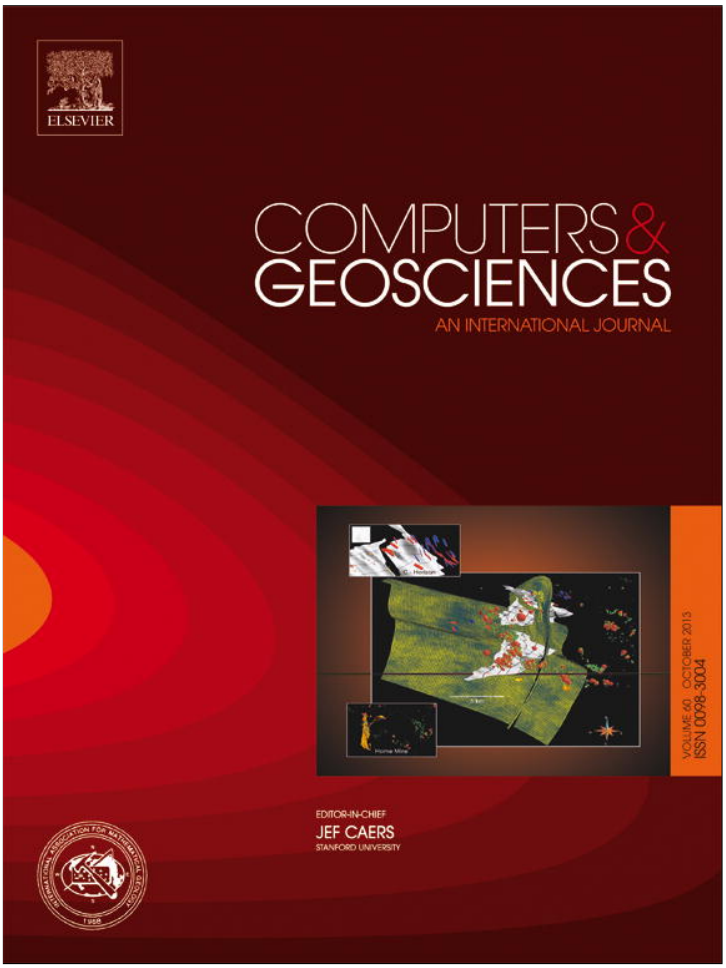
Provided for non-commercial research and education use.
Not for reproduction, distribution or commercial use.



This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

http://www.elsevier.com/authorsrights

Author's personal copy

Computers & Geosciences 60 (2013) 109–116

Contents lists available at ScienceDirect

Computers & Geosciences

journal homepage: www.elsevier.com/locate/cageo

Process virtualization of large-scale lidar data in a cloud computing environment

Haiyan Guan [a], Jonathan Li [b,a,*], Liang Zhong [c], Yongtao Yu [b], Michael Chapman [d]

[a] Department of Geography & Environmental Management, University of Waterloo, Waterloo, Ontario, Canada, N2L 3G1
[b] Key Laboratory for Underwater Acoustic Communication and Marine Information Technology (Xiamen University), Ministry of Education, School of Information Science and Engineering, Xiamen University, Xiamen, Fujian 3610005, China
[c] Changjiang Spatial Information Technology Engineering Co., Changjiang Institute of Survey Planning Design and Research, Wuhan, Hubei 430010, China
[d] Department of Civil Engineering, Ryerson University, Toronto, Ontario, Canada, M5B 2K3

ARTICLE INFO

Article history:
Received 14 December 2012
Received in revised form
12 July 2013
Accepted 15 July 2013
Available online 23 July 2013

Keywords:
Process virtualization
Lidar
Condor
Computing environment

ABSTRACT

Light detection and ranging (lidar) technologies have proven to be the most powerful tools to collect, within a short time, three-dimensional (3-D) point clouds with high-density, high-accuracy and significantly detailed surface information pertaining to terrain and objects. However, in terms of feature extraction and 3-D reconstruction in a computer-aided drawing (CAD) format, most of the existing stand-alone lidar data processing software packages are unable to process a large volume of lidar data in an effective and efficient fashion. To break this technical bottleneck, through the design of a Condor-based process virtualization platform, we presented in this paper a novel strategy that uses network-related computational resources to process, manage, and distribute vast quantities of lidar data in a cloud computing environment. Three extensive experiments with and without a cloud computing environment were compared. The experiment results demonstrated that the proposed process virtualization approach is promisingly applicable and effective in the management of large-scale lidar point clouds.

© 2013 Elsevier Ltd. All rights reserved.

1. Introduction

Light detection and ranging (lidar) technologies, including airborne, mobile and terrestrial laser scanning (ALS, MLS, and TLS), have gradually become common mapping practices in three-dimensional (3-D) data acquisition. Existing lidar systems can provide point clouds with a point density of up to thousands of points per square metre. For example, an Optech® Lynx Mobile Mapper captures a total of 144 million points of five blocks in 20 min (Conforti and Zampa, 2011). This revolutionary data acquisition technology allows for various applications, such as transportation, utility, forestry, mining, and urban planning. In addition to the huge quantity of point clouds acquired by a laser scanner, multiple high-spatial-resolution cameras (as common components) provide a significantly large quantity of image data. For example, a Trimble® MX-8 system, integrating two RIEGL® VQ-250 laser scanners and four CCD cameras, collects a total of 35 Gigabytes in 20 min (Guan et al. , 2013a).

Furthermore, it is intricate and sophisticated to efficiently store, manage, and process lidar data and images for customized products, such as object identification (Secord and Zakhor, 2007; Guan et al., 2013b), 3D building models (Habib et al., 2005; Zhang et al., 2005; Mitishita et al., 2008; Nebiker et al., 2010) and Digital Orthophoto Maps (DOMs) (Liu et al., 2007) because creating these lidar-derived products are costly and intensive computation due to a variety of methods and a diverse selection of parameters. Current lidar data processing software packages (e.g. QT Modeler, LasTools, InPho, LiDAR Explorer for ArcGIS, Terrasolid, and TopPIT) are with stand-alone and task-oriented features that considerably limit lidar data applications (GIM, 2012); thus, much attention has been paid to research on organizing lidar data more effectively and efficiently.

Although a group of level-of-detail (LOD) variants have proven to be suitable for adaptive visualization of a notably large volume of lidar data, these variants are time-consuming in pre-processing and progressively inefficient in query performance because of their unbalanced tree structures (Pfister et al., 2000; Rusinkiewicz and Levoy, 2000). Moreover, a variety of tree structures, ranging from binary-tree, quad-tree, octree to their combinations, have been presented to accelerate the lidar data processing procedure,

* Corresponding author at: Key Laboratory for Underwater Acoustic Communication and Marine Information Technology (Xiamen University), Ministry of Education, School of Information Science and Engineering, Xiamen University, Xiamen, Fujian 3610005, China. Tel.: +86 059 225 80003.
E-mail addresses: junli@xmu.edu.cn, junli@uwaterloo.ca (J. Li).

0098-3004/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cageo.2013.07.013

Author's personal copy

(Lu and He, 2008; Liu et al., 2008; Kreylos et al., 2008; Elseberg et al. 2011). For example, the binary-tree is employed for building extraction (Sohn et al., 2008); the octree is used for split-and-merge segmentation (Wang and Tseng, 2010). Those two-dimensional (2-D) tree-based algorithms are limited by their unbalanced index structures and one-dimensional (1-D) space partitioning.

To overcome the above two limitations, 3-D R-trees, the most well-known index structure for spatial data, have been applied to the real data by adaptively adjusting index structures (Zhu et al., 2007). In Gong et al. (2012), a hybrid spatial index method (3DOR-Tree) that integrates R-Tree and Octree structures is used to overcome the unbalanced data distribution. Similar to B-tree, R-tree is a height balanced tree that hierarchically splits spaces into possibly overlapping subspaces. However, the tree-node overlapping and complexity of 3-D R-trees cause multipath queries, resulting in lower query efficiencies in lidar data processing.

In most cases, to achieve final lidar-derived products, existing established lidar processing algorithms and software tools must divide substantial amounts of lidar points into a number of data blocks (Pu et al., 2011) and thin out or rasterize the lidar data for a series of post-processing procedures (Van Gosliga et al., 2006; Mongus and Zalik, 2012). Besides software limitations, computer hardware, such as the amount of computer memory ranging from a few hundred megabytes to gigabytes, is usually unable to support intensive calculation requirements if couples of threads are simultaneously implemented for lidar data processing on a multi-processor computer. Thus, a lidar data processing system requires an open, shared and interoperated environment, where data processing, management, and distribution are automatic, intelligent, and real-time. To this end, advanced techniques like parallel processing are used to improve large-scale lidar data processing efficiency by distributing them among multiple shared-servers (Wand et al., 2008; Ma and Wang, 2011). First, through data index structures, such as grid, quadtree, and octree, those techniques, regarding spatial relationship of the datasets, uniformly divide and distribute mass remotely sensed data into data servicers,. A client retrieves the data from the data servers directly and efficiently according to data spatial locations and boundaries, which enables the client to maximize the capability of parallel computing. With advances in Grid Computing, Cloud Computing is a promising choice of massive remotely sensed data processing (Xue et al., 2011). Therefore, besides a high throughput computation and grid workflow for remote sensing quantitative retrieval applications, a cloud computing environment is motivated by the requirement for customized remote sensing products, especially lidar-derived products.

In this paper, we design a Condor-based virtual platform of massive remotely sensed data processing for lidar-derived products in a cloud computing environment, and analyze the platform's performance on three common uses of lidar processing techniques: filtering, DEM interpolation, and DOM generation. Specifically, we take advantage of cloud computing, one of the newest internet-based paradigms in computation in the field of lidar data processing to accomplish the following: (1) solve the expanding data and task-intensive computation problems encountered within customized lidar-derived products as the demand for these products increases; (2) develop a prototype of a Condor-based process virtualization platform for large-scale remotely sensed data processing, analysis, and intensive computation; (3) provide an efficient method for users to make full use of various idle internet resources and established lidar-relevant data processing algorithms; (4) explore the potential for improving the parallel efficiency of our Condor-based process virtualization platform by synchronizing computation and communication procedures.

The remainder of the paper is organized as follows: Section 2 introduces cloud computing. Section 3 presents a Condor-based middleware design for lidar data processing. Section 4 discusses extensive experimental results and evaluates the performance of the proposed process virtualization platform in the cloud computing environment. Finally, Section 5 states the concluding remarks.

2. Cloud computing

The “cloud”, a natural evolution of distributed computing, is of the Web 2.0 protocol and is a particularly widespread virtualization technology. Associated with a new paradigm for the provision of computing infrastructure, cloud computing shifts infrastructure locations from the desktop to the network (Boss et al., 2007; Vaquero et al., 2008). Those network-related capabilities and resources are provided as services, via the on-demand and accessible internet without knowing the detailed knowledge of the underlying technology (Bolze and Deelman, 2010). Cloud computing in the early development period was called “Grid Computing” – a term that originated in the 1990s. The main research of Grid Computing ranged from Giga Ethernet testbed to Metacomputing. In Smarr and Catlett (1992), Metacomputing, focusing on managing and harnessing heterogeneous computational resources, is considered as the prototype of Grid Computing. Typical representative projects of Metacomputing included FAFNER (an internet-based sieving effort from Cooperating Systems Corporation), I-WAY, and Information Wide-Area Year (Foster et al., 1996). FAFNER was followed by distributed projects such as SETI@home (Korpela et al., 2001) and Distributed.Net; whereas, Globus (a toolkit of middleware components for Grid Computing infrastructure) (Foster et al., 2001) and Lehion (an object-based approach to Grid Computing) projects were based on I-WAY. The 1990s period of Grid Computing was characterised by the use of distributed interconnected computers and resources collectively to achieve high performance computational capabilities and resource sharing (Wilkinson, 2010). Grid computing technology subsequently evolved into a wider range of science and engineering disciplines, including biomedical research, industrial research, high-energy physics, bioinformatics, chemistry, earth science, and geometric modelling. In 2001, the Open Grid Services Architecture (OGSA), a new generation of grid structure as a standard of Grid Computing, was originally proposed by Foster et al. (2003) to integrate web services supported by industrial communities with computation services (Foster and Kesselman, 2003).

Network-based remotely sensed data management and distribution systems have achieved significant breakthroughs in commercial software (e.g., Lockheed Martin's Intelligent Library System, the Microsoft Terraserver, Z/I Imaging corporation's TerraShare) and scientific research platforms (e.g., Graz Distributed Server System(GDSS), Data and Information Access Link (DIAL)). However, to the best of our knowledge, most systems and commercial products concentrate on data retrieving, distribution, and storage. Little attention has been paid to fast and effective data processing and analysis services for such a huge volume of remotely sensed data, such as lidar point clouds in particular.

Virtualization is defined as creating something virtually or non-existent rather than having an actual physical version. The process virtualization of remotely sensed data in cloud computing environments is a system that connects internet-oriented service technology with a remote-sensing database for data retrieval, processing and feedback. There are two ways to virtualize lidar data in a cloud computing environment. One is a tightly coupled model for parallel computation; the other is a loosely coupled model in distributed computation. Physical and software interactions are highly inter-dependent with stable and robust communications in the tightly coupled model; while in a loosely coupled architecture, a significantly large number of interactions operate independently in different geographical positions, leading to costly and unreliable communications. Meanwhile, a successful remotely sensed data

Author's personal copy

processing platform not only handles a considerable amount of data with complex attributes and relationships, but also provides a variety of services to deal with a large number of analysis, modelling, and other computational requirements. Therefore, in view of a real network environment, data security and a variety of function requests, the process virtualization of lidar-derived products is built on the tightly coupled model in a cloud computing environment. With the process virtualization, what users need to do is send requests according to their demands. As the lidar-derived products are not standardized, carrying out user requests requires a number of processing procedures and a large quantity of computing resources, resulting in lidar-derived products that must be produced in Grid Computing environments.

3. Middleware component design for lidar data processing

To efficiently process a large volume of lidar data, a Condor-based process virtualization platform is designed to use network-related computational resources in a cloud computing environment. Based on the description of Condor, a middleware design of lidar data processing is introduced.

3.1. Condor

Condor, developed by the University of Wisconsin–Madison in 1998, is a software-development kit based on large collections of distributive computing resources to support high-throughput computing (Thain et al., 2005; Magoulès et al., 2009). Condor is characterized by having a high-throughput computing (Condor computation resource pool) formed by putting available idle computers together on the network as an integral part of many computational grids around the world. Thus, it has been widely used as a distributed batch computing system, and its name was changed to HTCondor in 2012 (http://research.cs.wisc.edu/htcondor/). In addition, Condor's advantages also include support of the following: multiple platforms, checkpoints and progress migration, remote procedure calling systems, internal connections of Condor pools, dynamic expansion, etc. Condor's portability also makes it easy to adjust to a computing pool.

Condor has two components: job and resource management. Job here is defined as a process, or set of processes, executed on the grid. The job management component is responsible for managing job execution. Users can inquire into job queuing or submit a new job. The resource management component focuses on policy scheduling, priority scheming, as well as resource monitoring, distributing, and managing. Through task scheduling and multi-task parallel mechanisms, a Condor-based cloud computing system accelerates task processing. With this fault-tolerant characteristic, the system survives crashes, network outages, or any single point of failure by flexibly transferring tasks to other available resources on the network (Chorafas, 2011).

In terms of the complexity and dynamism of lidar data processing on a large scale, the demand for fast shared and steady data processing software in the network environment requires a process virtualization platform to be an open server-oriented structure. With this structure, job submission, distribution, and management are performed while being closely monitored. From this viewpoint, we present the process virtualization platform of lidar-derived products shown in Fig. 1.

The process virtualization platform of lidar-derived products consists of multiple computing pools (clusters) in the network segments with management systems installed at the frontend computers. The management systems are responsible for interconnecting the computer pools. Intermediate, or final, results of lidar data are stored in each computing pool's data centre. The advantages of storing lidar data, or lidar-derived products, in the data storage centres are two-fold: high data-processing efficiency due to infrequent lidar-data communication in the network and high service efficiency due to the reusability of the intermediate or final results of lidar data for other lidar-relevant algorithms.

3.2. Middleware components

A complete fault tolerant middleware (defined as a collection of software and packages used for the implementation of a grid)

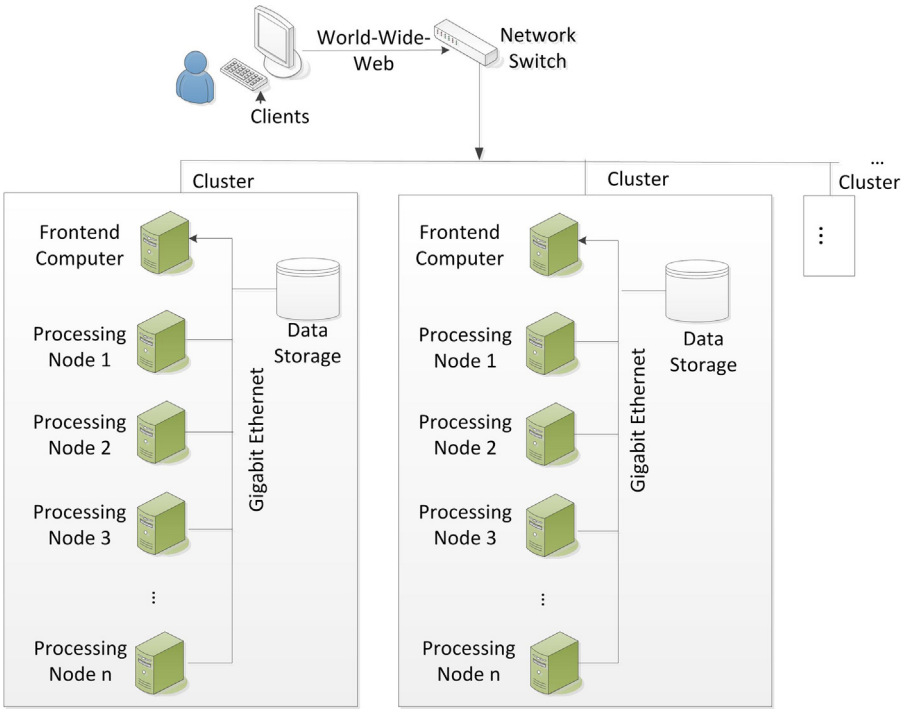


Fig. 1. Virtualization platform of lidar data products.

Author's personal copy

consists in many interconnected components. Because Condor technology focuses on the user concerns of job scheduling, job submission, job allocation, error recovery, and creation of a user-friendly environment, it provides solutions for both the frontend and backend of the middleware (Besserson et al., 2010). Condor-based middleware design is introduced and described here, with the key components shown in Fig. 2.

(1) Client
User-specific tasks and a variety of computational parameters are first submitted. Then, based on the description of these tasks, this model searches for specialized processing services on the network and submits the tasks and their corresponding data resources to the Condor computation resource pool for cloud computing.
(2) Lidar data processing server
Lidar data processing mainly includes the following services: data calibration, registration, DEM generation, land-use classification, digital orthoimage generation, and 3-D building reconstruction. These processing services have already been implemented in the stand-alone lidar data processing system. Under the distributed cloud computing environment, each lidar data processing service is treated as a single task implemented by a specific processing algorithm. Therefore, it is convenient for users to perform many kinds of algorithms to process lidar data, and upgrade processing software by changing only relevant processing models.
(3) Computational resource inquiring server
This model is responsible for collecting computational resources in the network and transferring tasks to resource offers.
(4) Data centre
The data centre stores, not only remotely sensed data (imagery, lidar point clouds, and GIS data), but also remotely sensed intermediate and final data processing results that have a high rate of repeated use. How all the remotely sensed data is stored in the data centre will support stability, rapidity and efficiency of data communication.
(5) Monitoring and dispatching
This model is in charge of monitoring computer nodes such as their CPUs, memories, and workload. The model also tracks the progress of tasks such as job completion, waiting, and failure.

Due to the unstructured nature of lidar points, lidar data can be processed hierarchically and separately. As a result, by integrating multiple idle computers on the internet, this distributed structure in cloud computing environments contributes to the processing of voluminous lidar data through the presented Condor-based middleware design that transfers a traditional single thread task into multi-thread parallel tasks. A combination of a high-throughput grid platform and unique data characteristics of remotely sensed data can accelerate computations and significantly increase data processing efficiency.

3.3. Task scheduling and load-balancing

Scheduling, a process of ordering the execution of a collection of tasks in a pool of resources, is one of the cores of the heterogeneously distributed cloud computing environment. When the size of the data to be processed increases, the overhead associated with it also increases; hence, the efficiency of the communication decreases. In cases of a huge volume of remotely sensed data, the data to be processed are usually satellite images, aerial images, and point clouds. Thus, the minimum granularity is usually a single image or a scanning strip of points as shown in the following two examples: (1) During the task of ortho-rectification,

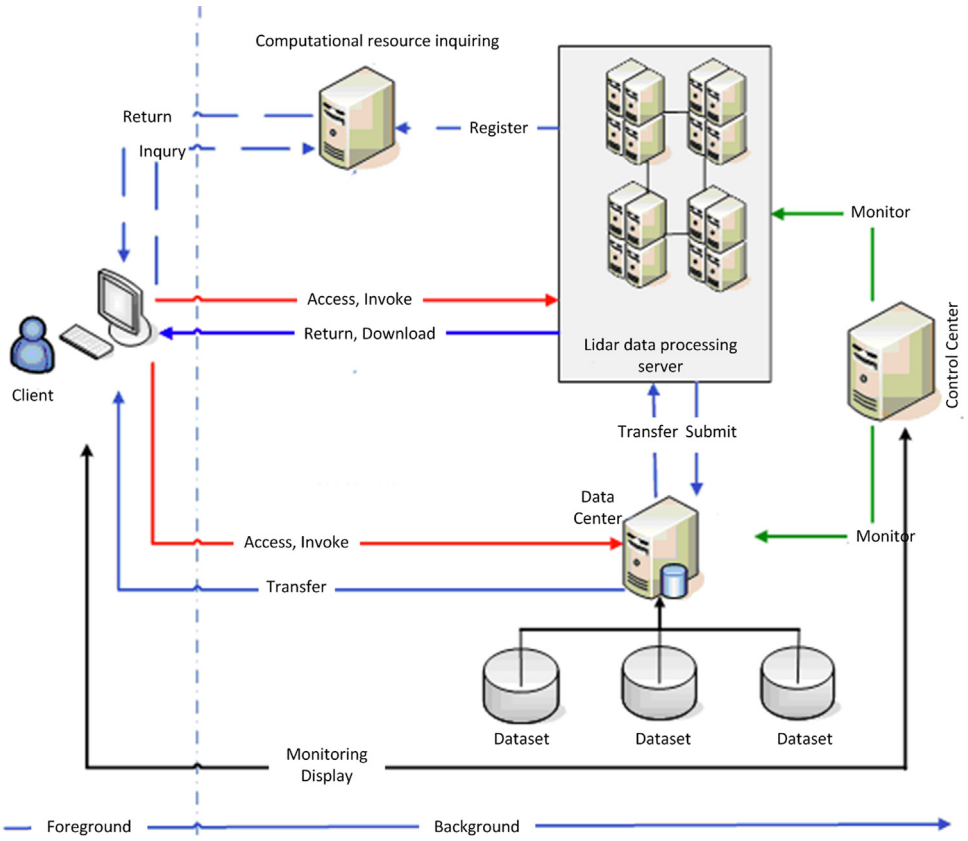


Fig. 2. The middleware design.

Author's personal copy

we subdivide a large task into a number of subtasks and distribute them in process units of one image (that is, a job) each to computational resources for ortho-rectification. (2) We divide lidar data into a number of blocks according to point density and minimum application requirements (e.g. an area of 1 square kilometre, or a scanning strip), and then distribute the blocks in process units of one block each to computational resources. After determining processing granularity, the job is added to its local job queue. Based on the recordings of the jobs in the local job queue, job allocation is performed to distribute those jobs to computational resources in the network.

Load-balancing provided by Condor supports online migration process between computational resources, that is, this Condor-supported load-balancing scheme ensures appropriate workload allocation by automatically migrating processes to idle computational resources for jobs. Thus, no human interaction is needed.

4. Results and discussion

To assess performance, we use two study sites in different cities in China. All lidar processing algorithms, used in this study, such as progressive Triangular Irregular Network (TIN) densification filtering, DEM interpolation, and ortho-rectification, have been successfully implemented in stand-alone stations.

4.1. Study sites and data description

The following two sites are included within this study, as shown in Fig. 3.

4.1.1. Dunhuang city

The first study site, Dunhuang City (a major stop on the ancient Silk Road) is located in northwestern Gansu Province, Western China. An arid, continental climate is typical in this mountainous area. Due to prolonged overgrazing of the surrounding land, Dunhuang City was gradually invaded by the expansion of the Kumtag Desert. Study area I, an area of about 280 km^2, is a mix of urban area, desert, and Mogao Caves. (Mogao Caves are well-known Buddhist cave sites, located 25 km southwest of Dunhuang City.) The lidar data were acquired in October, 2009, using a Leica ALS50-II with 15 scan strips and data volume of 16 GB at absolute altitudes ranging from 760 to 3200 m above Mean Sea Level. The laser sensor's specifications are designed according to the site-specific terrain. For example, Dunhuang City (A) is a city with features such as apartments, narrow streets, and industrial facilities, distinct from the desolate desert area (C) surrounding the city. Relatively, the area of Mogao Caves (B) is crowded with a number of Buddhist caves. Thus, to capture more terrain details and avoid occlusion, region B requires denser points than regions A and C. The detailed data-acquisition parameters are reported in Table 1.

4.1.2. Xi'an city

The second study site, Xi'an City, is the capital of Shanxi province (the province adjacent to Gansu). As part of the economic revival of interior China (especially for the central-northwest regions), Xi'an City features groups of ancient architecture that co-exist with modern high-rise buildings. Study area two is a typical urban area with larger-sized industrial or commercial buildings, small and large residential buildings, as well as some

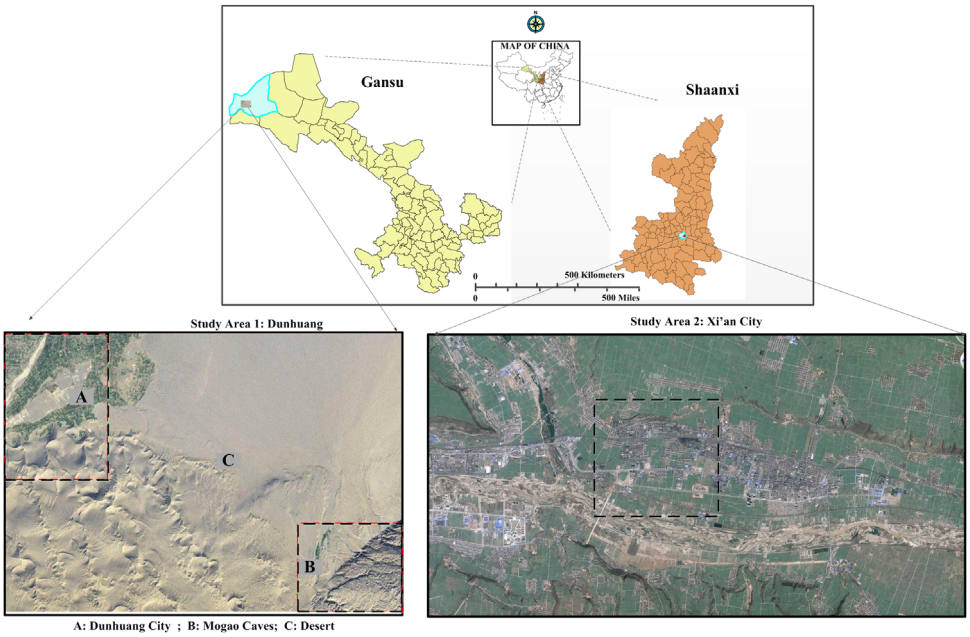


Fig. 3. Study areas for assessment of the proposed Condor-based process virtualization platform.

Table 1
The specifications of the study area in Dunhuang.

Regions	Flying height (m)	Scan angle (°)	Scan frequency (kHz)	Pulse repetition frequency (kHz)	Point density (points/m^2)
Mogao-caves area (A)	760	45.0	42.5	122.6	0.5
Urban area (B)	1310	45.0	34.8	82.2	0.9
Desert area (C)	3200	50	13.9	37.1	2.1

Author's personal copy

tessellated farmland. The Leica ALS50-II sensor was operated at a fixed wavelength (1064 nm) and flown at an absolute altitude of 1900 m above Mean Sea Level with a ground speed of about 120 km/h. The sensor had a laser divergence of 0.22 mrad, and generated points at nadir for a 45° field of view. The lidar data have average point spacing of 0.4 m in the along- and across-track directions, with a horizontal accuracy of ±27 cm and a vertical accuracy of ±15 cm. The currently available systems offer pulse repetition frequency values between 20 kHz and 160 kHz. A RCD105 digital frame camera was integrated with the Leica ALS50-II system, providing four photographic strips composed of a total of 49 images. Each aerial image has a size of 7162 by 5389 pixels with a ground sample distance (GSD) of 13 cm.

4.2. Results

All processing procedures were implemented by VC++ in Microsoft Visual 6.0 with the Condor 7.2 version download from Condor open source (http://research.cs.wisc.edu/htcondor/index.html). After customizing the configuration, we create a small cloud computing pool consisting of multiple computer nodes. Meanwhile, we develop a graphical task generation module for users to submit and monitor their jobs that produce the lidar-derived products using Application Programming Interfaces (APIs) provided by the Condor source. The test platform is composed of 12 computers as a cluster, among which, eight computers are configured with CPUs of 3.0 GHz and memories of 2 GB. The other four computers are deployed with CPUs of 2.66 GHz and memories of 1 GB. All 12 computers function as computational nodes. One of the nodes has the responsibility for resource inquiring and managing. To simplify the designed system structure, this node also coordinates computational tasks and management. The cluster's components are internally connected with each other through a one-GB Ethernet with maximum speed of 50 MB/s under the Microsoft Windows XP Operating System. In this study, we test the following three lidar data processing models, which have already been successfully implemented in stand-alone stations: filtering, DEM generation, and DOM services. Thus, based on distributed cloud computing environments, these algorithms, which can share resources on the internet to the maximum extent, can be directly invoked by constructing corresponding interface programs without any adjustments.

Based on the computational demand for performing a certain task, the proposed process virtualization platform searches the optimum idle hardware resources required for computation in the cloud computing environment. The optimum speed of the lidar data processing is obtained when computational resources on the internet are used. Table 2 compares the lidar data processing time for the above three services with four clusters of 3, 6, 9, and 12 nodes, respectively. Study area I was used to test the filtering and DEM generation services. Study area II, Xi'an City, was used to assess the computational performance of DOM generation in cloud computing environments. According to Table 2, for processing a large volume of lidar data, the proposed Condor-based process virtualization platform is more effective than that of the conventional stand-alone service.

Fig. 4(a) shows the results of DEM generation for study area I (Dunhuang dataset) using a filtering service called progressive TIN densification. A visual inspection confirms that the results represent the terrain features quite well. Quantitatively, an overall

Table 2
Tests of cloud computing.

Operations	Filtering	DEM generation	DOM generation
Algorithms	Progressive triangulated irregular network (TIN) densification	Moving surface interpolation	Differential rectification
Datasets	Study area I	Study area I	Study area II
Input dataset and size	Dunhuang dataset, 30 blocks (2.83 GB)	Dunhuang dataset, 30 blocks (2.83 GB)	Xi'an-city dataset, 40 images (4.31 GB) DEM (2.93 GB)
Output data size	2.83 GB	960 MB	4.30 GB
Total amount of data size	5.66 GB	3.79 GB	11.54 GB
Processing time of stand-alone	13′27″	91′06″	14′40″
Processing time of distributed cloud computing			
3 Nodes	9′31″	63′48″	12′13″
6 Nodes	7′20″	27′42″	10′22″
9 Nodes	6′18″	21′34″	9′46″
12 Nodes	5′42″	14′20″	9′20″

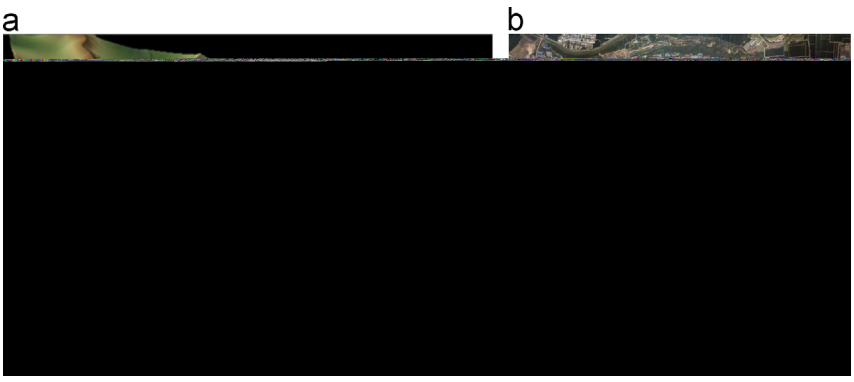


Fig. 4. The processing results; (a) DEM generation; (b) DOM.

Author's personal copy

accuracy of up to 20 cm is achieved for DEM over the desert area. Because the lidar data are uniformly divided into blocks, each of which is distributed to a computation node as a processing unit, there are some break-lines among the filtering results of the adjacent blocks. A post-adjustment is thus required to remove those sharp break-lines according to the accuracy required by the users.

Fig. 4(b) shows the DOM results obtained with the data from study area II. After using the indirect differential rectification method, the ortho-rectified image of 0.5 m GSD has no camera lens distortion.

4.3. Discussion

In general, the following two indicators are used to assess the performance of parallel computation in the Condor-based process virtualization platform: speedup and efficiency, defined as:

$$I_{Speedup} = T_S/T_p \quad (1)$$

$$I_{Efficiency} = T_S/(P \times T_p) \quad (2)$$

where, $T_S$is the time complexity in sequential data processing, $T_p$is the time complexity in parallel data processing, and P is the number of processors. However, because the number of processors on the internet is uncertain, we use $I_{Speedup}$to evaluate the performance of lidar data processing in the distributed cloud computing environment.

Table 3 displays the speedup differences of 3, 6, 9, and 12 nodes in the proposed cloud computing environment. As seen in Table 3, the efficiency of parallel processing is growing with an increase of the number of nodes. However, the speedup is not strictly proportional to the number of nodes owing to heavy data communication. In fact, data communication time, compared to computational time, accounts for a significant proportion of whole lidar data processing time in the cloud computing environment. As a result, the speedup of DOM generation from a large volume of images by an algorithm of differential rectification is smaller than that of DEM interpolation from a relatively small lidar dataset. In this study, the speedups of DEM interpolation are 1.24, 2.32, 2.81, and 4.04 times more than those of DOM generation at 3, 6, 9, and 12 nodes, respectively. Meanwhile, the ratio of the speedup to data volume (V) also explains that data volume plays an essential role in the performance of speedup, as seen in Fig. 5. In other words, the communication speed is proportional to the size of the input and output data. In view of the ratio of speedup to data volume, it seems that the Condor-based distribution platform in the field of remote sensing does not enhance processing efficiency as significantly as other science and engineering fields. The philosophy behind this phenomenon is that processing platforms of remotely sensed data are required not only to deal with large volume of input data, but also to output even larger volume of results, which is totally different from other disciplines that just require a small amount of computational results. Compared to stand-alone data processing modes, the Condor-based process virtualization in the cloud computing environment can enhance efficiency of data processing to a large extent. Besides filtering, DEM and DOM generation, the presented scheme could be extended to other more complicated and time-consuming algorithms in the lidar data processing systems, including decomposition of lidar waveform data, registration of lidar points with high-resolution images, feature extraction and 3D object reconstruction fusing lidar data with images. Aiming at the unusual situation of remotely sensed data, such as large-scale inputs and outputs, we could synchronize computation and communication to reduce network delay.

Table 3
A comparison of speedup between different nodes.

Speedup and ratio of speedup to data volume	Filtering		DEM generation		DOM Generation	
	$I_{Speedup}$	$I_{Speedup}$/ V	$I_{Speedup}$	$I_{Speedup}$/ V	$I_{Speedup}$	$I_{Speedup}$/ V
3 Nodes	1.4133	0.2497	1.4279	0.3768	1.2005	0.0998
6 Nodes	1.8340	0.3240	3.2888	0.8678	1.4148	0.1226
9 Nodes	2.1349	0.3772	4.2241	1.1145	1.5017	0.1301
12 Nodes	2.3596	0.4169	6.3558	1.6770	1.5714	0.1362

5. Conclusions

This paper introduced a Condor-based process virtualization platform on which we employed the concept of process virtualization for the processing, configuring, and managing of large-scale remotely sensed data in a cloud computing environment. The designed Condor-based middleware in this study is a fundamental research and prototype of information retrieval, extraction, and applications. With this middleware, data and task-intensive computations for a substantial amount of lidar data have been achieved. Extensive experiments showed that the Condor-based process virtualization platform of lidar data is applicable as well as flexible to customize lidar-derived products.

The parallel performance of the proposed process virtualization platform could be improved in the following two ways: synchronize computation and communication to reduce network delay; increase granular computing and decrease communication overhead using redundant computation. In addition, a user-friendly post-adjustment strategy could be improved to remove some break-lines among the adjacent blocks. Future research will incorporate such strategies into parallel distributed and network based processing workflows of remotely sensed data in cloud

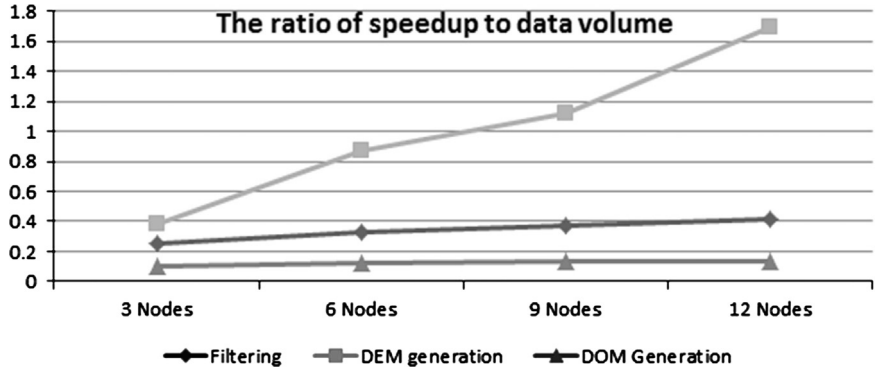


Fig. 5. The ratio of Speedup to data volume.

Author's personal copy

computing environments. There are some break-lines among the filtering results of the adjacent blocks. A post-adjustment is thus required to remove those sharp break-lines according to the accuracy required by the users.

Acknowledgements

The authors would like to acknowledge Mr. Michael McAllister, sessional lecturer at the College of Informatics, Xiamen University, China for proofreading the paper.

References

Besserson, X., Bouguerra, M., Gautier, T., Saule, E., Trystram, D., 2010. Fault-tolerance and availability awareness in computational grids. In: Ahson, S.A., Ilyas, M. (Eds.), Cloud Computing and Software Services: Theory and Techniques. CRC Press, Boca Raton, FL, p. 163.

Bolze, R., Deelman, E., 2010. Exploiting the cloud of computing environments: an application's perspective. In: Ahson, S.A., Ilyas, M. (Eds.), Cloud Computing and Software Services: Theory and Techniques. CRC Press, Boca Raton, FL, pp. 173–199.

Boss, G., Malladi, P., Quan, D., 2007. Cloud Computing, IBM White Paper. ⟨http://download.boulder.ibm.com/ibmdl/pub/software/dw/wes/hipods/Cloud_computing_wp_final_8Oct.pdf⟩.

Chorafas, D.N., 2011. Cloud Computing Strategies. CRC Press, Boca Raton, FL p. 241.

Conforti, D., Zampa, F., 2011. Lynx Mobile Mapper for Surveying City Centres and Highways. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 38(5/W16) 2011 ISPRS Trento 2011 Workshop, 2–4 March 2011, Trento, Italy.

Elseberg, J., Borrmann, D., Nuchter, A., 2011. Efficient processing of large 3D point clouds, 2011 XXIII International Symposium on Information, Communication and Automation Technologies (ICAT), 27–29 Oct. 2011.

Foster, I., Geisler, J., Tuecke, S., 1996. MPI on the I-WAY: a wide-area, multimethod implementation of the message passing interface. In: Proceedings MPI Developers Conference, Notre Dame – South Bend, 1–2 July,1996, pp. 10–17.

Foster, I., Kesselman, C., Tuecke, S., 2001. The anatomy of the grid: enabling scalable virtual organizations. The International Journal of High Performance Computing Applications 15, 200–222.

Foster, I., Kesselman, C., 2003. The Grid: Blueprint for a New Computing Infrastructure, 2nd ed. Morgan Kaufmann, San Francisco, CA p. 59.

Foster, I., Kesselman, C., Nick, J.M., Tuecke, S., 2003. The physiology of the grid. In: Berman, F., Fox, G., Hey, T. (Eds.), Grid Computing: Making the Global Infrastructure a Reality. John Wiley & Sons, Ltd, Chichester, UKhttp://dx.doi.org/10.1002/0470867167.ch8.

Gong, J., Zhu, Q., Zhong, R., Zhang, Y., Xie, X., 2012. An efficient point cloud management method based on a 3D R-Tree. Photogrammetric Engineering & Remote Sensing 78 (4), 373–381.

GIM, 2012, Airborne Lidar Processing Software, 2012. ⟨http://www.gim-international.com/⟩, (accessed 08.04.2013).

Guan H., Li, J., Yu, Y., 2013a. Geometric validation of a mobile laser scanning system for urban applications. In: The International Symposium on Mobile Mapping Technology 2013(MMT2013), Taiwan, China, 1–3 May, 2013.

Guan, H., Ji, Z., Zhong, L., Li, J., Ren, Q., 2013b. Partially supervised hierarchical classification for urban features from lidar data with aerial imagery. International Journal of Remote Sensing 34 (1), 190–210.

Habib, A., Ghanma, M., Morgan, M., Al-Ruzouq, R., 2005. Photogrammetric and LiDAR data registration using linear features. Photogrammetric Engineering & Remote Sensing 71 (6), 699–707.

Liu, H., Huang, Z., Zhan, Q., Lin, P., 2008. A database approach to very large LiDAR data management. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Beijing, China, 37(B1):463–468.

Liu, X., Zhang, Z., Peterson, J., Chandra, S., 2007. LiDAR-derived High Quality Ground Control information and DEM for image Orthorectification. GeoInformatica 11 (1), 37–53.

Lu, M.Y., He, Y.J., 2008. Organization and indexing method for 3D points cloud data. Geo-information Science 10 (2), 190–194.

Korpela, E., Werthimer, D., Anderson, D., Cobb, J., Lebofsky, M., 2001. SETI@home: massively distributed computing for SETI. Computing in Science and Engineering 3 (5), 78–83.

Kreylos, O., Bawden, G.W., Kellogg, L.H., 2008. Immersive visualization and analysis of LiDAR Data. In: Proceeding of the Fourth International Symposium on Advances in Visual Computing, Springer-Verlag, Berlin Heidelberg, pp. 846–855.

Ma, H., Wang, Z., 2011. Distributed data organization and parallel data retrieval methods for huge scanner point clouds. Computer & Geoscience 37 (1), 193–201.

Magoulès, F., Pan, J., Tan, K., Kumar, A., 2009. Introduction to Grid Computing. CRC Press, Taylor & Francis Group. (6000 Broken Sound Parkway NW, Suite 300, Boca Raton, FL).

Mitishita, E., Habib, A., Centeno, J., Machado, A., Lay, J., Wong, C., 2008. Photogrammetric and LiDAR data integration using the centroid of a rectangular roof as a control point. The Photogrammetric Record 23 (121), 19–35.

Mongus, D., Zalik, B., 2012. Parameter-free ground filtering of Lidar data for automatic DTM generation. ISPRS Journal of Photogrammetry and Remote Sensing 67, 1–12.

Nebiker, S., Bleisch, S., Christen, M., 2010. Rich point cloud in virtual golbes – a new paradigm in city modelling? Computer, Environment and Urban Systems 34, 508–517.

Pfister, H., Zwicker, M., Van Baar, J., Gross, M., 2000. Surfels: surface elements as rendering primitives. In: Proceedings of ACM SIGGRAPH 2000, 23–28 July 2000, New Orleans, Louisiana, USA, pp. 335–342.

Pu, S., Rutzinger, M., Vosselman, G., Elberink, S.O., 2011. Recognizing basic structures from mobile laser scanning data for road inventory studies. ISPRS Journal of Photogrammetry and Remote Sensing 66, S28–S39.

Rusinkiewicz, S., Levoy, M., 2000. QSplat: a multiresolution point rendering system for large meshes. In: Proceedings of ACM SIGGRAPH 2000, 23–28 July 2000, New Orleans, Louisiana, USA, pp. 343–352.

Secord, J., Zakhor, A., 2007. Tree detection in urban regions using aerial lidar and image data. IEEE Geoscience and Remote Sensing Letters 4 (2), 196–200.

Smarr, L., Catlett, C., 1992. MetaComputing. Communications of the ACM 35 (6), 44–52.

Sohn, G., Huang, X., Tao, V., 2008. Using binary space partitioning tree for reconstructing 3D building models from airborne LiDAR data. Photogrammetric Engineering & Remote Sensing 74 (11), 1425–1440.

Thain, D., Tannenbaum, T., Livny, M., 2005. Distributed computing in practice: the Condor experience. Concurrency and Computation: Practice and Experience 17, 323–356.

Van Gosliga, R., Lindenbergh, R., Pfeifer, N., 2006. Deformation analysis of a bored tunnel by means of terrestrial laser scanning. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 36 (5), 167–172.

Vaquero, L.M., Rodero-Merino, L., Caceres, J., Lindner, M., 2008. A break in the clouds: towards a cloud definition. SIGCOMM Computer Communication Review 39 (1). (December 2008).

Wand, M., Berner, A., Bokeloh, M., Jenke, P., 2008. Processing and interactive editing of huge point clouds from 3D scanners. Computer & Graphics 32 (2), 204–220.

Wang, M., Tseng, Y.H., 2010. Automatic segmentation of LiDAR data into coplanar point clusters using an octree-based split-and-merge algorithm. Photogrammetric Engineering and Remote Sensing 76 (4), 407–420.

Wilkinson, B., 2010. Grid Computing: Techniques and Applications. CRC Press, Taylor & Francis Group, Boca Raton, FL p. 1.

Xue, Y., Chen, Z., Xu, H., Ai, J., Jiang, S., Li, Y., Wang, Y., Guang, J., Mei, L., Jiao, X., He, X., Hou, T., 2011. A high throughput geocomputing system for remote sensing quantitative retrieval and a case study. International Journal of Applied Earth Observation and Geoinformation 13 (6), 902–911.

Zhu, Q., Gong, J., Zhang, Y.T., 2007. An efficient 3D R-tree spatial index method for virtual geographic environments. ISPRS Journal of Photogrammetry & Remote Sensing 62 (3), 217–224.

Zhang, Y., Zhang, Z., Zhang, J., Wu, J., 2005. 3D building modelling with digital map, lidar data and video image sequences. The Photogrammetric Record 20 (111), 285–302.