

Evaluating Efficiency of Spatial Analysis in Cloud Computing Platforms

Changlock Choi

Department of Geography, Kyung Hee University, Seoul, South Korea
hihi7100@khu.ac.kr

Yelin Kim


Department of Geography, Kyung Hee University, Seoul, South Korea
yelin910@khu.ac.kr

Youngho Lee

Department of Geography, Kyung Hee University, Seoul, South Korea
emfo0124@khu.ac.kr

Seong-Yun Hong

Department of Geography, Kyung Hee University, Seoul, South Korea
syhong@khu.ac.kr

 <https://orcid.org/0000-0001-5049-8810>

Abstract

The increase of high-resolution spatial data and methodological developments in recent years has enabled a detailed analysis of individuals' experience in space and over time. However, despite the increasing availability of data and technological advances, such individual-level analysis is not always possible in practice because of its computing requirements. To overcome this limitation, there has been a considerable amount of research on the use of high-performance, public cloud computing platforms for spatial analysis and simulation. In this paper, we aim to evaluate the efficiency of spatial analysis in cloud computing platforms. We compared the computing speed for calculating the Moran's I index between a local machine and spot instances on clouds, and our results demonstrated that there could be significant improvements in terms of computing time when the analysis was performed parallel on clouds.

2012 ACM Subject Classification Information systems → Geographic information systems

Keywords and phrases spatial analysis, parallel computing, cloud services

Digital Object Identifier 10.4230/LIPICs.GIScience.2018.24

Category Short Paper

Funding This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP; Ministry of Science, ICT & Future Planning) (No. 2017R1C1B5015090).

1 Introduction

The widespread use of social media and location-based services has produced a large amount of geospatial data [4]. Much of these data are made up of point data, such as OpenStreetMap's PoI and geotagged Twitter posts. Therefore, spatial analysis on point-based data is also widely used for practical and scientific purposes. Point data that involve millions of points are becoming common, and they cause a problem of storage space and memory shortage



© Changlock Choi, Yelin Kim, Youngho Lee, and Seong-Yun Hong;
licensed under Creative Commons License CC-BY

10th International Conference on Geographic Information Science (GIScience 2018).

Editors: Stephan Winter, Amy Griffin, and Monika Sester; Article No. 24; pp. 24:1–24:5

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

■ **Table 1** Test environments for the Moran’s I index.

Environment	Processor	Number of cores	RAM
Local machine	3.4GHz	3	6GB
Spot instance (t2xlarge)	2.3GHz–2.4GHz	4	16GB
Spot instance (m44xlarge)	2.3GHz–2.4GHz	16	64GB

of the computer due to the data scale. There is certainly a need for a new approach for analysing big geospatial data that are difficult to be handled by existing techniques [5].

Cloud computing is one of the alternatives for the analysis of big geospatial data. Many attempts have been made to solve the problem using cloud computing platforms, as it can provide a better analysis environment in terms of cost effectiveness, stability, and computing efficiency. The use of cloud services for spatial analysis is cost effective, because it allows users to lease hardware resources only when they are required. It can be more stable than running own high-performance servers because the cloud computing service providers maintain and manage the facilities. In this short paper, our purpose is to confirm the efficiency of spatial data analysis in cloud computing platforms. To achieve this goal, we compare the time taken for calculating the Moran’s I index on a local machine with those on virtual machines (or spot instances). We use the statistical software R for the experiments, but due to the fact that R utilises only one of the machines’ cores for its computation, the existing functions are adjusted to make the calculation parallel.

2 Background

2.1 Cloud computing with R

Cloud computing is a term that encompasses the hardware and system software in the data center that provide applications and services that are delivered as services over the Internet. These services have long been called Software as a Service (SaaS), and data centers, hardware and software are what we call the cloud. With the advent of cloud computing, developers are free to increase capital expenditures and operational costs, and use unprecedented, low-cost, resilient resources to deliver services [1].

Amazon Elastic Compute Cloud (EC2) is part of Amazon Web Services (AWS) and provides virtual computing environments called *instances*. There are many different types of instances available in EC2, each of which has a different combination of CPU, memory, storage, and networking capacity. Users can choose an instance based on their purpose—general purpose, computing optimisation, memory optimisation, accelerated computing, and storage optimisation. The use of such cloud computing platforms can be more cost effective than constructing a physical computing environment.

There are, however, limitations in using R on cloud services. Most cloud service providers increase the computing performance of spot instances by increasing the number of cores. However, since R can use only one core by default, the increasing number of cores on instances does not affect the computing performance of spatial analysis. To illustrate this point, we selected two spot instances from EC2 and compared the computing time between the instances and between a local machine and them (Table 1).

■ **Table 2** Computing time for single-core and multi-core environments.

Environment	Single-core (in seconds)	Multi-core (in seconds)
Local machine	26342.72	12655.43
Spot instance (t2xlarge)	37503.21	11889.89
Spot instance (m44xlarge)	36976.54	6458.94

2.2 Parallel computing with R

There have been many attempts to solve the problems caused by the size of spatial data in geography using the cloud platforms and parallel computing. Parallel computing is a technique for decomposing and concurrently manipulating data, or concurrently executing process components to complete a task [7]. A common method of parallel computing is to decompose a data set into smaller units, distribute it to multiple operators, and then collect and reconstruct the results after analysis [2].

In this work, we calculate the Moran's I index using Monte Carlo simulations, and each trial runs independently. This can be considered an application of *embarrassingly-parallel*, which means no interactions or communications exist between the operations during the parallel computing process [3]. We have modified the existing Moran's I function in R using the `parallel` package to enable this sort of parallel computing, and use it in each of the described computing environments to compare the efficiency.

3 Methods and results

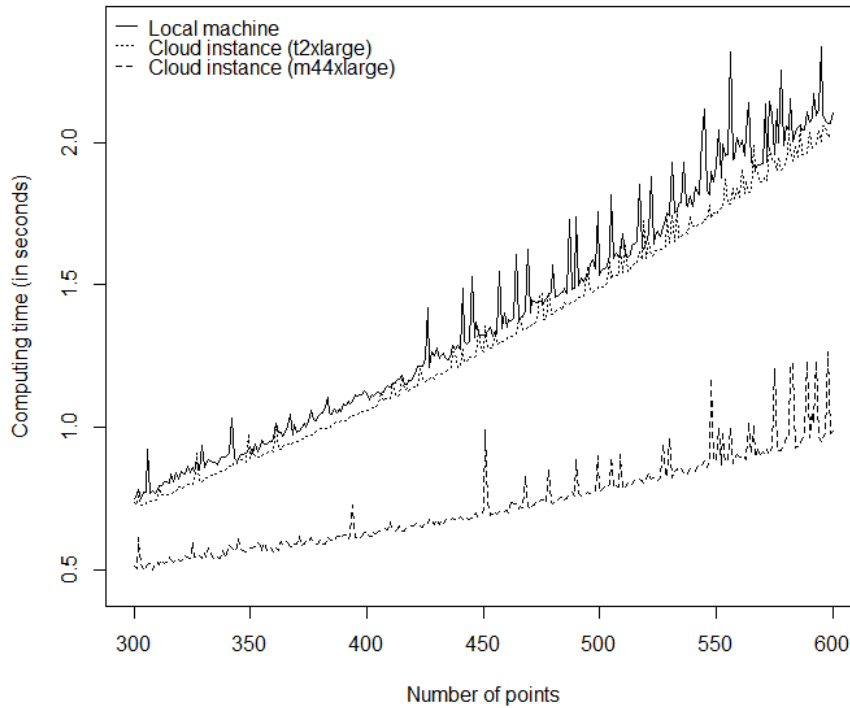
This paper uses Moran's I to compare the computational efficiency of spatial analysis on cloud services. Moran's I is an index for describing spatial autocorrelation of point patterns [6]. Theoretically, the range of Moran's I is from -1 to 1, with positive autocorrelation closer to 1, and negative autocorrelation closer to -1.

The Moran's I index is calculated for hypothetical data that contain 300–500 sets of coordinates and values. The coordinates and the values were generated from a uniform distribution, and the number of repetitions in the Monte Carlo simulation was set to 1,000. Each simulation was repeated 30 times. Table 2 presents the total computing time in each environment, and Figure 1 shows how the average computing time changes with the number of points (i.e., data size).

As shown in Table 2, the local machine took slightly over 26,000 seconds, while both spot instances, **t2xlarge** and **m44xlarge** took about 37,000 seconds. In addition, **m44xlarge** shows about four times more computational efficiency than **t2xlarge** in terms of catalog performance. These results seem to be derived from the performance enhancements of single-core and cloud computing services—a feature of R mentioned above.

On the other hand, in the case of parallel computing, it was confirmed that the computing time using parallel computing is less than that of the local machine. Also, as the size of data increases, the gap tends to increase more and more. However, when comparing **t2xlarge** and **m44xlarge**, there is less difference compared to actual performance difference. This is probably a problem of the parallel computing process. Parallel computing, when compared to a single-core computing, requires at least two additional processes, distribution of data and aggregation of results, and this might cause the difference in time.

Table 3 shows the minimum, mean, and maximum values for each environment, and it indicates a similar conclusion to that from Figure 1. When comparing the mean values, the



■ **Figure 1** Computing time by the number of points.

`t2xlarge` instance does not show a significant difference in the computing time with the local machine, but the `m44xlarge` instance has shortened the time from 1.5 to 2 times for the same number of points. However, when comparing the maximum values, the time taken for analysis fluctuates, possibly due to the instability of the system. When the calculation is repeatedly performed, the differences between the mean and the maximum values become clearly apparent.

4 Conclusions

As we have demonstrated, the use of single-core programs for big data analysis is limited, because it takes a considerable amount of time to operate or does not properly reflect the evolving computing environment. In particular, spatial analysis using spatial data requires a new approach, because the number of data increases and the computing resources required for analysis increase exponentially. Therefore, the need for high-performance computing technology that is capable of rapidly computing and processing large-scale data has begun to be emphasised.

This paper attempts to verify cloud computing as an alternative method to solve the above problems from the empirical point of view. Cloud computing platforms provide a better analysis environment in three ways. First, it is more economical to lease the hardware of the desired performance at the user's desired time through cloud computing than to build the high-performance resource at the initial cost. In general, users are tempted to perform high-performance analysis because their computing resources are time-sensitive and their replacement cycle is short. Second, cloud services meet the stability of analytics in the sense that the service providers take the responsibility for maintaining and servicing data. Finally,

■ **Table 3** Computing time by the number of points.

Environment		Number of points			
		300	400	500	600
Local machine	Max	0.94819	0.13022	0.16920	2.27118
	Mean	0.74900	1.11907	1.53280	2.10250
	Min	0.65706	0.92256	1.37751	1.86864
Spot instance (t2xlarge)	Max	0.73618	1.08571	1.51585	2.04415
	Mean	0.72592	1.05769	1.49446	2.02816
	Min	0.70898	1.04644	1.47618	2.01196
Spot instance (m44xlarge)	Max	0.61105	0.74626	0.92479	1.36301
	Mean	0.50443	0.63304	0.77270	0.98782
	Min	0.48263	0.60369	0.73902	0.94593

multi-core analysis on cloud computing platforms ensures the efficiency of analysis. In this paper, we demonstrated that the time for calculating Moran's I can be significantly improved (i.e., reduced) when parallel computing is used on cloud services.

In this study, a parallel processing structure of SIMD (Single Instruction Stream) method is used for the calculation. This means that the same operation is simultaneously performed on the data set assigned to each operator. When using multiple instruction streams (MIMD), different operations can be performed simultaneously on an allocated data set, resulting in more efficiency in parallel computing. In the future, it will be possible to verify the most effective approach to spatial analysis when various parallel processing structures are used.

References

- 1 Michael Armbrust, Armando Fox, Rean Griffith, Anthony D Joseph, Randy H Katz, Andrew Konwinski, Gunho Lee, David A Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. Above the clouds: A Berkeley view of cloud computing. Technical report, Electrical Engineering and Computer Sciences, University of California, Berkeley, 2009.
- 2 Yuemin Ding and Paul J Densham. Spatial strategies for parallel spatial modelling. *International Journal of Geographical Information Systems*, 10(6):669–698, 1996.
- 3 Ian Foster. *Designing and building parallel programs*, volume 78. Addison Wesley Publishing Company Boston, 1995.
- 4 Michael F Goodchild. Citizens as sensors: the world of volunteered geography. *GeoJournal*, 69(4):211–221, 2007.
- 5 Rob Kitchin. Big data and human geography: Opportunities, challenges and risks. *Dialogues in Human Geography*, 3(3):262–267, 2013.
- 6 Patrick A. P. Moran. Notes on continuous stochastic phenomena. *Biometrika*, 37(1/2):17–23, 1950.
- 7 Michael J Quinn. *Designing efficient algorithms for parallel computers*. McGraw-Hill, 1987.