Study on LBS for Characterization and Analysis of Big Data Benchmarks

AFTAB AHMED CHANDIO*, FAN ZHANG*, AND TAYAB DIN MEMON**

RECEIVED ON 07.02.2014 ACCEPTED ON 17.10.2014

ABSTRACT

In the past few years, most organizations are gradually diverting their applications and services to Cloud. This is because Cloud paradigm enables (a) on-demand accessed and (b) large data processing for their applications and users on Internet anywhere in the world. The rapid growth of urbanization in developed and developing countries leads to a new emerging concept called Urban Computing, one of the application domains that is rapidly deployed to the Cloud. More precisely, in the concept of Urban Computing, sensors, vehicles, devices, buildings, and roads are used as a component to probe city dynamics. Their data representation is widely available including GPS traces of vehicles. However, their applications are more towards data processing and storage hungry, which is due to their data increment in large volume starting from few dozen of TB (Tera Bytes) to thousands of PT (Peta Bytes) (i.e. Big Data). To increase the development and the assessment of the applications such as LBS (Location Based Services), a benchmark of Big Data is urgently needed. This research is a novel research on LBS to characterize and analyze the Big Data benchmarks. We focus on map-matching, which is being used as pre-processing step in many LBS applications. This paper also describes current status of Big Data benchmarks and our future direction.

Key Words: **Location Based Services, Map-Matching, Big Data Benchmark, Characterization and Analysis, GPS Data.**

1. INTRODUCTION

In recent years, large number of organizations are gradually diverting their applications and services to the Cloud environment for their data processing and storage needs. In the current survey of GigaSpaces [1], it is notified that only 20% of the IT professionals surveyed indicated that their company had no plan to move their Big Data to Cloud. It is proven that most of the organizations which deal with Big Data had adopted Cloud environment. This is because Cloud environment enables on-demand access and large data processing for their applications and users on Internet anywhere in the world [2]. In the Cloud environment, the large-volume data processing and storing facilities are provided by computer farms (i.e. a collection of computers). Currently, many Cloud RPs (Resource Providers) offer thousands of computational

* Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China
** Associate Professor, Department of Electronic Engineering, Mehran University of Engineering Technology, Jamshoro

nodes, and variety of services to facilitate end-users, such as Google, IBM, and Amazon.

Urban Computing is also one of the application domains, which has been rapidly adopted Cloud, such as, Urban Intelligent Transportation [3], Traffic Flow Forecasting [4] and Agent-based Urban Transportation Systems [5], etc. Cloud computing helps the above systems with the computing resources and large portion of storage required to define traffic strategy agent and to use mass transport data effectively [5]. In the concept of Urban Computing, sensors, vehicles, devices, buildings, and roads in the urban areas are considered as a component to analyze city dynamics.

In the past few years, a dramatic increase of GPS-embedded vehicles, handhelds, PDAs, and Smart phone devices has been noticed. GPS traces of vehicles are one of their data representations, and their data is periodically generated via GPS devices that store geo-location information of vehicles. An increasing usage of these devices produces a collection of huge amount of GPS trajectories. This results in GPS data increment in large volume ranging from TB to thousands of PT, called Big Data. The information from GPS data can be used in many LBS, such as traffic flow analysis, route planner, geographical social network, and hot route finder to achieve better QoS (Quality of Service) [6-7]. Specifically, GPS trajectory is comprised of a sequence of points which captures the information of longitude and latitude position, and timestamp of the moving object.

To increase development and assessment of the LBS applications, a Big Data benchmark for system management is urgently needed. However, it is big challenge for benchmark providers to assess and examine the Big Data and its application due to its large -volume, -variety and -velocity (i.e. Gartner [8] defined as 3Vs). To the best of our knowledge there are no any Big Data benchmarks to provide the facility for characterizing and analyzing the LBS applications for taxis. The field of characterization and analysis of Big Data and LBS applications need more study. In this paper, we identify the need for a yard-stick to measure the performance of Big Data systems which process the GPS-based data and employ this on LBSs.

This research aims to characterize and analyze the GPS traced workloads (i.e. Big Data) and LBS services (i.e. Big Data benchmark). A characterization and analysis of workload helps in understanding the overall behavior of the system highlighting load and performance of the system [9, 23]. For analyzing the GPS-based workloads and its services, we use Map-Matching algorithm for one of the LBS services, which aligns the observed GPS positions with the road network on a given digital map. Many trajectory-based applications e.g. moving objects management and driving directions use Map-Matching process as a fundamental pre-processing step. In this paper, we select ST-Matching [6] algorithm. We then employ the real world GPS-based workload, where the traces are taken from Shenzhen taxicabs. These traces generated 14 thousands trajectories in 24 hours per day, and their size is about 3.755 GB/Day. From the experimental work we found that ST-Matching algorithm increases running time when a trajectory has more number of GPS points, because the most expensive part of the ST-Matching algorithm is the shortest path query processing.

The rest of the paper is organized as follows. Section 2 briefs the introduction of Big Data. While in Section 3, we present the traditional and current benchmarks. Then in Section 4, we describe one of the LBS Map-Matching algorithm, which will be part of benchmark for analyzing and characterizing the GPS-based Big Data. Finally, Section 5 presents the results and discussion, and Section 6 concludes the paper and highlights future research directions.

2. INTRODUCTION OF BIG DATA

A huge and complex structured or unstructured dataset that is difficult to compute using basic and traditional DBMS tools, is called Big Data. Currently, a large number of organizations produce such kind of datasets, for example web logs, call detail records, social media, financial transactions and Internet text, sensor and machine generated data. The size of these datasets starts from TB, which is forced to make conventional data management tools failure for their processing. For instance, Wal-Mart produces one million transactions after every hour, which is estimated more than 2.5 petabytes of data [10].

Talend had reported in the summer 2012 on Big Data adoption survey [10] where a total of 231 professionals

were involved in the delivery of data solutions for their company. The survey revealed which kind of common applications derived Big Data needs. This is shown in Fig. 1.

2.1 Big Data Processing in Cloud Environment

New techniques based on Hadoop (Apache Hadoop, http://hadoop.apache.org), are emerging to analyze Big Data through MPP (Massively Parallel-Processing) technique. Hadoop framework is an Apache project that provides a library for distributed processing of Big Data using MapReduce programming model [11]. This model allows applications which may consist on multiple jobs. Each job is divided into several map- and reduce-tasks. After input, the map-task produces intermediate key-value pairs, which are distributed on nodes. Then an intermediate data is processed by reduce-tasks to generate final output key-value pairs. During this process, the input and output data are stored in HDFS (Hadoop Distributed File System). HDFS creates multiple copies of the data as data blocks for distributing them on nodes.

MapReduce only provides the batch processing and does not handle the real-time processing on the Cloud, thus other projects such as STORM and Apache Hama would complement MapReduce limitations. STORM (STORM, http://storm-project.net/) is a free distributed system which provides open-source and real-time computation environment. In STORM, a reliable process such as unbounded streams of data is processed into real-time processing.

On the other hand, Hama (Apache Hama Project, http://hama.apache.org) is a project introduced by Apache. This framework provides a pure BSP (Bulk Synchronous Parallel) computing environment on the top of HDFS. BSP performs massive scientific computations such as graph, network, and matrix algorithms.

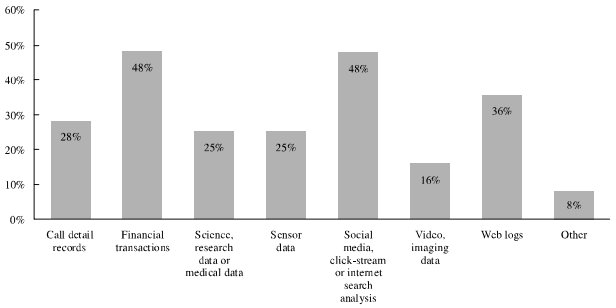

FIG. 1. WHICH APPLICATIONS ARE DRIVING BIG DATA NEEDS

3. BENCHMARK

Benchmarking is a process to evaluate the performance characteristics of a system including hardware, software and system environment. The process is to run a computer program, a set of programs or other operations in order to assess the system performance, which is called benchmark. These programs measure performance of system under a typical workload (i.e. workload may be either real-world or synthetic workload) with fixed configuration by using specific performance units called performance metrics, such as query throughput per time, MIPS (Millions of Instructions Per Seconds), data processed per time, etc.

There are many benchmarking organizations providing the SUT (System Under Test) as well CUT (Component Under Test) services to evaluate the average performance of system including application-, web-, and transactional database-servers, e-commerce applications, I/O disk performance, CPU performance, etc. TPC is one of them providing different workloads for different real-world applications, such as TPC-C for OLTP, TPC-H for OLAP, TPC-W for e-commerce applications. TPC is providing benchmarking in two basic areas including business operation area (TPC-A, TPC-C and TPC-W) and decision support area (TPC-D, TPC-H and TCP-R). It is known that all the current benchmarks are to evaluate deployed system under fixed configuration such as static hardware or software, so therefore their metrics are also static non-changing. Few major traditional benchmarks are discussed as follows:

3.1 Traditional Benchmarks

TPC-C (TPC-C Benchmark, http://www.tpc.org/tpcc): TPC-C is introduced by TPC (Transaction Processing Performance Council), TPC-C provides multiple transaction processing and more complex databases with different configuration of hardware and software to analyze their behavior. This benchmark measures the performance of OLTP by using tpmC (transaction-per-minute) metric. TPC-C benchmark provides five types of OLTP transactions (i.e. new order, delivery, payment, stock-level, and order-status) along nine tables, and their transaction includes insert, update, delete, and abort together with primary and secondary key access. Each type of transaction has response time ≤ 5 second except stock-level that is ≤ 20 second response time.

TPC-H (TPC-H Benchmark, http://www.tpc.org/tpch/)**:** It is also part of TPC, specially used for decision support benchmarking that consists of complete suite of business oriented ad-hoc queries and concurrent data modifications. Large business enterprises always need to access queries for data populating from database. To examine the large amount of data and to execute the query with ability of decision can be illustrated by TPC-H benchmark. This benchmark measures the performance by Query-per-hour-performance (QphH@Size) metric.

TPC-W (TPC-W Benchmark, http://www.tpc.org/tpcw/)**:** This benchmark is transactional benchmark used for web e-commerce working on the RBE (Remote Business Emulator) architecture. Most of the workloads on web including multiple on-line session, dynamic page generation, transaction integrity, and consistent web objects are measured by TPC-W benchmark. The number of WIPS (Web Interaction Processed Per Second) is the primary metric for measuring performance of shopping, while WIPSb and WIPSo both are used for measuring the performance of browsing and ordering, respectively.

SPEC (Standard Performance Evolution Corporation Benchmark suite, http://www.spec.org/)**:** SPEC was proposed in Oct.1988 by Apollo, HP, MIPS computer systems, and Sun Microsystems. The goal of SPEC is to provide accurate benchmark to measure the performance of computer system and to educate the consumer about vendor's product. SPEC is complete suit for performance of all major components in computer system such as hardware (CPU, floating point unit, I/O, graphics, network accelerator, peripherals and memory system) and applications (operating system, algorithms, libraries, compilers and application software).

3.2 Big Data Benchmarks

A program or a set of programs that process a Big Data in order to evaluate the system performance is called Big Data benchmark. It is highlighted into GigaSpaces survey [1] that majority of organizations viewed their Big Data processing as "mission-critical". Currently, very little number of benchmarks is present to evaluate the Big Data. Major Big Data benchmarks are: CloudRank-D, HiBench, TPC-DS, GridMix2-3, Hive Benchmark, PigMix, WL Suite, YCSB, and CloudSuite.

CloudRank-D (The CloudRank-D benchmark suite, http://prof.ict.ac.cn/CloudRank/)**:** CloudRank-D [11] is a benchmark suite for data processing to benchmark and rank private cloud computing system that are designed for running Big Data applications. **(a)** Data processed per second and **(b)** data processed per Joule are two complementary metrics used for evaluating the performance of cloud system. CloudRank-D benchmark suite comprises a set of 13 representative data analysis tools, including basic operations for data analysis, classification, clustering, association rule mining applications, and data warehouse operations. All the programs are implemented and used in Hadoop environment for distributed processing using MapReduce programming model. Experimental work has been tested on not more than 40 GB datasets with CloudRank-D benchmark in [11].

HiBench: HiBench [12] is a benchmark suite for evaluating the performance of Hadoop and implementation of MapReduce. The performance is assessed in terms of **(a)** throughput: the number of task completed per minute, **(b)** speed: job running time **(c)** system resource utilizations, **(d)** HDFS bandwidth, and **(e)** data access patterns. HiBench benchmark suite consists of a set of Hadoop programs including synthetic and real-world applications. The set of programs consists of eight workloads which is categorized into four categories. Seven of them are taken from their open source implementation while other one is an improved version. Their four categories include: **(a)** Micro Benchmark (Sort, WordCount, and TeraSort), **(b)** Web Search (PageRank and Nutch Indexing), **(c)** Machine Learning (K-means clustering and Bayesian Classification), and **(d)** HDFS Benchmark (EnhancedDFISO).

TPC-DS (April-2012) (TPC Decision Support benchmark (TPC-DS), http://www.tpc.org/tpcds/)**:** TPC-DS is the decision support benchmark offered by TPC (Transaction Processing Performance Council). This Benchmark provides an evaluation of the SUT (System Under Test's) performance and illustrates a decision support system, such as to examine large data, queries and data maintenance. An implementation of TPC-DS is used only in commercially available Database Management System software, and the queries executed through SQL interface. TPC-DS supports four classes of decision support queries, i.e.

adhoc queries, reporting queries, data mining queries, and iterative OLAP queries. All of these queries perform on the schema of sale business process of an organization, and a size of dataset is only scaled up to 100 TB. **(a)** Query throughput, **(b)** Price-Performance, and **(c)** System availability date are the three primary metrics used by TPC-DS benchmark for evaluating performance of the system.

YCSB (Yahoo! Cloud Serving Benchmark): [13] is framework uses PNUTS distributive database system to support yahoo's web applications and compares other cloud database systems.

CloudSuite (CloudSuite, http://parsa.epfl.ch/cloudsuite/)**:** CloudSuite [14] is a benchmark suite for scale-out applications. Currently, CloudSuite provides six different applications that have been selected based on their popularity in today's datacenters. A set of application includes Data Analytics, Data Serving (Cassandra and YCSB), Media Streaming, Software Testing, Web Search (Nutch search engine to benchmark the indexing process), and Web Serving (CloudStone to benchmark Web 2.0 applications).

From the above State-Of-The-Art of the benchmarks, we found that none of them has employed GPS workloads of vehicles. To the best of our knowledge there are no any benchmarks to provide the facility for analyzing and characterizing the applications of LBS, i.e. GPS-based data processing applications. The field of Big Data performance of GPS-based data and its applications needs more study and to employ a different methodology. We identify the need for a standard to measure the performance of Big Data systems to process the GPS-based data.

4. LOCATION BASED SERVICES

This section presents the Map-Matching service as one of the LBS service. This service is used in many applications of GPS based data.

4.1 Map Matching

Map-Matching is a process, which is needed to align the observed GPS positions with the road network on a given digital map. Many trajectory-based applications (e.g. moving objects management and driving directions) use map-matching process as a fundamental pre-processing step. This process is classified into three classes, i.e., local/incremental [15-16], global [6-7, 17-18], and statistical [19-20] method. In local/incremental method, the algorithms will try to find the local match of geometry where they consider the small portion of space that is close to the GPS point. This approach performs well when sampling frequency is very high, i.e. 2-5 seconds time interval between GPS points [15-16]. While in the global approach, the algorithms match the entire trajectory with road network. Unlike the local/incremental method, the global method can produce better result when low-sampling data rate is applied [17-18]. The map-matching algorithms that use statistical methods to match the GPS points with road network include Hidden Markov model, Bayesian classifies [19], cubic spline interpolation, and Kalman filter [20]. Most of the existing map-matching algorithms map GPS trajectories only with high-sampling-rate. We implemented ST-Matching 6, one of the global map-matching algorithms, which can deal low-level sampling-rate GPS data, i.e. 2-5 minutes time interval between GPS points. Before discussing the ST-Matching algorithm, few preliminary definitions for explaining map-matching problem statement are stated below.

Definition-1 (GPS Dataset): All GPS points are stored in GPS dataset $L=[p_1, p_2, p_3, ..., p_n]$. Each GPS point $p_i \in L$ contains the information including GPS position in latitude $p_1.lat$ and longitude $p_1.lon$, and timestamp $p_1.t$.

Definition-2 (GPS Trajectory): A taxi trip is called complete GPS trajectory T which has a sequence of GPS points, i.e. $T: p_1 \rightarrow p_2 \rightarrow ... p_n$, where $p_i \in L$ and $0 < p_{i+1}.t - p_i.t < \Delta T (1 \le i < n)$, ΔT time spent in a trip. We use ΔT sampling time rate of GPS trajectory with $\Delta T \ge 2$ min.

Definition-3 (Road Segment): It is directed edge e associated with unique id e.gid, road length e. l, travel speed e. v, starting point e.start, ending point e. end and intermediate points which describe the road polyline.

Definition-4 (Road Network): It is directed graph $G(V,E)$, where V are points intersecting to the road segments called vertices and E is set of road segments called edges.

Definition-5 (Path): A path P is list of connected road segments between two given vertices (V_i, V_j) in a road network G, i.e., $P: e_1 \rightarrow e_2 \ldots \rightarrow e_n$, where $e_1.start = V_i$, $e_n.end = V_j$, $e_k.end = e_{k+1}.start$, $1 \leq k < n$.

Map-Matching Problem Statement: Find the list of connected road segments between two given vertices in a road network, which must be a real path P of a given raw GPS trajectory T [6].

4.2 ST-Matching Algorithm for Map-Matching

ST-Matching [6] algorithm is a global map-matching algorithm to address the problem of sampling error. In particular there exists large amount of low-sampling-rate of GPS trajectories, such as, one point for every 2 minutes. For example, in Shenzhen most of the taxis (which are known to travel very frequently) are using GPS applications that enable sample rate reduction in order to save energy consumption and cost of communication. ST-Matching map-matching algorithm is comprised of basic four phases: **(a)** Candidate Preparation, **(b)** Spatial analysis, **(c)** Temporal Analysis, and **(d)** Result Matching. These phases and algorithms are well defined in [6].

In the candidate preparation phase, raw GPS trajectory is accepted from a user then an entire possible candidate points (point taken from candidate road segment) for each sampling point on the trajectory are retrieved. These candidate points c are available within given expanded distance between sampling points (i.e. 100 meters), and computed with line segment projections of p_i to the road segments e such that c = arg $\min_{\forall c_i \in e}$ dist (c_i, p), where dist (c_i, p) function returns the distance between sample point p and any candidate point c on road segment e. Finally, the problem becomes a selection of candidate sequence for sampling point sequence, i.e., $P: c_1^{j_1} \rightarrow c_2^{j_2} \rightarrow \ldots \rightarrow c_n^{j_n}$ with the best matches $T: p_1 \rightarrow p_2 \rightarrow \ldots \rightarrow p_n$.

The second phase evaluates the candidate points generated in previous phase and finds both geometric (observation probability) and topological (transmission probability) information of the road network. Observation probability defines likelihood method of candidate points for matching GPS sampling points based on distance between them dist (c_i^j, p). While transmission probability defines the likelihood method for true path from $p_{i-1} \rightarrow p_i$, which follows the shortest path from $c_{i-1} \rightarrow c_i$.

The temporal phase creates score of each candidate considering the average speed between two neighboring candidate points because the shortest path between candidate points denotes a list of road segments [6].

In the last phase, the results returned from all of the phases are able to generate a candidate graph $G'_T(V'_T, E'_T)$ for a trajectory. Each node V'_T in G'_T is a set of candidate points and associated with observation probability, and E'_T in G'_T is set of road segments in shortest path between two neighboring points and associated with two separate scores, i.e., spatial and temporal. This phase finds sequence of candidate points which matchs with true path for sampling points and the overall highest score of each neighboring candidate point.

5. EXPERIMENT AND RESULTS DISCUSSION

5.1 Dataset

In this experiment, we create our own simulation setup where we use two datasets, one for road network and second for taxi trajectory dataset. A road network dataset is used from Shenzhen city road system. Road network contains total 86335 vertices and 133397 road segments. Fig. 2 shows a snapshot of Shenzhen Road System.

We use real trajectories dataset which was collected from GPS-embedded taxis traveled in Shenzhen city road network. There are about 14 thousands taxi trajectories created in 24 hours per day in Shenzhen city. We have used few of trajectories in our experiment that are randomly selected. Each trajectory is diverged into spatial coverage (longitude and latitude), time interval, length, and number of road segments.

5.2 Setup

In the experimental work, we have created local setup on personal laptop. The simulation is based on our own simulator setup where we have implemented the above map-matching algorithm in Java [21] environment. Java provides full advantage of OOP (Object-Oriented Programming) technique. OOP is a software implementation centered on dynamic creation, storing, and manipulation and releasing of objects [22]. The environment setup also allows for connecting the Spatial Database tool (i.e. PostgreSQL, PostGis, pgRouting), where we store all datasets. We ran the program several times to get average running time to collect accurate result for the experiments. The full configuration of setup is explained in Table 1.

5.3 Result Comparison

For the comparison result, the running time is selected as a primary metric. We ran this program 20 times on four different trajectories which comprises of different number of GPS points. Each trajectory is tested with five different configuration setting, where we make change in number of candidates, as shown in Table 2. Here, we found that when we increased the candidate points for each GPS point in a trajectory with less number of GPS points, this does not have much effect on running time. On the other hand, it had rapidly increased the running time when a trajectory had more GPS points. In Fig. 3, it is clearly shown that a trajectory with 50 GPS points consume more time. (It must be noted that the running time has been taken on the aforementioned setup.) The reason of the increased running time is that the large number of shortest path queries had been processed. Accuracy of the algorithm is depicted in a snap shot of visualized picture in Fig. 4. A dark line shows a matched road, the light circles show its matched candidate points and the dark circle is the GPS point.

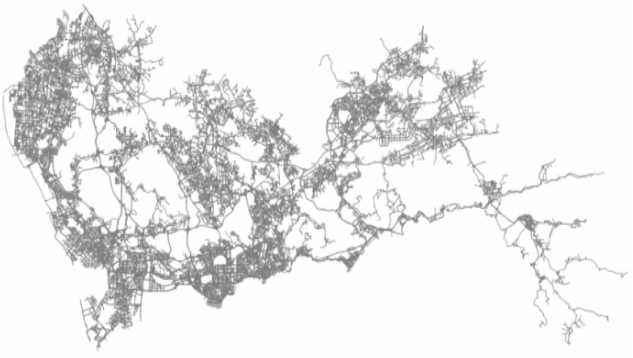

FIG. 2. A SNAPSHOT OF SHENZHEN ROAD SYSTEM

5.4 Discussion

In the above preliminary experimental work, we implement ST-Matching algorithm [6] which is suitable for low-sampling rate trajectories. In this algorithm, the most time consuming part is the shortest path query. We found that when a trajectory has more GPS points

TABLE 1. CONFIGURATION OF EXPREIMENTAL SETUP

Computer	Dell-2430M
CPU	Intel ® Core™ i5-@ 2.40 GHz
Memory	4 GB
Java	J2SE 1.7
DBMS	PostgreSQL 9.1
Spatial Tools	PostGis 2.0
Shortest Path Query Tool	pgRouting

TABLE 2. TESTED TRAJECTORIES WITH CANDIDATE CONFIGURATION

Trajectory	GPS-Points	Number of Candidates
1	10	2, 3, 4, 5, and 6
2	12	
3	22	
4	50	

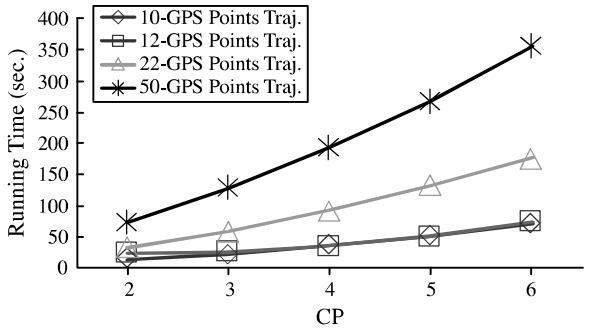

FIG. 3. RUNNING TIME METRIC

considering more candidate points the ST-Matching algorithm increases the running time under the aforementioned setup. However, we increased the number of candidate points that yields accurate result, and the trajectory could have more GPS points. Therefore, considering above factors a parallel processing and adaptive scheme is needed to boost up the running time. In the future work, we will analyze this algorithm implementing on cloud environment with huge number of trajectories, and then an adaptive map-matching strategy will be proposed which could efficiently handle the problem of shortest path queries in the map-matching process.

6. CONCLUSION

The rapid growth of urbanization in developed and developing countries leads to new emerging concept called Urban Computing, which is one of the application domains rapidly adapting to the Cloud environment. However, their applications are more data processing hungry because of their data increment in large volume and the range can start from few dozens of TB (i.e. Big Data). To increase the development and assessment of the applications of GPS-embedded (i.e. location based services), a benchmark for Big Data and system management is an urgently needed. This research makes a new born of Big Data research to characterize and analyze their benchmarks. We analyze the map-matching algorithm which is a very basic pre-processing step of many LBS. We implemented ST-Matching map-matching algorithm and found that by increasing the number of candidate points accurate results can beobtained. On the extreme, this algorithm increases the running time especially when the trajectory has more GPS points. The reason of the above statement is that it processes a large number of the shortest path queries. Therefore, considering above factors, a parallel processing and adaptive scheme is needed to boost up the running time, which is our future work.

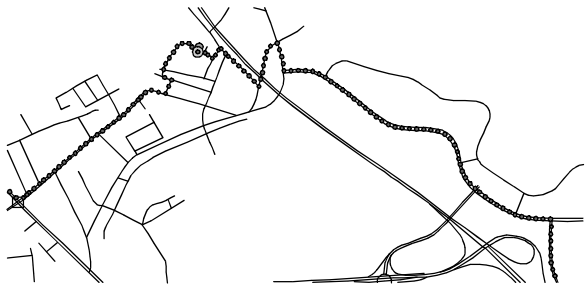

FIG. 4. ILLUSTRATION OF A REAL TRAJECTORY MATCHING ON ROAD NETWORK

ACKNOWLEDGMENTS

The work presented in this paper has been partly supported towards the Ph.D. studies of first author at Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, Guangdong, China. The authors would like to give special thanks to Yin Lou and Li Ye for their help and valuable support.

REFERENCES

[1] GigaSpaces Survey Summer, "Big Data Survey", 2012 http://www.gigaspaces.com

[2] Chandio, A. A., Korejo, I. A., Khuhro, Z. U. A., and Memon, F., "Clouds Based Smart Video Transcoding System", Sindh University Research Journal (Science Series), Volume 45, No. 1, pp. 123-130, Jamshoro, Pakistan, 2013.

[3] Wang, J.Z., and Wang, Z.J. , "Architecture Design of Urban Intelligent Transportation Using Cloud Computing", Advanced Materials Research, Volume 605, pp. 2549-2552, 2013.

[4] Chen, C., Liu, Z., Lin, W. H., Li, S., and Wang, K., "Distributed Modeling in a MapReduce Framework for Data-Driven Traffic Flow Forecasting", 2013.

[5] Li, Z., Chen, C., and Wang, K., "Cloud Computing for Agent-Based Urban Transportation Systems", Intelligent Systems, IEEE,.Volume 26, No. 1. pp. 73-79, 2011.

[6] Lou, Y., Zhang, C., Zheng, Y., Xie, X., Wang, W., and Huang, Y., "Map-Matching for Low-Sampling-Rate GPS Trajectories", Proceedings of 7th International Conference on Advances in Geographic Information Systems, 2009.

[7] Yuan, J., Zheng, Y., Zhang, C., Xie, X., and Sun, G. Z., "An Interactive-Voting Based Map Matching Algorithm", 11th International Conference on Mobile Data Management, 2010.

[8] Gartner, "3D Data Management: Controlling Data Volume, Velocity and Variety", February 2001 http://www.gartner.com

[9] Chandio, A. A., Xu, C. Z., Tziritas, N., Bilal, K., and Khan, S. U, "Comprative Study of Job Scheduling Strategies in Large-Scale Parallel Computational Systems", Proceedings of 11th IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA) co-located with TrustCom and IUCC, pp. 949-957, DOI:10.1109/TrustCom.2013.116, Melbourn, Australia, July 16-18, 2013.

[10] Talend Survey, "How Big Is Big Data Adoption?", Fall 2012 http://www.talend.com

[11] Luo, C., Zhan, J., Jia, Z., Wang, L., Lu, G., Zhang, L., Xu, C. Z., and Sun, N., "CloudRank-D: Benchmarking and Ranking Cloud Computing Systems for Data Processing Applications", Frontiers of Computer Science, Volume 6, No. 4, pp. 347-362, 2012.

[12] Huang, S., Huang, J., Dai, J., Xie, T., and Huang, B., "The HiBench Benchmark Suite: Characterization of the MapReduce-Based Data Analysis", IEEE 26th International Conference on Data Engineering Workshops, 2010.

[13] Cooper, B. F., Silberstein, A., Tam, E., Ramakrishnan, R., and Sears, R., "Benchmarking Cloud Serving Systems with YCSB", Proceedings of 1st ACM Symposium on Cloud Computing, 2010.

[14] Ferdman, M., Adileh, A., Kocberber, O., Volos, S., Alisafaee, M., Jevdjic, D., Kaynak, C., Popescu, A. D., Ailamaki, A., and Falsafi, B., "Clearing the Clouds: A Study of Emerging Scale-Out Workloads on Modern Hardware", Proceedings of 7th International Conference on Architectural Support for Programming Languages and Operating Systems, 2012.

[15] Greenfeld, J.S., "Matching GPS Observations to Locations on a Digital Map", National Research Council, Transportation Research Board, Washington, DC 2002.

[16] Wenk, C., Salas, R., and Pfoser, D., "Addressing the Need for Map-Matching Speed: Localizing Global Curve-Matching Algorithms", IEEE 18th International Conference on Scientific and Statistical Database Management, 2006.

[17] Alt, H., Efrat, A., Rote, G., and Wenk, C., "Matching Planar Maps", Proceedings of 14th Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, 2003.

[18] Brakatsoulas, S., Pfoser, D., Salas, R., and Wenk, C., "On Map-Matching Vehicle Tracking Data", Proceedings of 31st International Conference on Very Large Data Bases, 2005.

[19] Pink, O., and Hummel, B., "A Statistical Approach to Map Matching Using Road Network Geometry, Topology and Vehicular Motion Constraints", 11th International IEEE Conference on Intelligent Transportation Systems, 2008.

[20] Hummel, B., and Tischler, K., "Robust, GPS-only Map Matching: Exploiting Vehicle Position History, Driving Restriction Information and Road Network Topology in a Statistical Framework", GIS Research UK Conference, 2005.

[21] Horton, I., "Ivor Horton's Beginning Java 2, JDK 5th Edition, December, 2004.

[22] Chandio, A.A., Zhu, D., Sodhro, A.H., and Syed, M.U., "An Implementation of Web Services for Inter-Connectivity of Information Systems". International Journal of Computing and Digital Systems, Volume 3, No. 3, pp. 219-225, 2014.

[23] Chandio, A. A., Bilal, K., Tziritas, N., Yu, Z., Jiang, Q., Khan, S. U., & Xu, C. Z., "A comparative study on resource allocation and energy efficient job scheduling strategies in large-scale parallel computing systems", Cluster Computing, DOI:10.1007/s10586-014-0384-x, p. 1-19, 2014.