THE DESIGN OF LARGE SCALE DATA MANAGEMENT FOR SPATIAL ANALYSIS ON MOBILE PHONE DATASET

Apichon Witayangkurn[*1], Teerayut Horanont[2] and Ryosuke Shibasaki[3]
[1]PhD Candidate, Department of Civil Engineering, the University of Tokyo, 4-6-1, Komaba, Meguro-ku, Tokyo 153-8505, JAPAN; Tel: +81-3-5452-6412 Fax: +81-3-5452-6414;
E-mail: apichon@iis.u-tokyo.ac.jp

[2]Researcher, Institute of Industrial Science, the University of Tokyo, 4-6-1, Komaba, Meguro-ku, Tokyo 153-8505, JAPAN; Tel: +81-3-5452-6412 Fax: +81-3-5452-6414;
E-mail: teerayut@iis.u-tokyo.ac.jp

[3]Professor, Center for Spatial Information Science, the University of Tokyo, 5-1-5 Kashiwanoha, Kashiwa-shi, Chiba 277-8568, JAPAN; Tel: +81-4-7136-4291 Fax: +81-4-7136-4292;
E-mail: shiba@csis.u-toyko.ac.jp

KEY WORDS: GPS, Mobile Phone, Spatial Analysis, Cloud Computing, Hadoop

ABSTRACT: Mobile technology, especially mobile phone, is very popular nowadays. Increasing number of mobile users and availability of GPS-embedded mobile phones generate large amount of GPS trajectories. It is leading to have massive spatio-temporal dataset of people's trajectories. With accumulate large number of mobile phone data, we can mine and discover many useful knowledge and information such as human behavior, people mobility, and transportation planning. Favorite places, daily activity and life pattern are examples of the analysis results. However, how to handle such a large-scale dataset is a significant issue particularly in spatial analysis domain. A proper data management system is needed to allow smoothly processing in both scalability and performance on a whole large dataset rather than small or sampling dataset. In our research, we aimed to design a data management system for large scale mobile analysis based on three main features: big data, spatial support and data mining. Spatial database and cloud technique are considered in the design to handle large-scale data and support spatial query for spatio-temporal data analysis. As a core of the system, we used Cloud computing platform named Hadoop/Hive is an open source cloud computing software framework for data intensive and distributed application. Combining with spatial library called Java Topology Suite, Hadoop is able to perform spatial query and analysis. We evaluated the system by using a dataset of GPS trajectories of 1.5 million individual mobile phone users in Japan accumulated for one year. The total number was approximately 9.2 billion records.

1. INTRODUCTION

With the advancement of mobile phone and the increasing popularity of GPS embedded mobile devices, it is enable people to acquire their own location easily which can be used for various purposes such as location-based service application. Location information or geo-location is not only come from GPS but also from Wi-Fi and from cell tower which will be served when GPS has no signal. The analysis on mobile phone data has been researched for years by starting from using GPS log of mobile user (Ashbrook, 2003) and later using CDR in past few years (Huayong et al., 2010). GPS log represent geo-location of user in time constrain. Call Detail Records (CDR) are the data recorded each time a user initiated call and received call with mainly used for billing purpose; however, every CDR record has geo-location attached. However, in our research, we mainly focus on GPS data rather than CDR. For the past researches, they mostly involved with data mining techniques ranged in various topics such as extracting meaningful place of people (Zhou et al., 2007), understanding people moving pattern (Liao et al., 2005) transportation planning and etc. Nevertheless, those researches mainly focused on the algorithm with testing data rather than implementing real system with possible of real dataset which indeed, in the real world, data from mobile phone are very large and continuously increase. In our case, we used data of 1.5 million users in a year and size is 600GB. Hence, a large scale system is needed to handle in such case which should be scalable both in data storage and processing speed. Today, we are surrounded with a lot of data as seen in many social network systems. They have to handle such huge data generated and keeping more by users participating in social and finally go to terabytes and petabytes of data such as Google and Facebook. The question raised how to handle those data and provide response within concerned time. Cloud technology provides computation, software, data access and storage services that do not required to know physical location of the resources. It involved with dynamically scalable and virtualized resources. It normally expresses the following characteristic: Cost, Location dependence, Reliability, Scalability, Performance and Security (Yang et al., 2010). The firms like Google, Yahoo and Facebook are also used cloud technology as backend system to provide scalability (Lam, 2011).

Therefore, in this paper, we purposed a large scale data management using a cloud computing platform named Hadoop as a core system for mobile phone data analysis (Hadoop, 2008). We designed a comprehensive system with all necessary components such as data processing, data storage and virtualization with the aim to support three main features: big data, spatial support and data mining. We also emphasized on Hive, a data warehouse service built on top of Hadoop (Hive, 2008), to provide SQL-like query to users which is more familiar than MapReduce combining with several libraries such Java Topology Suite (JTS) and Java Machine Learning (Java-ML). With Java Topology Suite (JTS), a spatial function library, and User-Defined Function (UDF) on Hive, it allows Hadoop/Hive support spatial function. Java Machine Learning Library (Java-ML) allows Hive to process many data mining techniques. The experiments and examples of potential applications are also presented to prove and ensure performance of the system.

2. MOBILE PHONE DATA

In general, mobile phone data are obtained from two sources including user-based and operator-based. User-based data is the most popular method and used in many researches since it is easy to implement and acquire location data. It usually used an application installed in mobile device of volunteers to collect position data and archive data in device itself or report to server. For operator-based, data are collected by operator using auto-gps function embedded in mobile device or an application installed in the devices (Kiukkoneny et al., 2010). Since it has been done by operator, number of participants is much more than user-based; however, it is extremely difficult to convince operator participating in the research and also there have involved with many user privacy concerns. Regarding the accuracy of position data, it diverges based on sources of the location which can be calculated from GPS, Wi-Fi and Mobile cell tower location. GPS usually has the best accuracy among others. For data rate, it normally range from 1/sec to 1/5minutes depended on data collection methods and also battery consumption of the observed devices. Nevertheless, high data rate is more preferable since it is useful for detecting discovering mobility of people at detail level such daily trip and transportation mode (Leon et al., 2011). Our experimental dataset is presented in next subsection.

2.1 Mobile Dataset (Test dataset)

The dataset is collected anonymously from about 1.5 million real mobile phone users in Japan over one year period. A total record was 9,201 million and about 600GB in size. CSV was used as data format and the data are kept separating one file per day which is about 25 million records per day. Data collection has been done by mobile operator and private company under an agreement with mobile users. Positioning function including GPS is activated on mobile phone to send current location data to server in every 5 minutes; however, it is depended on several factors such as no signal and battery preserved function. For example, location sending function will be automatic turned off if there is no movement detected. That result in only 37 points collected a day in average. This was one drawback because location data might not be obtained at all-time and missing some important location data. Nevertheless, with this collecting method, it allows collecting large number of user data which are very useful for many cases. In addition, geo-locations were acquired and calculated from GPS, Wi-Fi and Mobile cell tower. The accuracy of position was defined in three levels range from 1 to 3 (the highest accuracy). With the accuracy defined by operator, 39% of points have accuracy in the 500 meters range. Another 23% have 100 meters range accuracy and the remaining 38% of points have accuracy in 10 meters. Figure 1 depicts the distribution of GPS data and number of points in difference cities (top 10). Density of point cloud was directly reflected with the size of the city. For example, Tokyo, the capital city of Japan, had the highest density of point data. Considering the privacy issues, we used these datasets anonymously.

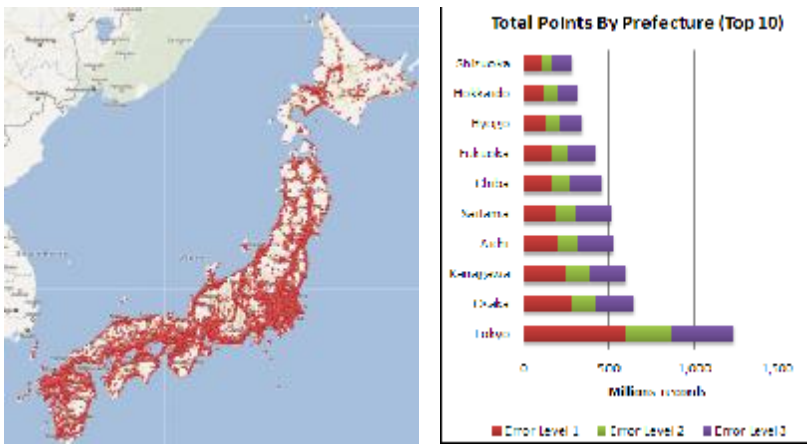

Figure 1: Data distribution in Japan

3. SYSTEM OVERVIEW

In this paper, we focus on designing as well as proofing a concept of large scale data management for mobile phone data which also include spatial processing to enable mobility analysis of mobile phone data. The system mainly takes into account of very large data issue lying on the limitation of data storage and poor performance on processing spatial data. It results in the following key features: large-scale support, scalable both in storage and processing speed, fast spatial processing, low-cost and easy use for data mining. Hadoop, a cloud computing platform, is used as a core of the system to enable large scale data support because it developed for data intensive and distributed application. It is also very famous and used by many well-known firms such as Google and Facebook (Lam, 2011). In the design, it covered all necessary components including data sources, data receiving service, data processing, persistent storage, spatial analysis, spatial user-interface and interface for further processing on other application as shown in Figure 2. In this section, the system architecture as well as detail description of each components of the large scale data management system are expressed and followed by the key technologies used for building the system.

3.1 System Components

This section describes each important components of the system as a diagram shown in Figure 2.

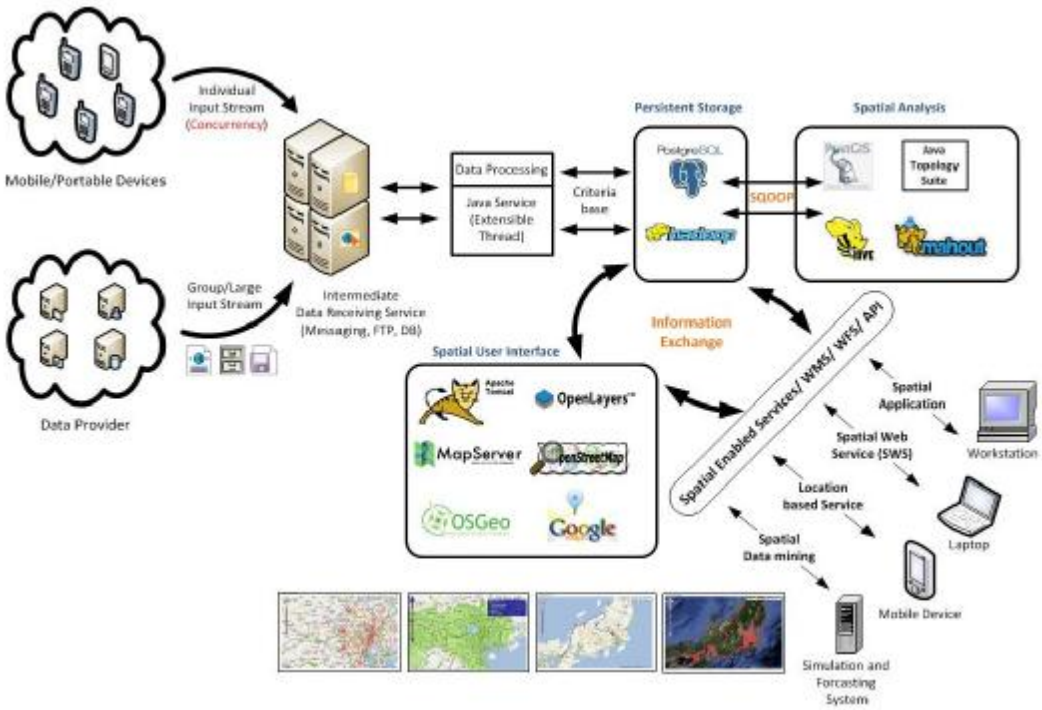

Figure 2: System Overview Diagram

Data Sources: The data are able to be collected in two ways. One is from mobile or portable device, which users are required to install specific software on the devices. The position data of an individual device are sent directly to intermediate data receiving service. With this technique, it is allow doing real-time analysis because data are fed to server all the time. Another way is collecting from data provider or mobile operator. In this case, mobile operator collect position data of mobile device and store in its server and after sometime, data will be export or transfer to intermediate data receiving service. The data are mostly come in CSV file format with compression.

Data Receiving Service: This is an intermediate service used for receiving data from multiple data sources and pass data to data processing service. The data did not persistently stored here but deleted after the processing finished. It provides multiple channels for retrieving data such as Messaging service, FTP and Database. As for the messaging service, we used JMS messaging service named ActiveMQ, open source messaging under apache project. It provides concurrent data pushing and also an asynchronous communication protocol meant that sender and receiver do not need to interact at same time as usual TCP connection. It also supports several of protocol such as TCP, UDP, SSL, REST, and HTTP allowed service to operate and bypass security in any network system. Moreover, it supports load balance and cluster which enhance large scale support of the system. For FTP and Database, we deploy standard open source software such as open-ftp and PostgreSQL since it does not need to involve much on concurrency issue.

Data Processing: It handles processing data before storing in the persistent storage. It involves several tasks including data cleaning, Data interpolation, Data sampling and Data conversion. Based on our experiences, many of data come with errors such as invalid format and invalid geo-location. Besides, data may arrive in difference format and time interval, which it need some conversion process before proceed to further steps. It is a java service with Multithread to increase performance and recommend running on the same machine as the data receiving service to reduce data transferring time.

Persistent Storage: In order to store large dataset with scalability feature, we use Hadoop, a cloud computing platform together with PostgreSQL. Hadoop is able to store large number of data and also fast processing since it is a combination of multiple computer nodes. Actual data are spitted into small files and store in different nodes. When, data processing is needed, job task is submitted to the particular nodes with related data are stored. This reduces data transferring cost since data does not move among cluster nodes but job task assign to the right node. More detail on Hadoop is described in Section 3.2. The main role of Hadoop is to archive large data including raw data and process the whole data; however, it did not play well in providing service to real-time application as Database did. PostgreSQL therefore take into account that issue. Once Hadoop finished processing data, the processed data are transferred to the database using Sqoop, a tool designed for efficiently transferring bulk data between Apache Hadoop and structured datastores such as relational databases.

Spatial Analysis: It is a combination of multiple libraries to enable effectively analysis on spatial data and data mining such as Hive, PostGIS and Java Topology Suite (JTS). The main part is Hive which is a data warehousing package providing SQL-like language called HiveQL for querying and processing data in Hadoop. Java Topology Suite (JTS) is a spatial function library developed using native java language. Enhanced with JTS, Hive is able to do spatial processing. PostGIS is a spatial database extension for PostgreSQL and we used it to serve spatial function for real-time application directly connected to PostgreSQL. In addition, we used Java Machine Learning Library (Java-ML) allowing Hive to process many data mining techniques such as classification and clustering.

Spatial User Interface (Virtualization): In order to provide interactive service to users, web-based interfaces are utilized. Apache Tomcat is used as web server together with map service such as Open Street Map and Google Map to provide interactive map. Also map libraries including OpenLayer is employed for interacting map services. Off-the-shelf software such as ArcGIS is also used for data virtualization.

3.2 Key Technologies

Hadoop: It is an open source software framework for data intensive and distributed application. It designed to work on commodity hardware meaning it does not require high performance server-type hardware but still preserve tolerance feature and also able to scale up with cost effective manner. Increasing system performance and storage can be simply done by adding new node without code modification required. The key distinctions of Hadoop are consisted of Accessible, Robust, Scalable and Simple. As for Accessible, Hadoop can runs on large clusters of commodity machines or on cloud computing services such as Amazon's Elastic Compute Cloud (EC2). For Robustness, because it is intended to run on commodity hardware, Hadoop is architected with the assumption of frequent hardware malfunctions. It can gracefully handle most such failures. With Scalable, Hadoop scales linearly to handle larger data by adding more nodes to the cluster. The last one, Simple, Hadoop allows users to quickly write efficient parallel code. There are many services and framework under Hadoop umbrella; however, in this research, we focused on Hadoop Distributed File System (HDFS) and Hive (Hadoop, 2008). To setup and use Hadoop for full operation mode, it required to run 5 components: NameNode, DataNodes, Secondary NameNode (SNN), JobTracker and TaskTrackers. NameNode is the bookkeeper of HDFS; it keeps track of how your files are broken down into file blocks, which nodes store those blocks, and the overall health of the distributed filesystem. DataNodes are the workhorses of the filesystem. They store and retrieve blocks when they are told to (by clients or the namenode), and they report back to the namenode periodically with lists of blocks that they are storing. Secondary NameNode (SNN) is an assistant daemon for monitoring the state of the cluster HDFS and the SNN help snapshots NameNode to help minimize the downtime and loss of data. JobTracker is the liaison between your application and Hadoop. Once you submit your code to your cluster, the JobTracker determines the execution plan by determining which files to process, assigns nodes to different tasks, and monitors all tasks as they are running. TaskTrackers is responsible for executing the individual tasks that the JobTracker assigns and manage the execution of individual tasks on each slave node.

Hive: Hive is a data warehousing package built on top of Hadoop (Hive, 2008). It target users who are familiar and comfortable with SQL to do ad hoc quires, summarization and data analysis. Web GUI and JDBC are provided for interacting with Hive by issuing queries in a SQL-like language called HiveQL. All data are stored in HDFS and Hive handles write and read data from HDFS. Hadoop/Hive do not support spatial query; however, Hive allow developers to create User-Defined Function (UDF) that could be any function based on user requirement and since Hive is natively developed by using Java language, Java is also used for UDF developing. Spatial library (JTS) described in

Section 3.1 is employed to make spatial function with UDF. With combination of those components, Hive is able to do spatial processing with fully utilized cloud computing platform (Wityangkurn et al., 2012).

3.3 Hadoop Cluster (Testing Platform)

Our Hadoop cluster is depicted in Figure 3. It consisted of five computers with the same specification that is 8-Cores Xeon 2.6GHz, 8GB memory, and two of 2TB disk to increase I/O performance. The operating system is CentOS 6.0 64-bit. Gigabit switch was used for communication among cluster nodes. One computer was for master node and other four computers were for task processing acting as DataNodes and TaskNodes. Hive service is run on the same machine with NameNode. In total, the cluster has 32 cores, 32GB memory and 16TB storages. However, we set number of concurrent tasks to 7 tasks for each node because one core was reserved for other process. Hence, it can have up to 28 tasks running at the same time (4*7 Cores). The version of Hadoop was 0.20.2, and the version of Hive was 0.8.0. The version of JTS was 1.12. For application server, the same specification of computer as Hadoop is used as well as same operating system. PostgreSQL 9.0.6 with PostGIS 1.5.3 was installed in the system to serve database service with spatial query.

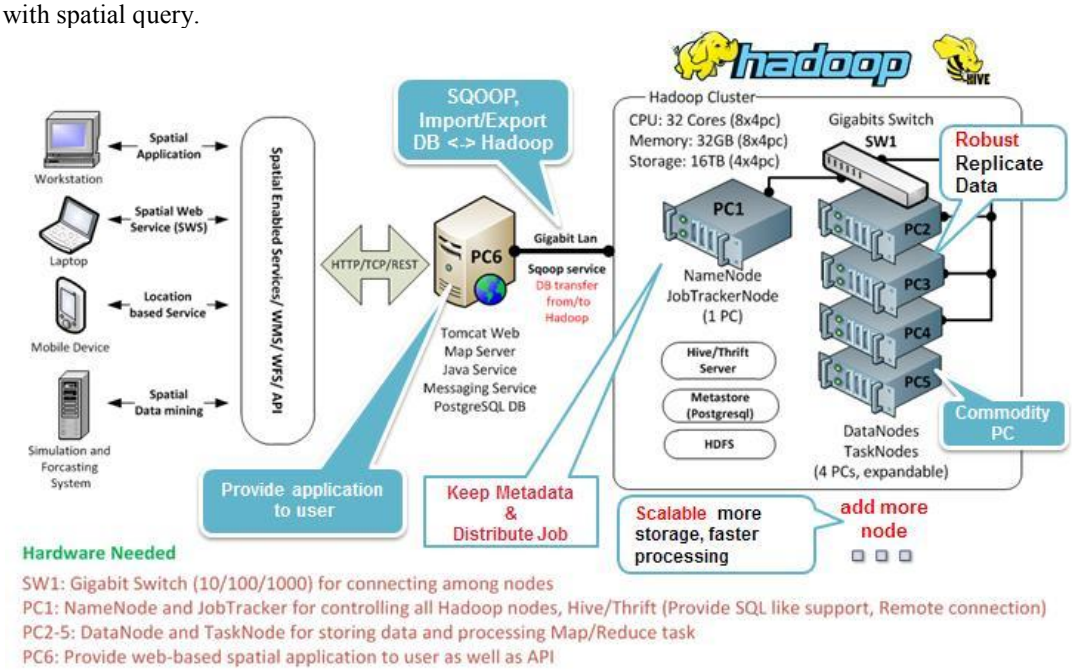

Figure 3: Testing platform for Hadoop Cluster

4. PERFORMANCE

With the result from our previous, we did a performance comparison of several techniques for processing large scale spatial data including spatial-enable database, spatial library-based application and cloud computing platform (Hadoop/Hive). We separated evaluation into two parts: preparation time and spatial computing time since some approaches required a lot of time for importing and loading data to system until they were ready for spatial query. Spatial computing time was the time used for executing one spatial query and for this experiment, we used "Within" function that was used for finding point geometry in polygon. For dataset, we first started from one day data stored in one CSV files, one line was one point. After that, we increased the number of files to measure the performance and effect of large scale data size. The result showed that Hadoop with Hive outperformed other techniques as shown in Figure 4. Hive could process data within a minute comparing with PostgreSQL; it took 1,269 minutes or 21 hours as stated in Table 2. In addition, we increased number of data by processing on one day (22million), two days (44 million) and five days (100 million). Hive method also beat other methods significantly. Lastly, we processed the whole dataset with a job to find prefecture and city that each GPS point was located. Total processing time was about 17 hours, and 9,201 million records are processed (Witayangkurn et al., 2012).

Table 1. Preparation time of all methods for one day data

Method	Task	Time (Sec)
Database (PostgreSQL)	Import Data, Create Geometry, Create Spatial Index	12,073 (3.3 hrs.)
Spatial library-based application	-	0
Cloud platform (Hadoop/Hive)	Import Data, Convert to Binary File	78

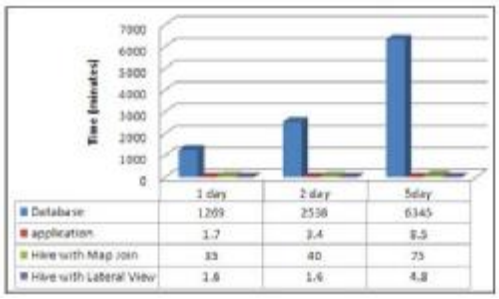

Table 2. Processing time on spatial query for one day data

Method	No. of processed records per second	Total Time (Mins)
Database (PostgreSQL)	227	1269
Spatial library-based application	171,490	1.7
Hive using Map Join (6 tasks, Normal)	8,148	35
Hive using Map Join (22 tasks)	21,361	13.5
Hive using Lateral View (6 tasks, Normal)	173,205	1.6
Hive using Lateral View (22 tasks)	288,675	1

*Remark: 1 day data = 20 million records

Figure 4: Performance comparison result

5. AN EXAMPLE OF POTENTIAL APPLICATIONS

5.1 Area-based Population Density Estimation

The process, shown in Figure 5, starts from loading raw data, which is GPS log, to Hadoop via Hive command. With Hive combined with spatial function from Wityangkurn et al, geo-location such as City, Ku and Grid cell is able to be associated with GPS log. After that, we calculate numbers of people in each area based on minute interval. One day has 1440 minutes. So far, we have a number of estimated people at different areas and minutes. We, therefore, export results to the ordinary database in this case is PostgreSQL with PostGIS so that it can be used for virtualization on Map and for further processing on GIS software such as ArcGIS.

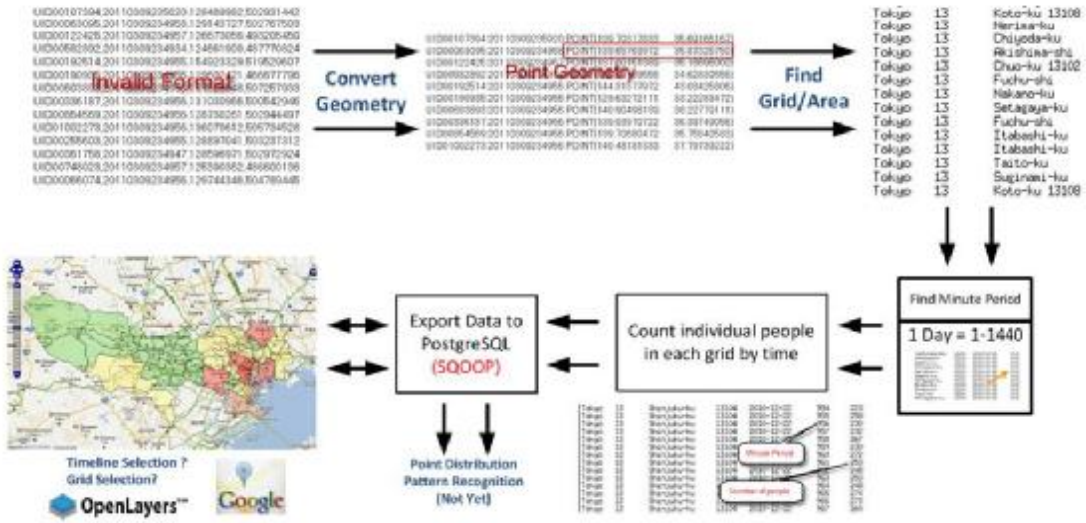

Figure 5: An example of processing steps

The Figure 6 depicted the result from processing data in Tokyo with 500*500 meter Grids. It is shown that in the morning time, there is the high density of people in the red color grid cells and when overlay with real map, those red areas are Shinjuku, Shibuya and Tokyo Station reflected to the real situation that many people go those train stations at morning time. The results showed that, we were able to find relationship between real population in each area and estimated population from mobile gps data which will gain a lot of possibility to further analysis on people mobility.

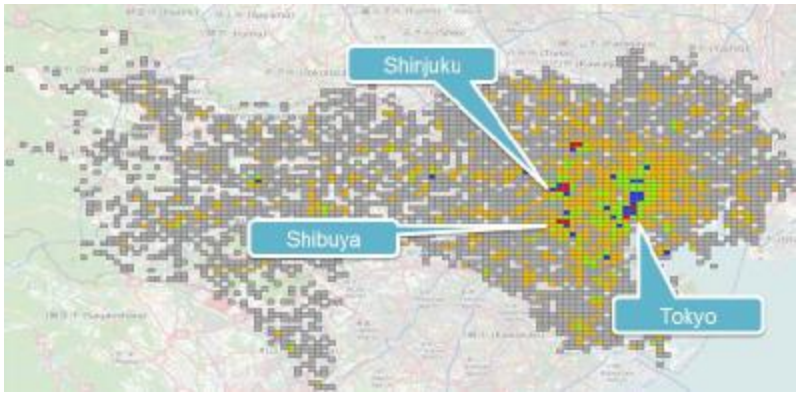

Figure 6: 500m-grid in Tokyo at morning time

6. CONCLUSIONS

The objective of this research is to design large scale data management for spatial analysis on mobile phone data. We have presented overall system architecture including the detail of each main component such as data processing, persistent storage, spatial analysis and virtualization. We have also introduced Hadoop, a cloud computing platform, together with Hive, a data warehouse module built on Hadoop, to handle large scale issue such as storage and processing speed. Testing platform of Hadoop cluster was built to prove the design and evaluate with very large scale real-world mobile phone dataset. A combination of Hive and Java Topology Suite (JTS) allowed Hadoop/Hive to support spatial analysis with significant improvement comparing with other traditional methods as presented in performance evaluation. Moreover, we demonstrated an example of potential application could make by this system. The result of area-based population density estimation showed meaningful and reasonable output of dynamic population at specific location and time. In combination of the design and testing platform as well as performance evaluation and example application, it proved that the system is able and suitable for processing large scale mobile phone data.

7. ACKNOWLEDGMENTS

The work described in this paper was conducted at Shibasaki Laboratory with an agreement from Zenrin Data Com to use mobile phone dataset of personal navigation service users for the research. This work was supported by GRENE (Environmental Information) project of MEXT (Ministry of Education, Culture, Sports, Science and Technology).

8. REFERENCES

References from Journals:
Zhou, C., Frankowski, D., Ludford, P., Shekhar, S., Terveen, L., 2007. Discovering Personally Meaningful Places: An Interactive Clustering Approach. In ACM Trans. on Information Systems, vol. 25(3).

References from Books:
Lam, C., 2011. Hadoop in Action. Manning, Connecticut, pp. 17-19.

References from Other Literature:
Ashbrook, D., and Starner, T., 2003. Using GPS to learn significant locations and predict movement across multiple users. Personal and Ubiquitous Computing 7(5), pp. 275-286.

Leon S., Ouri W., Philip S. Y., Bo Xu., 2011. Transportation Mode Detection using Mobile Phones and GIS Information. ACM SIGSPATIAL, Chicago, USA.

Liao, L., et al., 2005. Building Personal Map from GPS Data. In Proc. of IJCAI MOO05, Springer Press: pp. 249-265.

Hive Project: http://hive.apache.org, 2008.

Huayong W., Francesco C., Giusy Di L., and Carlo R., 2010. Transportation mode inference from anonymized and aggregated mobile phone call detail records. In Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems, Washington, DC, USA.

Kiukkoneny N., Blom J., Dousse O., Gatica-Perez D., and Laurila J., 2010. Towards Rich Mobile Phone Datasets: Lausanne Data Collection Campaign. In ICPS.

The Hadoop Project, http://hadoop.apache.org, 2008.

Wityangkurn, A., et al., 2012. Performance Comparisons of Spatial Data Processing Techniques for a Large Scale Mobile Phone. In Proc. of 3rd Com.Geo.

Xiaoqiang, Y., Yuejin D., 2010. Exploration of cloud computing technologies for geographic information. In: proceeding of The 18th International Conference on Geoinformatics, Beijing, pp. 1-5.

Yang, J., Wu, S., 2010. Studies on Application of Cloud Computing Techniques in GIS. In: proceeding of The International Conference on Geoscience and Remote Sensing, China, Vol.2, pp. 492-495.