Performance Analysis of Spatial Indexing in the Cloud

Lamiaa Said El-Sayed
Information System dep.,
Faculty of Computers and Information, Benha University
Egypt

Hatem M. Abdul-Kader
Information System dep.,
Faculty of Computers and Information, Menofia University
Egypt

Salah M. El-Sayed
Scientific Computing dep.,
Faculty of Computers and Information, Benha University
Egypt

ABSTRACT
The widespread of geospatial services produces massive volumes of spatial data. Cloud computing is a necessity for big data management. Efficient retrieval algorithms of such data are a prerequisite. In this paper, we evaluate the efficiency of spatial indexing for huge datasets at cloud computing environment. Two of the most common data structures are selected for this study namely the R-Tree and the priority R-tree (PR-Tree). R-Tree is one of the most common access methods for spatial data. Priority R-tree is an optimal variation of the R-tree and more efficient for extreme datasets. We implemented the two data structures then we deployed them on various cloud instances with different resources. We evaluated the performance of running these applications with different spatial datasets. The query response time is also measured for both data structures. We reported the results which can be useful in retrieving huge datasets on the cloud.

General Terms
Cloud computing, Spatial Index

Keywords
Cloud computing, Spatial Index, R-Tree, PR-Tree

1. INTRODUCTION
On the Internet, geographical information is constantly increasing. Big Data is a critical issue regarding to storage and computing resources. 1.2 zettabytes of data were produced in 2010 and will increase to 8 zettabytes in 2015 referring to a market research study of IDC [1]. Traditional relational databases can't get the ride of this amount of data. Spatial databases store data of the objects that defined in a geometric space; they are used in many applications which require retrieving the data rapidly. Consequently, spatial indexing has become essential for fast retrieval of results [2].

Cloud computing is the use of resources that are delivered as a service over a network. Due to the flexibility and scalability in cloud computing, now cloud computing plays an important role to handle a large-scale data analysis [3]. Using it in Geo-Spatial sciences can be very useful. Performance analysis for Big Data in the Cloud is significant, because its applications can include various thousands of programs running on thousands of machines [4].

In this paper, we will present the performance estimation of using cloud computing for indexing huge spatial datasets. We implemented the R-Tree and Priority R-Tree data structures for spatial data. R-tree is the most common spatial index. PR-Tree is an optimal variation of the R-tree and more efficient for extreme datasets. We studied the performance of the two data structures on different cloud instances and provide the analysis. We also measured the query response time for the both data structures. Two types of queries will be tested, these types are Range query and nearest neighbor query. Our experiments were done on Amazon EC2 cloud platform.

The rest of the paper is organized as follows. Section II discusses the related work of spatial indexing and cloud computing. Section III presents the setup of the experiments. Section IV presents the experimental results to evaluate the performance. Finally, Section V includes the conclusion and the future work plans.

2. RELATED WORK

2.1 Spatial Indexing
Indexing spatial data is a significant research area nowadays. Many existing applications store and manipulate spatial data. The fast development of data acquisition technology makes large amounts of spatial to be gathered and deployed in distributed computers [5]. In spatial data management it is important to efficiently retrieve objects based on their location by using spatial access methods [6]. A spatial database contains a huge collection of objects that are located in multi-dimensional space [7]. Spatial index is the main technology of spatial database; it is able to improve data retrieve efficiency by using an appropriate data structure to construct the relationship between spatial position and spatial object [6]. Spatial indexing is useful for many applications like Geographical Information System, Computer Aided Designing, Multimedia and others [8]. Many spatial index structures have been proposed.

R-Tree is the most widely used index structure for multi-dimensional data because of its performance efficiency and simple implementation. In this index structure, the objects are represented as Minimum Bounding Rectangles (MBRs). R-Tree consists of root, non-leaf and leaf nodes. The nodes in R-Tree have a maximum and minimum capacity. When a node reaches the maximum capacity, the node is split such that the number of objects in each node lies between the minimum and maximum limit [2].

Many variants of R-Tree have been proposed to give better performance in case of some queries or applications. Proposed R-tree variants like R+-Tree, R*-tree, Hilbert R-tree and PR-Tree. None of the existing R-tree variants has worst-case query performance guarantee except PR-Tree. In the worst-case, a query can visit all nodes in the tree even when the output size is zero [2].

PR-Trees were derived from Pseudo Priority trees. It defines MBRs as a point in N-dimensions, represented by the ordered pair of the rectangles. The term priority comes from the introduction of four priority-leaves that represents the most extreme values of each dimensions, included in every branch of the tree. Before answering a window-query by traversing the sub-branches, the priority R-tree first checks for overlap in its priority nodes. The sub-branches are traversed and constructed by checking whether the least value of the first dimension of the query is above the value of the sub-branches. This gives access to a quick indexation by the value of the first dimension of the bounding box [9].

PR-Tree is the first R-tree variant that is not only practically efficient but also optimal. In case of query performance, the PR-trees performance in well distributed real data has no significant difference compared to the existing R-tree variants. In contrast, with extreme data PR-trees are much more robust than all other R-tree variants, as they assure worst-case performance [9]. We used PR-Tree in this paper to perform our tests as Geo-Spatial data are extreme.

2.2 Cloud Computing

Cloud computing is a novel technological evolution in computer science, allowing groups to host, process, and analyze large volumes of data [10]. The U.S. National Institute of Standards and Technology (NIST) defines Cloud computing as a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. Because cloud computing is assured to be convenient, save the budget and energy, the US government impose that all agencies over the next several years must migrate to cloud computing or explain why they did not use it. Therefore it will become the future computing infrastructure for supporting geospatial sciences [11]. Cloud Computing allows businesses to experiment and implement new services rapidly [12]. It is easier and faster to rent computing power through cloud service than to buy the needed servers. The implementation of new services that base on computing power or storage can be completed faster [1].

Cloud computing characteristics are rapid elasticity, measured service, on-demand self service, broad network access, and resource pooling. Cloud computing has three service models and they are Software as a Service (SaaS) like Amazon EC2, Platform as a Service (PaaS) like Microsoft windows Azure and Google Apps, and Infrastructure as a Service (IaaS) like Salesforce.com. Its Deployment Models are Public Cloud, Private Cloud, and Hybrid Cloud which is a combination of private and public cloud [13].

The majority of cloud service providers use machine virtualization to provide flexible and cost-effective resource sharing. By using of virtualization, clouds promise to satisfy a massive number of users with different needs with the same shared set of physical resources. Using virtualization and resource time sharing may affect the performance of the demanding scientific computing workloads [14].

Spatial cloud computing refers to the cloud computing model that is used by geospatial sciences. Spatial Cloud Computing depends on the computing infrastructure like CPU speed, RAM size, network bandwidth, storage volume and other computing resources [11]. In the present work, our main use of Cloud computing technology is in evaluating the performance of running PR-Tree spatial index with huge dataset on cloud instances.

3. EXPERIMENTAL SETUP

The experiments were conducted on existing R-Tree and PR-Tree packages [15] [16]. The code is written in java and the implementation tries to be memory efficient.

Our results are based on a two random datasets. The first dataset is with 1,000,000 spatial data. The second dataset is with 2,000,000 of spatial data. Some modifications were done on the two packages to accept the same form of datasets. Each line in the dataset file has four numbers x1, y1, x2, y2 which represents one MBR.

We measured the average time of loading the tree, the intersection query and the nearest neighbor query. The intersection query finds all objects that intersect the given rectangle. The nearest neighbor query gets the nearest neighbor of the given point.

To perform our experiments one personal computer with 4 CPU cores and 8 GB memory was used. For cloud tests, Amazon EC2 platform was used to perform our experiments. Experiments have been run on various cloud instances with different RAM size and different CPU speed.

The first type of instances for general purpose GPU compute applications represented in "g2.2xlarge" instance with 8 CPU cores and 15 GB memory. The second type of instances is for memory-intensive applications represented in "r3.large" instance with 2 CPU cores and 15.25 GB memory and "r3.xlarge" instance with 4 CPU cores and 30.5 GB memory. The third type of instances provides a balance of compute, memory, and network resources, and it is a good choice for many applications represented in "m3.large" instance with 2 CPU cores and 7.5 GB memory and "m3.xlarge" instance with 4 CPU cores and 15 GB memory. The last type of instances is compute-optimized instances that provide customers with the highest performing processors represented in "c3.xlarge" instance with 4 CPU cores and 7.5 GB memory [17].

4. EXPERIMENTAL RESULTS

For testing the performance of running the two data structures we measured the average time of the indexing performance represented in loading the tree, the intersection query which finds all objects that intersect a given rectangle and The nearest neighbor query which gets the nearest neighbor of a given point.

Figure 1 shows the results of loading R-Tree and PR-Tree with 1,000,000 data size for indexing.

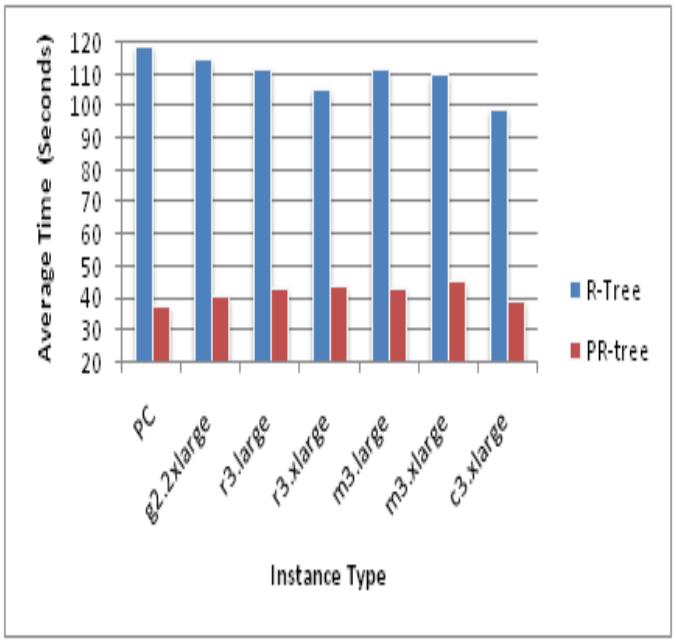

Fig 1: Loading Tree with one Million dataset

Figure 2 shows the results of making an Intersection query in the tree with 1,000,000 spatial data.

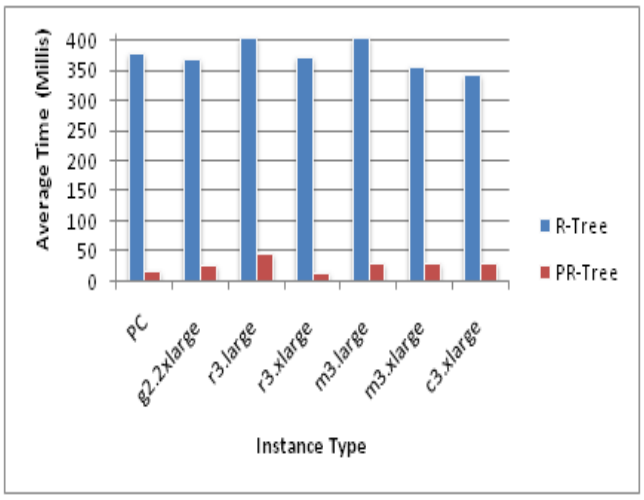

Fig 2: Intersection Query with one Million dataset

Figure 3 shows the results of making a nearest neighbor query in the tree with 1,000,000 spatial data.

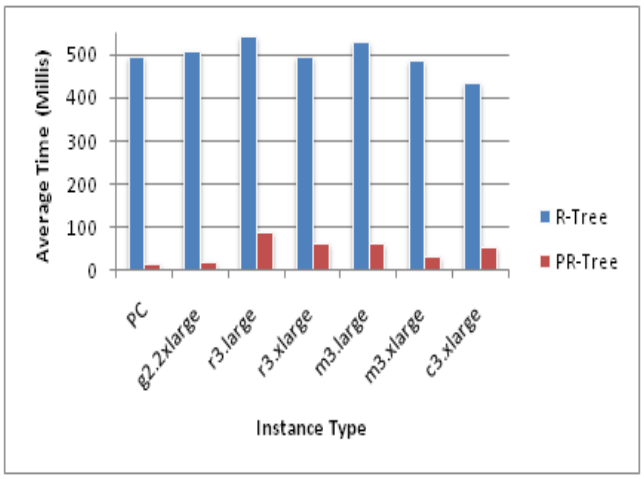

Fig 3: Nearest Neighbour Query with one Million dataset

Figure 4 shows the results of loading R-Tree and PR-Tree with 2,000,000 spatial data for indexing.

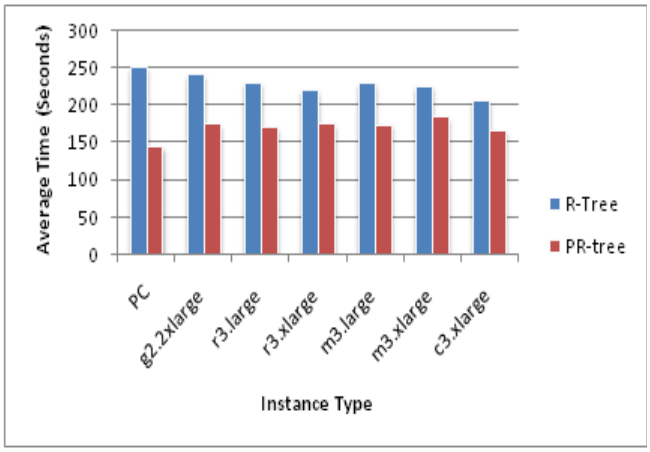

Fig 4: Loading tree with two Million dataset

Figure 5 shows the results of making an Intersection query in the tree with 2,000,000 spatial data.

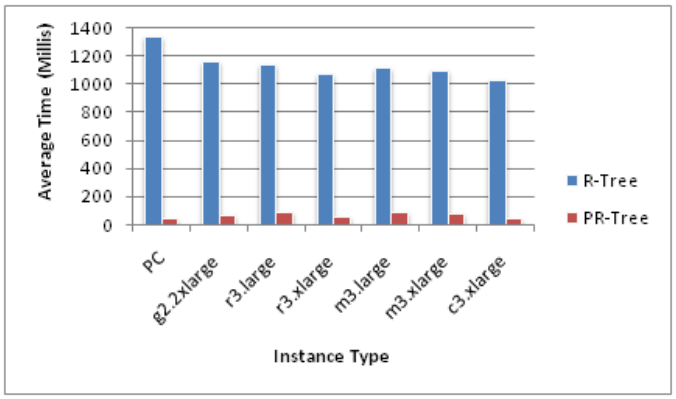

Fig 5: Intersection Query with two Million dataset

Figure 6 shows the results of making a nearest neighbor query in the tree with 2,000,000 spatial data.

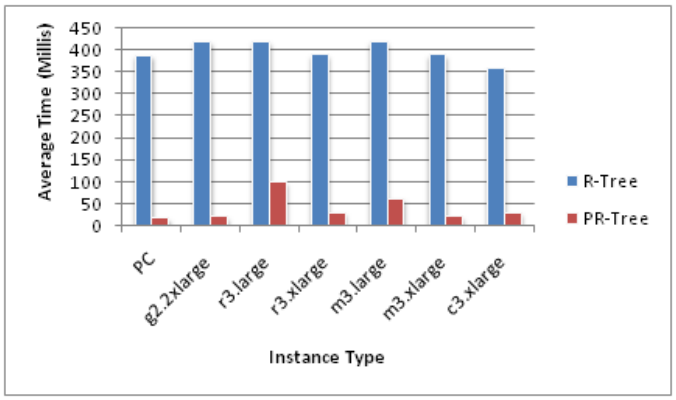

Fig 6: Nearest Neighbour Query with two Million dataset

The results show that the performance of loading the R-Tree and PR-Tree or executing any of the two queries for both of the two data structures is better when using compute optimized c3.xlarge instance and memory optimized r3.xlarge instance than the other instances at most cases. When using a personal computer with similar specifications to the cloud instance, the performance is a little better in the cloud than the PC. The code of the R-Tree and PR-Tree applications is not optimized to run on configurations with multiple CPU cores which affected the performance.

5. CONCLUSION AND FUTURE WORK

In this work we studied the performance of R-Tree and PR-Tree applications on various instances of Amazon EC2 cloud platform. The results show that when using an instance with greater resources a better performance will be gained. The performance on the cloud is a little better because the application is not optimized to run on configurations with multiple CPU cores.

As a future work, we plan to use a parallel programming model to be compatible with parallel processing implementations for improving the performance. Also we plan to use a real dataset.

6. REFERENCES

[1] Leimbach T. et al., "Potential and impacts of cloud computing services and social network websites," Science and Technology Options Assessment, 2014.

[2] Balasubramanian L. and Sugumaran M., "A State-of-Art in R-Tree Variants for Spatial Indexing," *International Journal of Computer Applications*, vol. 42, no. 20, 2012.

[3] Wang Y., Wang S., and Zhou D., "Retrieving and Indexing Spatial Data in the Cloud Computing Environment," in *the First International Conference on Cloud Computing, Springer-Verlag*, 2009.

[4] Dai J., Huang J., Huang, B. Huang S., and Liu Y., "HiTune: Dataflow-Based Performance Analysis for BigData Cloud," in *USENIX annual technical conference*, 2011.

[5] Yang Y., Wu L., Guo J., and and Shanjunliu, "Research on Distributed Hilbert R tree Spatial Index Based on BIRCH Clustering," *IEEE*, 2012.

[6] Moreau M. and Osborn W., "mqr-tree: A 2-dimensional Spatial Access Method," *Journal Of Computer Science And Engineering*, vol. 15, no. 2, 2012.

[7] Sharifzadeh M. and Shabani C., "VoR-tree: R-tree with Voronoi Diagrams for Efficient Processing of Nearest Neighbour Queries," *Proc. VLDB Endowment*, vol. 3, no. 1, 2010.

[8] Manolopoulos Y., Nanopoulos A., and Papadopoulos N., "R-Trees: Theory and Applications," *London: Springer, 1st Edition*, 2006.

[9] Arge L., deBerg M., Haverkort H.J., and Yi K., "The Priority R-Tree: A Practically Efficient and Worst-Case Optimal R-Tree," in *Proceedings of ACM SIGMOD Conference on Management of Data*, 2004.

[10] Gannon D. and Reed D.A., *Parallelism and the cloud*, The Fourth Paradigm ed., 2009.

[11] Yang C. et al., "Spatial cloud computing: how can the geospatial sciences use and help shape cloud computing?," *International Journal of Digital Earth*, 2011.

[12] Fielder A., Brown I., Frank A., Kara S., and Weber V., "Cloud Computing Study," European Parliament's Committee, Internal Market and Consumer Protection 2012.

[13] Office of the privacy Commissioner of Canada. (2014, March) Introduction to Cloud Computing. [Online]. http://www.priv.gc.ca/resource/fs-fi/02_05_d_51_cc_e.pdf

[14] Wang G. and Ng TSE, "The impact of virtualization on network performance of amazon ec2 data center," *Proceedings IEEE*, 2010.

[15] (2014, February) R-Tree in Java with storage capabilities and an api. [Online]. https://github.com/moubarak/spatial-index

[16] (2014, February) PRTree, a Priority R-Tree, a spatial index for java code. [Online]. http://www.khelekore.org/prtree/

[17] (2014, July) Amazon EC2 Instances. [Online]. http://aws.amazon.com/ec2/instance-types/