

Article

# Building an Elastic Parallel OGC Web Processing Service on a Cloud-Based Cluster: A Case Study of Remote Sensing Data Processing Service

Xicheng Tan <sup>1,\*</sup>, Liping Di <sup>2</sup>, Meixia Deng <sup>2</sup>, Jing Fu <sup>3</sup>, Guiwei Shao <sup>3</sup>, Meng Gao <sup>4</sup>, Ziheng Sun <sup>2</sup>, Xinyue Ye <sup>5</sup>, Zongyao Sha <sup>4</sup> and Baoxuan Jin <sup>6</sup>

<sup>1</sup> Spatial Information and Digital Technology Department, International School of Software, Wuhan University, 37, Luoyu Road, Wuhan 430079, China

<sup>2</sup> Center for Spatial Information and Science Systems, George Mason University, 4087 University Dr, Fairfax, VA 22030, USA; E-Mails: ldi@gmu.edu (L.D.); mdeng@gmu.edu (M.D.); szhwhu@gmail.com (Z.S.)

<sup>3</sup> China Electric Power Research Institution, 143, Luoyu Road, Wuhan 430079, China; E-Mails: fujing@epri.sgcc.com.cn (J.F.); shaoguiwei@epri.sgcc.com.cn (G.S.)

<sup>4</sup> International School of Software, Wuhan University, 37, Luoyu Road, Wuhan 430079, China; E-Mails: gmshepard@foxmail.com (M.G.); zongyaosha@163.com (Z.S.)

<sup>5</sup> Department of Geography, Kent State University, Kent, OH 44242, USA; E-Mail: xye5@kent.edu

<sup>6</sup> Yunnan Provincial Geomatics Center, 404, Huanchengxi Road, Kunming 650034, China; E-Mail: jinbx163@163.com

\* Author to whom correspondence should be addressed; E-Mail: xctan@whu.edu.cn; Tel.: +86-27-6877-8770.

Academic Editors: Marc A. Rosen and Yichun Xie

Received: 3 September 2015 / Accepted: 14 October 2015 / Published: 21 October 2015

---

**Abstract:** Since the Open Geospatial Consortium (OGC) proposed the geospatial Web Processing Service (WPS), standard OGC Web Service (OWS)-based geospatial processing has become the major type of distributed geospatial application. However, improving the performance and sustainability of the distributed geospatial applications has become the dominant challenge for OWSs. This paper presents the construction of an elastic parallel OGC WPS service on a cloud-based cluster and the designs of a high-performance, cloud-based WPS service architecture, the scalability scheme of the cloud, and the algorithm of the elastic parallel geoprocessing. Experiments of the remote sensing data processing service demonstrate that our proposed method can provide a higher-performance WPS service

that uses less computing resources. Our proposed method can also help institutions reduce hardware costs, raise the rate of hardware usage, and conserve energy, which is important in building green and sustainable geospatial services or applications.

**Keywords:** Open Geospatial Consortium (OGC); geospatial service; cloud computing; parallel computing

---

## 1. Introduction

The implementation of the OGC Web Services (OWS) has promoted the sharing and interoperability of geospatial data. Many applications have been built based on the OWS. The geospatial Web processing service (WPS) is used to implement web-based geoprocessing. However, in practical applications, the WPS defines an interface, which facilitates the publishing and discovery of geospatial processes. Numerous geospatial Web services are built on OWS, such as the Web services available on GeoBrain [1]. WPS services have also been used in the geospatial service chain or aggregation, which can chain the existing distributed OWS services into a composite service or workflow to complete a complex geospatial task [2–5]. The majority of research on OWS-based distributed geospatial processing has focused on seamless data sharing, interoperability, and service chaining. With the increasing use of geospatial applications and the expanding volume of geospatial data, performance issues of the OWS service have attracted considerable attention in recent studies [6,7]. However, constructing a high-performance WPS on a physical cluster via parallel computing technology means that the WPS will occupy the entire cluster. To fulfill the WPS request, resources are required for the peak workload, but the resources are idle at non-peak times. More pronounced variations result in more waste [8]. For a provider of geospatial services, purchasing a computer cluster for a single purpose and wasting considerable energy during the non-peak period is a significant burden. This paper proposes a cloud-based elastic WPS service that can provide a high-performance geospatial WPS on a cloud and also helps to reduce hardware costs and energy usage.

## 2. Background and Related Work

### 2.1. Geospatial Processing Service

Geospatial processing on local machines has played a significant role in traditional GIS applications before 2000 [9–11]. With the development and wide application of Internet technology, Internet GIS and Web GIS have become the main forms of GIS applications in the early years of the 21st century [12–14]. Since it is easy to use and can be widely accessed over the Internet, Internet GIS, and Web GIS offered a useful distributed work environment that is easy for the public to access. Since OGC released the Web Processing Service 1.0 (WPS 1.0) in 2007, OWS has been used widely in SOA-based geospatial applications. Many studies based on the OGC WPS have been conducted [15,16]. NASA, USGS, and the USDA have built a series of SOA-based applications for EOS data sharing and remote sensing analysis [17,18]. For the outstanding characters of the Web service, SOA-based distributed geospatial processing has become a major method of distributed geospatial processing [19,20]. WPS has also been

explored to construct the geospatial service chain (or service aggregation) [3–5], which aggregates the atomic service into a composite service to solve more complex issues. The geospatial workflow is executed sequentially via OGC OWS and includes a geospatial WPS service [21].

## 2.2. Parallel Computing

Parallel computing has had a significant impact on many areas, such as physics [22], biology [23], meteorology, and global climate studies [7,24]. Parallel computing divides a task into multiple sub-tasks that can be executed within a number of processes upon several processors [25]; in this manner, large problems can be solved efficiently by using massively parallel processor (MPP) systems [26]. Parallel computing technologies have been recommended in the geosciences. Various techniques for the high-performance parallel processing and analysis of remote sensing data and geospatial datasets based on clusters have been proposed [27–29]. Geospatial applications based on parallel computing have also been designed and implemented [25,30]. Parallel algorithms based on GPUs have also received considerable attention as GPUs are more efficient than CPUs [31]. GPU-based parallel computing techniques have been widely applied in the geosciences to improve the performance of spatial data processing and can obtain much better efficiency than CPU-based methods because of the massively parallel programmable units at their cores [32–34]. Parallel computing techniques have been predicted to become the dominant geocomputation techniques in the near future [35].

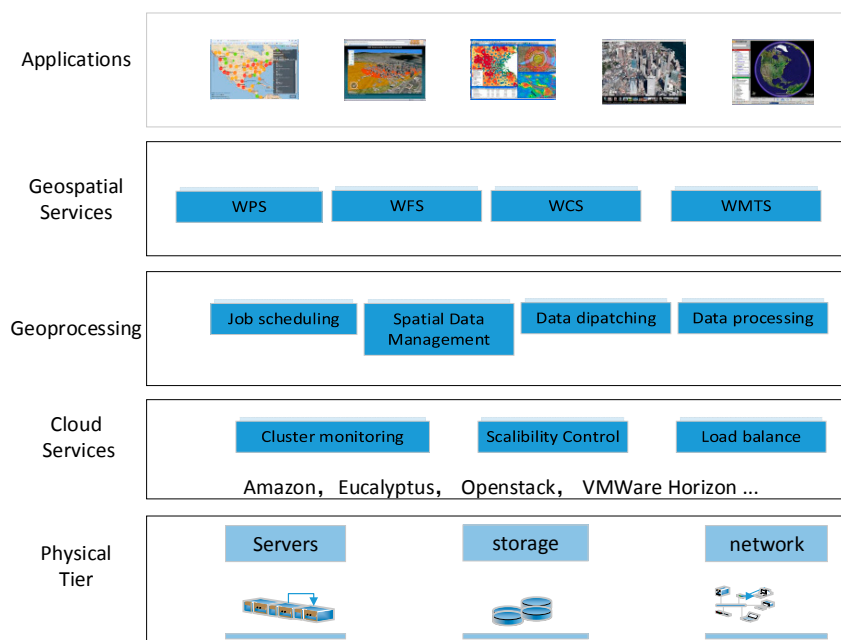
## 2.3. Cloud Computing

Cloud computing has had a significant impact on computer science and other domains. Cloud computing stems from grid computing, distributed computing, and hardware virtualization technology. In cloud, the infrastructure can act as a service (IaaS) [36,37], the platform can act as a service (PAAS) [38], and the software can act as service (SAAS) [39,40]. For the scalability and pay-as-you-go model of cloud computing, countries and many companies have decided to transfer the traditional system onto clouds, such as AWS, Google Cloud, and Microsoft Azure. The new features of cloud computing allow governments and commercial groups to reduce the amount of hardware invested and save energy. Cloud computing can reduce company investments at the start of a business and can also reduce wasted CPU time when the number of concurrent requests is dynamically changed [8]. Cloud computing has been utilized to optimize the performance, reliability, and usability of computing tasks [41,42] and to solve the existing geoscience challenges. Yang [43] reported the utilization of cloud computing to support the intensities of geospatial sciences. Research projects and applications have also demonstrated that cloud-based geospatial applications and services can obtain high performance and better intelligence by utilizing the scalability and virtualization capacity of clouds [7,40,44]. Currently, research on geospatial-oriented cloud computing is a topic of considerable interest, and cloud computing is expected to play a greater role in the geoscience field.

### 3. Methodologies

#### 3.1. Logical Architecture

To achieve a better performance in elastic computing, infrastructure is designed to have five tiers, as shown in Figure 1.



**Figure 1.** Logical architecture.

The base tier is the hardware and network, which includes clusters, storage, and network equipment. They all support elastic utilization under the control of the cloud service tier.

The second tier is the cloud service. It is the management infrastructure of the environment. The cloud will dynamically manage these resources by the IaaS service. This tier is fundamental to elastic geoprocessing and can offer the needed virtual machines (VM), storage, load balance, and elastic control service to support the geoprocessing.

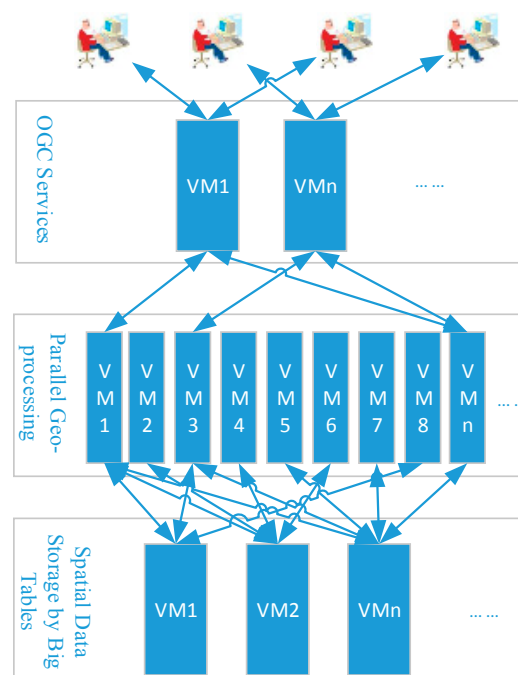
The geoprocessing tier aims at parallel geoprocessing task execution. This tier offers various cloud services, including task schedules, spatial data management, and parallel geoprocessing. By using the cloud services offered by the cloud services tier, this tier can complete the task with high efficiency and excellent elastic performance.

The geospatial service tier is the container of the published OGC geospatial services. It includes the spatial data services, such as WFS, WCS, WMS, and WMTS, as well as WPS. This tier can offer basic services for the application tier.

The top tier is the application tier. All of the geospatial applications will be built in this tier by using or aggregating the geospatial services. It is the interactive interface between the user and the spatial information services.

### 3.2. Scalability Scheme for Parallel Geoprocessing

The purpose of scalability research is to maximize the performance and reduce the costs. With this scheme, one can automatically extend or shrink the cloud scale by increasing or decreasing the number of the specific VMs during the execution of the WPS service. As shown in Figure 2, there are three types of VMs: the OGC services interface VMs, geoprocessing VMs, and spatial data VMs. Each VM works for a specific purpose: the OGC services interface VMs accept the WPS service request, the geoprocessing VMs conduct the elastic parallel geoprocessing, and the spatial data VMs manage the spatial data.



**Figure 2.** VMs for geoprocessing.

If the geoprocessing tasks are beyond the capacity of the cluster, the new VMs will be launched, but if the resources needed by the tasks are less than the capacity, the VMs will be terminated to shrink the cluster scale. Specifically, the geoprocessing VMs and spatial data VMs follow the following rules:

(1) Cluster extension condition:

There are jobs that are delayed to execute for more than 3 min.

(2) Cluster shrinkage condition:

No Jobs are delayed to execute for less than 2 min.

(3) Cluster stable condition:

The jobs must be delayed to execute for more than 2 min and less than 3 min.

The geoprocessing service vendor can change the conditions of the rules according to how rapidly the vendor wants to offer a service. For example, the vendor may set the cluster extension condition to “no jobs are delayed to execute” to offer the users high-performance services.

For the timing policy and charging rules of the cloud platform (e.g., EC2), a running VM with a duration of less than one hour will be charged for one hour; thus, it is recommended to stop the VMs that are close to passing a whole hour in duration.

The jobs are submitted to the cluster via the Portable Batch System (PBS). The sub-tasks of the jobs will be scheduled to the VMs that have resources for the computation. During the processing of the remote sensing data, the task will decompose to some sub-tasks with the same area. The sub-tasks will be executed in the separated threads, and the result will be collected through MPI messages.

The VMs in the cloud cluster environment, such as EC2, Eucalyptus, and Openstack, are created from the prepared images. VMs will be created by a command when a larger capacity is required. An excessive number of VMs will have a high cost according to the charging rules of the cloud computing service vendor. Accordingly, some VMs will need to be stopped according to the rule stated above if the overhead of the cloud cluster declines below a pre-determined value. When new VMs are created, the sub-tasks will be assigned to the original VMs and the newly created VMs. When the cluster is going to shrink, the new geoprocessing tasks will not be assigned to the VMs that have the fewest tasks, and the VM will stop when no more tasks exist. The other VMs will be stopped in this manner until the cluster fulfills the stable condition.

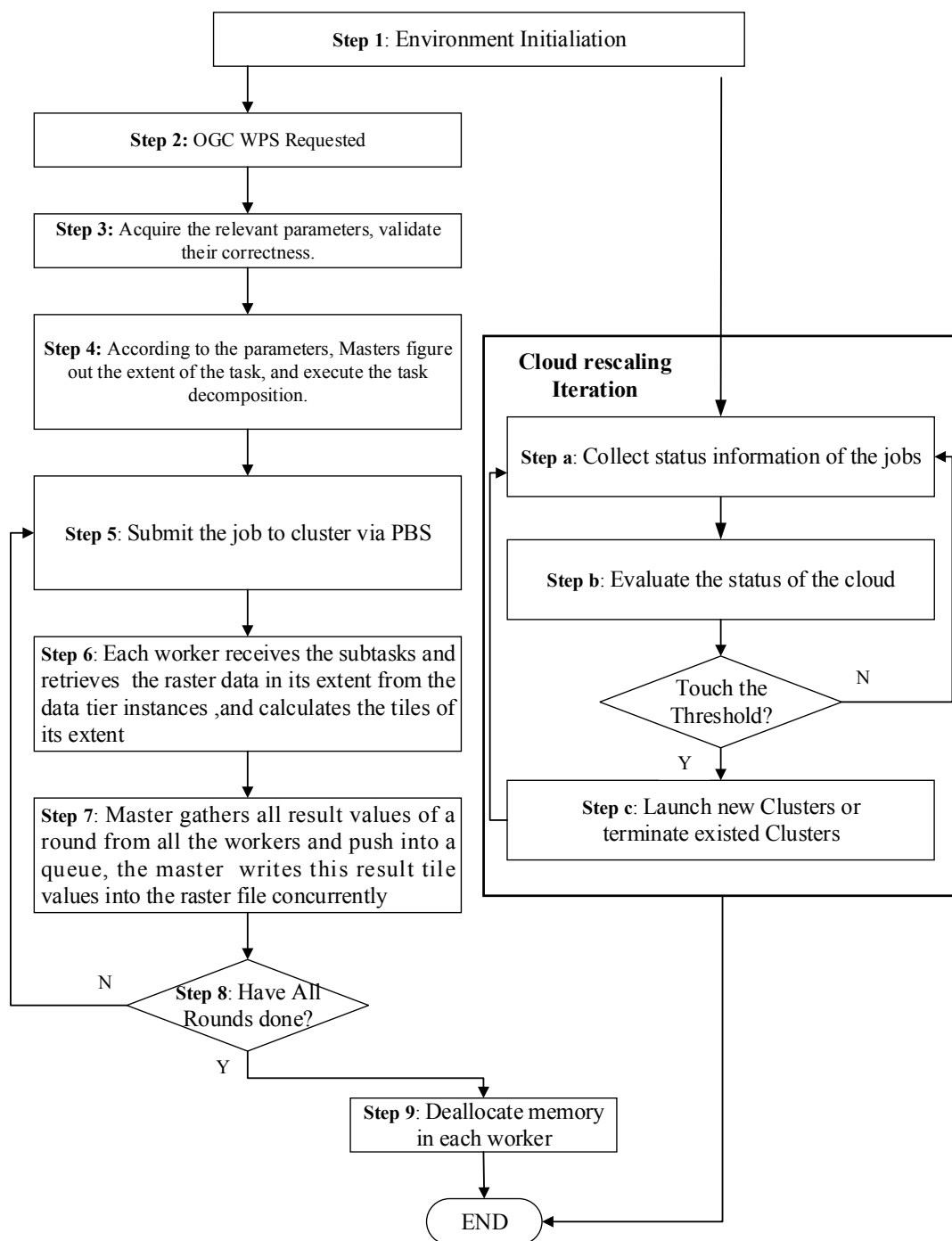
### 3.3. Algorithm of the Elastic Parallel WPS Service

This study proposes and designs the Elastic parallel WPS service algorithm, as shown in Figure 3. The algorithm contains nine steps.

- Step 1:** The cloud and OGC WPS environment is initiated;
- Step 2:** OGC WPS receives a request from a client;
- Step 3:** OGC WPS acquires the parameters from the request and validates them. If problems with the parameters exist, the WPS will return an error message;
- Step 4:** Based on the parameters that are received, the master will determine the extent of the task and then execute the task decomposition;
- Step 5:** The job will be submitted to the cloud-based cluster via the PBS;
- Step 6:** Each worker receives the sub-tasks, retrieves the raster data in its extents from the data tier VMs, and calculates the tiles within its extents. In this step, the data will be transferred to the workers concurrently by the multi-thread method. Except for the first round, waiting for the required data is unnecessary;
- Step 7:** The master gathers all the result data from all the workers and places the data into a queue. The master then writes the result data values into the raster file. For better performance, the non-blocking communication method is used such that workers can continuously perform computations;
- Step 8:** If all rounds are completed, the algorithm will go to step 9; otherwise, the algorithm returns to step 5;
- Step 9:** After all the rounds are completed, all the VMs de-allocate the used memory.

From Steps 5 to 8, the iteration will not break until all the rounds are completed. The cloud-rescaling iteration, provided below, is also followed:

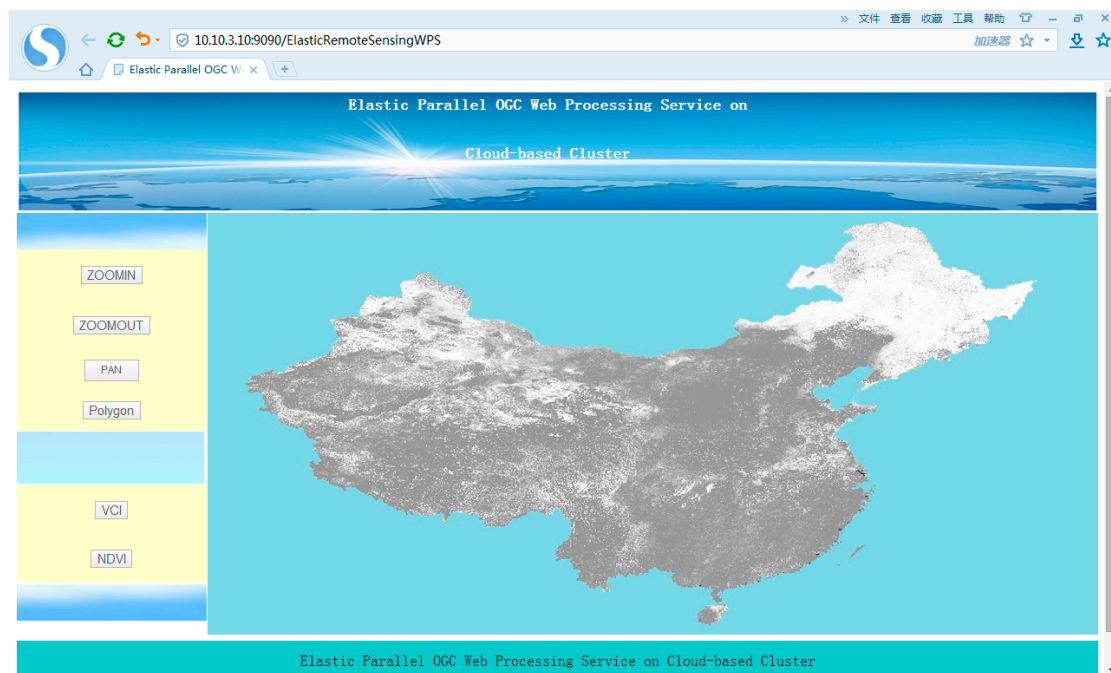
- Step a:** The cloud-watching node collects the status information of the submitted jobs via PBS commands;
- Step b:** The status data are evaluated. If the status reaches the threshold, the algorithm goes to step c; otherwise, it returns to step a;
- Step c:** The cloud-watching node will launch new VMs or terminate existing VMs according to the scalability scheme rules in Section 3.2; then, the algorithm returns to step a.



**Figure 3.** Algorithm of the elastic parallel WPS service.

#### 4. Prototype Implementation and Analysis of the Results

To illustrate the feasibility and advantages of the proposed elastic parallel WPS service, we build a prototype system that offers an elastic parallel remote sensing data-processing WPS service. The prototype system is built on a private Openstack cloud platform. Up to 60 VMs can be launched dynamically. Each VM has two 2.0 GHz virtual CPUs (vCPU), 4 GB of RAM and a 1 TB disk. WPS services are deployed on the cloud image, which can create VMs via Openstack commands. The WPS service calculates the Vegetation Condition Index (VCI) result data by using the Terra satellite's NDVI 10-day synthetic product with 1 km resolution in China from 2001 to 2010. Openlayer is used to display the VCI analysis result data and original NDVI data, as shown in Figure 4.



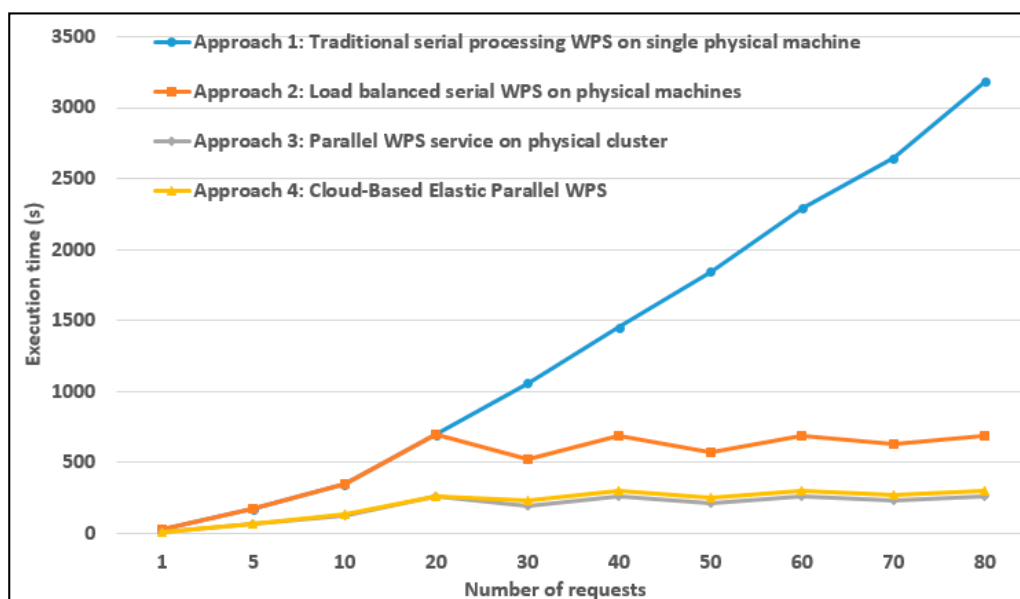
**Figure 4.** Prototype of the elastic parallel remote sensing data processing WPS service.

To demonstrate that the proposed method can greatly improve the performance and feasibility of the cloud-based parallel WPS service, comparisons were conducted between four different WPS services: (1) Approach 1: traditional serial WPS on a single physical machine, which has a 2.0 GHz CPU, 64 GB of RAM, and a 10-TB Raid0 disk array; (2) Approach 2: load-balanced serial WPS on physical machines, where each machine is the same as that in Approach 1; (3) Approach 3: parallel WPS service on a physical cluster, which consists of the 4 machines from Approaches 1 and 2 and is connected to a 1000 Mbps network; and (4) Approach 4: cloud-based elastic parallel WPS, where an Openstack cloud is built upon the machines from Approaches 2 and 3 and each VM has 2 2.0 GHz vCPUs, 4 GB of RAM and a 1 TB disk.

Figure 5 shows the test results and execution time for the four approaches when there are different numbers of requests. The results indicate that the proposed method (*i.e.*, Approach 4) outperforms Approach 1. The execution time for Approach 1 increases dramatically with the number of concurrent requests, whereas the performance of the proposed method remains fairly stable. For example, the execution time for Approach 1 increases from several seconds to 3000 s when the number of requests



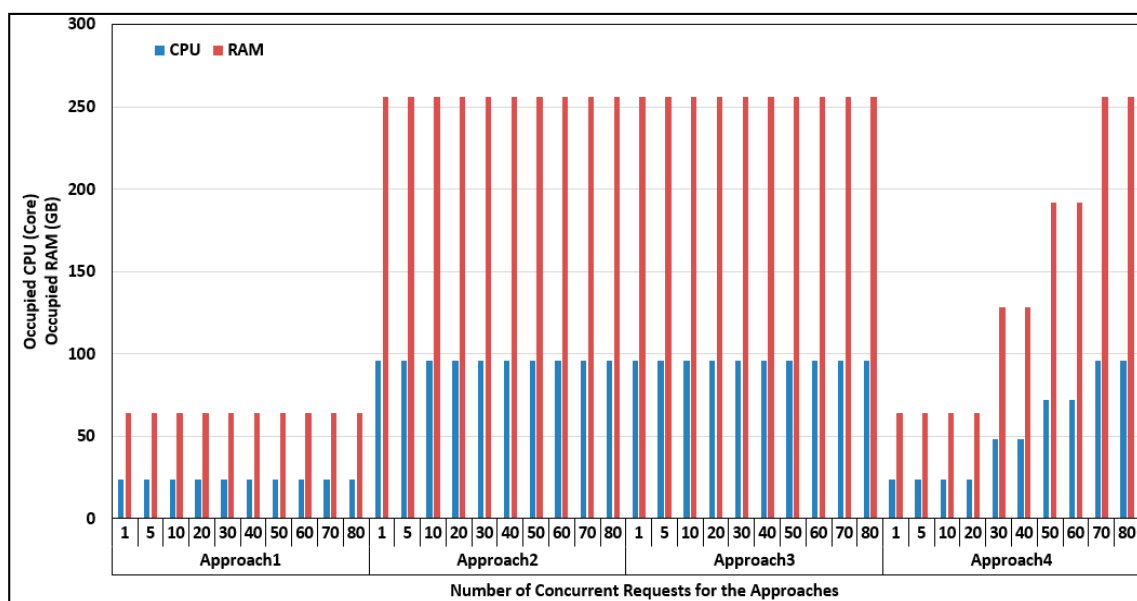
increases from one to 80. In contrast, the execution time for our proposed method only increases from several seconds to approximately 300 s. Furthermore, although there are an increasing number of concurrent requests, the elastic parallel WPS can maintain a stable performance, which benefits from the scalability property of the cloud. The proposed method is also better than the load-balanced serial WPS service (*i.e.*, Approach 2) with the same hardware configurations because of the superiority of the parallel computing, which indicates that the cloud-based elastic parallel WPS is a superior method when facing a large number of concurrent requests. The proposed method and the parallel method on the physical machine (*i.e.*, Approach 3) display similar performances, but the performance of Approach 3 is slightly better than with the proposed method because Approach 3 runs directly on the physical machines, whereas our proposed method runs on VMs of the cloud.



**Figure 5.** Execution times of the four approaches.

Figure 6 shows the resources that are occupied by the four methods. If Approach 1 only runs on a single server, it has a poor performance. If more than 20 concurrent requests are made, the requests must be processed sequentially; otherwise, the system would crash. With Approach 1, the system takes more than 3000 s to finish all the processing tasks when 80 concurrent requests are made; thus, the load-balanced method (*i.e.*, Approach 2) has been widely used in real applications. Approach 2 can schedule the concurrent requests to multiple servers to keep the load balanced and obtain better performance compared to Approach 1. However, with Approach 2, the service vendor must purchase a sufficient number of machines in advance to fulfill the resource requirement for the maximum amount of requests. For example, in this test, we assume that there is a maximum of 80 concurrent requests to give the users an acceptable experience (e.g., no more than 10 min); we must provide four servers with 24 cores, and the servers must configure the same system and software environment. Thus, the servers are occupied by the application exclusively. When adopting Approach 3, we can speed up the execution time to half that of Approach 2 by making use of the capacity of parallel computing, but because Approach 2 requires all the physical nodes in the cluster to be configured with same software, it still occupies the resources exclusively. With Approach 4, the cloud platform can launch new VMs

dynamically according to the number of concurrent requests; thus, when there are only a few concurrent requests (e.g., less than 20 or 40), the idle resources can be used by other applications by launching new VMs with the required system and software environment. As a result, Approach 4 can obtain similar performance to Approach 3 by occupying less computing resources. Moreover, if public cloud services are used, the WPS service vendor can purchase the cloud resources in a “pay-as-you-go” model and hence reduce the investment in computing resources [43]. In conclusion, the cloud-based elastic parallel WPS service can not only improve the performance but also share the computing resources with other applications and reduce the hardware investment, which is a key factor for many research institutions and researchers. Furthermore, the physical machines of the idle cloud computing resources can be powered off to save energy; hence, our proposed method can provide more sustainable and green WPS services and geospatial applications by exploring the capability of cloud computing.



**Figure 6.** Computing resource requirements of the approaches.

## 5. Conclusions

We proposed a cloud-based elastic parallel computing method to obtain high-performance WPS services while using fewer computing resources. Experiments on a remote sensing data processing prototype system demonstrated the feasibility and efficiency of the proposed method. Given the scalability capacities of cloud computing, this method can provide a more efficient WPS service. Scalability is also beneficial for research institutions, particularly small research organizations and individual researchers, because it helps reduce hardware costs, which is often a burden for research activities. Furthermore, this method can also provide a WPS service with excellent performance by utilizing the virtualization and elasticity capacity of the IaaS service of cloud computing, even for a large number of concurrent requests. The WPS does not occupy all the computing resources, and when the number of concurrent requests is less than the maximum number, the idle resources can be shared by other applications or shut down automatically to save energy; thus, our approach is more sustainable than other approaches.

In the future, efforts will be made to build WPS services for Big Data analysis and applications on the cloud. The high-performance WPS can be utilized to improve the performance of geospatial service chains and the efficiency of geospatial-based events, particularly emergency and disaster response decision-making and activities.

## Acknowledgments

We are grateful to the anonymous reviewers; to Xiaohui Cui for his detailed comments, which helped improve the quality of the paper; and to Justin Poupart and Aijun Chen for their revisions. This work was supported in part by NSFC project 51277167 and the Guangzhou Science and Technology Project on the Supercomputing Applications Research: 2012Y2-00035, 2013Y2-00031. The data sets for the experiments were provided by the International Scientific & Technical Data Mirror Site, Computer Network Information Center, Chinese Academy of Sciences (<http://www.gscloud.cn>).

## Author Contributions

Xicheng Tan designed the study and wrote the paper. Liping Di, Meixia Deng and Xinyue Ye helped to write the introduction and conclusion and polish the English. Jing Fu, Guiwei Shao, Ziheng Sun and Zongyao Sha analyzed the data, Meng Gao and Baoxuan Jin contributed to environment configuration, data collection and processing. All authors have read and approved the final manuscript.

## Conflicts of Interest

The authors declare no conflict of interest.

## References

1. Deng, M.; di, L. GeoBrain Online Resources for Supporting College-Level Data-Intensive Geospatial Science and Engineering Education. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Boston, MA, USA, 7–11 July 2008; pp. IV-443–IV-446.
2. Friis-Christensen, A.; Lucchi, R.; Lutz, M.; Ostländer, N. Service chaining architectures for applications implementing distributed geographic information processing. *Int. J. Geogr. Inf. Sci.* **2009**, *23*, 561–580.
3. Di, L.; Zhao, P.; Yang, W.; Yue, P. Ontology-driven automatic geospatial-processing modeling based on web-service chaining. In Proceedings of the Sixth Annual NASA Earth Science Technology Conference, College Park, MD, USA, 27–29 June 2006; pp. 27–29.
4. Yue, P.; di, L.; Yang, W.; Yu, G.; Zhao, P. Semantics-based automatic composition of geospatial web service chains. *Comput. Geosci.* **2007**, *33*, 649–665.
5. Yue, P.; Gong, J.; Di, L. Augmenting geospatial data provenance through metadata tracking in geospatial service chaining. *Comput. Geosci.* **2010**, *36*, 270–281.
6. Sorokine, A. Implementation of a parallel high-performance visualization technique in GRASS GIS. *Comput. Geosci.* **2007**, *33*, 685–695.

7. Huang, Q.; Yang, C.; Benedict, K.; Rezgui, A.; Xie, J.; Xia, J.; Chen, S. Using adaptively coupled models and high-performance computing for enabling the computability of dust storm forecasting. *Int. J. Geogr. Inf. Sci.* **2013**, *27*, 765–784.
8. Armbrust, M.; Fox, A.; Griffith, R.; Joseph, A.D.; Katz, R.; Konwinski, A.; Lee, G.; Patterson, D.; Rabkin, A.; Stoica, I.; *et al.* A view of cloud computing. *Commun. ACM* **2010**, *53*, 50–58.
9. Briggs, D.J.; Collins, S.; Elliott, P.; Fischer, P.; Kingham, S.; Lebre, E.; Pryl, K.; van reeuwijk, H.; Smallbone, K.; van der veen, A. Mapping urban air pollution using GIS: A regression-based approach. *Int. J. Geogr. Inf. Sci.* **1997**, *11*, 699–718.
10. Chuvieco, E. Integration of linear programming and GIS for land-use modelling. *Int. J. Geogr. Inf. Sci.* **1993**, *7*, 71–83.
11. Bojórquez-Tapia, L.A.; Diaz-Mondragón, S.; Ezcurra, E. GIS-based approach for participatory decision making and land suitability assessment. *Int. J. Geogr. Inf. Sci.* **2001**, *15*, 129–151.
12. Mathiyalagan, V.; Grunwald, S.; Reddy, K.R.; Bloom, S.A. A WebGIS and geodatabase for Florida's wetlands. *Comput. Electron. Agric.* **2005**, *47*, 69–75.
13. Kraak, M. The role of the map in a Web-GIS environment. *J. Geogr. Syst.* **2004**, *6*, 83–93.
14. Jia, Y.; Zhao, H.; Niu, C.; Jiang, Y.; Gan, H.; Xing, Z.; Zhao, X.; Zhao, Z. A WebGIS-based system for rainfall-runoff prediction and real-time water resources assessment for Beijing. *Comput. Geosci.* **2009**, *35*, 1517–1528.
15. Castronova, A.M.; Goodall, J.L.; Elag, M.M. Models as web services using the Open Geospatial Consortium (OGC) Web Processing Service (WPS) standard. *Environ. Model. Softw.* **2013**, *41*, 72–83.
16. Chen, Z.; Chen, N.; Yang, C.; Di, L. Cloud computing enabled Web processing service for Earth observation data processing. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2012**, *5*, 1637–1649.
17. Di, L.; Chen, A.; Yang, W.; Liu, Y.; Wei, Y.; Mehrotra, P.; Hu, C.; Williams, D. The development of a geospatial data Grid by integrating OGC web services with globus-based Grid technology. *Concurr. Comput. Pract. Exp.* **2008**, *20*, 1617–1635.
18. Zhang, T.; Tsou, M. Developing a grid-enabled spatial web portal for internet GIServices and geospatial cyberinfrastructure. *Int. J. Geogr. Inf. Sci.* **2009**, *23*, 605–630.
19. Han, W.; Yang, Z.; Di, L.; Mueller, R. CropScape: A web service based application for exploring and disseminating US conterminous geospatial cropland data products for decision support. *Comput. Electron. Agric.* **2012**, *84*, 111–123.
20. Granell, C.; Díaz, L.; Gould, M. Service-oriented applications for environmental models: Reusable geospatial services. *Environ. Model. Softw.* **2010**, *25*, 182–198.
21. Chen, N.; Di, L.; Yu, G.; Gong, J. Geo-processing workflow driven wildfire hot pixel detection under sensor web environment. *Comput. Geosci.* **2010**, *36*, 362–372.
22. Giersch, J.; Weidemann, A.; Anton, G. ROSI—An object-oriented and parallel-computing Monte Carlo simulation for X-ray imaging. *Nuclear Instrum. Methods Phys. Res. Sect. A Accel. Spectrom. Detect. Assoc. Equip.* **2003**, *509*, 151–156.
23. Gröger, T.; Zimmermann, R. Application of parallel computing to speed up chemometrics for GC×GC-TOFMS based metabolic fingerprinting. *Talanta* **2011**, *83*, 1289–1294.
24. Vital, J.A.; Gaurut, M.; Lardy, R.; Viovy, N.; Soussana, J.F.; Bellocchi, G.; Martin, R. High-performance computing for climate change impact studies with the Pasture Simulation model. *Comput. Electron. Agric.* **2013**, *98*, 131–135.

25. Huang, F.; Liu, D.; Tan, X.; Wang, J.; Chen, Y.; He, B. Explorations of the implementation of a parallel IDW interpolation algorithm in a Linux cluster-based parallel GIS. *Comput. Geosci.* **2011**, *37*, 426–434.
26. Deslippe, J.; Samsonidze, G.; Strubbe, D.A.; Jain, M.; Cohen, M.L.; Louie, S.G.; Berkeley, G.W. A massively parallel computer package for the calculation of the quasiparticle and optical properties of materials and nanostructures. *Comput. Phys. Commun.* **2012**, *183*, 1269–1289.
27. Wang, L.; Ma, Y.; Zomaya, A.Y.; Ranjan, R.; Chen, D. A Parallel File System with Application-Aware Data Layout Policies for Massive Remote Sensing Image Processing in Digital Earth. *IEEE Trans. Parallel Distrib. Syst.* **2015**, *26*, 1497–1508.
28. Yang, C.; Wong, D.W.; Yang, R.; Kafatos, M.; Li, Q. Performance-improving techniques in web-based GIS. *Int. J. Geogr. Inf. Sci.* **2005**, *19*, 319–342.
29. Ma, Y.; Wang, L.; Zomaya, A.Y.; Chen, D.; Ranjan, R. Task-Tree based Large-Scale Mosaicking for Massive Remote Sensed Imageries with Dynamic DAG Scheduling. *IEEE Trans. Parallel Distrib. Syst.* **2013**, *25*, 2126–2137.
30. Tabik, S.; Romero, L.F.; Zapata, E.L. High-performance three-horizon composition algorithm for large-scale terrains. *Int. J. Geogr. Inf. Sci.* **2011**, *25*, 541–555.
31. Garland, M.; Le Grand, S.; Nickolls, J.; Anderson, J.; Hardwick, J.; Morton, S.; Phillips, E.; Zhang, Y.; Volkov, V. Parallel computing experiences with CUDA. *IEEE Micro* **2008**, *28*, 13–27.
32. Liu, P.; Yuan, T.; Yan, M.A.; Wang, L.; Liu, D.; Yue, S.; Kołodziej, J. Parallel processing of massive remote sensing images in a gpu architecture. *Comput. Inform.* **2014**, *33*, 197–217.
33. Christophe, E.; Michel, J.; Inglada, J. Remote sensing processing: From multicore to GPU. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2011**, *4*, 643–652.
34. Zhang, T.; Li, J. Online task scheduling for LiDAR data preprocessing on hybrid GPU/CPU devices: A reinforcement learning approach. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 386–397.
35. Clarke, K.C. Geocomputation's future at the extremes: High performance computing and nanoclients. *Parallel Comput.* **2003**, *29*, 1281–1298.
36. Nathani, A.; Chaudhary, S.; Somani, G. Policy based resource allocation in IaaS cloud. *Future Gener. Comput. Syst.* **2012**, *28*, 94–103.
37. Xia, H.; Karimi, H.A.; Meng, L. Parallel implementation of Kaufman's initialization for clustering large remote sensing images on clouds. *Comput. Environ. Urban Syst.* **2014**, doi:10.1016/j.compenvurbsys.2014.06.002
38. Tang, W.; Feng, W. Parallel map projection of vector-based big spatial data: Coupling cloud computing with graphics processing units. *Comput. Environ. Urban Syst.* **2014**, doi:10.1016/j.compenvurbsys.2014.01.001
39. Cusumano, M. Cloud computing and SaaS as new computing platforms. *Commun. ACM* **2010**, *53*, 27–29.
40. Huang, Q. Development of a SaaS application probe to the physical properties of the Earth's interior: An attempt at moving HPC to the cloud. *Comput. Geosci.* **2014**, *70*, 147–153.
41. Ekanayake, J.; Fox, G. High Performance Parallel Computing with Clouds and Cloud Technologies. In *Cloud Computing*; Springer Verlag: Berlin, Germany, 2010.

42. Mauch, V.; Kunze, M.; Hillenbrand, M. High performance cloud computing. *Future Gener. Comput. Syst.* **2013**, *29*, 1408–1416.
43. Yang, C.; Goodchild, M.; Huang, Q.; Nebert, D.; Raskin, R.; Xu, Y.; Bambacus, M.; Fay, D. Spatial cloud computing: How can the geospatial sciences use and help shape cloud computing? *Int. J. Digit. Earth* **2011**, *4*, 305–329.
44. Tan, X.; Di, L.; Deng, M.; Chen, A.; Huang, F.; Peng, C.; Gao, M.; Yao, Y.; Sha, Z. Cloud-and agent-based Geospatial service chain: A case study of submerged crops analysis during flooding of the Yangtze River basin. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 1359–1370.

© 2015 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>).