, appears to be an ideal solution to address the performance problem of GIServices and geoprocessing tasks. Current multicore CPU Web servers or clusters may provide a partial solution for computationally intensive geospatial applications. However, most multicore CPU servers still cannot match the performance, flexibility, and scalability of grid computing.

This article introduces a HPC framework (with a grid-enabled Web portal) for Internet GIServices and geospatial cyberinfrastructure to address the limitations of traditional three-tier GIService frameworks. We designed a feasible implementation framework and established a grid-enabled high-performance spatial Web portal and evaluated several performance criteria.

2. The foundation technologies for supporting geospatial cyberinfrastructure

A comprehensive geospatial cyberinfrastructure requires several key technologies including high-performance geocomputation, grid computing, Internet GIServices, and spatial Web portals.

2.1 *High-performance geocomputation and GIS grid*

HPC is an approach for handling computing-intensive tasks that are *intractable* (not solvable in a reasonable amount of time) by a single processor. The experiments by Li (1992) on primitive map analysis operations were early HPC endeavors to parallelize raster-based map analysis. Marc Armstrong, another HPC pioneer, implemented research projects on parallel domain decomposition (Armstrong and Densham 1992), terrain analysis (Rokos and Armstrong 1992), network analysis (Ding *et al.* 1992), and spatial autocorrelation (Rokos and Armstrong 1993). His further studies in applying parallel processing methods in spatial statistics (Armstrong and Marciano 1995, Armstrong and Marciano 1996) and spatial interpolation (Armstrong and Marciano 1997) contributed to the development of parallel processing in geocomputation.

The design of parallel GIS algorithms is important for geographic data partition and geometry operation tasks. The two major GIS data models (vector and raster) lead to distinct design strategies and methodologies for parallel GIS algorithms (Healey *et al.* 1998). Recently, several studies of parallel GIS algorithm were published in the special issue of *Parallel Computing* (Clematis *et al.* 2003). A few of these articles highlighted the adoption of grid computing technologies as the next step in the development of HPC GIS applications (Aloisio and Cafaro 2003; Chervenak *et al.* 2003; Wang and Armstrong 2003).

Grid computing technologies have evolved rapidly since the mid 1990s, with a focus on the integration of large-scale computational resources and services via networks (Baker *et al.* 2002). An ideal grid framework will integrate distributed and dynamic computing resources. Therefore, service-oriented architecture (SOA) (Foster *et al.*

2002) can offer an improved platform for integrating grid computing resources. The Globus Toolkit^{TM} (Foster and Kesselman 1997), developed by the Globus Alliance, provides a good set of middleware for grid computing.

GIS researchers are also attracted to the grid computing paradigm. Several ongoing geospatial grid projects share a similar technical foundation but address different problems, such as the *Geosciences Network: Building Cyberinfrastructure for the Geosciences)* (http://www.geongrid.org/), the *Earth System Grid* (http://www.earthsystemgrid.org/), the *GISolve* (https://gisolve-portal.ncsa.uiuc.edu/), and the *Linked Environments for Atmospheric Discovery* (https://portal.leadproject.org/).

2.2 *Internet GIServices*

Internet GIServices are network-based information services that enable users to access geographic information, spatial analytical tools, and GIS Web services via the Internet (Peng and Tsou 2003). Early examples of Internet GIServices can be traced back to several Web-based maps in the 1990s, such as MapQuest and the Xerox Map Viewer. Component-based GIServices became popular in the early 2000s and offered new insights for building distributed GIServices online (Li 2000). Within *GIServices*, the SOA supports flexible and customizable information services. Service chaining mechanisms (Friis-Christensen *et al.* 2009) and workflow solutions (Chen *et al.* 2009) are keys to such SOA-based solutions. Tsou (2004) identified three levels of Internet GIServices from a user's perspective: data services, information services, and spatial analysis (knowledge) services. To date, most Internet GIServices operate at the data services and information services levels. Under the traditional client/server architecture, it is difficult to resolve the performance problems of geospatial Web services and the dilemma of client-side computing or server-side computing (Tsou and Buttenfield 2002). Geospatial cyberinfrastructure might provide solutions by allowing users to dynamically build GIS functions and spatial models at the knowledge level of GIServices. For model-level sharing, the identification of common functions among models (Hu and Bian 2009) is a potential solution.

2.3 *Spatial Web portals*

A Web portal offers a centralized, uniform interface to access distributed and heterogeneous resources and services (Tang and Selwood 2005). Some researchers have designed standard Web portal frameworks that host portal-based applications by providing basic Web portal administration tools such as user authentication, customizable user interface, and user profile management. The framework can be seen as a container to encapsulate simple portal applications, called *portlet* programs.

Within the GIS community, spatial (GIS) Web portals will be an important component of geospatial cyberinfrastructure. The development of science gateways has been encouraged by the needs of conducting scientific research through Web portals using specific domain-oriented tools and data. In addition, based on common science gateways, GIScience researchers develop comprehensive solutions to support GIScience research and education (e.g. TeraGrid GIScience Gateways) (Wang and Liu 2009). Spatial Web portals can facilitate geospatial data access and enhance spatial data infrastructure (Maguire and Longley 2005) but must address issues such as system usability, high availability, and lack of content standards (Tait 2005).

A noticeable weakness of current spatial Web portals is that they rarely provide powerful analytical tools beyond data searching and map viewing, except for a few pioneering prototypes (e.g. PGIST portal) (Nyerges *et al.* 2006). Recent technical advancements in Web Services and grid computing encouraged GIScience researchers to design comprehensive spatial Web portal solutions for DGIP applications. For example, Yang *et al* (2007) detailed (1) how to utilize interoperable DGIP services, such as Web map service (WMS) and WFS, for interoperability solution and (2) how to utilize Java specification request (JSR)-168/268 compliant portlets and Web services for remote portlets to integrate a spatial Web portal.

3. A new framework for grid-enabled Internet GIServices

This article introduces a new grid-enabled Internet GIService framework for geospatial cyberinfrastructure, which is flexible, interoperable, and scalable. This framework (Figure 2) contains four tiers: a *presentation tier*, *logic tier*, *service tier*, and *grid tier*. The presentation tier is a counterpart to the client in the traditional client/server architecture. The logic tier incorporates the roles of map servers and GIS servers in the traditional architecture. The logic tier has an additional role to coordinate grid-enabled Web portal services according to user requests (converting them to geospatial problem semantics). The service tier replaces traditional centralized GIS databases

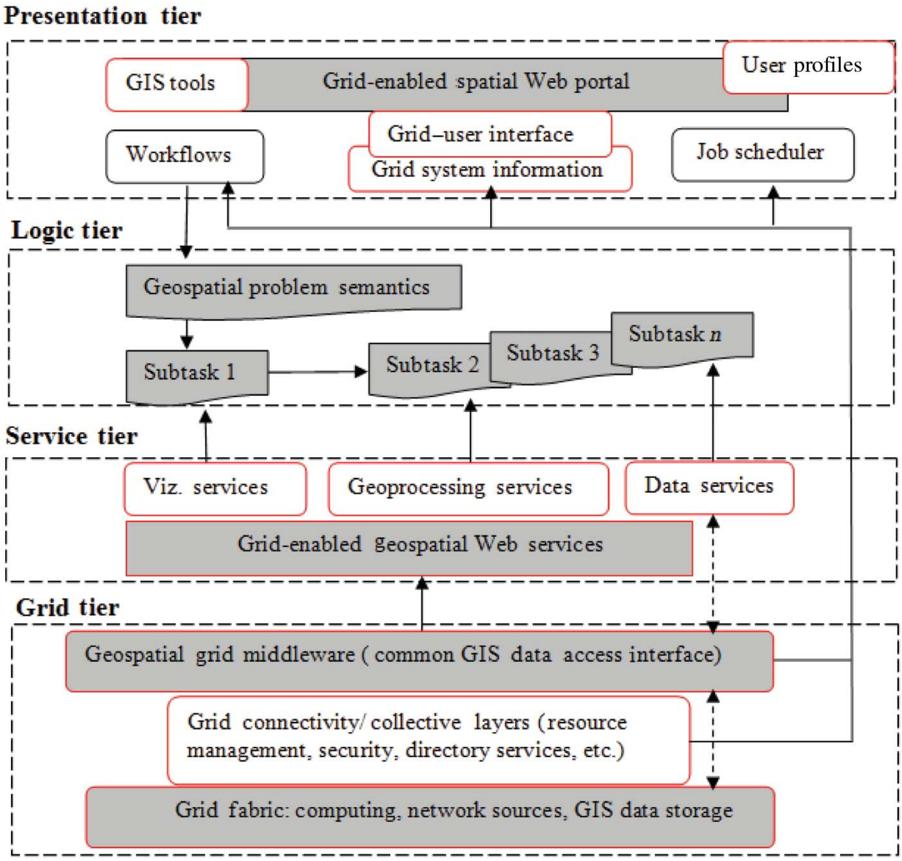


Figure 2. A grid-enabled Internet GIService framework.

with new grid-enabled geospatial Web data services. The grid tier supports the other three tiers with distributed Web services, HPC, network resources, GIS data storage, and other components. Due to the technical challenges of semantic implementation, the grid-enabled Web portal prototype does not implement real geospatial problem semantics but uses ad hoc programs to mimic the results of semantics and identify Web services for spatial analysis methods. The major components outlined in red in Figure 2 have been implemented and developed in the prototype. Most functions provided by the grid-enabled framework have been tested except workflows, job scheduler, and geospatial semantics. The dashed arrow lines illustrate dependent/indirect data flow. Communications among d*ata service*, c*ommon GIS data access interface*, and *GIS storage* also rely on other middleware components and grid services. For example, grid security protocols are essential to enable grid data services. However, logically, *GIS data physical storage*, *GIS data access interface*, and *GIS data services* are interconnected. Solid arrow lines indicate direct and independent information exchange.

The presentation tier works as a simplified integrated interface for users to access intelligent and distributed Internet GIServices at the service tier. Also, the presentation tier offers mechanisms to view information about submitted grid jobs and grid resources from the grid tier. The workflow view stores the semantics for geospatial problems and is directly connected to the logic tier. The logic tier, which is located between the presentation and the service tiers, defines a set of geospatial problem semantics that formalize geospatial problem-solving procedures. The address and identifier of the located GIServices can then be returned to the presentation tier to dynamically build a real-time application. The service tier hosts distributed grid-enabled GIServices that are connected to the subtasks defined in the logic tier. The last grid tier underpins the other tiers to provide grid-based computing and geospatial resources through geospatial middleware.

3.1 *The presentation tier*

The presentation tier is the interface between a user and the underlying distributed grid computing environment. A grid-enabled spatial Web portal is the major component in the presentation tier, which can provide interactive GIS tools for data searching, geoprocessing, and the final display of GIS operation results. This tier is a thin client because most intensive analysis procedures and data transfer tasks are migrated to the high-end grid computing platforms. Users can monitor current progress of grid-enabled GIS jobs, track the computing resources used, interrupt running jobs if needed, and interact with (connect to) grid systems via grid–user interfaces.

Users can implement their personalized spatial analysis tasks in this tier. Several windows (views) are available to examine the computational problems from different perspectives, such as a workspace view, data view, map view, and workflow view. The workspace view provides a high-level vision of the spatial analysis problems with basic information, including participants, problem-solving history, progress, and calendars. The data view presents a list of georeferenced data used in the current project. A map view allows users to visualize the given data, to evaluate geocomputation results, and to directly manipulate maps generated from grid-enabled GIS operations. The workflow view is presented as a multistep GIS procedure flow chart with different diagram shapes and colors to describe different elements, operations, and priority levels. Users can visually and interactively manipulate Internet GIServices modules (e.g. spatial querying, buffering, layering, autocorrelation, and cluster and outlier detection). The workflow view stores the semantics for the geospatial problems, which is updated when the contents are modified in other views.

3.2 *The logic tier*

A logic tier is required to handle the complexity of multiple geospatial problem-solving methods. Different people may consider the same geospatial problem from different perspectives, and this may lead to multiple problem definitions and descriptions. These inconsistencies would produce different problem-solving procedures that require different geospatial data, geoprocessing tools, and GIS operations. Ideally, the visualization of workflows in the presentation tier will connect to the logic tier where geospatial problem semantics can be formalized by the user.

The semantics in the logic tier will be described by formal semantic languages, such as Web ontology language or other ontology frameworks. Usually, a workflow consists of multiple subtasks. These multilevel structures are stored in the form of geospatial semantics in the logic tier. Users with specialized privileges (defined in user profiles) can modify the geospatial semantics by adding, deleting, or changing the subtask elements. Subtasks are supported by customizable GIServices in the next tier (the service tier). The automated and dynamic GIService discovery, selection, and composition mechanisms are also operated in the logic tier. Given specific geospatial problem-solving semantics, the logic tier mechanisms will describe what GIServices are needed given current contexts and the preference criterion, such as data requirements, performance, visualization preferences, and credibility of service providers. With intelligent matching mechanisms, GIService locators and filters should be developed at the logic tier to query, locate, and select the most appropriate GIServices for geospatial applications.

3.3 *The service tier*

The major goal of the service tier is to provide grid-enabled geospatial Web services based on the subtasks in the logic tier. Three types of geospatial Web services are hosted in this tier: visualization services for geovisualization and mapping tasks, data services for archiving, searching, and transferring geospatial data, and geoprocessing services for complex geocomputation tasks. Data services will replace the database servers in traditional client/server architecture. Users can utilize integrated geospatial data services remotely rather than access distributed GIS databases. This arrangement will provide more robust data services and mitigate network traffic congestion problems.

For all three types of geospatial Web services to be interoperable, geospatial Web services should follow XML-based (extensible markup language) standards and specifications, which include SOAP, WSDL (Web service description language), and UDDI (universal description, discovery, and integration) standards. Geospatial Web services also need to follow Internet GIS specifications, such as open geospatial consortium (OGC) standards (Web mapping service, Web feature service, and Web coverage service) and the emerging DGIP standards (e.g. OGC Web processing service). Users will be able to automatically locate the most suited geospatial Web services given their preferences and task requirements with help from the Web service locators and filters in the logic tier. Use of these Web service standards will ensure the interoperability of GIS operations and data processing tasks. OGC conducted a series of test-bed experiments to combine SOAP, UDDI, and WSDL with OGC interoperability standards as a portion of OGC Web Services, Phase 2 (OWS-2) (http://www.opengeospatial.org/projects/initiatives/ows-2/).

One important aspect of the service tier is the security consideration for GIServices. The new framework must ensure the integrity and confidentiality of GIService messaging. Authentication, authorization, delegation, accounting, access control, encryption, and data integrity are all critical to enhance the security of geospatial Web services. By adopting the Web service security (WS security) protocol, the service tier provides a better protection for geospatial Web services.

3.4 *The grid tier*

The foundation of the new framework is the grid tier, the fundamental element of high-performance geospatial cyberinfrastructure. This tier includes *grid fabric*, *connectivity*, *and collective layers* (Foster *et al*. 2001), and combines all the underlying grid computing hardware and software components. High-speed interconnection networks, high-end computing, and large storage spaces are the basic fabric elements. Coupled with grid middleware, multiple computers can be connected to provide aggregated manageable resources in the grid fabric. To manage multiple computers, several administrative rules, such as data access permissions, job management, failover, authorization, authentication, service discovery, and diagnostic mechanisms, have to be integrated by a geospatial grid middleware. The grid middleware provides coordinated connections between geospatial Web services and the grid fabric. The geospatial grid middleware extends generic grid software to facilitate the access of grid fabric resources such as computing units, network resources, and GIS data storage.

The geospatial grid middleware provides a common GIS data access interface based on geospatial interoperability principles and standards. Popular GIS data storage models, such as file-based data sets (e.g. ESRI shapefile$^{\mathrm{TM}}$), GIS proprietary databases (e.g. ESRI ArcSDE$^{\mathrm{TM}}$), and open source GIS databases (e.g. PostGIS), should be supported in the grid middleware. Data service requests will be sent via the common GIS data access interface to the grid fabric to retrieve data from the GIS data storage devices.

4. The implementation of a grid-enabled spatial Web portal

We implemented a spatial Web portal prototype to demonstrate the feasibility of the new grid-enabled Internet GIService framework. This prototype provides high-performance DGIP capabilities to support an accessibility analysis task with a unified Web portal interface.

4.1 *A transport accessibility analysis case study*

A simplified transport accessibility analysis case study was introduced to demonstrate the advantages of the grid-enabled spatial Web portal. Transport accessibility refers to the capabilities of individuals to reach destinations to perform specific errands. This case study computed zone-based accessibility measures at multiple scales and provided Web-based accessibility maps. Given a geographic zone (e.g. traffic analysis zone or census block), a generic accessibility measure is

$$\mathrm{Accessibility}_i = \sum_j \sum_m \mathrm{opportunities}_j \times f\left(C_{ijm}\right) \tag{1}$$

where

i is the index of the zone where accessibility is measured,
j is the index of the zones comprising potential travel destination,

m is the transportation modes,
Accessibility$_i$ is accessibility measured at zone i,
Opportunities$_j$ is opportunities in zone j,
$f(C_{ijm})$ is the generic travel cost function between zone i and j by transportation mode m.

Travel cost functions, $f(C_{ijm})$, calculate time or other types of generalized costs (i.e. money, congestion, and risk) between zones for a given transportation mode. In the prototype, two classical accessibility measures are implemented: *cumulative opportunity* measure and *gravity-based* measure. The travel cost function for the cumulative opportunity measure (Wachs and Kumagai 1973) is

$$F_{jm} = \begin{cases} 0 & \text{cost}_j > \text{threshold} \\ 1 & \text{cost}_j < \text{threshold} \end{cases} \tag{2}$$

Opportunities in zone j are counted only if the associated costs do not exceed the thresholds.

The gravity-based measure (Hansen 1959) defines accessibility by calculating potential accessibility with $f(C_{ijm})$ describing the impedance between zones i and j by transportation mode m, where $f(C_{ijm}) = C_{ijm}{}^{-2}$. In this study, travel cost is measured by the distance between zones, using the (x, y) coordinates of the zone centroids.

4.2 *The implementation framework of the spatial Web portal prototype*

The spatial Web portal is a proof-of-concept demonstration of the four-tier framework for geospatial cyberinfrastructure. The portal was constructed using grid computing technologies, commercial off-the-shelf software, and open source programming tools. Figure 3 presents the technical implementation framework for the spatial Web portal. The portal server hosts a Web portal framework that adopts

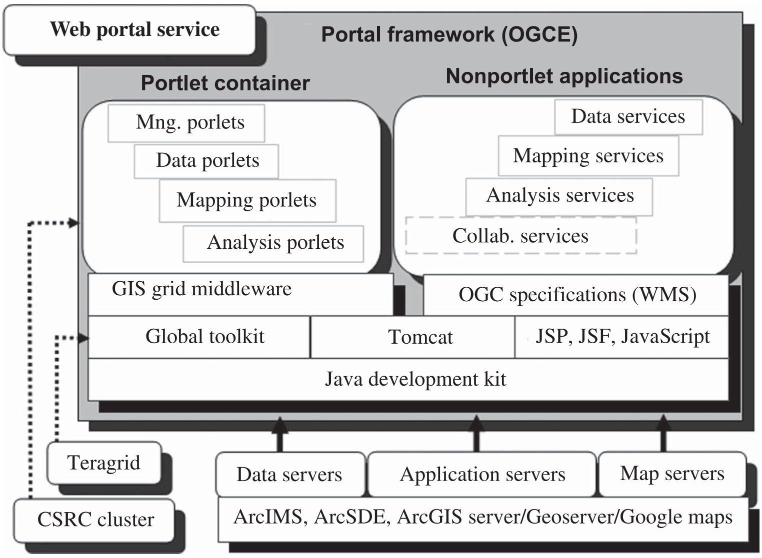


Figure 3. A technical framework for the accessibility analysis Web portal.

the Open Grid Computing Environments (OGCE) Portal and Gateway Toolkit (version 2.2.1) as the container for JSR-168-compatible portlets (http://www.collab-ogce.org).

The Web portal framework consists of two types of components: JSR 168-compatible portlets and nonportlet applications. The JSR 168 portlets provide management, data, mapping, and analysis services. For nonportlet applications, multiple data services, Web mapping tools, and spatial analysis services were implemented by a hybrid combination of ESRI™ Web GIS solutions (ArcIMS™, ArcWeb Services™, ArcGIS Server™), GeoServer, and Google® Maps. Google maps application programming interfaces (APIs) were used to display accessibility analysis maps on Google Maps using the WMS protocol via an ArcIMS WMS connector and the GeoServer WMS server. The accessibility analysis results were also published by ArcIMS and ArcWeb REST (representational state transfer) services. GeoServer, a popular open source GIS server package, was used to generate various accessibility maps in multiple formats (e.g. PDF, OpenLayer, geography markup language).

The OGCE portal framework has built-in management portlets for user account management and user profiling. This prototype installed new georeferenced data portlets, which are mainly responsible for providing the data view window and metadata services. Accessibility mapping and analysis portlets were also created in the portlet container to provide map viewer functions and grid computing configuration for the accessibility analysis model. Scalable Vector Graphics (SVG)™ was adopted as a map export format for visualizing accessibility analysis results. GIS grid middleware was created using GeoTools (a Java GIS development library) (http://geotools.codehaus.org/). The major tasks of the GIS grid middleware are to (1) establish connections to remote computing resources, (2) initiate computing sessions, (3) process geospatial data, (4) coordinate network communications, (5) fetch and store the computing results, and (6) generate visualization maps. The Globus Toolkit™ (4.0.3) was installed on the Web portal server to offer grid programming capability for geospatial cyberinfrastructure.

All of these Web-based services are powered by the Apache Tomcat Web application container. JavaServer Pages™, JavaServer™ Faces, and JavaScript technologies were used to develop Web interfaces for both portlet and nonportlet applications. The Java™ Development Kit provided the major programming environment for creating the grid-enabled GIS applications and programs.

The prototype deploys accessibility analysis portlets in three test-bench environments: a stand-alone Web server (GeoGrid), a PC cluster in the Computational Science Research Center (CSRC) at San Diego State University (SDSU), and the TeraGrid. The spatial Web portal executes GIServices using the message passing interface (MPI) as the parallel programming model on multiple processors. Georeferenced data and MPI programs were transferred to remote machines to calculate transport accessibility based on two accessibility measures. Returned analysis results are visualized either in SVG or in a shapefile format on the spatial Web portal.

The GeoGrid server (http://geogrid.sdsu.edu) runs on a DELL™ Dimension™ E310 workstation computer (Intel™ Pentium™ 4 CPU 3.06 GHz. 1 GB DDR2 SDRAM at 400 MHz) located in the Department of Geography at SDSU. The PC cluster (http://dolphin.sdsu.edu) is located in the CSRC at SDSU. It is a Beowulf-style Linux cluster with 40 batch nodes. Every node hosts an Intel™ P4/Xeon™ Dual 2.4 GHz CPU and 2 GB RAM. The MPI compilers of the CSRC cluster use portable batch system (PBS). PBS is a popular parallel job scheduler that is a widely used queuing system for PC clusters and supercomputers.

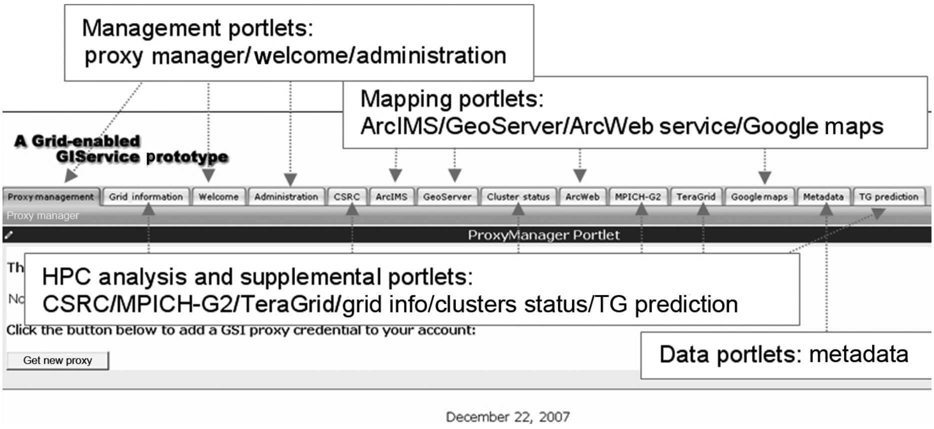
TeraGrid (http://www.teragrid.org/) is a nationwide grid computing platform that integrates the most advanced supercomputer resources at geographically distributed sites. Three sites are chosen for the TeraGrid testing experiments: the San Diego Supercomputer Center (SDSC), the National Center for Supercomputing Applications (NCSA), and the Argonne National Laboratory (UC/ANL). For the purpose of performance comparison, three DTF IA-64 Linux Clusters at the three sites are the major TeraGrid resources used in the experiments. All sites offer IBM™ Dual Intel™ Itanium™ processors and have comprehensive job submission and scheduling mechanisms, such as PBS, Condor-G, Globus, MPICH-G2, MyCluster, and Multi-site VMI (http://www.teragrid.org/userinfo/jobs/index.php). We selected PBS, Globus, and MPICH-G2 as the three grid submission job methods for the spatial Web portal.

4.3 *The spatial Web portal prototype*

We created a grid-enabled GIService Research prototype, which is a spatial Web portal for transport accessibility analysis tasks. Each user type (e.g. administrator, super user, TeraGrid user, or CSRC user) has associated security privileges to access various functions in the spatial Web portal. The Web portal adopted a '*single sign-on*' mechanism in which users log on once to gain access to multiple computing resources, distributed database servers, and remote storage devices. The spatial Web portal provides a Web-based graphical user interface, containing multiple portlet tabs (Figure 4). Four groups of portlets are designed for GIServices: management portlets, HPC analysis and supplemental portlets, mapping portlets, and GIS data portlets.

Three management portlets incorporate user and group account management (the Administration portlet), portlet deployment (the Proxy Manager portlet), and user login functions (the Welcome portlet).

Six HPC analysis and supplemental portlets have been implemented for this prototype (CSRC, MPICH-G2, TeraGrid, grid information, cluster status, and TG Prediction). The CSRC portlet allows users to execute transport accessibility analysis



Click the button below to add a GSI proxy credential to your account:

December 22, 2007

Figure 4. The user interface of the grid-enabled GIService research portal.

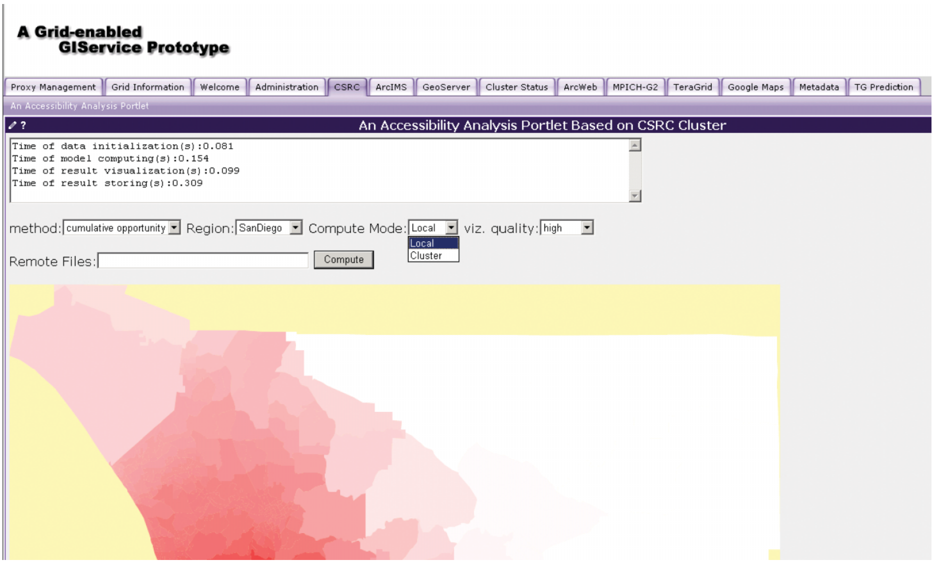
tasks using the local server (GeoGrid) or the CSRC cluster (Figure 5). Two TeraGrid-based portlets (TeraGrid and MPICH-G2) are implemented using three MPI standards: PBS, Globus, and MPICH-G2. The PBS and Globus job submission mechanisms are provided in the TeraGrid portlet, which can only submit GIService tasks to a single grid computing site on the TeraGrid. MPICH-G2 is a separate portlet that enables cross-site job submission, enabling users to submit a GIService task to multiple TeraGrid computing sites concurrently.

Three other portlets, grid information, cluster status, and TG (TeraGrid) prediction, are supplemental components to update real-time TeraGrid system information, CSRC cluster status, and TeraGrid batch queue prediction services, respectively. This information can be used to determine where to submit the next GIService job and how many grid 'nodes' should be requested from the spatial Web portal.

Figure 6 shows a screenshot of the CSRC portlet used in the transport accessibility analysis. The accessibility analysis task was executed by the CSRC cluster (16 nodes), and the result map was displayed using SVG for the continental US.

The accessibility analysis results were also visualized in four Web mapping portlets: Google maps, ArcIMS, ArcWeb services, and GeoServer. The purpose of including these Web mapping tools is to compare the advantages of these Web mapping technologies for the spatial Web portal. The Google Maps portlet tested the adoption of the Google Map API and the mashup application. The mapping tools adopted the OGC WMS to provide interoperable mapping services and ensure the integrity of mapping information from multiple map servers.

The GIS data portlet delivers GIS metadata services using ArcGIS Metadata Explorer. Users can register and upload geospatial data to this portlet which later will publish them as downloadable data sets. These data sets can be directly processed by the accessibility analysis portlets.

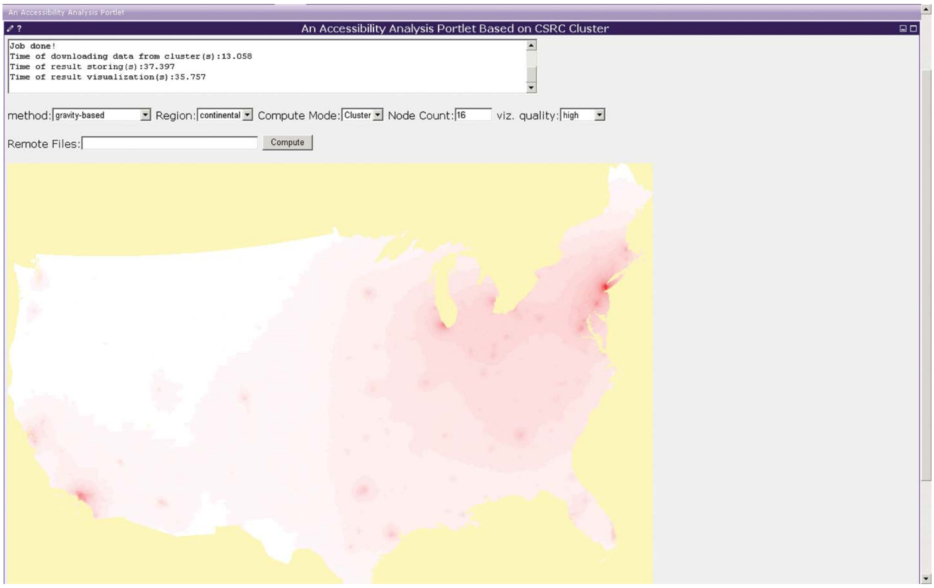


```
Time of data initialization(s):0.081
Time of model computing(s):0.154
Time of result visualization(s):0.099
Time of result storing(s):0.309
```

method: cumulative opportunity Region: SanDiego Compute Mode: Local viz. quality: high

Remote Files:

Figure 5. The local-based/cluster-based accessibility analysis portlet.



```
Job done!
Time of downloading data from cluster(s):13.058
Time of result storing(s):37.397
Time of result visualization(s):35.757
```

method: gravity-based Region: continental Compute Mode: Cluster Node Count: 16 viz. quality: high

Remote Files:

Figure 6. An accessibility analysis result using the CSRC cluster.

4.4 *A usage scenario and stress test for the spatial Web portal prototype*

This section illustrates a usage scenario to demonstrate the advantages of the spatial Web portal prototype and the needs of HPC. The usage scenario simulates a GIS analyst from the California Department of Transportation conducting a transport accessibility analysis for the San Diego region. The analyst can log on to the portal and specify a service area (e.g. San Diego) and transportation planning tasks (e.g. measuring accessibility to job by transit). The portal provides an integrated working environment to access original geospatial data, Web-based mapping tools, and other geoprocessing services from a united Web-based interface.

This integrated spatial Web portal can become a problem-solving environment in which multiple government employees and common citizens enter and (with sufficient privileges) conduct personalized analysis with active online Internet GIServices. This scenario demonstrates a typical geospatial cyberinfrastructure tool enabling multiple users to conduct personalized geospatial analysis for solving complex geospatial problems. Part of the scenario has been simulated in our high-performance spatial Web portal prototype, which is composed of multiple data sets, mapping services, and geoprocessing services.

One major advantage of this grid-enabled portal is that it enables many concurrent users, a capability not generally possible with traditional three-tier GIS servers. Therefore, we conducted stress tests for the portal using Apache JMeter (http://jakarta.apache.org/jmeter/), a Java-based testing tool for system performance. The test results indicate that the system can support 3000 simultaneous user sessions. The spatial Web portal system crashed when 3100–3400 virtual users were simulated. With 3000 simulated users, the total throughput for selected portlets can reach 16.9 requests per second, with an average time per request of 8.2 seconds. The stress test results demonstrate the advantage of scalable user capacity in the new grid-enabled GIService framework.

4.5 *Experimental GIS data*

Three groups of georeferenced data objects were used in the prototype experiments: San Diego County, the State of California, and the continental United States. For both San Diego and California data sets, the attribute information includes census 2000 population data and employment data at the census block group level. For the continental US, only census 2000 population data are available. The Web portal prototype uses these three sets of data to compare the performance of grid-enabled Web portlets in various settings. The data reside in ESRI shapefiles$^{\mathrm{TM}}$. The San Diego County shapefiles have 1762 polygons, the State of California shapefiles have 22,195 polygons, and the continental US shapefile has 207,755 census block group polygons. The two accessibility measures were used to examine the transport accessibility to jobs and to residential homes based on employment and population data.

Ideally, indexed, optimized GIS databases should be used for grid-enabled GIServices rather than file-based shapefiles. However, current grid computing environments in both PC clusters and the TeraGrid do not support GIS data formats (either GIS database or shapefiles). Consequently, we created a geospatial grid middleware to convert the GIS data sets from shapefiles to a text-based format that were then transferred to remote grid computing platforms for accessibility model computation. Based on the data conversion experience, we suggest that geospatial cyberinfrastructure support common GIS data models for the grid computing and parallel programming environments. The next section demonstrates the advantages of the grid-enabled framework compared to traditional local GIS servers and PC clusters. The major evaluation criterion is the run time to conduct the accessibility analysis with varying experimental settings.

4.6 *Computational performance results*

The first experiment used the GeoGrid portal server (a stand-alone server) to compare the performance between a local stand-alone Java program and a Web portal portlet developed in the prototype. The accessibility analysis algorithm was executed in both frameworks (Java and portlet). The run time comparison for the accessibility analysis task (the cumulative opportunity measurement method) with multiple GIService tasks is shown in Table 1.

From Table 1, the following observations were made:

- There is no significant performance difference between a stand-alone Java program and a portlet inside a Web portal (a portlet container) in terms of execution time of the spatial model computation.
- The run time of the spatial model computation increases nonlinearly with data set size. For the large-size continental GIS data set (286 Mb), the computation run time (2054 seconds) is almost 12,000 times longer than the run time (0.17 seconds) of the county-level data set (2.1 Mb).
- The SVG-based visualization method is very effective for the spatial Web portal. The continental data set, with more than 220,000 polygons, can be rendered in 30y seconds to provide high-quality SVG-based mapping output.

In addition to testing on a Java platform and a portlet container, we applied the same GIS algorithms on a high-performance single GIS server (2 × AMD$^{\mathrm{TM}}$ Opteron$^{\mathrm{TM}}$ 2224 SE Dual Core 3.2 GHz, 16 Gb RAM). The run time for the continental US data on the single high-performance GIS server was 5.39 (data reading), 624.56 (GIS

Table 1. The run-time comparison between Java program and portlets (time in seconds).

Data			Data reading	Model computation	Results storage	Visualization	Total
San Diego population (2.1 Mb)	Java	0.44	0.18	0.47	0.18	1.27	
	Portlet	0.09	0.16	0.31	0.10	0.66	
San Diego employment (3.0 Mb)	Java	0.49	0.17	0.70	0.18	1.54	
	Portlet	0.13	0.16	0.57	0.16	1.02	
California population (27.2 Mb)	Java	1.45	24.44	3.75	2.07	31.71	
	Portlet	1.07	24.42	3.94	2.22	31.65	
California employment (38.6 Mb)	Java	2.03	24.50	6.90	2.00	35.43	
	Portlet	1.64	24.40	6.94	2.24	35.22	
Continental population (286.0 Mb)	Java	10.92	2054.22	36.19	26.33	2127.66	
	Portlet	10.64	2054.63	39.10	31.05	2135.42	

model computation), 18.43 (storing results), and 12.89 (visualization). Clearly, a single high-performance server can boost the computation performance of GIS tasks significantly.

The second experiment examined the scalability of the CSRC cluster for parallel computing frameworks. The test used two data sets, California employment and continental US population data sets. Table 2 and Figure 7 present the execution times of the two large data sets. However, the US population case could not be executed successfully with a single node. This test illustrated that parallel computing can significantly reduce the run time of GIS tasks with large data sets by using multiple computer nodes.

In a third experiment, we measured the run time of accessibility analysis tasks on the TeraGrid using PBS and Globus GRAM (Grid Resource Allocation and Management) submission methods. The continental US population data set was used, and high-quality SVG maps were adopted for the default output visualization format. Table 3 presents the test results for TeraGrid PBS and Globus. This test only utilized one grid computing site (NCSA) in the TeraGrid because PBS and Globus methods cannot submit GIService tasks to multiple grid computing sites

Table 2. Model computing time with California/US data using CSRC cluster.

		Number of nodes (each node has 2 processors) time (seconds)					
Data	Measures	1	4	8	12	16	20
California employment (CA data)	Gravity based	25.05	3.85	3.21	2.22	1.77	1.05
	Cumulative opportunity	12.13	2.39	1.97	0.98	1.25	0.78
Continental US population (continental data)	Gravity based		338.29	157.95	104.30	77.55	61.97
	Cumulative opportunity		199.26	94.10	77.71	57.83	38.05

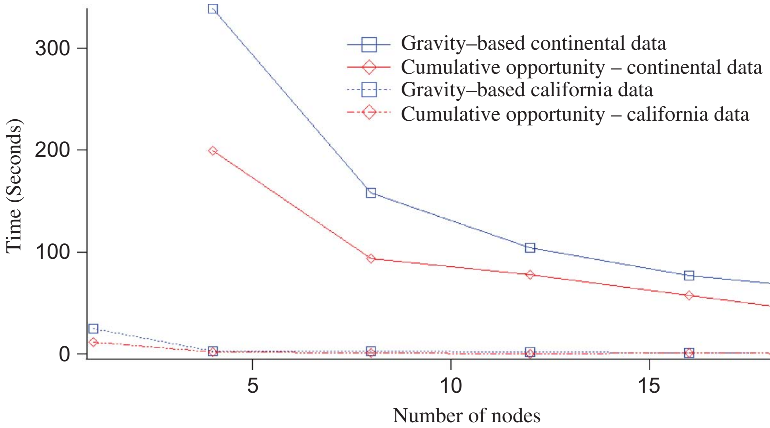


Figure 7. Model computation time with California/US data using CSRC cluster.

Table 3. Model computation time with US data using TeraGrid PBS and Globus.

		Number of nodes (each node has two processors) time (seconds)					
Submission methods	Measures	4	8	12	16	20	24
Globus	Gravity based	129.16	71.83	47.48	36.63	29.28	23.24
PBS	Gravity based	124.35	65.23	43.59	34.76	31.42	25.69
Globus	Cumulative opportunity	52.77	26.43	19.04	14.94	12.63	11.18
PBS	Cumulative opportunity	52.61	26.51	18.76	15.13	13.23	11.21

concurrently. The execution times of PBS and Globus GRAM on the TeraGrid are illustrated in Table 3 and Figure 8.

Based on the third performance test, we observed the following:

- The two-grid computing job submission methods, PBS and Globus GRAM, achieved similar performances. Grid computing can significantly reduce the run time of GIS tasks by increasing the number of computer nodes in the grid framework.
- Experimental results were compared between the CSRC cluster and the TeraGrid. Overall, the TeraGrid outperformed the CSRC cluster. For example, the eight-node CSRC cluster required 157.95 seconds to process the continental US population data set, while the eight-node TeraGrid only needed 71.83 seconds to process the same data set using Globus. The TeraGrid has more powerful computing resources than the CSRC cluster, and its more advanced load-balancing mechanism helps achieve better performance.
- The Globus GRAM job submission method reduced the job queuing time in our experiments and provides a standard interface to execute remote jobs (Czajkowski *et al.* 1998). Our experiments used the Pre-Web service GRAM2 instead of GRAM4 (based on Web service interfaces) because of the slower performance of Web Services.

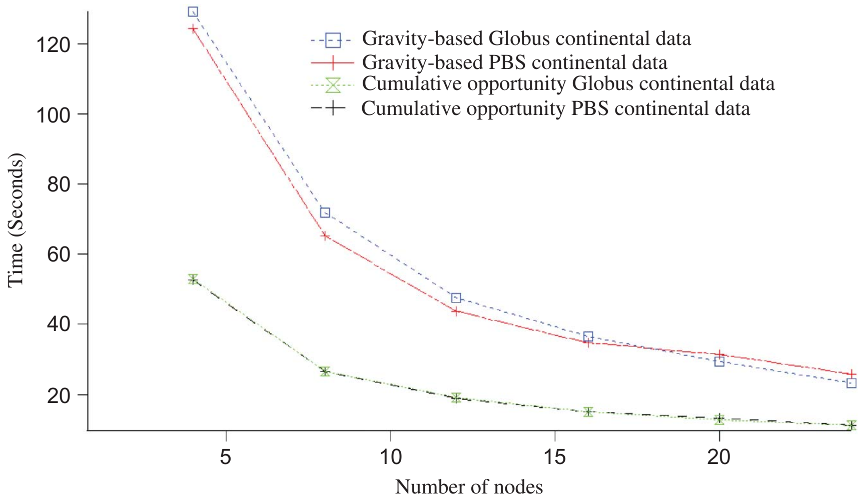


Figure 8. Model computation time with US data using TeraGrid PBS and Globus.

The next experiment used the MPICH-G2 submission method to run a cross-site grid computation for the continental US population data at two TeraGrid sites (SDSC and NCSA). The number of nodes from each site is the same. For example, four nodes from each site were used in the first test. Table 4 and Figure 9 illustrate the comparison of execution run times for the accessibility measurements between MPICH-G2 and PBS.

Based on the fourth performance test, we observed the following:

- The MPICH-G2 method required more time to complete the GIS model computation compared to the PBS method (Figure 9). For example, 118.04 seconds are needed for the eight-node MPICH-G2 compared to 65.23 seconds for the eight-node PBS. Similar results are reported by Dong *et al.* (2005).
- The slower performance of cross-site grid computing experiments can be explained by the high network latency in its wide area network between SDSC (San Diego, California) and NCSA (Urbana-Champaign, Illinois). However, the MPICH-G2 performance (118.04 seconds) is still much better than the CSRC cluster (157.95 seconds).

Table 4. Model computation time with US data using TeraGrid MPICH-G2 and PBS.

Submission methods	Measures	Number of nodes and nodes from each site time (seconds)				
		8 (4 + 4)	12 (6 + 6)	16 (8 + 8)	20 (10 + 10)	24 (12 + 12)
MPICH-G2	Gravity based	118.04	84.88	61.83	54.97	46.15
PBS	Gravity based	65.23	43.59	34.76	31.42	25.69
MPICH-G2	Cumulative opportunity	52.62	47.78	35.02	21.71	18.63
PBS	Cumulative opportunity	26.51	18.76	15.13	13.23	11.21

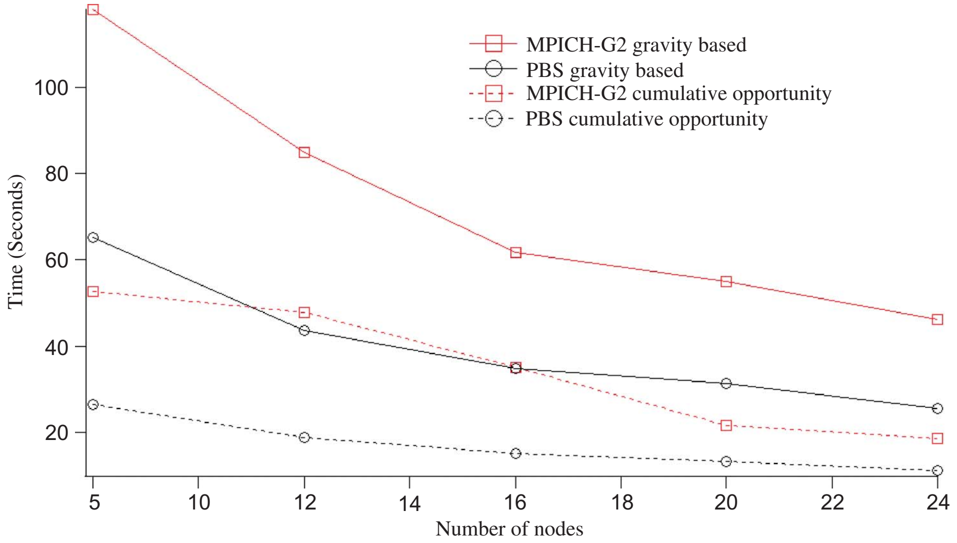


Figure 9. Model computation time with US data using TeraGrid MPICH-G2 and PBS.

We also compared the job queuing time (and any executable staging time) for the CSRC cluster, TeraGrid PBS, Globus GRAM, and MPICH-G2 methods. The job queuing times for the continental US population data set are shown in Table 5.

Based on Table 5, we observed the following:

- The job queuing time was affected by several factors including the current load of the TeraGrid sites, available CPU processors and memory, and job characteristics (e.g. requested number of nodes). The availability of grid computing resources, network traffic, and the job submission time has significant impacts on the job waiting (queuing) time.
- The wait times for the NCSA PBS and SDSC MPICH-G2 sites were much longer than on the CSRC cluster. NCSA and SDSC sites are very popular TeraGrid hubs and typically have many users. In contrast, the CSRC cluster is used only by the SDSU Computer Science Department.

Table 5. Comparison of job queuing time costs.

Submission methods	Measures	Number of nodes (each node has two processors) time (seconds)					
		4	8	12	16	20	24
CSRC PBS	Gravity based	2.96	2.61	3.16	3.33	6.28	
	Cumulative opportunity	2.06	2.42	1.01	6.02	2.57	
TeraGrid PBS	Gravity based	81.65	44.77	104.41	131.93	154.58	178.31
	Cumulative opportunity	20.39	61.49	34.24	75.87	78.77	39.80
TeraGrid globus	Gravity based	27.84	53.17	47.53	58.37	65.72	74.76
	Cumulative opportunity	23.23	38.57	29.96	80.01	51.38	52.82
TeraGrid MPICH-G2	Gravity based		65.96	69.12	80.18	36.03	40.85
	Cumulative opportunity		69.38	63.22	65.98	99.29	61.37

- The job queuing time increased as the number of requested processors increased. For example, user wait time was 178.31 seconds for a 24-node job submission and 44.77 seconds for an eight-node job submission. However, smaller jobs might require a longer waiting time depending on the availability of computing resources.

4.7 *Parallel computing programming in the prototype*

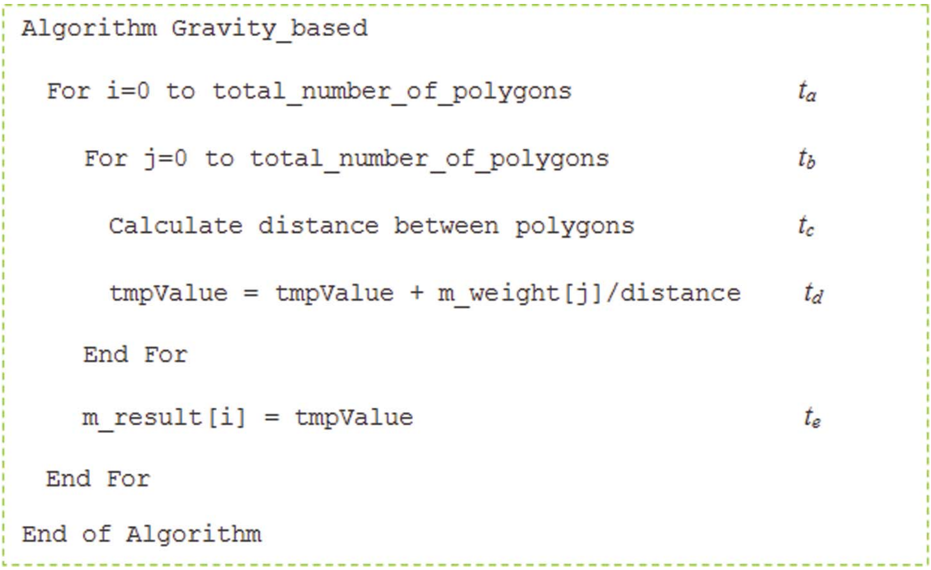
The spatial Web portal prototype ran the gravity-based accessibility algorithm and the cumulative opportunity measure algorithm to compare the performance of various grid computing parameters. Figure 10 illustrates a pseudocode for the gravity-based measure algorithm.

The *m_weight* array represents accessibility opportunity (i.e., population or employment) data and the *m_result* array stores accessibility result values. The time complexity of the gravity-based measures is $O(n^2)$:

$$T_{\mathrm{g}}(n) = t_{\mathrm{a}}n^2(t_{\mathrm{b}} + t_{\mathrm{c}} + t_{\mathrm{d}}) + t_{\mathrm{e}}n \tag{3}$$

To compare the performance between PC clusters and the TeraGrid, the two accessibility algorithms were converted into the MPI parallel codes shown in Figure 11.

This MPI program-enabled data parallelism, in that every processor can perform the same accessibility computation on different data portions (also called single program multiple data). The size of individual data sets for each machine can be reduced as the whole data set is divided and distributed to parallel processors. The observed speedup for the continental US population data set computation is a factor that is close to the number of processors N, for small N (i.e., less than 10 nodes/20 processors). This speedup factor decreases as N increases. This nonlinear effect may be due to the overhead introduced by data communications when a relatively large number of processors are deployed.



```
Algorithm Gravity_based
  For i=0 to total_number_of_polygons                 $t_a$
    For j=0 to total_number_of_polygons               $t_b$
      Calculate distance between polygons             $t_c$
      tmpValue = tmpValue + m_weight[j]/distance      $t_d$
    End For
    m_result[i] = tmpValue                            $t_e$
  End For
End of Algorithm
```

Figure 10. The gravity-based measure algorithm.

```
Read  x, y and weight data
MPI_INIT()
MPI_COMM_RANK (MPI_COMM_WORLD,rank)
MPI_COMM_SIZE (MPI_COMM_WORLD,size)
Blocksize = number_of_polygon/(size -1)
If (rank = 0)
  MPI_BCAST (x, y and weight data to every processor)
For (i= rank * Blocksize to (rank+1) * Blocksize)
  Calculate accessibility
If (rank = 0)
 MPI_GATHER (computation results from every processor)
 Store the results
MPI_FINALIZE()
```

Figure 11. The pseudo codes of MPI programs.

According to Amdahl's law (1967), the speedup factor can be calculated using

$$\text{Speedup}(N) = \frac{T_\text{s} + T_\text{p}}{T_\text{s} + \frac{T_\text{p}}{N}} \tag{4}$$

where T_s is the amount of time used on serial sections of the entire program and T_p is the time on parallel sections. In our experience, the design of the MPI programs can maximize the parallel sections of the program, which can speed up the performance of geoprocessing tasks.

4.8 *GIS middleware*

The spatial Web portal prototype converted a complete GIS process into four component subtasks: data reading/preparation, model computation, results storage, and output visualization. For the cluster or TeraGrid test bed, there are two additional subtasks, data uploading and result downloading. As shown in Table 1, the model computation will be very time intensive if large amounts of geospatial data are processed. In this prototype, only the model computation subtask utilized the parallel computing programming model and was executed by remote clusters and TeraGrid sites. The GIS grid middleware processed the four subtasks of grid computing: data reading/processing, data uploading, data downloading, and results storage (Figure 12).

The geospatial grid middleware performed well in our experiments. The middleware run times for the NCSA Globus experiments (cumulative opportunity measurement) with the continental US population data are shown in Table 6 and Figure 12. The data downloading subtask time varied slightly, while the other three subtasks had very similar run times as the number of nodes was increased from 4 to 24. Among the four subtasks, data reading and processing, and results storage are conducted locally on the GeoGrid server, while data uploading and downloading tasks required dynamic connections to the CSRC cluster and the TeraGrid. The connection's

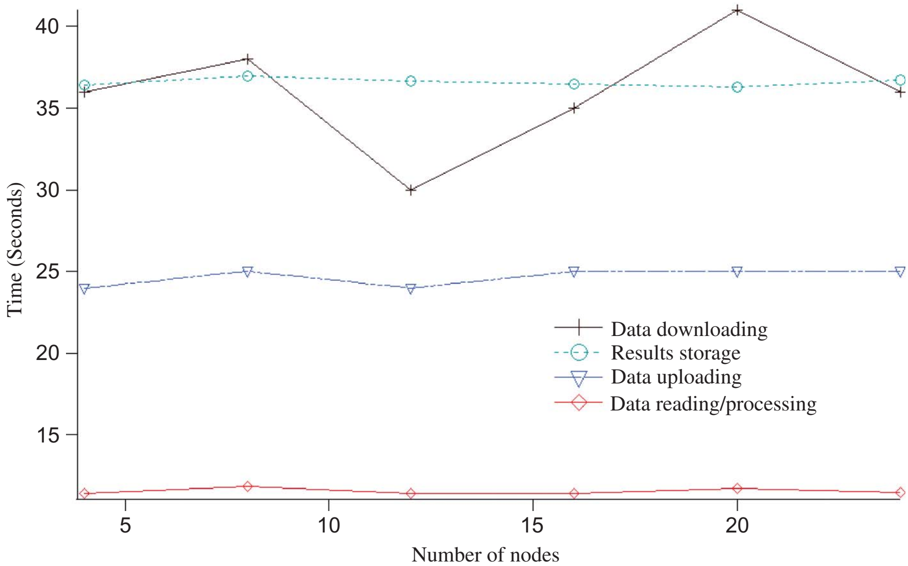


Figure 12. Performance comparison of GIS middleware subtasks.

Table 6. Performance comparison of middleware subtasks.

	Number of nodes (each node has two processors) time (seconds)					
Tasks	4	8	12	16	20	24
Data reading/processing	11.42	11.87	11.41	11.42	11.72	11.49
Data uploading	24.0	25.0	24.0	25.0	25.0	25.0
Data downloading	36.0	38.0	30.0	35.0	41.0	36.0
Results storage	36.41	36.93	36.64	36.46	36.27	36.68

stability between the local machine and the TeraGrid may have a negative impact on the stability of data transfer.

5. Discussion

This section discusses the advantages of the new four-tier grid-enabled GIService framework and the technical challenges of developing a grid-enabled spatial Web portal.

5.1 *The advantages of the grid-enabled GIService framework*

Compared to the current prevailing three-tier client/server Internet GIS architecture, the four-tier framework introduced in this article has several advantages that can be categorized as conceptual, organizational, operational, and implementation aspects.

At the conceptual level, the new framework illustrates the trend toward building a comprehensive geospatial cyberinfrastructure. High-performance grid computing frameworks and geospatial grid middleware can empower Internet GIServices.

Customizable and user-friendly spatial Web portals allow GIS users to build comprehensive GIServices according to their individual needs.

At the organizational level, the framework adopts an open and distributed grid computing framework that can be utilized by numerous Internet users concurrently. The new framework can facilitate the collaboration of geospatial problem solving across multiple communities and disciplines. This framework provides an opportunity for anyone to publish high-performance GIServices and to collaborate in grid-enabled *virtual organizations* (Foster *et al.* 2001) across institutional boundaries.

At the operational level, our framework can support the integration of multiple distributed GIServices. With the traditional client/server architecture, it is difficult to integrate multiple GIS servers or Web services. In the future, multiple traditional GIS servers could be replaced by a single spatial Web portal offering multiple services. A spatial Web portal can direct user requests to multiple grid-enabled computing resources on the grid computing platform while providing better disaster recovery and fault tolerance solutions.

At the implementation level, the new framework provides GIS developers a user-friendly and flexible development environment for creating DGIP applications. Any GIS application can be packaged into a portlet and become widely accessible via a spatial Web portal. The dilemma of thick-client or thin-client solutions no longer exists because most spatial Web portal interfaces will be very thin and connected to very powerful grid computing resources. Interoperable standards (e.g. OGC™ standards) are easy to implement in the grid computing framework.

5.2 *The technical challenges of creating a grid-enabled spatial Web portal*

Several technical challenges remain in producing a spatial Web portal. The lack of commonly accepted geospatial ontology frameworks in GIService discovery and semantic composition is a major obstacle for the implementation of the logic tier. For this reason, the prototype did not deploy a logic tier based on the four-tier GIService framework. In the presentation tier, the spatial Web portal prototype adopted generic Web portal frameworks. We made a special effort to develop and customize geospatial-oriented interfaces with basic mapping functions (i.e., zoom and pan). The generic Web portal framework did not provide any GIS mapping function or GIS data connectivity. One possible mapping solution is to utilize open source Web mapping clients (e.g., Mapbuilder, Ka-map, and CartoWeb), which can provide highly interactive GIS Web portal interfaces. The primary barrier in the service tier is the very limited number of online interoperable GIServices available. Many Internet GIServices are heterogeneous. In the grid tier, the crucial challenge is the development of effective geospatial grid middleware. Geospatial grid middleware should provide a standard Web service interface (based on Web services and OGC standards) to bridge grid-enabled GIS tasks and the underlying grid computing resources (at the TeraGrid sites).

6. Conclusions

This article introduces a new grid-enabled Internet GIService framework that can provide an integrated HPC environment for the future development of geospatial cyberinfrastructure. The four-tier framework was implemented and demonstrated by a Web portal prototype. The performance testing results demonstrated that grid-enabled spatial Web portals can provide high-performance GIServices and geoprocessing functionality.

Our performance experiments generated the following valuable findings:

- When computation tasks get more complex or data size is very large, the run time savings in grid computing systems becomes significant. Therefore, advanced GIS applications may benefit from the use of grid computing systems due to the increasingly complicated nature of GIS studies and use of large data sets.
- In the cross-site grid computing job submission method (MPICH-G2), performance is slower than the single-site job submission method (Globus and PBS) due to the wide area network latency. In the future, as advanced grid computing systems utilize very high-speed networks, such as the National LambdaRail (http://www.nlr.net/), the cross-site grid computing latency problems should be resolved.
- Although the run time of GIS tasks in grid computing can be improved significantly, job queuing time (waiting time) in the TeraGrid is not predictable. Sometimes the job waiting time greatly exceeds the actual run time of GIS tasks. The problem of job submission waiting time may be alleviated by building more grid computing sites/machines or creating a dedicated geospatial grid computing facility to meet computational needs.
- To improve the overall performance of grid-enabled Internet GIServices, we must investigate the bottleneck of GIS middleware, including GIS database access, data transfer methods, GIS model computation, and visualization tools. Advanced high-speed grid-enabled GIS data transfer mechanisms and remote grid-enabled geovisualization methods could improve the overall performance of geoprocessing tasks within the new framework.

Future research topics based on our prototyping experience include the following:

- The new GIService framework should be tested for other geospatial applications (e.g., emergency management or urban planning) to further demonstrate the feasibility and advantages of grid computing for various DGIP applications.
- GIS professionals need to develop new, high-performance geovisualization methods suitable for grid-enabled Web services and mapping services.
- Geospatial grid middleware should be more robust and capable to meet the needs of more diversified DGIP Web GIServices.
- The semantics for GIS Web portals should be fully developed and formalized to provide the mechanisms for match making, discovery, and service chaining for semantic-based Web GIServices.

In summary, the new grid-enabled Internet GIService framework can facilitate the development of low-cost, scalable, and powerful on-line GIS applications and geoprocessing functions. A grid-enabled spatial Web portal can deliver high-performance GIS solutions in a cost-effective manner. We hope that the GIS community, in collaboration with other disciplines, can work together to build a comprehensive and high-performance geospatial cyberinfrastructure for all geographers and other users deploying geographic applications.

Acknowledgments

This work was partially funded by the NASA REASoN-0118-0209 project (A Border Spatial Decision Support System) at San Diego State University and the National

Science Foundation Advanced Technology Education program (NSF DUE #0801893). The authors are grateful to Dr. Qinghua Qiao (Chinese Academy of Surveying and Mapping) for his suggestions and contributions in the implementation of the prototype. The work was also partially supported by the National Basic Research Program of China (2005CB724205), National Natural Science Foundation of China (40830530), and the Open Research Fund of State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing, Wuhan University (09P01). Comments from anonymous reviewers are greatly appreciated.

References

Aloisio, G. and Cafaro, M., 2003, A dynamic earth observation system. *Parallel Computing*, **29**, pp. 1357–1362.

Amdahl, G., 1967, Validity of the single processor approach to achieving large-scale computing capabilities. In *AFIPS Conference*, 18–20 April 1967, Atlantic City, NJ, pp. 483–485 (Reston, VA: AFIPS Press).

Anderson, G. and Moreno-Sanchez, R., 2003, Building web-based spatial information solutions around open specifications and open source software. *Transactions in GIS*, **74**, pp. 447–446.

Anselin, L., Kim, Y.W. and Syabri, I., 2004, Web-based analytical tools for the exploration of spatial data. *Journal of Geographical Systems*, **6**, pp. 197–218.

Armstrong, M.P. and Densham, F.J., 1992, Domain decomposition for parallel processing of spatial problems. *Computers, Environments and Urban Systems*, **16**, pp. 497–513.

Armstrong, M.P. and Marciano, R., 1995, Massively parallel processing of spatial statistics. *International Journal of Geographical Information Systems*, **9**, pp. 169–189.

Armstrong, M.P. and Marciano, R., 1996, Local interpolation using a distributed parallel supercomputer. *International Journal of Geographical Information Systems*, **10**, pp. 713–729.

Armstrong, M.P. and Marciano, R., 1997, Massively parallel strategies for local spatial interpolation. *Computer and Geosciences*, **23**, pp. 859–867.

Baker, M., Buyya, R. and Laforenza, D., 2002, Grid and grid technologies for wide-area distributed computing. *Software: Practice and Experience*, **32**, pp. 1437–1466.

Chen, A., Di, L., Wei, Y. and Bai, Y., 2009, Use of grid computing to model virtual geospatial products. *International Journal of Geographical Information Science*, **23**(5), pp. 581–604.

Chervenak, A., Deelman, E., Kesselman, C., Allcock, B., Foster, I., Nefedova, V., Lee, J., Sim, A., Shoshani, A., Drach, B., Williams, D. and Middleton, D., 2003, High-performance remote access to climate simulation data: a challenge problem for data grid technologies. *Parallel Computing*, **29**, pp. 1335–1356.

Clematis, A., Mineter, M. and Marciano, R., 2003, High performance computing with geographical data. *Parallel Computing*, **29**, pp. 1275–1279.

Czajkowski, K., Foster, I., Karonis, N., Kesselman, C., Martin, S., Smith, W. and Tuecke, S., 1998, A resource management architecture for metacomputing systems. In *4th IPPS/SPDP Workshop on Job Scheduling Strategies for Parallel Processing*, 30 March, Orlando, FL, Lecture notes in Computer Science, Vol. 1459, pp. 62–82 (London: Springer-Verlag).

Ding, Y., Densham, F.J. and Armstrong, M.F., 1992, Parallel processing for network analysis: decomposing shortest path algorithms on MIMD computers. In *5th International Symposium on Spatial Data Handling*, 3–7 August, Charleston, SC, IGU Commission on GIS, pp. 682–691.

Dong, S., Karniadakes, G.E. and Karonis, N.T., 2005, Cross-site computations on the TeraGrid. *Computing in Science & Engineering*, **7**, pp. 14–23.

Foster, I. and Kesselman, C., 1997, Globus: a metacomputing infrastructure toolkit. *International Journal of Supercomputer Applications*, **11**, pp. 115–129.

Foster, I., Kesselman, C., Nick, J. and Tuecke, S., 2002, The physiology of the grid: an open grid services architecture for distributed systems integration. *Open Grid Service Infrastructure Working Group, Global grid Forum*, available online at: http://www.globus.org/alliance/publications/papers/ogsa.pdf

Foster, I., Kesselman, C. and Tuecke, S., 2001, The anatomy of the grid: enabling scalable virtual organizations. *International Journal of Supercomputer Applications*, **15**, pp. 200–222.

Friis-Christensen, A., Lucchi, R., Lutz, M. and Ostlander, N., 2009, Service chaining architecture for implementing distributed geoprocessing applications. *International Journal of Geographical Information Science*, **23**(5), pp. 561–580.

Hansen, W.G., 1959, How accessibility shapes land use. *Journal of American Institute of Planners*, **25**, pp. 73–76.

Healey, R., Dowers, S., Gittings, G. and Mineter, M. (Eds), 1998, *Parallel Processing Algorithms for GIS* (London: Taylor & Francis).

Hu, S. and Bian, L., 2009, Interoperability of functions in environmental models: a case study in hydrological modeling. *International Journal of Geographical Information Science*, **23**(5), pp. 657–681.

Li, B., 1992, Opportunities and challenges of parallel spatial data analysis: initial experiments with data parallel map analysis. In *GIS/LIS 1992*, 10–12 November, San Jose, CA, pp. 445–448 (Bethesda, MD: American Congress on Survey and Mapping).

Li, B., 2000, A component perspective on geographic information services. *Cartography and Geographic Information Science*, **27**, pp. 75–86.

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**, pp. 3–14.

NSF (National Science Foundation), 2003, *Revolutionizing Science and Engineering through Cyber-infrastructure: Report of the National Science Foundation Blue-Ribbon Advisory Panel on Cyberinfrastructure*, available online at: http://dlist.sir.arizona.edu/897

Nyerges, T., Ramsay, K. and Wilson, M., 2006, Design considerations for an Internet portal to support public participation in transportation improvement decision making. In S. Balram and S. Dragicevic (Eds), *Collaborative Geographic Information Systems*, pp. 208–230 (Hershey: Idea Publishing).

Peng, Z.R. and Tsou, M.-H., 2003, *Internet GIS: Distributed Geographic Information Services for the Internet and Wireless Network* (New Jersey: John Wiley & Sons).

Rokos, D. and Armstrong, M.P., 1992, Parallel terrain feature extraction. *In*: *GIS/LIS 1992*, 10–12 November 1992, San Jose, CA. Bethesda, MD: American Congress on Survey and Mapping, 652–661.

Rokos, D. and Armstrong, M.P., 1993, Computing spatial autocorrelation parallel: a MIMD implementation for polygon data. In *GIS/LIS 1993*, 2–4 November, Minneapolis, MN, pp. 621–630 (Bethesda, MD: American Congress on Survey and Mapping).

Scholten, M., Klamma, R. and Kiehle, C., 2006, Evaluating performance in spatial data infrastructure for geoprocessing. *IEEE Internet Computing*, **10**, pp. 34–41.

Tait, M.G., 2005, Implementing geoportals: applications of distributed GIS. *Computers, Environment and Urban Systems*, **29**, pp. 33–47.

Tang, W. and Selwood, J., 2005, *Spatial Portals: Gateways to Geographic Information* (Redlands: ESRI Press).

Tsou, M-H., 2004, Integrating web-based GIS and on-line remote sensing facilities for environmental monitoring and management. *Journal of Geographical Systems*, 6, pp. 155–174.

Tsou, M.-H. and Buttonfield, B.P., 2002, A dynamic architecture for distributing geographic information services. *Transactions in GIS*, **6**, pp. 355–381.

YANG, P., EVANS, J., COLE, M., ALAMEH, N., MARLEY, S. and BAMBACUS, M., 2007, The emerging concepts and applications of the spatial web portal. *Photogrammetric Engineering and Remote Sensing*, **73**, pp. 691–698.

YANG, C. and RASKIN, R., 2009, Introduction to distributed geographic information processing (DGIP) research. *International Journal of Geographical Information Science*, **23**(5), pp. 553–560.

WACHS, M. and KUMAGAI, T., 1973, Physical accessibility as a social indicator. *Socioeconomic Planning Science*, **7**, pp. 437–456.

WANG, S. and ARMSTRONG, M.P., 2003, A quadtree approach to domain decomposition for spatial interpolation in grid computing environments. *Parallel Computing*, **29**, pp. 1481–1504.

WANG, S. and LIU, Y., 2009, TeraGrid GIScience gateway: bridging cyberinfrastructure and GIScience. *International Journal of Geographical Information Science*, 23(5), pp. 631–656.

ZHANG, T. and TSOU, M.-H., 2005, The integration of grid-enabled internet GIServices and geographic semantic web technologies. *Geographic Information Science*, **11**, pp. 15–23.