

# GEOPROCESSING IN THE MICROSOFT CLOUD COMPUTING PLATFORM - AZURE

Jianya Gong, Peng Yue, Hongxiu Zhou

State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing (LIESMARS), Wuhan University, 129 Luoyu Road, Wuhan, China, 430079  
geogjy@163.net, geopyue@gmail.com, zhouhx0623@yahoo.com.cn

**Commission IV, WG IV/4**

**KEY WORDS:** Geoprocessing, Cloud Computing, Geospatial, Service, GIS, Azure

## **ABSTRACT:**

Geoprocessing functions are important for discovering hidden and useful geospatial information, and are widely used in Earth science modeling and applications. The growth of the Web has resulted in the Web-based sharing of large volumes of distributed geospatial data and computational resources. Processing heterogeneous and distributed data into information requires a powerful, scalable, and interoperable information infrastructure. The emergence of Cloud Computing technology brings a new computing Information Technology (IT) infrastructure to general users. Using these Cloud Computing platforms, typical IT resources such as storage, computing utilities, and database are available as services. This paper presents an implementation of geoprocessing service that integrates geoprocessing functions and Microsoft Cloud Computing technologies to provide geoprocessing capabilities in a distributed environment. The integration combines the conventional powerful geospatial analysis functions with the scalable Cloud Computing technology to provide Web-based geoprocessing functionalities. Challenges and approaches on the integration are discussed. The implementation is demonstrated in the Microsoft Cloud Computing Platform – Azure.

## **1. INTRODUCTION**

Geoprocessing functions are important for discovering hidden and useful geospatial information, and are widely used in Earth science modeling and applications. The geospatial analysis functions in the Geographical Information Science (GIS) have been developed over decades and can be used in the desktop-based environment. The growth of the Web has resulted in the Web-based sharing of large volumes of distributed geospatial data and computational resources. Processing heterogeneous and distributed data into information requires a powerful, scalable, and interoperable information infrastructure.

The emergence of Cloud Computing technology brings a new computing information technology (IT) infrastructure to general users. Some Cloud Computing platforms are already available, including Microsoft Azure, Google App Engine, Amazon EC2 & S3. Using these Cloud Computing platforms, typical IT resources such as storage, computing utilities, and database are available as services. The key feature of the Cloud Computing technology is that it hides the underlying complexity of using IT resources, while at the same time bring the scalable, reliable, sustainable, ready-to-use, on-demand and cost-effective IT services to the general public.

This paper presents an implementation of geoprocessing service that integrates geoprocessing functions and Microsoft Cloud Computing technologies to provide scalable geoprocessing capabilities in a distributed environment. The integration combines the conventional powerful geospatial analysis

functions with the scalable Cloud Computing technology to provide Web-based geoprocessing functionalities. Challenges and approaches on the integration are discussed. The implementation is demonstrated in the Microsoft Cloud Computing Platform – Azure. The reminder of this paper is organized as follows: Section 2 introduces related work. Section 3 presents challenges to be solved when moving conventional geoprocessing functions into the Microsoft Azure platform. Section 4 discusses approaches on the integration and Section 5 describes the system implementation details. Conclusions and pointers to future work are given in Section 6.

## **2. RELATED WORK**

Geoprocessing services provide building blocks for higher geoprocessing modeling and complex geoprocessing tasks, and are the fundamental utilities for the Geospatial Processing Web (Yue et al., 2009). Geoprocessing is also a core component in Geospatial Cyberinfrastructure (GCI) (Yang et al., 2010), which must support geospatial data processing within and across scientific domains. Cloud Computing is associated with a new paradigm to provide computing infrastructure (Vaquero et al. 2009), which can support geoprocessing over the Web. Several new aspects of the information infrastructure enabled by Cloud Computing are the illusion of infinite computing resources available on demand, the elimination of an up-front commitment by cloud users, and the ability to pay for use of computing resources on a short-term basis as needed (Armbrust et al., 2009).

A cloud could be public or private, depending on the deployment approaches. The public cloud is available to the public, while the private cloud is used inside an organization. According to the type of provided capabilities, major categories of Cloud Computing include Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) (NIST, 2009). Several large Information Technology (IT) companies such as Google and Amazon have already created their own cloud products to offer services in these categories. The well-known successful examples are Elastic Compute Cloud (EC2) and Simple Storage Service (S3) from Amazon Web Services (AWS). The Google App Engine can host Web applications developed using Java and Python, and allow users to develop, deploy, and try new applications.

Several efforts on using Cloud Computing in geospatial applications have been reported. Baranski et al. (2009) creates a cloud enabled spatial buffer analysis service in the Google App Engine and conducts a stress test for scalability evaluation. The interface of the service follows the WPS specification and the implementation is based on the Java-implemented 52North open source WPS software. Wang et al. (2009) proposes the use of OGC as Well Known Binary (WKB) and Well Known Text (WKT) for data exchange in the Cloud Computing environment, and demonstrates how spatial indexes can be created in the Google App Engine. Schaeffer et al. (2010) discusses the cloud-enabled Spatial Data Infrastructures (SDIs). The Amazon Web Services (AWS) together with an OGC Web Processing Service implementation hosting a buffer and intersection process is used in a public risk management use case. Our work follows the same idea in using interoperable WPS interface for geoprocessing services, yet choosing the Windows Azure platform as the Cloud Computing platform for discussion and proof-of-concept implementation.

### 3. CHALLENGES

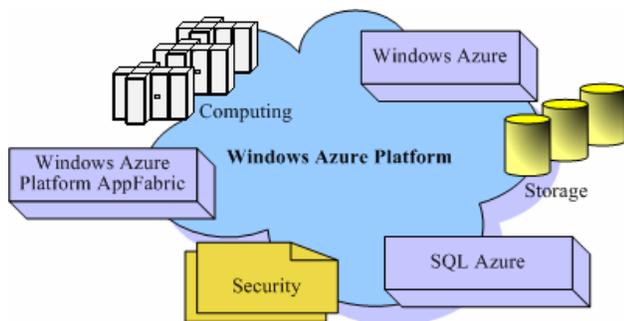


Figure 1. The Windows Azure platform

The spatial analysis functions in professional GIS software have been developed for over decades. However, the GIS software cannot support an open analysis environment. The analysis functions can only be used in their own proprietary environments. Taking a terrain slope computation as an example, supposing a raster map of slope can be generated from a raster map for the Digital Elevation Model (DEM) data, using

an analysis algorithm written as a Java application. This legacy application works only in a Java Runtime Environment (JRE).

The Microsoft Azure platform consists of the following components (Chappell, D., 2009) (Figure 1):

- Windows Azure: Provides a platform for running Windows applications and storing their data in the cloud. Windows Azure runs on a large number of servers in Microsoft data centres, which are connected into a unified whole by Windows Azure Fabric. Windows-based compute and storage services for cloud applications are built on top of this fabric.
- SQL Azure: Provides data services in the cloud based on SQL Server. Although the eventual goal of SQL Azure includes a range of data-oriented capabilities, including data synchronization and reporting, the first SQL Azure component to appear is SQL Azure Database, which provides a cloud-based database management system (DBMS).
- Windows Azure platform AppFabric: Provides cloud services for connecting applications running in the cloud or on premises. The functions provided by AppFabric are called cloud-based infrastructure services including Service Bus and Access Control Service.

One challenge is that the conventional spatial analysis application needs to be adapted to the running environment in the Cloud Computing environment. In particular, how various components in the Microsoft Azure platform can be used to support functional requirements and processes in scientific analysis and discovery, such as data retrieval and geocomputation?

The other challenge is about the interoperable service interface that geospatial applications interact with cloud services. Geospatial data is collected by diverse means and is highly complex and heterogeneous. Often, the temporal and spatial coverage, resolution, origin, format, and map projections are incompatible. Because of the multidisciplinary nature of Earth science modeling and applications, geoprocessing functions involved are always diverse. Interoperability issue, therefore, is important in a distributed environment. One possible solution is to promote the use of the Open Geospatial Consortium (OGC) standards, which allows the seamless access to distributed geospatial data and geoprocessing functions in a distributed environment, regardless of heterogeneity of involving geospatial data and geoprocessing platforms.

### 4. APPROACHES

Based on the preliminary analysis of the components in Microsoft Azure platform and geospatial analysis applications, we propose that the integration of geoprocessing functions and Microsoft Azure platform takes place at three points, in order to address challenges in Section 2. Figure 2 shows the proposed architecture of the integrated system.

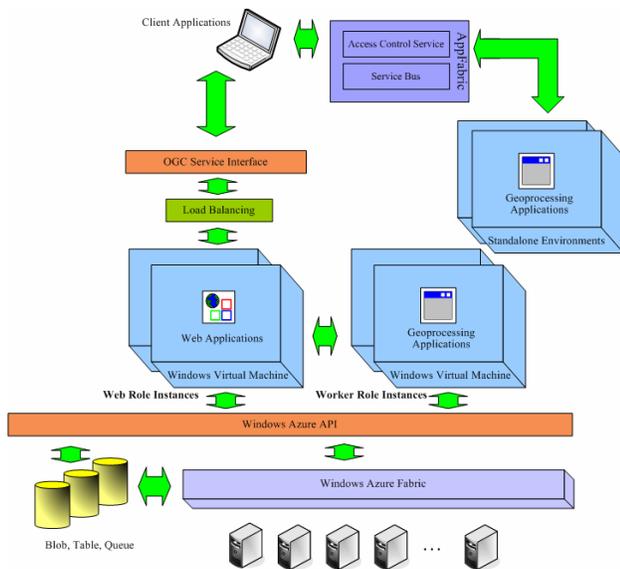


Figure 2. A high level framework for geospatial analysis in Windows Azure Platform

The first one is to move computational applications into the compute service in the Microsoft Azure. Conventional analysis applications work in their own standalone environments. The applications in the Windows Azure platform differentiate the Web roles and Worker roles. An application with a Web role is a Web application supported by Internet Information Services (IIS), which can be implemented using ASP.NET or Windows Communication Foundation (WCF). An application with a Worker role is used for general development of applications in the Cloud, and can perform background processing for a web role. The implementation of Worker role applications can use .Net code such as C#, Visual Basic, or F#. Both Web and Worker role applications can have multiple instances on Windows Azure, each running in its own Windows virtual machine (VM). The VMs are maintained by Windows Azure in the background and allows role applications to have access to resources in the data servers managed by Windows Azure Fabric via Windows Azure API. One advantage of using Windows Azure for providing geoprocessing services is that Windows Azure provides built-in load balancing to Web applications when multiple Web role instances of the same application are running.

The key development to integrate geospatial analysis functions with Windows Azure are then the development of Worker role applications for background geoprocessing and expose the Web service interface via Web role applications. The development of Worker and Web role applications uses the Windows Azure API. The service interface for geoprocessing cloud services can adopt the OGC Web Processing Service (WPS) standard. OGC WPS specifies a standard interface and protocol for discovering and executing distributed geoprocessing processes. The three mandatory operations included in the standard WPS interface are GetCapabilities, DescribeProcess, and Execute (Schut 2007). The GetCapabilities operation allows a client to request and

receive a service capabilities document that describes the operations and processes of a specific WPS implementation. The DescribeProcess operation allows a client to get detailed information, such as input and output parameter types, about specific processes. The Execute operation allows the client to run a specific process in a WPS server. The use of HTTP protocol, in particular the GET and POST operations, to exchange XML-based request and response messages, is compatible with the RESTful API of services in Windows Azure.

The second point is to use storage services in the Microsoft Azure to manage the application data. The storage service in Microsoft Azure offers several ways including blobs, tables, and queues to store data. The blobs store binary or text data. The queues store messages for communication between components of Windows Azure applications such as the communication between Web role instances and Worker role instances. The tables provide a structured way to store non-relational data. For the relational data, the Microsoft SQL Azure extends SQL Server capabilities to the cloud and offers a relational database service. All stored data are replicated in the Windows Azure platform to allow the fault tolerance. Geospatial data that is required for geoprocessing, therefore, can be stored in Windows Azure for robust storage services. The access to Windows Azure storage uses the RESTful API.

The third one is to connect cloud-based services and legacy geoprocessing applications using the AppFabric. The Service Bus component in the AppFabric provides a facility for identifying service endpoints, publishing them, routing messages among them, and connecting them despite common Internet connectivity challenges such as translations between external and internal network addresses. The Access Control Service helps application services authenticate and receive identity information for their client applications. These cloud-based infrastructure services allows users to connect easily distributed applications, whether in a standalone environment or in the cloud. Legacy geoprocessing applications, therefore, can be reused by exposing itself via the Service Bus.

## 5. IMPLEMENTATION

The Windows Azure Tools for Microsoft Visual Studio 1.2 (June 2010 version) is used to develop geoprocessing services in Windows Azure. The tools is combined with Visual Studio 2010, SQL Server 2005 Express, and IIS 7.0 to provide a Windows Azure development environment including development fabric and storage utilities, which can simulate the Windows Azure compute and storage services in the desktop machines. Thus the tools allow the creation, building, debugging, and running of cloud applications in a simulated Windows Azure environment. Once these services pass the tests in the development environment and are ready to be used by public, they can be packaged and deployed on Windows Azure by using utilities in the tools. Since Windows Azure platform is a commercial product and does not have free offers, the implementation here is tested only in the Windows Azure development environment for academic purposes.

The implementation includes the following steps, using the terrain slope computation as an illustration.

### 5.1 Creating application framework for Web and Worker role

The Windows Azure Tools for Microsoft Visual Studio provides a project template for developing a cloud service. A cloud service solution containing a cloud service project can be generated using the project template. Additional Work and Web role projects for development of background geoprocessing and Web service interface are added when creating the cloud service solution. The cloud service project in the cloud service solution includes the service definition and service configuration files. The service definition file specifies the roles that comprise a service and optional local storage resources, and the service configuration file.

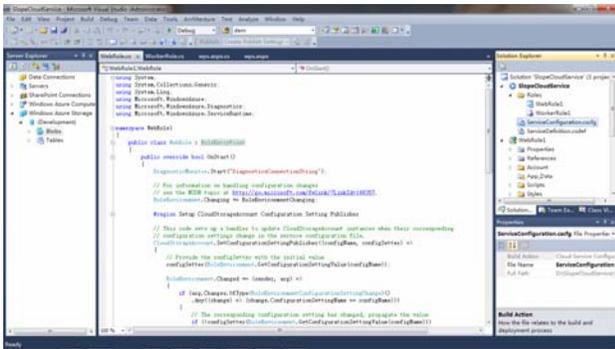


Figure 3. Interface of cloud service development in Microsoft Visual Studio

As shown in Figure 3, three projects, namely SlopeCloudService, WebRole1, and WorkerRole1, are included in the solution for the cloud service of terrain slope computation. The files with names ServiceDefinition.csdef and ServiceConfiguration.cscfg are service definition and service configuration files respectively.

### 5.2 Developing WPS interface for Web role application

According the WPS specification, the interface includes three operations: GetCapabilities, DescribeProcess, and Execute, which can be invokes via HTTP GET method with a Key-Value-Pair (KVP) encoded request, or HTTP POST method with a XML-based encoding. The following is an implementation of the KVP encoded DescribeProcess request for retrieving the process description of the terrain slope computation:

```
http://127.0.0.1:81/
wps.aspx?Request=DescribeProcess&Service=WPS&Version=
0.4.0&Identifier= DEM2SlopeProcess
```

Table 1 shows the XML encoding of the Execute request for invoking the WPS process DEM2SlopeProcess. The processing of HTTP requests in the Web role can use a HTTP handler from

ASP.NET, which calls the ProcessRequest method to process WPS requests and returns the execution response in XML. One example on execution using HTTP GET in a Web browser is shown in Figure 4. The input data and output data are shown in the Geomatica FreeView software, attached on sides of the Web browser.

Table 1. WPS Request in XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Execute xmlns="http://www.opengeospatial.net/wps"
xmlns:ows="http://www.opengeospatial.net/ows"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
service="WPS" status="false" store="false"
version="0.4.0">
  <ows:Identifier>DEM2SlopeProcess</ows:Identifier>
  <DataInputs>
    <Input>
      <ows:Identifier>InputData</ows:Identifier>
      <ows:Title>DEM data</ows:Title>
      <Complex Value
schema="http://www.lmars.whu.edu.cn/ws/iodata.xsd">
        <ComplexData Value
xmlns="http://www.lmars.whu.edu.cn/ws"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:ows="http://www.opengis.net/ows">
          <Reference xlink:href="
http://127.0.0.1:10000/devstoreaccount1/dem/163419559244
0084157_654d2f37-899a-4c04-bda2-c118265fa3f8.bil">
            <ows:Format>BIL(.bil)</ows:Format>
          </Reference>
        </ComplexData Value>
      </Complex Value>
    </Input>
  </DataInputs>
  <OutputDefinitions>
    <Output>
      <ows:Identifier>OutputData</ows:Identifier>
    </Output>
  </OutputDefinitions>
```

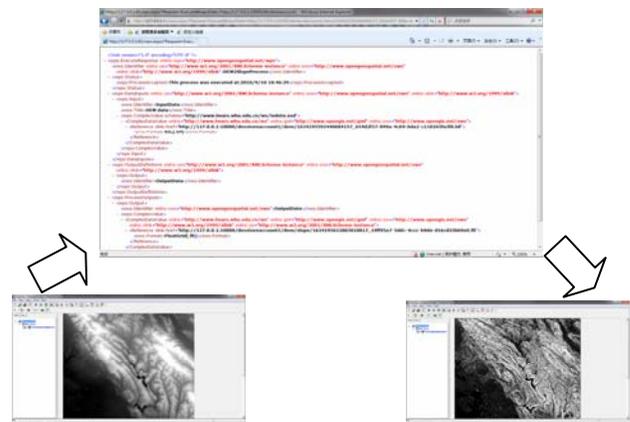


Figure 4. WPS Execution request using HTTP GET method

### 5.3 Providing message queue and blob storage

Figure 5 shows the sequence diagram of processing a WPS Execution request. The Blob service is used for robust data storage. In the blob service, blobs are organized into a container beneath a storage account. The geospatial data can be available as either a Web URL to a downloadable data product or URI to the address of a blob. In the former case, the data need to be downloaded and stored into blobs. In the latter one, the blob can be access directly in the same storage account. The Queue service is used to store communication messages between Web and Worker roles. When an Execution request comes, container and queue are created if they do not already existed. The data in the WPS request is loaded into blobs of the container. The path to the blob is recorded into a message and added into the queue.

The Worker role detects the message in the queue as a job, and handles it using domain business logics, including extracting the path to the blob from the message, accessing data in the blob, invoking a new process for geoprocessing, uploading processing result into the blob belonging to a new container, and deleting the old message. The Windows Azure Tools provides the Windows Azure Storage Explorer to view read-only blob and table data, as shown in the left panel of Figure 3.

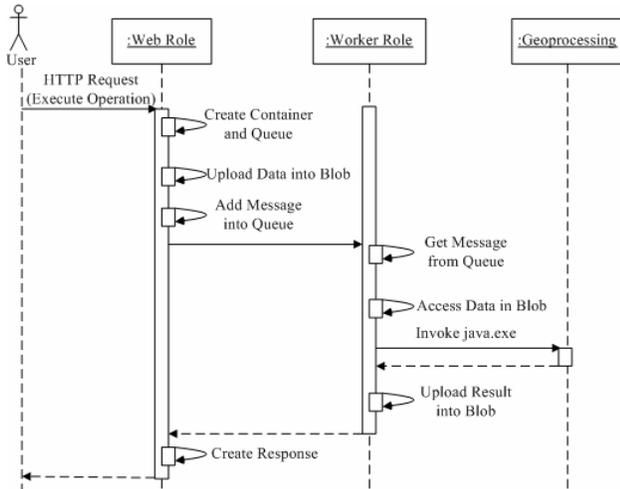


Figure 5. Workflow on using storage services in Web and Worker roles

### 5.4 Implementing geoprocessing algorithm in the Worker role application

It usually requires the rewriting, compiling, or building computer source code, when migrating legacy geoprocessing programs into Cloud Computing platforms. For example, the Windows Azure does not support Java directly. However, Marx (2009), a developer of Microsoft Windows Azure, suggests an approach to run Java applications in Windows Azure by taking advantage of VMs in Windows Azure. The JRE is copied and deployed into the Windows Azure so that the Java code can work by invoking java.exe using the system process. Thus Java code for the slope computation does not need to be modified

and can work in Windows Azure. It is noted that we do not choose the Java when implementing WPS operations in the Web role, because it is inefficient to invoke a Java process on each Web request, while in the worker role, invoking an executable is reasonable (Marx, 2009).

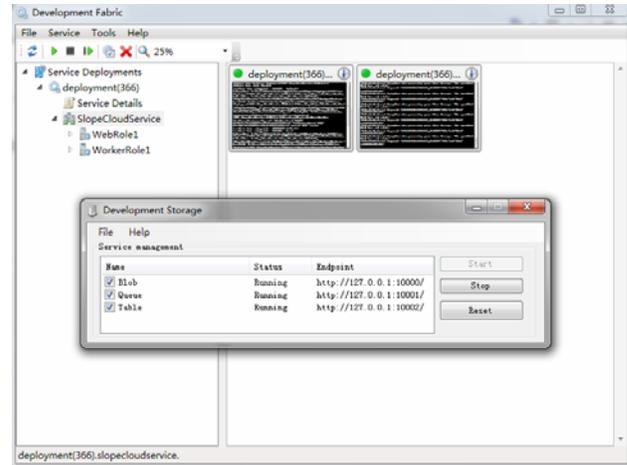


Figure 6. Development fabric user interface

Figure 6 shows user interfaces (UI) for the development fabric and development storage, which demonstrates the local test for running the cloud service in a simulation environment. The left panel of the development fabric UI shows cloud services deployed and role instances that are running. Clicking on each of them can display the log information in the right pane.

## 6. CONCLUSIONS AND FUTURE WORK

This paper describes the design and implementation of geoprocessing services in the Microsoft Cloud Computing platform – Azure. Approaches on how various components in the Microsoft Azure platform can be used to address the storage and computing demands of geoprocessing services are presented. A prototype system demonstrates the implementation of Worker role for background processing, the use of Web role for exposing standard interface, and the development of message queue and blob storage for internal message exchange and data access.

Although the implementation is tested currently in the Windows Azure development environment, the implementation paves the way for deployment into the commercial Windows Azure platform. By adopting standard interfaces for cloud-based geoprocessing services, the system will enhance the common good of Earth science research and educational users in that they will have a geoprocessing platform that is easily accessible and enjoy the benefits of the new IT infrastructure including the load balancing, data replication, and resource management.

The migration of conventional geoprocessing functions into a Cloud Computing platform requires considerable work, since various Cloud Computing platforms have their own APIs and

deployment requirements. The vendor specific APIs complicate the development of geoprocessing services in different Cloud Computing platform. Although the Cloud Computing interoperability is still an open issue, we take advantage of the platform independent language – Java and write the geoprocessing functions in Java. The work demonstrates that Java Runtime Environment (JRE) can be deployed into the Windows Azure with Java applications. It also shows that a platform independent programming language in the conventional computing environment will play an important role in the Cloud Computing environment. Although many traditional geoprocessing algorithms are written in other programming languages such as C for execution efficiency, the on-demand allocation of computing resources in Cloud Computing environment allows us to change focus on system development: from efficiency to interoperability. Since other major cloud vendors including Google and Amazon support the Java developers, future work will test the Java implementations in other Cloud Computing platforms to demonstrate the easy migration of domain business logic.

## REFERENCES

- Armbrust, M., Fox, A., Griffith, R., Joseph, Anthony D., Katz, Randy H., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., and Zaharia, M., 2009. Above the Clouds: A Berkeley View of Cloud Computing. Technical Report. EECS Department, University of California, Berkeley. 23 pp.
- Baranski, B., Schaeffer, B., and Redweik, R., 2009. Geoprocessing in the Clouds. In: GeoInformatics, van Rees, E. (Ed.), Volume 12, pp. 386-395.
- Chappell, D., 2009. Introducing the windows azure platform. Chappell & Associates. 21pp. Whitepapers, Available from: <http://www.microsoft.com/windowsazure/> [accessed 23 March 2010].
- Marx, S., 2009. Does Windows Azure Support Java? Available from: <http://blog.smarx.com/posts/does-windows-azure-support-java> [accessed 10 July 2010].
- NIST, 2009. National Institute of Standards and Technology (NIST) Definition of Cloud Computing. , Available from: <http://csrc.nist.gov/groups/SNS/cloud-computing/index.html> [accessed 16 August 2010].
- Schaeffer, B., Baranski, B., and Foerster, T., 2010. Towards Spatial Data Infrastructures in the Clouds. In : *Proceedings of the 13th AGILE International Conference on Geographic Information Science*, Guimarães, Portugal, Painho, M., Santos, M., and Pundt, H., (Eds.), Geospatial Thinking, Lecture Notes in Geoinformation and Cartography, Springer Verlag, pp. 399-418.
- Schut, P., 2007. OpengGIS® Web processing service. Version 1.0.0, OGC 05-007r7, Open Geospatial Consortium, Inc.. 87 pp
- Vaquero, L. M., Rodero-Merino, L., Caceres, J., and Lindner, M., 2009. A break in the clouds: towards a cloud definition.

SIGCOMM Computer Communication Review, Vol. 39, pp. 50-55.

Wang, Y., Wang, S., and Zhou, D., 2009. Retrieving and Indexing Spatial Data in the Cloud Computing Environment, In: *Proceedings of the 1st International Conference on Cloud Computing (CloudCom 2009)*, Jaatun, M.G., Zhao, G., and Rong, C., (Eds.), Lecture Notes in Computer Science (LNCS), vol 5931, pp. 322–331.

Yang, C. Raskin, R., Goodchild, M., and Gahegan, M., 2010. Geospatial cyberinfrastructure: past, present and future. *Computers. Environment and Urban Systems*, 34(4), pp. 264-277.

Yue, P., Gong, J., Di, L., Yuan J., Sun L., and Wang Q., 2009. GeoPW: Towards the Geospatial Processing Web. In: *9th International Symposium on Web and Wireless Geographical Information Systems (W2GIS 2009)*. 7 & 8 December 2009, Maynooth, Ireland, Lecture Notes in Computer Science (LNCS), vol 5886. Springer, Berlin, pp 25-38.

## ACKNOWLEDGEMENTS

This work was funded jointly by National Natural Science Foundation of China (No. 40801153, 40721001), National Basic Research Program of China (No. 2006CB701304), 863 Program of China (2007AA120501), LIESMARS and SKLSE (Wuhan University) Special Research Fundings.