

Low-cost cloud computing solution for geo-information processing

GAO Pei-chao(高培超)^{1,2}, LIU Zhao(刘钊)², XIE Mei-hui(谢美慧)², TIAN Kun(田琨)²

1. Department of Land Surveying and Geo-Informatics, Hong Kong Polytechnic University,
Kowloon, Hong Kong, China;

2. Institute of Geomatics, Department of Civil Engineering, Tsinghua University,
Beijing 100084, China

© Central South University Press and Springer-Verlag Berlin Heidelberg 2016

Abstract: Cloud computing has emerged as a leading computing paradigm, with an increasing number of geographic information (geo-information) processing tasks now running on clouds. For this reason, geographic information system/remote sensing (GIS/RS) researchers rent more public clouds or establish more private clouds. However, a large proportion of these clouds are found to be underutilized, since users do not deal with big data every day. The low usage of cloud resources violates the original intention of cloud computing, which is to save resources by improving usage. In this work, a low-cost cloud computing solution was proposed for geo-information processing, especially for temporary processing tasks. The proposed solution adopted a hosted architecture and can be realized based on ordinary computers in a common GIS/RS laboratory. The usefulness and effectiveness of the proposed solution was demonstrated by using big data simplification as a case study. Compared to commercial public clouds and dedicated private clouds, the proposed solution is more low-cost and resource-saving, and is more suitable for GIS/RS applications.

Key words: cloud computing; geo-information processing; geo-processing

1 Introduction

Cloud computing [1–2] is an emerging paradigm for storing and analyzing geographic information (geo-information) [3]. In this paradigm, distributed resources (including networks, servers, storage, applications, and services) are gathered as a shared pool of resources, and then provided to consumers on a pay-per-use basis [4]. Since cloud computing has characteristics of on-demand service, broad network access, resource pooling, rapid elasticity, and measured service [4], an increasing number of geo-information processing (geo-processing) tasks are running on clouds [5].

Cloud computing makes it possible to process big geo-information in an effective and efficient way. For example, as a powerful decision support systems tool, spatial on-line analytical processing (SOLAP) was utilized to explore the multidimensional perspective of spatial data, but it failed to deal with big data. A cloud-based SOLAP successfully analyzed spatial data which is long time-series, wide-range and large-scale [6]. For existing geo-processing tasks, the attraction of cloud computing is its powerful computing capability, which helps to improve the efficiency of these tasks. For instance, ADDAIR et al [7] accelerated the analysis of

seismic signals, a global data set of seismograms, based on cloud computing. WANG et al [8] proposed a cloud-based method for rapid processing of remote sensing images. They compared the maximum likelihood classification (MLC) in three modes, i.e., sequential, parallel, and cloud computing, and found that the cloud-based approach performed best.

There are also many geo-processing studies that adopted cloud storage services, which are not only scalable but also stable. For example, LEE and LIANG [9] developed a location data service, Geopot, based on cloud storage. Geopot enables global access, that is, users all over the world can access the location data with the same quality of service. LI et al [10] successfully utilized cloud storage and parallel computing when dealing with urban traffic data. FUJIOKA et al [11] stored their ocean biogeographic information which contains 31.3 million observations in private cloud storage. LIN et al [12] stored remote sensing data in a novel platform which was characterized by cloud computing. TANG and FENG [13] deployed vector-based big spatial data in cloud storage, in order to improve the efficiency of a map projection task for this data.

There is no doubt that the efficiency of geo-processing tasks has been improved by adopting a

cloud-based strategy. However, this strategy has drawbacks as well as merits. Adopting a cloud-based strategy means to rent commercial public clouds or establish dedicated private clouds, both of which require investment and lead to a waste of resources to a certain degree. In order to solve this problem, this work proposes a low-cost cloud computing solution, which is able to save computing resources by making full use of existing physical hardware in a geographic information system/remote sensing (GIS/RS) computer laboratory.

The rest of this paper is organized as follows. The next section outlines the challenge to cloud-based geo-processing tasks. Section 3 proposes a novel and convenient solution. Section 4 uses big data simplification as a case study to illustrate the usefulness and effectiveness of the proposed solution. The merits of the proposed solution are discussed in Section 5. Finally, Section 6 draws conclusions.

2 Challenge to cloud-based geo-processing

Even though there are an increasing number of geo-processing tasks completed or being completed using platforms based on cloud computing, cloud-based geo-processing still has challenges. Table 1 lists some cloud-based geo-processing studies reported in recent years. An investigation was conducted on the properties of the cloud computing environments adopted by these studies, including the type of cloud computing (public/private), configuration and whether parallel computing is enabled

Table 1 Some cloud-based geo-processing studies reported in recent years

Ref.	Public/ Private cloud	Configuration	Parallel- enabled
[10]	Public	(ECU×4+7.5 GB RAM)×28	Yes
[11]	Public	(ECU×4+15 GB RAM)×1	No
[14]	–	(8 CPUs)×8+(16 CPUs)×1+ (32 CPUs)×1+(64 CPUs)×1	Yes
[8]	Private	(Pentium 4 CPU, 0.5 GB RAM)×4	Yes
[15]	Public	(ECU×4+7.5 GB RAM)×1	No
[12]	–	(3.10 GHz CPU, 12 GB)×1+ (3.10 GHz CPU, 12 GB)×3	Yes
[16]	Public	(ECU×4+7.5 GB RAM)×1	No
[6]	–	(2.4 GHz CPU×8, 12 GB RAM)×8	Yes
[13]	Private	(2.93 GHz CPU, 4 GB RAM)×1	No
[17]	Both	(3.6 GHz CPU, 7.8 GB RAM)×1+ (2.4 GHz CPU, 5.0 GB RAM)×1+ (2.4 GHz CPU, 7.5 GB RAM)×1+ (2.4 GHz CPU, 7.5 GB RAM)×7	Yes

Note: ECU is short for Amazon EC2 Compute Unit. One ECU provides the equivalent CPU capacity of a 1.0–1.2 GHz 2007 Opteron or 2007 Xeon processor [18]; “()” stands for a virtual machine and “–” denotes “unknown”.

or not.

Except for a few studies that utilized cloud computing for a data storage purpose only, most of the cloud-based geo-processing applications shown in Table 1 also enabled parallel computing. Applications that are parallel-enabled can take advantage of computing power, which is the most important strength of clouds. The common grounds of the majority of cloud-based geo-processing tasks are as follows:

1) The cloud platform is established based on high-end servers, as opposed to ordinary desktops.

2) The task runs on a cluster of physical or virtual machines, in order to deal with geo-information in a parallel diagram.

Therefore, it is wise to set up a temporary computing environment by renting public cloud resources, when executing cloud-based tasks for processing geo-information, especially large-scale geo-information. Public clouds are off-premises and relatively low-cost. In the case that geo-information is highly privacy-sensitive, it is necessary to establish an on-premises cloud computing platform for private usage [19], by purchasing and virtualizing expensive hardware including servers, switches, and storage. In the field of GIS/RS, however, most users do not deal with big data every day. They are likely to process big data only a few times a year, so it is not economical to build a dedicated cloud, neither a public one nor a private one, for the following reasons:

1) Public cloud computing is an inexpensive choice as reported, but it is usually not cost-efficient. Take amazon web services (AWS) for example, as the first and largest public cloud provider in the world [20], AWS has launched a variety of cloud services. The most cost-efficient cloud service is called a “reserved instance”, which can be rented by consumers for one or three-year terms. However, the term is too long for common use. If, for example, GIS/RS researchers execute a parallel computing task on a cluster of reserved instances, they need to pay for at least one year, even though the task is completed in several minutes.

2) The cost of establishing a private cloud is higher than that of a public cloud. Users are required to purchase both hardware and software to establish a cloud. Hardware refers to minimal machines, blade servers, ordinary rack servers, high-end rack servers, switches, storage, cabinets, etc. Software includes virtualization software, databases, operating systems, and GIS professional software. In the case of privacy-sensitive geo-information, private cloud computing is likely to be the only option for utilizing cloud resources. However, it is a waste to establish a dedicated private cloud for a temporary task, because the cloud resources will become idle after completing the task.

Therefore, the use of clouds not only failed to save, but also resulted in a waste of computing resources. From the economic point of view, the challenge to cloud-enable geo-processing is how to employ cloud computing on a “real” pay-per-use basis, which is one of the core ideas behind clouds.

3 A novel convenient framework

3.1 Design

Inspired by volunteer computing, we propose a hosted Hadoop-based cloud computing framework termed “GRCloud”, which is designed for large-scale geo-information processing tasks in GIS/RS.

Volunteer computing allows Internet users to share their idle computing resources voluntarily, in order to help accelerate compute-intensive scientific projects [21]. The most well-known platform developed for volunteer computing is BOINC [22], which has been utilized for many scientific projects. For example, SETI@home [23], a project that employed volunteers’ personal computers to search for extraterrestrial intelligence. To participate in volunteer computing, Internet users only need to install a piece of software, which runs in the background or as a screensaver. The software does not interfere with normal usage. Since a volunteer computing project requires a

large number of volunteer nodes, a lack of available nodes can severely limit the potential of volunteer computing. On the other hand, most GIS/RS departments in universities and colleges have set up laboratories with a large number of computers installed with professional GIS/RS software. These computers can serve as volunteer computing nodes for large-scale GIS/RS projects. In this work, we employ these computers as a part of the GRCloud, in order to establish a low-cost cloud computing environment for geo-processing tasks. The GRCloud consists of four layers: an infrastructure layer, a cloud computing layer, an application layer and a management layer, as shown in Fig. 1.

The infrastructure layer refers mainly to the computers in a GIS/RS laboratory. These computers are usually installed with 32-bit Windows operating systems, and are equipped with professional software, such as ArcGIS and Erdas. All the computers are connected to a switch with Ethernet cables. By utilizing the idle resources of these computers, this infrastructure layer provides upper layers with a Windows-based software operating environment and resources including computing, storage and network.

The cloud computing layer (or cloud layer) virtualizes the underlying physical hardware, and provides a unified interface to employ the virtualized

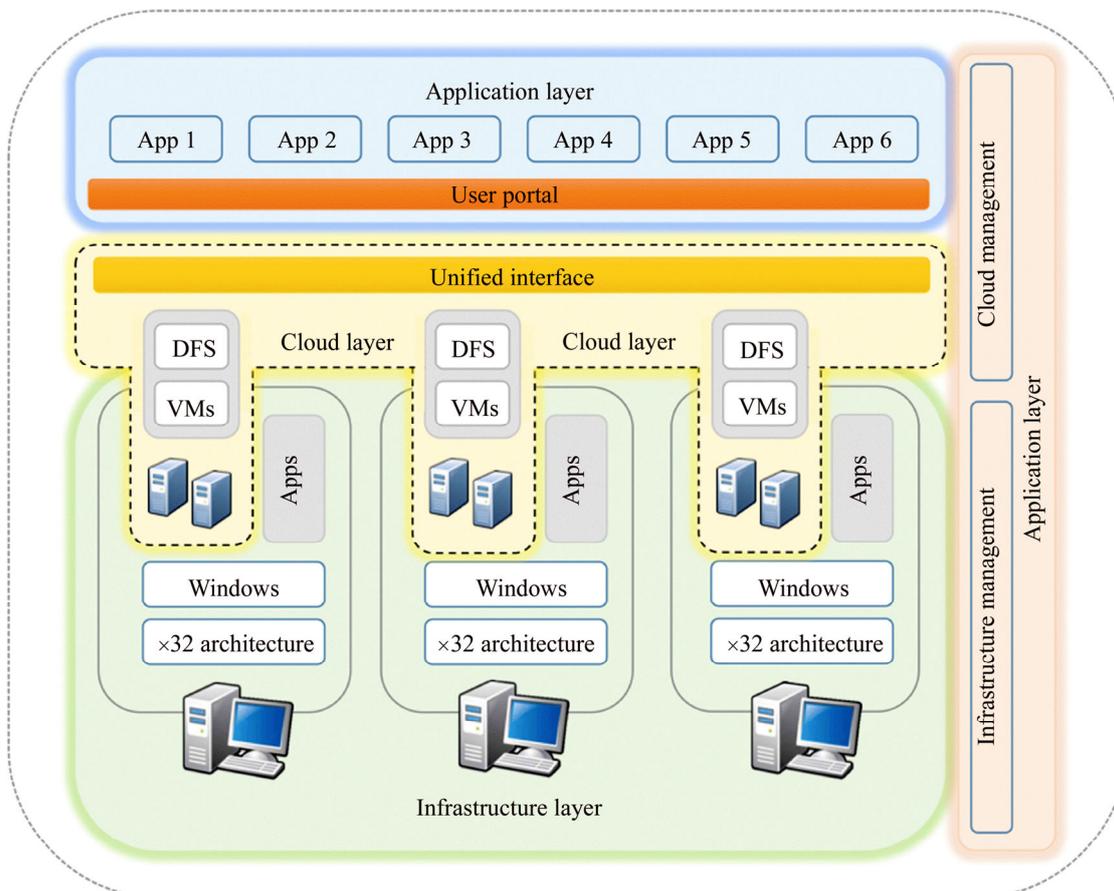


Fig. 1 Framework of GRCloud (VM: virtual machine; DFS: distributed file system)

hardware. The cloud computing layer can be further divided into three sub-layers: computation, storage, and network. The computation sub-layer consists of a large number of virtual machines (VMs), which are hosted in Windows operating systems. In contrast, the storage sub-layer is a distributed file system (DFS), which is made up of the data storage provided by VMs. In addition, the network sub-layer enables connections between VMs and between data storage.

The application layer includes a user portal and user-developed applications. The portal provides users with an easy-to-use operating environment, by abstracting the underlying infrastructure layer and cloud computing layer away from users. Users are able to deploy their applications with the portal, and run the applications in a cloud-based environment.

The management layer is responsible for monitoring and configuring the GRCloud, and is divided into an infrastructure management component and a cloud management component. The infrastructure management component is utilized for starting up/shutting down computers and enabling/disabling network connections. This component can be replaced by the central control unit for a GIS/RS laboratory. The cloud management component is where users can configure the cloud environment, for example, to change the hosts file. The configuration of this component is to be synchronized to all VM nodes.

The GRCloud has the same advantage as common cloud computing models, that is, executing computing tasks on a multi-node cluster and making full use of idle computing resources. Meanwhile, compared to common cloud computing models, the GRCloud has the following strengths: 1) there is no need to change a computing node's operating system from Windows to Linux/UNIX; thus Windows-based applications are not affected; and 2) processing tasks on the GRCloud do not interfere with the normal usage of computing nodes, and a computing node can stop being involved in the GRCloud anytime. The ending of its involvement will not affect the outcomes of the GRCloud tasks.

For most GIS/RS departments in universities and colleges, there is no need to purchase any extra hardware when using the GRCloud for cloud-based geo-processing tasks. A cloud computing platform can be established easily with computers in a GIS/RS laboratory. Moreover, the platform will not affect the normal use of the laboratory by students.

3.2 Implementation

The most significant component in the GRCloud is VMs. In order to improve the efficiency of deploying VMs, we firstly created a VM template. The template is a useful tool for VM management and rapid deployment.

With a template, users are able to rapidly “clone” a large number of VMs with the same operating system, applications and configuration from a VM. Therefore, in this study, VM templates were employed to avoid repeatedly configuring computing nodes (i.e., VMs).

To create a VM template, a Window-based virtualization software “VMware” was adopted. We installed the VMware workstation (VMW) on a computer with a Windows operating system, and created a VM by VMW. The configuration of this VM is 1.80 GHz CPU, 2.0 GB RAM and 20.0 GB Disk, where the performance of the CPU and RAM is half that of the physical machine. The operating system installed on this VM is Ubuntu, a commonly used Linux system. Additionally, some other tools were installed on Ubuntu, including Java Development Kit (JDK), Secure Shell Protocol (SSH), Remote Synchronize (RSync) and Hadoop. JDK is a program development environment for writing Java applications and applets. SSH is used for remote login and file copying between VMs. RSync is a utility for synchronizing files over a network. Hadoop is a framework for running applications/tasks on clouds. Hadoop-based applications/tasks are executed in parallel, in order to make full use of the computing power of clouds. The version of Hadoop utilized is 1.2.1, a stable version which is recommended for standard use. The number of duplicate data in Hadoop is set to 1.

The VM was exported as a template in the format of OVA. There are two frequently used formats for VM templates, i.e., OVF and OVA. The template in an OVF format is a set of files in several formats, such as “.ovf”, “.mf” and “.vmdk”. In contrast, the OVA template is a single file with a suffix of “.ova”, and thus it is more portable. Therefore, the format we adopted was OVA.

We employed 16 laboratory computers as the infrastructure resources for the GRCloud. All the 16 computers are Windows-based, and each computer is equipped with a 32-bit processor, a 4.0 GB RAM and a 200.0 GB Disk. We installed VMW on these computers, and deployed VMs therein by using the pre-created template. It is time-saving to create VMs with a template; the creation is almost completed as soon as the file copying of the template is complete. After deploying all the VMs, we performed the following steps: 1) rename all the VMs, otherwise the names of VMs would be the same as that of their template, resulting in routing errors; 2) configure the hosts file of one VM, in order to associate all VM names with their corresponding IP addresses, and distribute this hosts file to all the other VMs to replace their original one; 3) synchronize the system time of all VMs with an Internet time server; and 4) set one VM as the Hadoop master node, and set the rest of VMs as Hadoop slave nodes.

It can be seen that the architecture adopted for implementing the GRCloud is a hosted one, as opposed to a bare-metal architecture [24]. As a result, VMs in the GRCloud can utilize the idle resources of their host machines, by existing in a hosted form.

4 Case study: big data simplification

4.1 Data source

The big data utilized for simplification is the data set of T-Drive Taxi Trajectories (T-Drive), which was released by Microsoft Research Asia [25]. The size of T-Drive is 778 MB, and it consists of a large number of sub-files. The data of each sub-file is a spatial-temporal trajectory, which was collected by GPS devices on taxis in Beijing over a one week period from February 2nd, 2008 to February 8th, 2008. The number of taxis involved was 10357, and the total length of their trajectories was approximately 9×10^6 km. A typical trajectory is composed of a series of sequential spatial-temporal points; the total number of such points in T-Drive is approximately 1500 million.

A program was developed to extract spatial information (i.e., longitudes and latitudes) from each sub-file in T-Drive. The extracted information was gathered in a text file (denoted as T-Drive-One) with the following format:

```
ID-1 Lon.1 Lat.1 Lon.2 Lat.2 Lon.3 Lat.3 ... Lon.n
Lat.n ...
```

```
ID-2 Lon.1 Lat.1 Lon.2 Lat.2 Lon.3 Lat.3 ... Lon.n
Lat.n ...
```

where ID-1 and ID-2 represent different taxis; Lon. n is the longitude of the n -th spatial point of a taxi, and Lat. n is the corresponding latitude. T-Drive-One contains in total 10357 rows, and each row represents a trajectory for a taxi over the monitoring period. Compared to that of the T-Drive, the size of the T-Drive-One is smaller (315 MB), since the information relating to date and time was removed.

Even though the size of T-Drive-One does not reach GB or TB level, which is common for big data, we regard T-Drive-One as big data, for the following reasons: 1) hundreds of megabytes is already quite a large amount for geo-information stored in a vector form (e.g., text file), as opposed to a raster form; and 2) the trajectories contained in T-Drive-One are complex, with an average node number of 1709 (the node number of a few trajectories is approximately 100000).

4.2 Algorithm design

The algorithm we adopted for big data simplification is “multi-scale visual curvature” (MVC) [26]. In fact, the most frequently used algorithm for

vector geo-information simplification is Douglas-Peucker (DP) [27], but its simplification quality is not always satisfactory [28]. One of the reasons for this is that the salient vertices detected by the DP are sometimes incorrect. On the other hand, in the field of computer vision, a novel salient vertex-detecting algorithm termed MVC was proposed by LIU et al [26]. The MVC algorithm can exactly determine the importance of vertices, and thus can be also utilized to simplify vector data (polylines and polygons) in the field of GIS/RS.

However, it is challenging to simplify big vector data based on MVC. In the MVC algorithm, salient vertices are detected through a process which simulates the human visual system. The process is accompanied by a large amount of calculation and a long processing time. The computational complexity of MVC is much higher than that of commonly used algorithms in GIS/RS. It is therefore a challenging job to perform data-intensive tasks by using such a compute-intensive method with ordinary computers.

We presented a parallel MVC algorithm to run on the GRCloud. To parallelize MVC, a computational paradigm called MapReduce was adopted. A typical MapReduce job is divided into a Map phase and a Reduce phase, both of which run on a Hadoop cluster. In the Map phase, each Hadoop node executes a Map task to process a part of the input data. All the Maps run in parallel and independently. In the Reduce phase, correlative results from different Map tasks are collected and then transferred to the same Reduce task, which produces a final output dataset. A possible solution for MapReduce-based MVC is to calculate the MVC values of all vertices in the Map phase, and then to detect salient vertices according to MVC values in the Reduce phase. However, this solution will cause a large amount of data transmission from Map tasks to Reduce tasks, which is very time-consuming. Taking into account this fact, we directly detected salient vertices in the Map phase, without using Reduce tasks.

4.3 Experiments

4.3.1 Experiment I

In this experiment, we tested the availability of the GRCloud. To this end, we simplified the T-Drive-One data set based on the GRCloud with the MVC algorithm, and then compared its execution time to that based on an ordinary personal computer (PC). The data set was copied 1–5 times in order to study the effect of data size. Each simplification was executed 3 times, and the average execution time was recorded in Table 2. All simplifications based on the GRCloud were completed and took shorter than the simplifications based on the PC,

Table 2 Execution time of simplifying T-Drive-One on an ordinary PC/ GRCloud, and time saved by GRCloud

Type	Execution time of simplifying x copies of T-Drive-One(s)				
	$x=1$	$x=2$	$x=3$	$x=4$	$x=5$
Ordinary PC	1812	3379	6034	7669	9688
GRCloud	775	814	956	1056	1204
Time saved	57%	76%	84%	86%	88%

demonstrating the availability of the GRCloud. Furthermore, as can be seen from Table 2, the execution time saved by the GRCloud increased with the data size, implying that the GRCloud is more suitable to deal with big data.

4.3.2 Experiment II

In this experiment, we evaluated the stability of the GRCloud. We simulated three unexpected situations where some nodes in the GRCloud broke down during the simplification process.

Situation 1: Shut down one node which is not executing Map tasks when the MapReduce job is 80% completed;

Situation 2: Shut down one node which is executing Map tasks when the MapReduce job is 80% completed;

Situation 3: Shut down two nodes which are executing Map tasks when the MapReduce job is 80% completed.

Situation 0: Normal situation, without any errors.

The simplification tasks in each situation were executed 3 times, and the average execution time was shown in Table 3. It can be seen that the execution time of Situations 1–2 shows little difference to that of the normal one (i.e., Situation 0), no matter whether the broken-down node was executing tasks or not. The execution time increased markedly when two working nodes were shut down deliberately, because the number of duplicate data was set to 1 (as mentioned in Section 3.2). All the geo-processing tasks in simulated situations were completed and all the simplified data was the same as that of the normal situation (though the execution time needed was different), demonstrating that the GRCloud is stable.

Table 3 Execution time of experimental GRCloud-based task in different situations

Situation	Execution time of simplifying x copies of T-Drive-One (s)				
	$x=1$	$x=2$	$x=3$	$x=4$	$x=5$
0	775	814	956	1056	1204
1	755	795	953	1064	1221
2	780	817	991	1066	1207
3	900	949	1069	1158	1390

5 Merits of novel framework

Located in the city of Changsha, China, the TianHe-1 supercomputer stands as a perfect example of high-performance data centers for cloud computing. Tianhe-1 consists of 2560 compute nodes, and its computing capability ranked the first in the world in November 2011 [29]. However, a local official press, the newspaper of Xiangjiang, reported in July 2014 that Tianhe-1 had been idle for nearly one year. This poses a question about the necessity of dedicated data centers: whether or not it is worthwhile to build a resource-centralized data center with a great deal of money.

Today, as the data size of geo-information increases and the efficiency is also a top concern, more and more geo-processing tasks are completed based on data centers with cloud-based strategies. The overwhelming majority of these data centers are reported to be set up based on either commercial public clouds or dedicated private clouds. However, as mentioned in Section 2, neither of the two clouds are low-cost and cost-efficient. For most GIS/RS users, a flexible but stable data center for temporary use, rather than a dedicated one for long-term use, is a wise choice for cloud-based geo-processing tasks. The GRCloud solution is proposed in response to this fact.

As shown in the case study, the GRCloud solution was utilized to accelerate the simplification of big geo-data. Experiments demonstrated that the framework is not only efficient but also stable. However, it should be noted that applications of the proposed GRCloud are not limited to this case study. The GRCloud can be considered as an alternative to dedicated clouds for geo-processing, when cloud computing is required for temporary use. Compared to dedicated clouds, the GRCloud has the following merits:

1) There is no need to purchase dedicated extra hardware or software when establishing a GRCloud-based data center in a GIS/RS computer laboratory; thus there is no need for extra investment.

2) The computing power and storage of the CRCloud is provided by ordinary PCs, which are not limited to 64-bit ones but also include common 32-bit computers.

3) Since the GRCloud adopts a hosted architecture, the establishment of the GRCloud does not change the original system of involved PCs, and the execution of the GRCloud does not stop the normal use of the involved PCs.

4) The GRCloud is characterized by being resource-saving, since it increases the utilization of involved PCs, avoiding computing power wastage.

In addition, compared with volunteer computing, the GRCloud is more suitable for GIS/RS researchers, who have access to many available ordinary PCs. The GRCloud places more emphasis on real-time computing than volunteer computing; the computing tasks running on the GRCloud are expected to complete in a shorter time.

6 Conclusions

This work proposes a low-cost and convenient cloud computing solution for geo-information processing tasks, especially for temporary processing tasks. Before this solution, cloud-based geo-processing tasks are implemented primarily based on two types of clouds, i.e., dedicated private clouds and commercial public clouds. For the former, it is expensive to establish and is not cost-efficient for temporary geo-processing tasks. For the latter, it is neither convenient nor time-efficient to use since the cloud is off-premises; all the geo-information related to processing tasks must be transmitted to the off-premises cloud, thereby wasting a large amount of time. Furthermore, the latter is also not cost-efficient for temporary geo-processing tasks, since the rent is generally paid on a month to month basis. In contrast, the proposed cloud computing solution avoids these problems, and thus is more low-cost and convenient than dedicated private clouds or commercial public clouds. The proposed solution demonstrates great potential in cloud-based GIS/RS applications. Note that the solution is based on Hadoop, which to some degree limits the performance of the solution. In the future, we plan to realize the proposed solution by other parallel approaches such as Spark.

Acknowledgments

The authors would like to thank John Olbrich for improving the English of the manuscript.

References

- [1] ZHOU Zhou, HU Zhi-gang, SONG Tie, YU Jun-yang. A novel virtual machine deployment algorithm with energy efficiency in cloud computing [J]. *Journal of Central South University*, 2015, 22(3): 974–983.
- [2] MA Hua, HU Zhi-gang. User preferences-aware recommendation for trustworthy cloud services based on fuzzy clustering [J]. *Journal of Central South University*, 2015, 22(9): 3495–3505.
- [3] GAO Pei-chao, LIU Zhao, XIE Mei-hui, TIAN Kun. CRG-index: A more sensitive Ht-index for enabling dynamic views of geographic features. *The Professional Geographer*, 2016, 68(4): 533–545.
- [4] MELL P, GRANCE T. The NIST definition of cloud computing (draft) [J]. *NIST Special Publication*, 2011, 800(145): 1–7.
- [5] GAO Pei-chao, LIU Zhao, HAN Fei, TANG Lei, XIE Mei-hui. Accelerating the computation of multi-scale visual curvature for simplifying a large set of polylines with Hadoop [J]. *GIScience & Remote Sensing*, 2015, 52(3): 315–331.
- [6] LI Ji-yuan, MENG Ling-kui, WANG F Z, ZHANG Wen, CAI Yang. A Map-Reduce-enabled SOLAP cube for large-scale remotely sensed data aggregation [J]. *Computers & Geosciences*, 2014, 70: 110–119.
- [7] ADDAIR T G, DODGE D A, WALTER W R, RUPPERT S D. Large-scale seismic signal analysis with Hadoop [J]. *Computers & Geosciences*, 2014, 66: 145–154.
- [8] WANG Peng-yao, WANG Jian-qin, CHEN Ying, NI Guang-yuan. Rapid processing of remote sensing images based on cloud computing [J]. *Future Generation Computer Systems*, 2013, 29(8): 1963–1968.
- [9] LEE D W, LIANG S L. Geopot: A cloud-based geolocation data service for mobile applications [J]. *International Journal of Geographical Information Science*, 2011, 25(8): 1283–1301.
- [10] LI Qing-quan, ZHANG Tong, YU Yang. Using cloud computing to process intensive floating car data for urban traffic surveillance [J]. *International Journal of Geographical Information Science*, 2011, 25(8): 1303–1322.
- [11] FUJIOKA E, BERGHE E V, DONNELLY B, CASTILLO J, CLEARY J, HOLMES C, MCKNIGHT S, HALPIN P. Advancing global marine biogeography research with open-source GIS software and cloud computing [J]. *Transactions in GIS*, 2012, 16(2): 143–160.
- [12] LIN Feng-cheng, CHUNG Lan-kun, WANG Chun-ju, KU Wen-yuan, CHOU Tien-yin. Storage and processing of massive remote sensing images using a novel cloud computing platform [J]. *GIScience & Remote Sensing*, 2013, 50(3): 322–336.
- [13] TANG Wen-wu, FENG Wen-peng. Parallel map projection of vector-based big spatial data: Coupling cloud computing with graphics processing units [J]. *Computers, Environment and Urban Systems*, 2017, 61: 187–197.
- [14] PURI S, AGARWAL D, HE Xi, PRASAD S K. MapReduce algorithms for GIS polygonal overlay processing [C]// *The 27th IEEE International Parallel and Distributed Processing Symposium Workshops & PhD Forum*. Cambridge, MA: IEEE, 2013: 1009–1016.
- [15] LEE K, KANG S. Mobile cloud service of geo-based image processing functions: A test ipad implementation [J]. *Remote Sensing Letters*, 2013, 4(9): 910–919.
- [16] KIM K, KANG S, LEE K. Geo-based image blending in a mobile cloud environment [J]. *Remote Sensing Letters*, 2013, 4(11): 1117–1126.
- [17] GAO Song, LI Lin-na, LI Wen-wen, JANOWICZ K, ZHANG Yue. Constructing gazetteers from volunteered big geo-data based on Hadoop [J]. *Computers, Environment and Urban Systems*, 2017, 61: 172–186.
- [18] EXPÓSITO R R, TABOADA G L, RAMOS S, TOURIÑO J, DOALLO R. Evaluation of messaging middleware for high-performance cloud computing [J]. *Personal and Ubiquitous Computing*, 2013, 17(8): 1709–1719.
- [19] DOELITZSCHER F, SULISTIO A, REICH C, KUIJS H, WOLF D. Private cloud for collaboration and e-learning services: From IAAS to SAAS [J]. *Computing*, 2011, 91(1): 23–42.
- [20] GAO Pei-chao, LIU Zhao, XIE Mei-hui, TIAN Kun. The development of and prospects for private cloud GIS in China [J]. *Asian Journal of Geoinformatics*, 2014, 14(4): 30–38.
- [21] NOUMAN D M, SHAMSI J A. Volunteer computing: Requirements, challenges, and solutions [J]. *Journal of Network and Computer Applications*, 2014, 39: 369–380.
- [22] ANDERSON D P. BOINC: A system for public-resource computing and storage [C]// *The 5th IEEE/ACM International Workshop on*

- Grid Computing. IEEE, 2004: 4–10.
- [23] ANDERSON D P, COBB J, KORPELA E, LEBOFKY M, WERTHIMER D. SETI@home: An experiment in public-resource computing [J]. *Communications of the ACM*, 2002, 45(11): 56–61.
- [24] CHANG Bao-rong, TSAI Hsiu-fen, CHEN Chi-ming. Evaluation of virtual machine performance and virtualized consolidation ratio in cloud computing system [J]. *Journal of Information Hiding and Multimedia Signal Processing*, 2013, 4(3): 192–200.
- [25] YUAN Jing, ZHENG Yu, ZHANG Cheng-yang, XIE Wen-lei, XIE Xing, SUN Guang-zhong, HUANG Yan. T-Drive: Driving directions based on taxi trajectories [C]// *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*. New York: ACM, 2010: 99–108.
- [26] LIU Hai-rong, LATECKI L, LIU Wen-yu. A unified curvature definition for regular, polygonal, and digital planar curves [J]. *International Journal of Computer Vision*, 2008, 80(1): 104–124.
- [27] SCHINDLER F, FÖRSTNER W. Dijkstrafraps: Graph partitioning in geometry and image processing [J]. *Photogrammetrie Fernerkundung Geoinformation*, 2013, 2013(4): 285–296.
- [28] WENZEL S, FÖRSTNER W. Finding poly-curves of straight line and ellipse segments in images [J]. *Photogrammetrie Fernerkundung Geoinformation*, 2013, 2013(4): 297–308.
- [29] LIAO Xiang-ke, YUNG Can-qun, TANG Tao, YI Hui-zhan, WANG Feng, WU Qiang, XUE Jing-ling. Openmc: Towards simplifying programming for tianhe supercomputers [J]. *Journal of Computer Science and Technology*, 2014, 29(3): 532–546.

(Edited by FANG Jing-hua)