International Journal of Database Theory and Application
Vol.9, No.8 (2016), pp.147-158
http://dx.doi.org/10.14257/ijdta.2016.9.8.15

Data Analysis Technique for Massive Spatial Data Using Hadoop

Minwuk Jeon[1] and Byoung-Woo Oh[2*]

[1,2]*Department of Computer Engineering, Kumoh National Institute of Technology, Gumi, Korea*
[1]*minwukjeon@gmail.com*, [2]*bwoh@kumoh.ac.kr*

Abstract

The spatial data set has much useful information, but the amount of volume is massive and the type is complex. It makes hard to analyze the spatial data. There are software tools for general data. Hadoop is one of the tools to process the big data. Hadoop can be used to analyze the large amount of spatial data. This paper proposed a data analysis technique for massive spatial data using Hadoop. We extend the grid based clustering algorithm to use Hadoop. The grid based clustering algorithm makes clusters with cells. Each cell has a number that counts contained objects. Only the cells who had the sufficient population can be join in clusters. The other cells ignored as noise. This paper proposed to enhance performance using Hadoop. In order to evaluate the enhancement of performance, the execution time is measured and compared. As the result, the proposed algorithm is 1.8 times faster than the original grid based clustering algorithm.

Keywords*: Spatial Data, Hadoop MapReduce, Clustering Algorithm, Data Mining*

1. Introduction

The spatial data is difficult to analyze by human because its size is massive and it represents by many types such as point, line and polygon. There were many researches for analyzing spatial data to produce meaningful information from the past. But, there was restriction on limitation in Hardware or Communications technology [1-3]. Become a Big Data era these days, many tools that analyzing the massive data efficiently such as R or Hadoop are released [4-6]. Especially Hadoop has strong point that it's possible to distributed processing the massive data in low cost, there's a drift towards research about Hadoop or System using Hadoop [7-9]. Hadoop consists of two parts, HDFS (Hadoop Distributed File System) and MapReduce framework. HDFS distributes the massive data and save it, and MapReduce framework distributes the massive data and processing it [10-11]. Hadoop's structure has one Master with many Slaves which processing distributed data. But in Hadoop MapReduce, JobTracker plays a role of Master and TaskTracker plays a role of Slaves to distribute processing. In this paper, we implement the Grid Based Clustering algorithm using Hadoop MapReduce to analyzing spatial data.

This paper is organized as follows. Chapter 2 introduces Hadoop MapReduce and Grid Based Clustering Algorithm. Chapter 3 describes the Grid Based Clustering Algorithm using Hadoop MapReduce. Chapter 4 reports the result of the experiment. Finally, Chapter 5 presents conclusion and future work.

[1] *Corresponding Author

ISSN: 2005-4270 IJDTA

2. Related Works

2.1. Hadoop MapReduce

Hadoop MapReduce is a distributed framework for processing massive data. Hadoop MapReduce consists of Map phase and Reduce phase. Each phase has input and output that has key and value. In Map phase, Map task transforms received data to intermediate key-value pair. The result of Map task is key-value pair. Reduce task extracts object data from the result of Map task. Figure 1 shows how MapReduce processes request from a client. A job is divided into many Map tasks and Reduce task in Hadoop MapReduce. TaskTracker handles many Map tasks and Reduce task to send heartbeat to JobTracker in certain time for notice the normal condition. The heartbeat is periodic signal. There is Map task or Reduce task should be handled, assigned the task to available JobTracker.

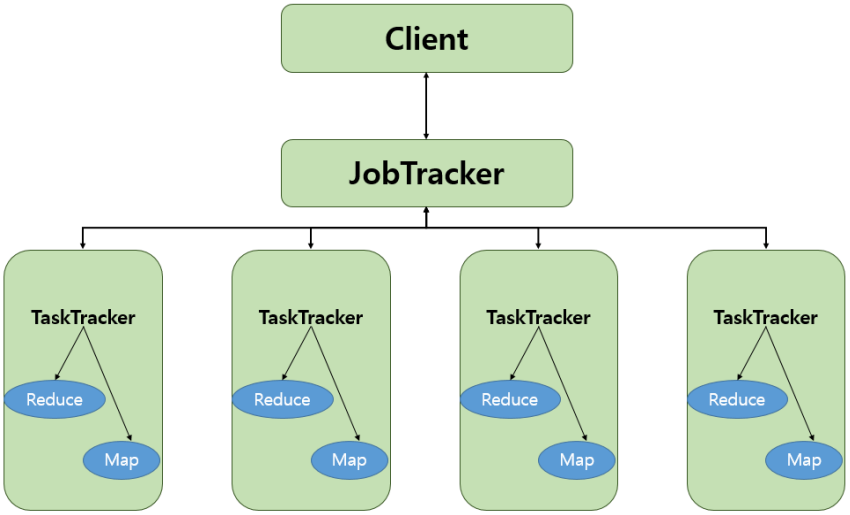

Figure 1. Grid Based Clustering Algorithm

2.2. Grid Based Clustering Algorithm

In this paper, we extend the grid based clustering algorithm [12]. The grid based clustering algorithm makes clusters by allocating objects in each cluster cells. Figure 2 shows the process of making clusters in the grid based clustering algorithm.

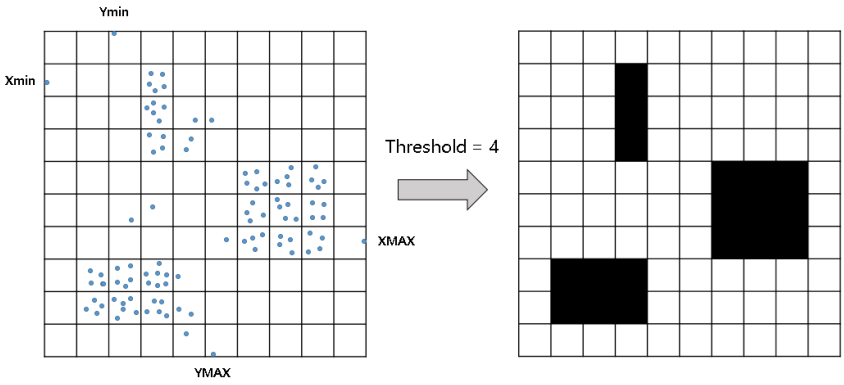

Figure 2. Grid Based Clustering Algorithm

The summary of the algorithm is as follows:
- Calculate the each objects maximum value and minimum value on each dimension
- Draw square using these four points
- Divide the square in equal size and counting number of objects contained in cells
- Compare the number of objects with threshold
- Classify as noise if the number of objects are less than threshold else classify as Cluster
- Connect adjacent Clusters

The algorithm can distinguish clusters from noise. In the Figure 2, the threshold for noise is set to 4. The algorithm can deal with not only spatial 2-dimensinal space but also n-dimensional data cube including aspatial (attribute) data.

3. Grid Based Clustering Algorithm Using Hadoop MapReduce

We propose a method for grid based clustering algorithm using Hadoop. The input of the algorithm is a set of spatial data. The type of the input spatial data can be point, line, and polygon. Figure 3 shows summary of the proposed algorithm. The process consists of map method, shuffling, reduce method, clustering method. The map and reduce method implements the Hadoop map function and reduce function, respectively. The clustering method is extended the original grid based clustering algorithm for Hadoop.

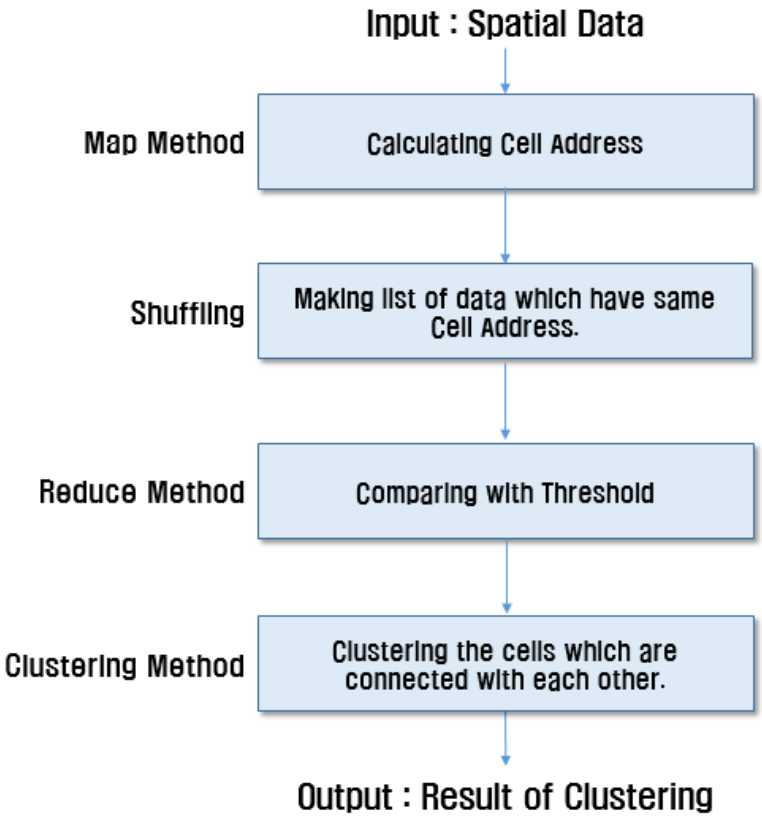

Figure 3. Processing of Proposed Algorithm

3.1. Map Method

Figure 4 shows Algorithm of Map Method. The Map Method extract the data to clustering in input data, calculate the Cell Address which has assigned data. First of all, read the data line by line to make pairs of key-value that has Row Number as key and Data in a Row as value. Split the Data in a Row to extract x and y value for calculate the address. The Map Method sets 1 (integer one) as value to make a key-value pair. Hadoop merges these returned values and makes <key, list of Integer 1> pair with the same Cell Address.

```
Input : <Row number, Data in a Row>
Output : <Cell Address, Integer 1>

Algorithm Map Method

   X <- Split the Data in a Row to extract X value for calculate the address
   Y <- Split the Data in a Row to extract Y value for calculate the address

   Integer cx, cy <- GetCellAddress(X, Y)
   Integer one <- 1

   Return <Cell Address, one>

End Algorithm
```

Figure 4. Algorithm of Map Method

Figure 5 shows GetCellAddress algorithm to calculate Cell Address.

```
Input : X, Y (X,Y-coordinate which calculate the Cell Address)
Output : cx, cy (Cell Address of X,Y)

Algorithm GetCellAddress

   xmin <- Minimum of whole X-coordinate
   xmax <- Maximum of whole X-coordinate
   ymin <-  Minimum of whole Y-coordinate
   ymax <- Maximum of whole Y-coordinate

   mx <- number of section of X-axis
   my <- number of section of Y-axis

   Integer cx <- (Integer) ((X- xmin) / ((xmax-xmin+1) / mx))
   Integer cy <- (Integer) ((Y - ymin) / ((ymax-ymin+1) / my))

   Return cx, cy

End Algorithm
```

Figure 5. Algorithm of GetCellAddress

3.2. Reduce Method

Figure 6 shows the Algorithm of Reduce Method. Reduce Method counts the number of assigned data on cell address with comparing with threshold to group candidate of cluster or noise. As explained in Section 3.1, Reduce Method receives key-value pairs that consist of cell address and list of integer 1. The cell address is key. The list of Integer 1 is value. To classify the candidate of cluster and noise, Reduce Method calculates the sum of list of Integer 1. If its result has same or bigger than the threshold it is classified as candidate of cluster. If it less than the threshold, it is classified as noise. At last, Reduce Method returns the key-value pairs. The key is cell address. The value is classified result which can be cluster or noise.

```
Input : <Cell Address, List of Integer 1>
Output : <Cell Address, State of Cell>

Algorithm Reduce Method

   Integer num_of_point <- 0

   While ( List.hasNext() )

      number_of_point <- number_of_point + Element of List

   End While

   IF (number_of_point >= Threshold)

         String state_of_cell <- Candidate of cluster

      Else

         String state_of_cell <- Noise

      End Else

   End IF

  Return <Cell Address, state_of_cell >

End Algorithm
```

Figure 6. Algorithm of Reduce Method

3.3. Clustering Method

Clustering Method checks the connection between candidates of clusters. It assigns cluster number to each cluster. To assign the number of cluster, it uses variable cnt. When the cluster number is assigned to a cluster, value of variable cnt increased by one. If a cell is classified as a noise, it assigns cluster number to -1. If a cell is classified as a candidate

of cluster, it assigns cluster number to the value of variable cnt. Figure 7 shows algorithm of clustering method.

```
Input : <Cell Address, State of Cell>
Output : <Cell Address, Cluster Number>

Algorithm Clustering Method

   Integer cluster_number <- 0
   Integer cnt <- 1

   For (From Start to End of Cell)

      If(State of Cell = Noise)

           cluster_number <- -1

      End If

      If(State of Cell = Candidate of Cluster)

           cluster_number <- cnt

           While(Cell.hasNeighbor())

              cluster_number <- cnt

           End While

      End If

   cnt <- cnt+1

   End For

   Return <Cell Address, cluster_number>

End Algorithm
```

Figure 7. Algorithm of Clustering Method

Figure 8 shows condition of the connection.

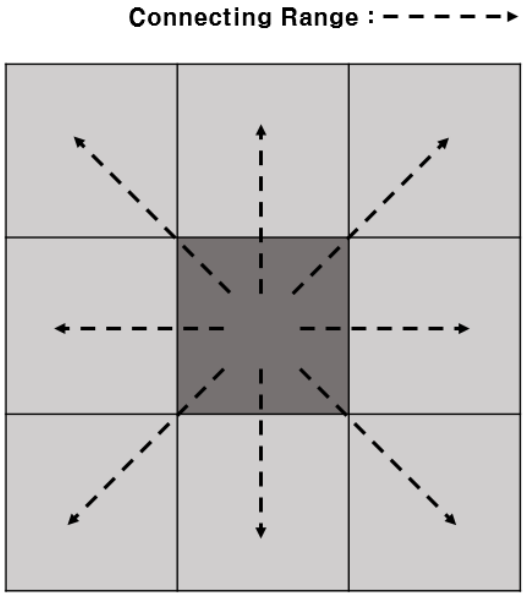

Figure 8. Condition of the Connection

4. Experiment and Result

All experiments are performed on a Hadoop Server with one Namenode and two Data node. Operating system which is operated on Namenode and Datanodes is Ubuntu 12.04. Namenode is equipped with Intel® Core 2 CPU 2.40GHz and 4GB of memory. Datanodes are equipped with Intel® Core™i3 CPU 3.60GHz and 4GB of memory. Table 1 shows composition of Hadoop Server.

Table 1. Composition of Hadoop Server

Name	CPU	RAM	OS	Role
PC 1	Intel Core 2 CPU 2.40GHz	4GB	Ubuntu 12.04	NameNode
PC 2	Intel Core™i3 CPU 3.60GHz	4GB	Ubuntu 12.04	DataNode
				Secondary NameNode
PC 3	Intel Core™i3 CPU 3.60GHz	4GB	Ubuntu 12.04	DataNode

4.1. Data Set and Implementing the Experiment

In order to implement the experiment, we generated the data for the experiment. We generated point-data randomly and constructed data set. Figure 9 shows a data set which consists of 500,000,000 point-data. The spatial data are much large in general. It reduces the performance of analyzing the spatial data. In order to prove the performance of proposed algorithm, we generated 4 data sets for experiment. Data sets consist of 1,000,000 10,000,000 100,000,000 and 500,000,000. In order to gather reliable result, we implemented experiment 100 times each data set. The experiment for proposed algorithm

was executed on Hadoop Server. The experiment for Grid Based Clustering Algorithm was executed on a machine with Intel® Core™i3 CPU 3.60GHz and 4GB of memory.

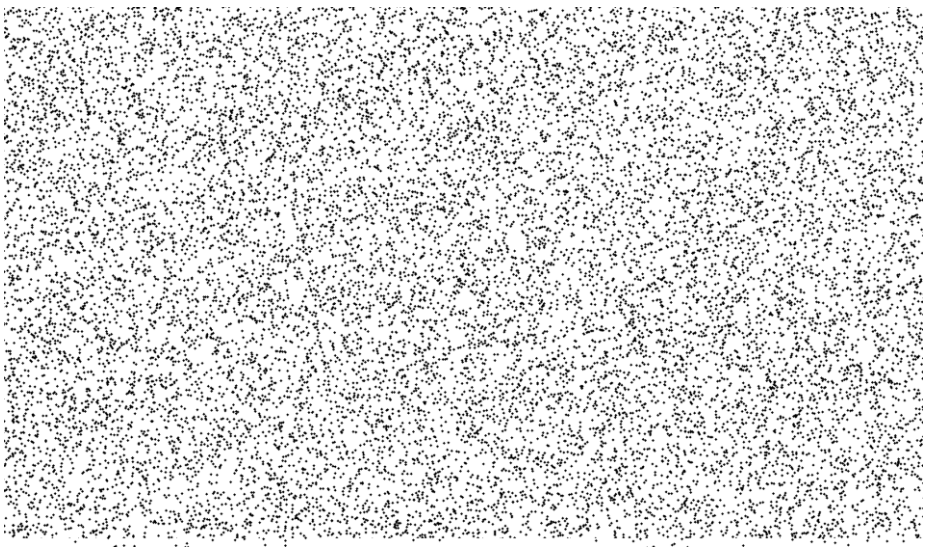

Figure 9. Data Set for the Experiment

4.2. Result

Table 2 shows the result of the experiment each data set.

Table 2. Result of the Experiments

Number of Points	File Size	Grid Based Clustering Algorithm	Proposed Algorithm	Difference
1,000,000	13.1MB	792 ms	432 ms	460 ms
10,000,000	131MB	11228 ms	6221 ms	5007 ms
100,000,000	1.28GB	38412 ms	21282 ms	17130 ms
500,000,000	7.34GB	213016 ms	116624 ms	96392 ms

As can be seen in Figure 10, proposed algorithm is an about 1.8 times faster than Grid Based Clustering Algorithm. In contrast, proposed algorithm is executed ordinarily and performed clustering.

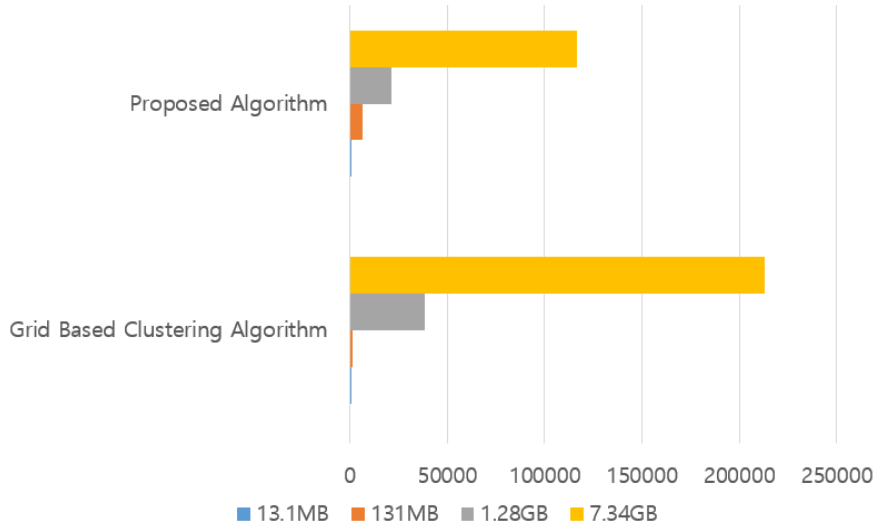

Figure 10. Performance Comparison

Figure 11 shows difference of running time between two algorithms. In Figure 11, difference of running time becomes wider as experiment data grows. The more efficiency gained, the bigger data input by the clustering with the proposed method in this paper.

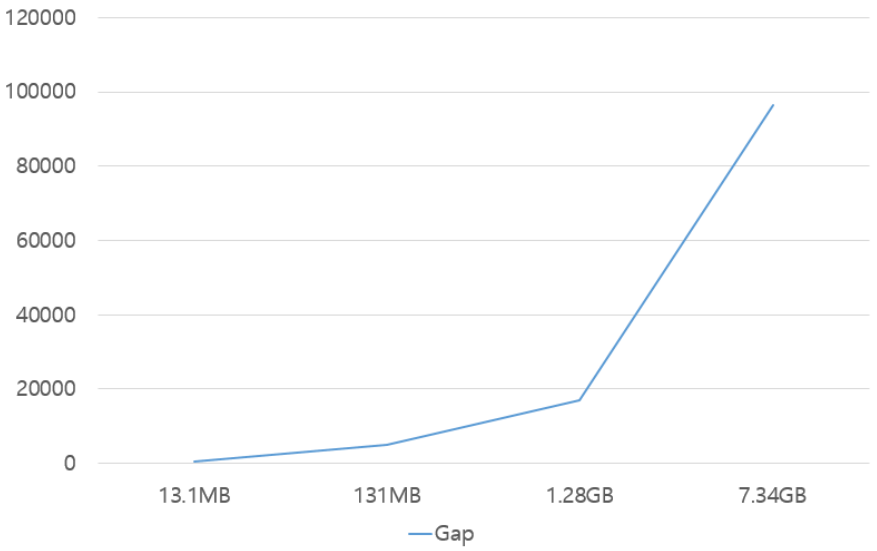

Figure 11. Difference Between Two Algorithms

Figure 12 shows the part of visualization result that clustered 5,000,000,000 points data set.

Figure 12. Result of Clustering

5. Conclusion

This paper proposed a technique for analyzing massive spatial data using Hadoop. It extended the grid based clustering algorithm to use Hadoop. We performed the experiment by using the spatial data in order to evaluate the performance of the proposed algorithm. In order to implement the experiment, we generated the data for the experiment. We generated point-data randomly and constructed Data Set. With respect to performance time, proposed algorithm is 1.8 times faster than not to use Hadoop. A between of running time becomes wider when Experiment data grows. The more efficiency gained, the bigger data input by the clustering with the proposed method in this paper. Proposed algorithm processes the spatial data efficiently by clustering the spatial data at lower cost.

Our future works could focus on Data Mining and Deep Learning for spatial data using Hadoop.

Acknowledgments

This Paper was supported by the Research Fund, Kumoh National Institute of Technology.

References

[1] J. Refonaa, M. Lakshmi and V. Vivek, "Analysis and prediction of natural disaster using spatial data mining technique", Proceedings of the Circuit, Power and Computing Technologies (ICCPCT), Nagercoil, India, **(2015)**.

[2] Z. Lijun, Q. Yaochen, Z. Jinping and L. Chaojun, "Spatial differentiation of urban carbon emissions — An exploratory spatial data analysis in Beijing", Proceedings of 2013 21st International Conference on Geoinformatics, Kaifeng, China, **(2013)**.

[3] Z. Chunqiu, L. Hongtai, W. Huibo, G. Yudong, L. Chang, Z. Mingjin, C. Shuching and R. Naphtali, "Optimizing online spatial data analysis with sequential query patterns", Proceedings of Information Reuse and Integration (IRI), 2014 IEEE 15th International Conference, Redwood City, USA, **(2014)**.
[4] H. Ruizhu and X. Weijia, "Performance evaluation of enabling logistic regression for big data with R", Proceedings of Big Data (Big Data), 2015 IEEE International Conference, Santa Clara, USA, **(2015)**.
[5] Y. Feng, W. Zhijian, Z. Fachao, W. Yapu and Z. Yuanchao, "Cloud-Based Big Data Mining & Analyzing Services Platform Integrating R", Proceedings of Advanced Cloud and Big Data (CBD), 2013 International Conference, Nanjing, China, **(2013)**.
[6] R. Pandey, N. Srivastava and S. Fatima, "Extending R Boxplot Analysis to Big Data in Education", Proceedings of Communication Systems and Network Technologies (CSNT), 2015 Fifth International Conference, Gwalior, India, **(2015)**.
[7] H. Yicheng, L. Xingtu, C. Xing and G. Wenzhong, "Towards Model Based Approach to Hadoop Deployment and Configuration", Proceedings of 2015 12th Web Information System and Application Conference (WISA), Jinan, China, **(2015)**.
[8] S. Narayan, S. Bailey and A. Daga, "Hadoop Acceleration in an OpenFlow-Based Cluster", Proceedings of High Performance Computing, Networking, Storage and Analysis (SCC), 2012 SC Companion, Salt Lake City, USA, **(2012)**.
[9] V. Ubarhande, A. M. Popescu and H. G. Vélez, "Novel Data-Distribution Technique for Hadoop in Heterogeneous Cloud Environments", Proceedings Complex, Intelligent, and Software Intensive Systems (CISIS), 2015 Ninth International Conference on, Blumenau, Brazil, **(2015)**.
[10] S. Saba, M. Grant, S. Pengju, W. Jun and B. John, "Diagnostic Ultrasound: Principles and Instruments", Journal of IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 1, **(2012)**, pp. 158-169.
[11] S. T. Ahmed and D. Loguinov, "On the performance of MapReduce: A stochastic approach", Proceedings of Big Data (Big Data), 2014 IEEE International Conference, Washington, USA, **(2014)**.
[12] B. W. Oh and K. J. Han, "H-SCAN: A Hash-based Spatial Clustering Algorithm for Knowledge Extraction", Journal of KISS (B): Software and Applications, vol. 26, no. 7, **(1999)**, pp. 857-869.

Authors

Minwuk Jeon, He received his B.S degree in computer engineering from the Kumoh National Institute of Technology, Korea in 2014. He is in course of M.S degree. His research interests include Big Data Processing and Mobile Software.

Byoung-Woo Oh, He received his B.S., M.S. and Ph.D degree in computer engineering from the Kon-Kuk University, Korea in 1993, 1995 and 1999 respectively. From 1999 to 2014, he was a senior researcher in the Electronics and Telecommunications Research Institute (ETRI), Korea. He is a Professor in Department of Computer Engineering, Kumoh National Institute of Technology, Korea since 2004. His research interests include spatial database, mobile software, parallel programming, Geographic Information System (GIS) and Location Based Service.