Journal of the Korean Association of Geographic Information Studies, 21(4) 2018, pp. 119-131 ISSN 1226-9719(Print)
https://doi.org/10.11108/kagis.2018.21.4.119 ISSN 2287-6952(Online)

클라우드 컴퓨팅 기반 공간분석의 연산 효율성 분석*

최창락[1]·김예린[1]·홍성연[1※]

Evaluating Computational Efficiency of Spatial Analysis in Cloud Computing Platforms*

Changlock CHOI[1]·Yelin KIM[1]·Seong-Yun HONG[1※]

요 약

휴대용 기기와 다양한 위치 기반 서비스의 확산으로 공간데이터의 양적 팽창이 가속화됨에 따라 대용량의 공간데이터를 효율적으로 다룰 수 있는 기술의 중요성이 점차 커지고 있다. 클라우드 컴퓨팅은 인터넷을 통해 스토리지, 메모리, 애플리케이션 등 다양한 전산 자원을 공유할 수 있는 서비스 환경으로, 최근 이를 활용해 대용량의 공간데이터를 처리, 분석하는 방법과 그 필요성에 관한 연구가 활발히 수행되어 왔다. 그러나 아직까지 대용량 공간데이터의 분석에 클라우드 컴퓨팅 플랫폼을 활용했을 때 어느 정도의 성능 향상을 기대할 수 있는지에 대한 실증적 연구는 비교적 많이 이루어지지 않았으며, 본 연구의 목표는 이러한 논의의 공백을 채우는 것이다. 이를 위해 연구에서는 클라우드 컴퓨팅 플랫폼에서 병렬 연산을 사용했을 때 모란지수와 지리가중회귀분석의 연산 속도가 어느 정도 향상되는지 살펴보았으며, 그 결과를 통해 클라우드 컴퓨팅을 활용한 공간분석의 효율성을 평가하였다. 실험 결과, 중앙처리장치의 클록 수가 더 높은 로컬 컴퓨터에 비해 병렬 연산에 적합한 환경을 갖춘 공용 클라우드 컴퓨팅 플랫폼에서 좀 더 효율적인 연산이 가능했으며, 데이터의 규모가 클수록 격차가 더욱 크게 나타났다.

주요어 : 클라우드 컴퓨팅, 병렬 연산, 공간분석, 공간데이터, 효율성, 효과성

ABSTRACT

The increase of high-resolution spatial data and methodological developments in recent years has enabled a detailed analysis of individual experiences 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

2018년 10월 25일 접수 Received on October 25, 2018 / 2018년 12월 13일 수정 Revised on December 13, 2018 / 2018년 12월 13일 심사완료 Accepted on December 13, 2018
* 이 논문은 2018년 8월 30일 GIScience 2018 학술대회에서 발표한 내용을 확장, 발전시킨 것으로 한국연구재단의 지원을 받아 수행된 연구임(NRF-2017R1C1B5015090).
1 경희대학교 이과대학 지리학과 Dept. of Geography, Kyung Hee University
※ Corresponding Author E-mail : syhong@khu.ac.kr

amount of research on the use of high-performance, public cloud computing platforms for spatial analysis and simulation. The purpose of this paper is to empirically evaluate the efficiency and effectiveness of spatial analysis in cloud computing platforms. We compare the computing speed for calculating the measure of spatial autocorrelation and performing geographically weighted regression analysis between a local machine and spot instances on clouds. The results indicate that there could be significant improvements in terms of computing time when the analysis is performed parallel on clouds.

KEYWORDS : cloud computing, parallel computing, spatial analysis, spatial data, efficiency, effectiveness

서 론

휴대용 기기와 다양한 위치 기반 서비스의 확산은 개개인이 정보의 제공자가 되어 데이터를 생산하는 것을 가능케 했으며, 이는 공간데이터의 급속한 양적, 질적 팽창을 가져왔다(Goodchild, 2007). 한국정보화진흥원에 의하면 2011년 한 해 동안 약 1조 기가바이트 정도의 데이터가 생성되었으나, 2020년에는 이 수치가 약 50배 이상 급증할 것으로 추산된다(Cho, 2013). 특히 급증하는 공간데이터는 교통, 치안, 범죄, 보건, 의료, 부동산 등 국민 생활과 밀접한 분야에서 널리 활용될 수 있을 것으로 전망되며, 따라서 대용량의 공간데이터를 효과적으로 처리하고 분석할 수 있는 기술과 서비스의 중요성이 점차 커지고 있다(Park *et al.*, 2015).

공간분석에 사용되는 데이터 규모에 비례한 연산량의 증가는 컴퓨팅 측면에서 여러 가지 어려움을 수반하게 된다. 특히 국지적 모란지수(Local Moran's I)나 지리가중회귀분석(Geographically Weighted Regression; GWR)과 같은 국지적 통계기법은 각 공간 단위에서 반복적으로 연산을 수행해야 하므로 높은 수준의 알고리즘 최적화와 많은 전산 자원을 필요로 한다. 그러나 일반적으로 사용되는 많은 GIS 프로그램에서는 한 번에 처리 가능한 데이터의 규모가 일정 이하로 제한되는 경우가 많고(Zhang, 2016), 따라서 대용량 공간데이터의 효율적인 분석과 활용을 위해서는 새로운 접근법을 모색하는 것이 필요하다(Kitchin, 2013).

본 연구에서는 최근 빅데이터 분석을 위한 합리적인 대안으로 활발하게 논의되어 온 클라우드 컴퓨팅 플랫폼에서 실제 공간분석을 수행할 때 기대할 수 있는 효율성과 효과성을 실험을 통해 확인하고자 한다. 클라우드 컴퓨팅이란 사용자가 필요로 하는 소프트웨어와 스토리지, 메모리 등의 전산 자원을 필요한 만큼 사용하고, 이에 따라 비용을 지급하는 서비스 모델을 의미한다(Kim *et al.*, 2012). 클라우드 컴퓨팅을 활용한다면 기존의 슈퍼컴퓨터 기반 분석의 단점으로 꼽히는 막대한 시설 구축 비용과 유휴 자원에 의한 낭비를 최소화할 수 있다. 즉, 빅데이터 분석에 요구되는 방대한 연산을 위해 고성능의 하드웨어를 직접 구축하지 않고 필요할 때에만 자원을 임대해서 사용함으로써 관련 비용을 절감할 수 있는 것이다. 또한, 전산 자원의 유지와 보수, 관리를 서비스 제공자에게 맡기고, 사용자는 안정된 환경에서 분석에만 집중할 수 있다는 장점이 있다(Choi and Noh, 2011). 이에 따라 본 연구에서는 클라우드 컴퓨팅 플랫폼에서 크기가 다른 여러 개의 공간데이터를 분석하고, 그 결과를 바탕으로 공간데이터의 크기에 따라 클라우드 컴퓨팅의 효율성이 어떻게 변화하는지 살펴볼 것이다.

이를 위해 아마존 웹 서비스(Amazon Web Services; AWS)에서 제공하는 가상 서버에 통

계 프로그램 R을 설치하여 대표적인 공간통계 기법인 모란지수와 지리가중회귀분석을 수행하고, 연산에 걸린 시간을 일반적인 로컬 컴퓨터에서의 결과와 비교할 것이다. 분석에 사용한 데이터는 *n*개의 점으로 구성된 가상의 포인트 데이터이며, 점의 수를 단계적으로 증가시켜 데이터 크기와 연산 속도 간의 관계를 알아본다. 분석에는 CPU의 코어 수와 메모리 용량이 다르게 구성된 두 개의 가상 서버(인스턴스 유형)를 활용하며, 각 서버에서 도출된 연산 결과를 비교함으로써 공간분석에 상대적으로 영향을 크게 미치는 전산 자원을 추가적으로 확인할 수 있을 것이다.

논문의 구성은 다음과 같다. 우선 다음 장에서는 클라우드 컴퓨팅의 기본적인 개념과 클라우드 컴퓨팅 환경에서의 공간분석 사례를 정리하고, 관련된 선행연구를 살펴볼 것이다. 이어서 본 연구의 실험 방법과 연산 속도의 비교 분석에 사용된 공간통계 기법에 관해 설명한다. 다음으로는 실험 결과를 통해 클라우드 컴퓨팅 플랫폼에서 공간데이터 분석의 효율성을 확인하고, 경제성과 안정성 측면에서 클라우드 컴퓨팅이 갖는 장점에 대해 논할 것이다. 결론에서는 실험을 통해 도출된 결과를 다시 정리하고, 본 연구의 한계점과 향후 연구 방향에 대해 논의한다.

클라우드 컴퓨팅 기반의 공간분석

클라우드 컴퓨팅 플랫폼에서 대용량의 공간데이터를 처리, 분석하는 방법과 그 필요성에 관한 연구는 지난 10여 년간 꾸준히 수행되어 왔다(Wang *et al.*, 2009; Xiaoqiang and Yuejin, 2010; Yang *et al.*, 2011; Yang *et al.*, 2013; Yue *et al.*, 2013; Zhou *et al.*, 2015; Tang and Feng, 2017; Yang *et al.*, 2017). 최근 자료 수집 기술의 발전과 다양화로 공간데이터의 양적 팽창이 가속화되고, 이러한 데이터를 처리하기 위한 분석 알고리즘과 모형도 복잡해지면서 연산에 필요한 시간이 큰 폭으로 증가하게 되었다. 특히 지리가중회귀분석과 같은 국지적 통계기법을 넘어, 특정 공간에서 개인의 경험을 계량화하는 방법론이 활발히 개발되기 시작하면서 원활한 분석 수행을 위해 대규모의 전산 자원을 구축하고 운용할 필요성이 커지고 있다. 그러나 분석을 위한 전산 자원의 확대에는 상당한 비용이 발생하게 되며, 대규모 분석 작업이 상시로 이루어지지 않는다면 이는 유휴 자원으로 인한 손실로 이어지게 된다(Yang *et al.*, 2017).

이러한 문제에 대한 대안으로 주목받는 클라우드 컴퓨팅은 인터넷을 통해 스토리지, 메모리, 애플리케이션 등 다양한 전산 자원을 공유할 수 있는 서비스 환경으로, 제공하는 전산 자원의 범위와 공유 방식에 따라 표 1과 같이 구분될 수 있다. 대용량의 공간데이터 분석에 이 같은 클라우드 컴퓨팅 플랫폼을 활용한다면 데이터 규모에 가장 적절한 전산 자원을 탄력적으로 임대하여 효율적인 작업 수행이 가능하고, 같은 성능의 컴퓨팅 환경을 직접 구축하는 것과 비교

TABLE 1. Comparisons of cloud computing platforms

Cloud computing platforms		Characteristics	Examples
Purpose	Infrastructure-as-a-Service	Provides the infrastructure components, such as servers and storage	Amazon Web Services EC2, S3
	Platform-as-a-Service	Provides a platform where users can develop, run, and manage applications	Google App Engine
	Software-as-a-Service	Provides software on a subscription basis	Oracle CRM, Google Docs
Types	Public clouds	Makes computing resources available to the public over the Internet	Amazon Web Services
	Private clouds	Maintains the services and infrastructure on a private network	–
	Hybrid clouds	Combines a public and private cloud options	–

하여 운용 및 유지 보수에 드는 비용은 절감할 수 있다(Armbrust *et al.*, 2009). 또한, 여러 명의 사용자가 데이터를 공유하고 공동작업을 수행하는 것이 용이하기 때문에, 처리해야 하는 데이터와 분석 규모에 따른 작업 인원 증가에도 효과적으로 대응할 수 있다.

Lee *et al.*(2011)는 특히 대규모의 병렬 연산(parallel computing)이 필요할 때 클라우드 컴퓨팅의 작업 환경이 유용하다고 밝혔다. 병렬 연산은 데이터 또는 연산과정에서 반복적으로 계산이 필요한 부분을 분리하고 이를 여러 개의 전산 자원이 동시에 나누어 수행하는 기술을 말하는데(Quinn, 1987), 독립적인 연산이 가능한 자원이 많을수록 효율이 높아진다. 일반적으로 공용 클라우드 컴퓨팅 플랫폼에서는 매우 많은 수의 가상 서버를 동시에 운용할 수 있으므로 대용량 데이터를 분산시켜 병렬로 연산하는 것이 상대적으로 수월하며, 따라서 최근의 많은 연구에서는 클라우드 컴퓨팅과 병렬 연산을 접목하여 대용량 공간데이터 분석 과정에서 발생하는 특정한 문제점을 해결하고자 시도하였다(Healey *et al.*, 1997).

예를 들어, 도시의 토지이용 변화를 시뮬레이션 하는 SLEUTH 모형의 경우 도시의 공간적 범위가 확장되고 데이터 규모가 증가하면서 예측에 필요한 시간이 크게 상승하였으나, 시뮬레이션 과정에 병렬 연산을 적용함으로써 처리 속도는 물론 결과의 정확성도 높일 수 있었다(Guan and Clarke, 2010). 대표적인 내삽(interpolation) 기법인 역거리가중법의 경우에도 소규모 클러스터에서의 병렬 연산만으로 상당한 성능 향상 효과를 볼 수 있었고(Fang *et al.*, 2011), 가시권역(viewshed) 분석에서도 기존의 순차적 알고리즘을 사용해 다루기 어려웠던 대용량의 공간데이터를 GPU의 고대역폭 메모리를 활용한 병렬 연산으로 처리한 사례가 있다(Zhao *et al.*, 2013).

최근에는 클라우드 컴퓨팅과 병렬 연산을 결합하여 대용량 벡터 데이터의 투영 변환 속도를 개선하는 방안이 제시되었다(Tang and Feng, 2017). 공간통계 분야에서도 고전적인 크리깅(Kriging) 알고리즘을 확장하여 시공간 크리깅 알고리즘을 개발하고 이를 병렬 처리함으로써 결과의 적합성과 연산 속도의 향상을 볼 수 있었으며(Zhang *et al.*, 2018), 지리가중회귀분석의 연산에도 기존 알고리즘 구조에서 메모리 사용을 최적화하고 병렬 알고리즘을 도입하여 상당한 연산 속도의 향상을 불러일으켰다(Li *et al.*, 2018). 원격탐사 분야에서도 초분광(hyperspectral) 영상의 병렬형 K-means 알고리즘을 클라우드 컴퓨팅에 접목한 사례가 있고(Haut *et al.*, 2017), 공간 연산 과정에서 발생할 수 있는 희소 행렬의 연산도 불필요한 연산 제거와 병렬화를 통해 효율성을 높일 수 있음을 확인했다(Azad and Buluc, 2017). 이처럼 새로운 기술에 기반을 둔 분석 프레임워크의 적용 분야는 점차 다양해지고 있으며 그 활용 가치 또한 커지고 있다.

병렬 처리를 통한 연산 속도의 향상은 공간분석 과정을 정량적으로 가속할 뿐만 아니라, 분석을 통해 해결할 수 있는 문제의 범위를 확대한다는 점에서 의미가 있다(Turton and Openshaw, 1998). 그러나 아직까지 대용량 공간데이터의 분석에 클라우드 컴퓨팅 플랫폼을 활용했을 때 어느 정도의 성능 향상을 기대할 수 있는지에 대한 실증적 연구는 활발히 이루어지지 않고 있으며, 따라서 본 논문에서는 이를 중점적으로 다룸으로써 논의의 공백을 채우고자 한다. 본 연구에서는 클라우드 컴퓨팅 플랫폼에서 병렬 연산을 적용했을 때 대표적인 공간분석 기법인 모란지수와 지리가중회귀분석의 연산 속도가 어느 정도 향상되는지 살펴보고 그 결과를 바탕으로 클라우드 컴퓨팅을 활용한 공간분석의 효율성을 평가하고자 한다. 병렬 연산은 통계 프로그램 R에 parallel, foreach와 같은 추가 패키지를 설치하여 수행하며, 계산 과정에서 프로세서 간 상호작용 및 영향이 없는 간단한 형태의 처치 곤란 병렬(embarrassingly parallel) 알고리즘이 사용될 것이다.

실험 설계

1. 클라우드 컴퓨팅 플랫폼

TABLE 2. Computing environment for experiments

Platform		Central Processing Unit (CPU)			Memory (GB)
		Model	Freq. (GHz)	# Cores	
Local machine		Intel Core™ i5 7th generation processor	3.4	3	6
Cloud instance	Type I	Intel® Xeon® Scalable processor	2.3 – 2.4	4	17.18
	Type II	Intel® Xeon® E5-2686 v4 (Broadwell) processor	2.3 – 2.4	16	68.72

본 연구는 공용 클라우드 서비스 중 가장 사용자 수가 많은 AWS의 아마존 일래스틱 컴퓨트 클라우드(Amazon Elastic Compute Cloud; Amazon EC2)를 기반으로 분석을 수행한다. EC2는 AWS의 핵심적인 요소로, 물리적인 전산 자원을 인스턴스라 불리는 여러 대의 가상 서버로 나누어 제공한다. EC2 내에는 CPU, 메모리, 스토리지 및 네트워킹 용량의 조합에 따른 다양한 유형의 인스턴스가 존재하며, 사용자는 데이터의 크기나 분석 알고리즘에 적합한 유형을 선택하고 사용시간에 따라 요금을 지급할 수 있다. 또한 사용자의 필요에 따라 자유로운 유형의 변경이 가능하기 때문에 탄력적이며 즉각적인 연산 환경의 재구성이 가능하며, 이는 비용을 절감하는 효과를 가져온다. 따라서 최근 고성능 컴퓨팅 서버의 직접 구축에 대한 대안으로서 EC2가 언급되고 있다.

인스턴스 유형을 선택하기에 앞서 중요한 것 중 하나는, 클라우드에서 수행하고자 하는 작업과 사용하는 소프트웨어의 특징을 파악하는 것이다. EC2와 같은 대부분의 공용 클라우드 서비스에서는 중앙처리장치(CPU)의 클록 수를 높이는 대신, 같은 성능의 코어 개수를 늘리는 방식으로 인스턴스의 성능을 향상한다. 따라서 수행하고자 하는 작업이 멀티 코어 환경에 적합하지 않거나 소프트웨어가 이를 지원하지 못한다면, 코어 수가 많은 상위 유형의 인스턴스를 사용한다 해도 성능 차이를 체감할 수 없을 것이다. 본 연구에서 사용하는 통계 프로그램 R 또한 단일 코어 연산에 최적화된 소프트웨어이기 때문에, 실험에 앞서 우선 멀티 코어를 활용한 병렬 연산이 가능하도록 함수를 일부 수정하는 과정을 거쳤다.

클라우드 컴퓨팅 플랫폼과 로컬 컴퓨터에서의 분석 성능을 비교하기 위해 본 연구에서는 두 개의 EC2 인스턴스를 활용하였다. 표 2는 실험에 사용된 인스턴스와 로컬 컴퓨터의 기본적인 전산 자원을 나타내고 있다. 로컬 컴퓨터는 일반적인 연구 환경을 대표하기 위해 가격 대비 효율이 높은 것으로 알려진 대중적인 부품들로 구성하였으며, 가상 서버 인스턴스도 개인이 충분히 감당 가능한 중저가의 서비스를 선택해 비교 분석을 수행하였다. 이를 통해 클라우드 컴퓨팅 환경에서 공간분석을 수행할 때 체감할 수 있는 현실적인 성능 향상 효과를 파악하는 것이 본 연구의 일차적인 목표가 된다.

2. 비교 분석 방법

서론에서 언급한 바와 같이 본 연구에서는 대표적인 공간통계 기법인 모란지수와 지리가중회귀분석의 몬테카를로 시뮬레이션 시행을 통해 공용 클라우드 컴퓨팅 플랫폼과 로컬 컴퓨터 간의 연산 속도를 비교할 것이다.

모란지수는 공간데이터에서 전반적으로 나타나는 공간적 자기상관성(spatial autocorrelation)을 측정하기 위해 사용하는 대표적인 지표로 기존의 상관관계 측정을 공간적 맥락에 투영한 기법이다. 모란지수는 기본적으로 속성값의 유사성 또는 상이성이 객체 간 물리적인 거리와 얼마나 밀접한 관련이 있는지를 평가하며, 수식으로는 아래와 같이 정의된다(Moran, 1950).

$$I=\left[\frac{n}{\sum_{i=1}^{n}(y_i-\bar{y})^2}\right]\times\left[\frac{\sum_{i=1}\sum_{j=1}w_{ij}(y_i-\bar{y})(y_j-\bar{y})}{\sum_{i=1}\sum_{j=1}w_{ij}}\right] \quad (1)$$

여기서, y_i와 y_j는 i 번째 지역과 j 번째 지역의 속성값을 의미하며, n은 지역의 총 개수를 나타

낸다. w_{ij}는 두 지역 간의 공간적 관계를 나타내는 정보로 본 논문에서는 점 간의 기하학적 거리가 사용될 것이다. 모란지수의 이론적인 범위는 -1에서 1 사이이며, 일반적으로 0.4보다 큰 값을 나타낼 때 양의 상관관계, 그리고 −0.4보다 작은 값을 가질 때 음의 상관관계를 가진다고 말한다.

지리가중회귀분석은 1970년대 말부터 Cleveland (1979)와 Cleveland and Devlin(1988)이 제안한 국지적가중회귀분석(Locally Weighted Regression)을 발전시킨 것으로, 전체 연구지역을 여러 개의 단위 지역으로 세분화하고, 각 단위 지역에서 개별적인 회귀모형을 도출하는 방법이다. 지리가중회귀분석은 지역에 따라 독립변수가 갖는 영향력이 다를 수 있음을 전제로 하고, 일반적인 다중선형회귀모형에 좌표를 더함으로써 위치에 따른 회귀계수의 변화를 파악한다.

$$y_i = \sum_k \beta_k(u_i, v_i)x_{ik} + \epsilon_i \qquad (2)$$

여기서, y_i는 i 지점의 종속변수 값을, x_{ik}는 같은 지점에서 k번째 독립변수 값을 나타내며, (u_i, v_i)는 i 지점의 좌표를 의미한다. β_k는 i 지점의 k번째 독립변수에 대한 회귀계수이며 ϵ_i는 해당 지점의 오차항을 나타낸다.

모란지수와 지리가중회귀분석은 spdep 패키지의 moran.mc () 함수와 GWmodel 패키지의 gwr.montecarlo () 함수를 각각 사용하여 수행하였으나, 병렬 연산을 위해 함수를 일부 수정하였다. 함수의 수정은 각 분석에 대한 몬테카를로 시뮬레이션 과정의 병렬화에 중점을 두고 진행되었다. 그림 1은 실험에서 몬테카를로 시뮬레이션 상의 n번의 반복 횟수를 m개의 CPU 코어로 병렬 연산하는 알고리즘을 설명한다. 이는 1번부터 m번째의 시뮬레이션을 각 CPU 코어에 할당한 다음, m+1번째의 시뮬레이션을 다시 첫 번째 CPU 코어에 할당하는 방식으로 이루어진다. 각각의 코어에 순차적으로 시뮬레이션을 할당하기 때문에 CPU 코어의 개수가 증가할수록 더욱 효율적인 연산이 가능해진다.

효율성 및 효과성 평가

1. 연산 속도의 비교

비교 분석에 사용된 데이터는 300개에서 600개의 점으로 구성된 포인트 데이터이다. 점의 위치와 속성값은 모두 0에서 10,000 사이의 값을 갖는 균일 난수로 설정하였으며, 이렇게 생성된 가상의 데이터에 모란지수와 지리가중회귀분석을 각각 30번씩 수행하여 평균 소요시간을 산출하였다. 모란지수와 각 회귀계수의 통계

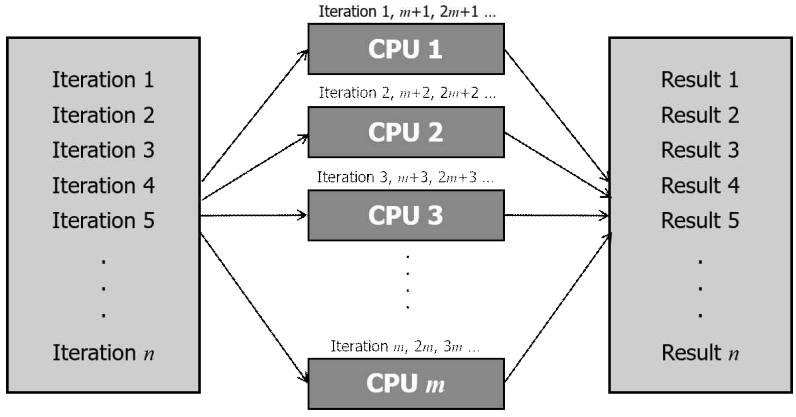

FIGURE 1. An illustration of a parallel computing algorithm implemented in the experiments

TABLE 3. Time spent (in seconds) for completing analysis on different computing platforms

Platform			Moran's I				Geographically Weighted Regression			
			Number of points (n)				Number of points (n)			
			300	400	500	600	300	400	500	600
Local machine	Single-core	Mean	1.2911	2.2372	3.4685	4.9163	1.0535	1.7141	2.4795	3.3297
		S. dev.	0.0057	0.0157	0.0383	0.0185	0.0234	0.0383	0.0443	0.1022
	Multi-core	Mean	0.7490	1.1191	1.5328	2.1025	0.7498	1.0209	1.7169	2.8523
		S. dev.	0.0683	0.0912	0.0906	0.1290	0.1828	0.1613	0.2012	0.1615
Cloud instance (Type I)	Single-core	Mean	1.8735	3.1819	4.9592	7.0206	1.3216	2.1054	2.8651	3.9205
		S. dev.	0.0474	0.0056	0.0821	0.0564	0.0557	0.0243	0.0311	0.0300
	Multi-core	Mean	0.7360	1.0577	1.4945	2.0282	0.4903	0.6797	1.0156	1.4070
		S. dev.	0.0673	0.0072	0.0100	0.0080	0.0496	0.0506	0.1116	0.1041
Cloud instance (Type II)	Single-core	Mean	1.8358	3.1447	4.8378	6.8968	1.3110	1.9252	2.7666	3.6664
		S. dev.	0.0543	0.0232	0.0315	0.0016	0.1383	0.0785	0.0202	0.0664
	Multi-core	Mean	0.5118	0.6330	0.7727	0.9878	0.3441	0.4397	0.5733	0.7273
		S. dev.	0.0863	0.0266	0.0317	0.0995	0.0137	0.0429	0.0862	0.1211

적 유의성 검증을 위한 몬테카를로 시뮬레이션은 로컬 컴퓨터와 가상 서버 인스턴스 모두 동일하게 999번 시행하였다.

모란지수와 지리가중회귀분석의 연산 속도에 관한 실험 결과는 다음의 표 3과 같다. 300개의 점으로 구성된 공간데이터의 경우, 단일 코어만을 활용했을 때에는 중앙처리장치의 클록 수가 높은 로컬 컴퓨터에서 가장 소요시간이 짧았으나 병렬 연산이 가능하도록 함수를 수정한 이후에는 코어 수에 비례해 연산 속도가 증가함을 알 수 있었다. 모란지수의 경우 코어가 세 개인 로컬 컴퓨터에서는 약 0.7490초가 걸렸으나, 코어가 네 개인 유형 I 인스턴스에서는 약 0.7360초, 그리고 코어가 16개인 유형 II 인스턴스에서는 약 0.5118초밖에 소요되지 않았다. 로컬 컴퓨터와 유형 I 인스턴스는 클록 수가 다르기 때문에 직접적인 비교가 어렵지만, 클록 수가 같은 유형 I 인스턴스와 유형 II 인스턴스를 비교하면 코어의 수가 4배 증가함에 따라 연산 소요시간이 3분의 2 정도로 감소하는 것을 확인할 수 있다.

코어 수에 따른 연산 소요시간의 감소 효과는 데이터의 크기에 비례해 증가했다. 점의 개수가 400개일 때는 유형 II 인스턴스에서의 모란지수 연산 시간이 유형 I 인스턴스 대비 약 59% 정도이나, 점의 개수가 500개, 600개로 증가함에 따라 연산 소요시간은 각각 52%, 49%로 감소했다. 실제 데이터 크기와 연산 소요시간 간의

TABLE 4. Ordinary least squares (OLS) regression results

Platform		Moran's I				Geographically Weigted Regression			
		Estimate	Std. Error	t-value	Pr (>|t|)	Estimate	Std. Error	t-value	Pr (>|t|)
Local machine	Intercept	−0.7493	0.0234	−32.03	≤0.0001	−1.8809	0.0504	−37.29	≤0.0001
	Slope	0.0048	0.0001	93.15	≤0.0001	0.0075	0.0001	67.91	≤0.0001
	Adj. R2	0.9667				0.9391			
Cloud instance (Type I)	Intercept	−0.6767	0.0138	−48.96	≤0.0001	−0.5128	0.0150	−34.19	≤0.0001
	Slope	0.0044	≤0.0001	146.89	≤0.0001	0.0031	≤0.0001	95.97	≤0.0001
	Adj. R2	0.9863				0.9686			
Cloud instance (Type II)	Intercept	−0.0341	0.0168	−2.025	0.0438	−0.0895	0.0079	−11.29	≤0.0001
	Slope	0.0017	≤0.0001	46.047	≤0.0001	0.0014	≤0.0001	80.31	≤0.0001
	Adj. R2	0.8764				0.9557			

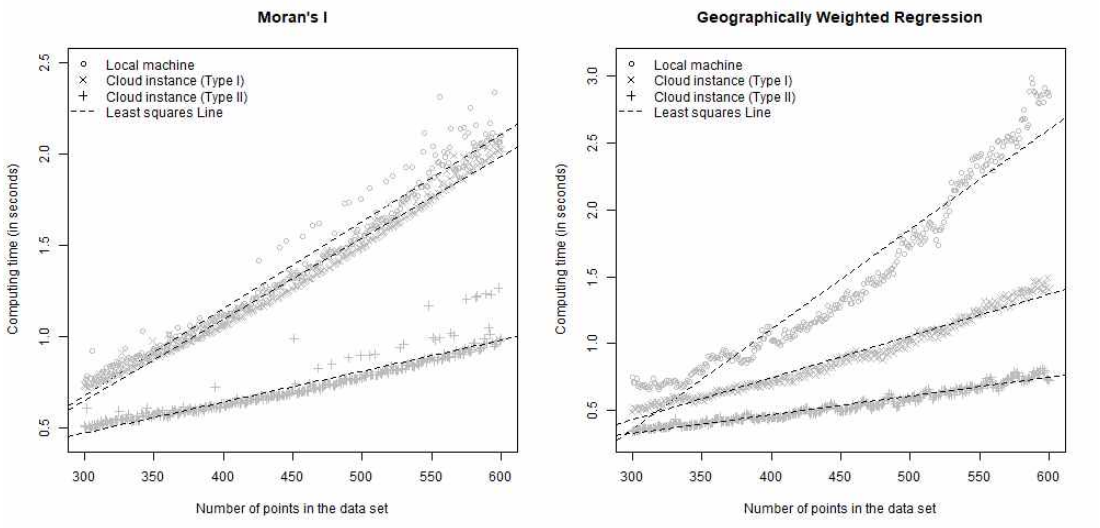

FIGURE 2. Linear relationships between the number of points and the computing time

관계를 선형회귀분석을 통해 살펴보면, 유형 II 인스턴스의 회귀계수는 0.0017로 유형 I 인스턴스의 3분의 1 정도에 불과하며(표 4), 그림 2를 통해서도 유형 II 인스턴스에서 데이터 크기에 따른 연산 소요시간의 증가 추세가 훨씬 완만함을 알 수 있다.

이와 같은 패턴은 지리가중회귀분석의 수행결과에서도 전반적으로 유사하게 나타났다. 표 3을 통해 알 수 있듯, 병렬 연산을 수행했을 때 코어 수가 미치는 영향은 클록 성능으로 인한 차이보다 크게 나타났고 코어가 증가함에 따라 분석에 걸리는 시간은 감소하였다. 그러나 모란지수와 비교했을 때 인스턴스 유형 간 기울기 차이는 상대적으로 적은 반면(표 4), 로컬 컴퓨터와 클라우드 컴퓨팅 플랫폼에서의 연산 소요시간은 크게 차이가 났는데, 이것은 지리가중회귀분석을 수행하기 위해 사용한 함수가 클록 성능보다 코어 수에 더 크게 영향을 받고 있음을 시사한다.

분석 결과를 종합했을 때, 모란지수와 지리가중회귀분석 같은 공간통계 기법은 병렬 연산을 사용해 효율적인 수행이 가능하며 클록 자체의 성능보다 코어의 수가 연산 속도에 크게 영향을 미침을 알 수 있다. 코어 수에 따른 연산 소요시간의 차이가 데이터 크기와 비례하는 추세를 보였기 때문에(그림 2), 코어의 수가 많은 고사양의 전산 자원에 기반한 병렬 연산은 최근 급증하는 대용량의 공간데이터를 분석하는데 특히 효과적일 수 있을 것이다. 그러나 후술하는 바와 같이 일정 개수 이상의 코어를 갖춘 전산 환경을 구성하는 것은 많은 비용을 요구하며, 따라서 데이터 규모와 분석 빈도를 고려할 때 상시 분석이 필요한 경우가 아니라면 공용 클라우드 컴퓨팅 플랫폼의 활용이 경제적이고 접근성 높은 대안일 수 있다.

2. 경제성 및 안정성에서의 효율

앞서 실험에서는 로컬 컴퓨터와 클라우드 컴퓨팅 플랫폼에서의 연산 시간을 비교하고, 그 결과를 바탕으로 데이터 크기가 클수록 코어의 수가 많은 가상 서버 인스턴스에서 분석을 수행하는 것이 효율적임을 밝혔다. 물론 로컬 컴퓨터도 실험에서의 유형 II 인스턴스와 동일한 하드웨어로 구성한다면, 연산 소요시간 역시 동일하게, 혹은 그 이상으로, 단축될 수 있을 것이다. 하지만 높은 성능의 전산 환경을 별도로 구

TABLE 5. Estimated costs for purchasing computing components equivalent to cloud instances (Unit: Korean Won)

Platform	Estimated costs for purchase				Cloud pricing per hour (B)	Available Time1)
	Components			Total (A)		
	CPU	Memory	M/B			
Cloud instance (Type I)	122,900	161,300	88,040	372,240	63.25	5,885.5
Cloud instance (Type II)	645,000	766,000	332,900	1,743,900	161.35	10,807.9

1) The amount of hours for which the equivalent cloud instance can be used instead of purchasing the computing components (i.e., (A)/(B))

축하는 데에는 상당한 경제적 비용이 소요되며, 향후 더 높은 성능을 필요로 하는 분석이 발생한다면 이는 추가적인 지출로 이어지게 된다. 반면 일반적으로 클라우드 컴퓨팅 플랫폼에서는 작업 규모에 따라 하드웨어의 성능과 각 구성 요소의 사용 여부 등을 탄력적으로 조정할 수 있고, 인스턴스를 사용한 시간에 대해서만 과금되기 때문에 많은 경우에는 비용을 낮출 수 있다(Garfinkel, 2007). 표 5는 실험에 사용된 인스턴스와 동일한 성능을 가지는 하드웨어를 로컬 컴퓨터로 구축하였을 때 필요한 경제적 비용을 나타낸다. 표에서 (A)는 해당 성능의 로컬 컴퓨터를 구축하였을 때의 비용이며 (B)는 동일한 성능의 인스턴스를 한 시간 동안 사용하는 비용에 해당한다. 가용 시간은 클라우드 인스턴스와 같은 성능의 로컬 컴퓨터를 구축하는 비용으로 해당 인스턴스를 사용할 수 있는 시간을 의미하는데, 인스턴스의 성능이 높을수록 클라우드 컴퓨팅 플랫폼의 경제적 효용성이 더 높음을 확인할 수 있다.

이 밖에 클라우드 컴퓨팅 플랫폼의 활용은 전산 자원의 운영 및 유지에 비용을 투자할 필요가 없어진다는 점과 다양한 매체를 통해 데이터와 분석 결과를 빠르게 공유할 수 있다는 장점을 갖는다. 하드웨어와 소프트웨어 인프라를 자체적으로 운용할 경우 이를 구입하고 유지, 관리하기 위한 비용과 추가 인력이 요구되는데(Leavitt, 2009), 이러한 과정을 클라우드 컴퓨팅 플랫폼에 위탁할 경우 고정된 비용으로 안정적인 관리를 받을 수 있다. 또한, 네트워크를 통해 컴퓨터, 스마트폰과 같은 다양한 장치에서 클라우드 컴퓨팅 환경에 접속하고 분석을 수행할 수 있기 때문에, 학제 간 공동연구와 같은 협업을 추진하는 데에도 유리할 수 있다(Jang and Park, 2011).

다만 공용 클라우드 컴퓨팅 플랫폼은 본질적으로 불특정 다수의 사용자가 공유하는 것이고, 따라서 내부적으로 구축된 폐쇄적 전산 환경에 비해 외부 공격에 취약할 수밖에 없다. 때문에 클라우드 컴퓨팅 플랫폼을 활용할 때에는 데이터 손실이나 오염, 외부 유출과 같은 문제가 발생할 수 있음을 인지하고, 이러한 문제를 최소화할 수 있는 방안에 대해서도 고민이 필요할 것이다.

결 론

데이터 수집 기술의 발달과 대용량의 공간데이터 처리, 가공 기술의 발전은 공간분석에 사용할 수 있는 데이터의 양적 증가를 이끌었다. 클라우드 컴퓨팅 플랫폼에서는 데이터 크기나 연산의 복잡도에 따라 전산 자원을 탄력적으로 재구성할 수 있기 때문에, 병렬 연산을 통해 대용량의 공간데이터도 효과적인 처리가 가능하다(Yang *et al.*, 2017). 또한, 분석의 규모가 일정하지 않은 경우에도 유휴 자원에 의한 비용 손실이 발생하지 않아, 자체적인 전산 환경을 구축하는 것보다 경제적일 수 있다. 그러나 데이터 규모나 분석의 종류에 따라 클라우드 컴퓨팅 플랫폼에서의 공간분석이 갖는 효율성과 효과성은 다를 수 있음에도 불구하고 이에 관한 체계적인 연구는 아직까지 활발히 이루어지지 않고 있다. 이에 따라 본 연구에서는 일반적으로 많이 사용되는 공간통계 기법인 모란지수와

지리가중회귀분석을 대표적인 공용 클라우드 컴퓨팅 플랫폼인 아마존 웹 서비스의 가상 서버와 로컬 컴퓨터에서 각각 수행한 후 데이터 크기에 따른 연산 속도 증가량과 분석에 소요되는 비용을 비교하였다.

앞서 살펴본 바와 같이 시뮬레이션을 통해 생성된 가상의 데이터를 사용해 실험한 결과, 중앙처리장치의 클록 수가 높은 로컬 컴퓨터와 비교하여 병렬 연산에 적합한 환경을 갖춘 공용 클라우드 컴퓨팅 플랫폼에서 보다 효율적인 연산이 가능함을 알 수 있었다. 특히 데이터 크기에 따른 연산 시간의 증가 정도를 살펴보았을 때, 데이터의 규모가 클수록 클라우드 컴퓨팅 플랫폼이 로컬 컴퓨터와 비교하여 상대적으로 높은 효율성을 보여주었다. 실험에 사용된 데이터는 실제 대규모 데이터라고 할 수는 없지만, 연산 시간에 대한 회귀분석을 통해 공간데이터의 크기가 클수록 클라우드 기반의 분석이 효과적일 수 있음을 확인했다.

물론 클라우드 컴퓨팅 플랫폼과 동일한 성능의 전산 환경을 자체적으로 구축, 운용할 수 있다면 이와 같은 비교는 의미가 없을 것이다. 그러나 매우 높은 성능의 전산 자원을 필요로 하는 대규모 분석이 상시로 이루어지는 것이 아니라면 공용 클라우드 컴퓨팅 서비스의 활용은 경제적인 측면에서도 유리하다. 높은 성능의 전산 자원을 직접 구축하는 데에는 상당한 비용이 소요되나, 실제 이를 모두 필요로 하는 분석은 일시적인 경우가 많고, 향후 더 높은 성능을 요구하는 분석이 발생한다면 이는 추가적인 지출로 이어지게 된다. 반면 일반적으로 클라우드 컴퓨팅 플랫폼에서는 하드웨어의 성능과 각 구성 요소의 사용 여부를 탄력적으로 조정할 수 있기 때문에, 대규모의 분석이 지속적으로 이루어지는 경우를 제외하면 상대적으로 비용을 절감할 수 있다.

본 연구의 가장 큰 의의는 실험을 통해 클라우드 컴퓨팅 플랫폼에서의 공간분석이 갖는 효율성과 효과성을 실증적으로 밝혔다는 점일 것이다. 그러나 본 연구에서는 무작위로 분포된 임의의 포인트 데이터를 활용하여 분석을 수행했으며, 몬테카를로 시뮬레이션의 원활한 실행을 위해 데이터의 규모 또한 일정 정도로 제한할 수밖에 없었다. 실험에 사용된 데이터 규모 내에서는 데이터의 크기와 연산 속도 간 뚜렷한 선형 관계가 나타났고, 이를 토대로 대용량 공간데이터의 가공과 분석에 클라우드 컴퓨팅 플랫폼이 효율적이라는 결론을 도출하였으나 데이터의 크기가 실험 범위 이상으로 커질 경우 이와 같은 관계가 유효하지 않을 수도 있다. 또한, 로컬 컴퓨터와 클라우드가 장치의 성능이라는 관점에서 정확히 동일한 조건에서 이루어지지는 않았기 때문에, 환경 특성에 따른 성능 차이를 보여주지 못하였다는 한계를 갖는다. 이와 같은 연구의 한계를 보완하기 위해서는 가상의 포인트 데이터에 국한되지 않은, 다양한 공간 빅데이터를 사용한 후속 연구가 필요하며, 동일한 성능을 가진 장치 간의 연산 속도 차이를 비교하는 후속 연구 역시 필요하다. 덧붙여 본 연구에서는 비교적 단순한 형태의 병렬 연산인 처치곤란 병렬 방식 연산을 사용하였는데, 공간데이터의 분석에 있어서 연산량을 기하급수적으로 증가시키는 공간 연산 특성에 대하여 거리 가중치 및 데이터 행렬 관계에 새로운 방식의 병렬 연산을 접목했을 때 나타날 수 있는 효율성 향상 효과에 관해서도 추후 연구를 통해 검증해야 할 것이다. KAGIS

REFERENCES

Armbrust, M., A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica and M. Zaharia. 2009. Above the clouds: a berkeley view of cloud computing. Technical Report No. UCB/ EECS-2009-28.

Azad, A. and A. Buluc. 2017. A work-efficient parallel sparse matrix-sparse vector multiplication algorithm. Proceedings of the 2017 IEEE International Parallel

and Distributed Processing Symposium. Florida, FL, USA, 31 May 2017. pp.688-697.

Cho, Y.I. 2013. Understanding big data and its major issues. Journal of Korean Associastion for Regional Information Society 16(3):43-65 (조영임. 2013. 빅데이터의 이해와 주요 이슈들. 한국지역정보화학회지 16(3):43-65).

Choi, J.G. and B.N. Noh. 2011. Security technology research in cloud computing environment. Journal of Security Engineering 8(3):371-384 (최재규, 노봉남. 2011. 클라우드 컴퓨팅 환경에서의 보안 평가 요소. 보안공학연구논문지 8(3):371-384).

Cleveland, W.S. 1979. Robust locally weighted regression and smoothing scatterplots. Journal of the American Statistical Association 74(368):829-836.

Cleveland, W.S. and S.J. Devlin. 1988. Locally weighted regression: an approach to regression analysis by local fitting. Journal of the American Statistical Association 83(403):596-610.

Fang, C., C.J. Yang, Z. Chen, X.J. Yao and H.T. Guo. 2011. Parallel algorithm for viewshed analysis on a modern GPU. International Journal of Digital Earth 4(6):471-486.

Garfinkel, S. 2007. An evaluation of Amazon's grid computing services: EC2, S3, and SQS. Harvard Computer Science Group Technical Report TR-08-07.

Goodchild, M.F. 2007. Citizens as sensors: the world of volunteered geography. GeoJournal 69(4):211-221.

Guan, Q. and K.C. Clarke. 2010. A general-purpose parallel raster processing programming library test application using a geographic cellular automata model. International Journal of Geographical Information Science 24(5):695-722.

Haut, J.M., M. Paoletti, J. Plaza and A. Plaza. 2017. Cloud implementation of the K-means algorithm for hyperspectral image analysis. The Journal of Supercomputing. 73(1):514-529.

Healey, R., S. Dowers, B. Gittings and M. J. Mineter. 1997. Parallel processing algorithms for GIS. CRC Press. Florida, FL, USA. 460pp.

Jang, E.Y. and C.S. Park. 2011. A study of modeling and simulation for the availability optimization of cloud computing service. Journal of the Korea Society for Simulation 20(1):1-8 (장은영, 박춘식. 2011. 클라우드 컴퓨팅 서비스의 가용성 최적화를 위한 모델링 및 시뮬레이션. 한국시뮬레이션학회논문지 20(1):1-8).

Kim, T., I. Kim, C. Min and Y.I. Eom. 2012. Trends in cloud computing security technology. Communications of the Korean Institute of Information Scientists and Engineers 30(1):30-38 (김태형, 김인혁, 민창우, 엄영익. 2012. 클라우드 컴퓨팅 보안 기술 동향. 정보과학회지 30(1):30-38).

Kitchin, R. 2013. Big data and human geography: opportunities, challenges and risks. Dialogues in human geography 3(3):262-267.

Leavitt, N. 2009. Is cloud computing really ready for prime time?. Computer 42(1): 15-20.

Lee, K.H., H. Choi and Y.D. Chung. 2011.

Massive data processing and management in cloud computing: a survey. Journal of KISE 38(2):104-125 (이경하, 최현식, 정연돈. 2011. 클라우드 컴퓨팅에서의 대용량 데이터 처리와 관리 기법에 관한 조사. 정보과학회논문지 38(2):104-125).

Li, Z., A.S. Fotheringham, W. Li and T. Oshan. 2018. Fast Geographically Weighted Regression (FastGWR): a scalable algorithm to investigate spatial process heterogeneity in millions of observations. International Journal of Geographical Information Science. 33(1):155-175.

Moran, P.A. 1950. Notes on continuous stochastic phenomena. Biometrika 37 (1/2):17-23.

Park, J.M., M.H. Lee, D.B. Shin and J.W. Ahn. 2015. Deduction of the policy issues for activating the geo-spatial big data services. Journal of Korea Spatial Information Society 23(6):19-29 (박준민, 이명호, 신동빈, 안종욱. 2015. 공간 빅데이터 서비스 활성화를 위한 정책과제 도출. 한국공간정보학회지 23(6):19-29).

Quinn, M.J. 1987. Designing efficient algorithms for parallel computers. McGraw-Hill, Inc. New York, NY, USA. 288pp.

Tang, W. and W. Feng. 2017. Parallel map projection of vector-based big spatial data: coupling cloud computing with graphics processing units. Computers, Environment and Urban Systems 61:187-197.

Turton, I. and S. Openshaw. 1998. High-performance computing and geography: Developments, issues, and case studies. Environment and Planning A 30(10): 1839-1856.

Wang, Y., S. Wang and D. Zhou. 2009. Retrieving and indexing spatial data in the cloud computing environment. In: Jaatun, M.G., G. Zhao and C. Rong(ed.). Cloud Computing. Springer Berlin Heidelberg. Berlin, pp.322-331.

Xiaoqiang, Y. and D. Yuejin. 2010. Exploration of cloud computing technologies for geographic information services. Proceedings of the 18th International Conference on Geoinformatics. Beijing, China, 18-20 June 2010. pp.1-5.

Yang, C., M. Goodchild, Q. Huang, D. Nebert, R. Raskin, Y. Xu, M. Bambacus and D. Fay. 2011. Spatial cloud computing: how can the geospatial sciences use and help shape cloud computing?. International Journal of Digital Earth 4(4):305-329.

Yang, C., Q. Huang, Z. Li, K. Liu and F. Hu. 2017. Big Data and cloud computing: innovation opportunities and challenges. International Journal of Digital Earth 10(1):13-53.

Yang, C., Y. Xu and D. Nebert. 2013. Redefining the possibility of digital Earth and geosciences with spatial cloud computing. International Journal of Digital Earth 6(4):297-312.

Yue, P., H. Zhou, J. Gong and L. Hu. 2013. Geoprocessing in cloud computing platforms—a comparative analysis. International Journal of Digital Earth 6(4):404-425.

Zhang, J., S. You and L. Gruenwald. 2016. High-performance polyline intersection based spatial join on GPU-accelerated clusters. Proceedings of 2016 ACM SIGSPATIAL International Workshop on

Analytics for Big Geospatial Data. San Francisco, CA, USA, 31 October 2010. pp.1-8.

Zhang, Y., X. Zheng, Z. Wang, G. Ai and Q. Huang. 2018. Implementation of a Parallel GPU-Based Space-Time Kriging Framework. ISPRS International Journal of Geo-Information. 7(5):193-205

Zhao, Y.L., A. Padmanabhan and S.W. Wang. 2013. A parallel computing approach to viewshed analysis of large terrain data using graphics processing units. International Journal of Geographical Information Science 27(2):363-384.

Zhou, X., C. Xu and B. Kimmons. 2015. Detecting tourism destinations using scalable geospatial analysis based on cloud computing platform. Computers, Environment and Urban Systems 54:144-153.