Enhanced GNSS Positioning Solution on Android for Location Based Services Using Big Data

Anwar ul Haque, Tariq Mahmood, Muhammad Saeed

Institute of Business Administration Karachi, University of Karachi, Pakistan
ahaque@iba.edu.pk, tmahmood@iba.edu.pk, saeed@uok.edu.pk

Abstract

GNSS based positioning is common and widely utilized positioning system in world. The only limitation beside of accuracy a GNSS receiver has is the availability of GNSS signal. Closed environments where direct view of sky is not available e.g., inside building or shopping mall or under tunnel do not offer valid GNSS positioning. These types of environments are generally termed as GNSS denied environments (GDEs). In such scenario a modified technology called Dead Reckoning System (DRS) is used for positioning. A DRS is a sophisticated and costly system which is not widely available for general public usage. In this paper, we propose a novel and state of the art DRS technology. Specifically, we develop the DRS on Android platform for providing positioning services in real time using state of the art, open-source big data technologies. To ensure effective and accurate positioning with minimum power and resource consumption. We use the in-memory computing and streaming of Apache Spark and data storage of Aerospike. We test our DRS in three different GDEs along with the PoC in which we compare our DRS result with actual GPS-based positions. In each experiment, our system has given a promising circular error probability (CEP) of less than 3 meters, irrespective of object state. The potential domains of applications are security, location-based service, and indoor navigation.

Keywords: GNSS, Big data, Sensor fusion, Apache spark, AeroSpike

1 Introduction

Positioning is the process of giving a distinct coordinate relative to some reference coordinates. Positioning systems are based on Global Navigation Satellite System (GNSS). There are a number of GNSS operational today, among which GPS is a well-known and commonly used. GNSS is a satellite-based navigation system which provides coded information in the form of a RF signals. This information consists of ranging code, navigation data and carrier signals [16]. The GNSS receiver decodes the captured signal and extracts navigation information. Then, by the method of triangulation it computes its position [7]. The basic idea is to determine the relative position of object using the geometry of triangles. The location of a point is described in terms of fixed distances to three known points, and the efficiency of this process depends upon a geographical orientation of these points. The proper alignment of three points can describe a position of receiver with the help of three equations which are given in Equations (1), (2) and (3).

$$d_1^2 = x^2 + y^2 \quad \textbf{(1)}$$

$$d_2^2 = (x-p)^2 + y^2 \quad \textbf{(2)}$$

$$d_3^2 = (x-q)^2 + (y-r)^2 \quad \textbf{(3)}$$

Here, x and y are the points of origin, p is the ordinate of second fixed point, and q and r are the coordinates of third point. By solving the above three equations for x and y we get the object's coordinates as shown in Equations (4) and (5):

$$x = \{d_1^2 - d_2^2 + p^2/2p\}^{1/2} \quad \textbf{(4)}$$

$$y = +- \{d_1^2 - [(d_1^2 - d_2^2 + p^2)/2p]^2\}^{1/2} \quad \textbf{(5)}$$

Figure 1 demonstrates the method for computing an object's position in 3D space. Sat 1, 2, and 3 are satellites and the black dot represents the receiver.

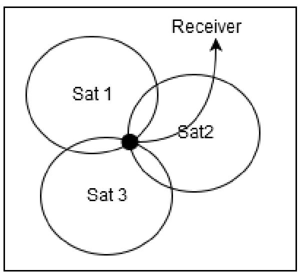

Figure 1. A GNSS receiver is triangulating three satellites for calculating its position

However, positioning becomes difficult when the receiver is in a GDE i.e. inside a building, shopping mall or subway station. Due to GDE receiver neither

*Corresponding Author: Anwar ul Haque; E-mail: ahaque@iba.edu.pk
DOI: 10.3966/160792642019032002009

will obtain satellite signals nor will resolve position. In such scenario the receiver will wait in an endless loop until the hardware moves out from GDE. To solve such problems, typically a DRS is used [8].

Generally a DRS employees a predictive algorithm i.e. Kalman position filter, which predicts the object's next position using its previous position and sensory data (accelerometer, gyroscope, magnetometer and orientation sensor) data [5, 11].

DRS is used in various applications like network games, animal navigation, marine navigation, and robotics. Figure 2 shows an implementation of DRS for positioning a vehicle inside a tunnel which is a GDE. The yellow circles represent DRS based position while green circles show GNSS based position. Although DRS is effective for GDEs, but it's usage is relatively low and not widespread in general public navigation systems due the following reasons:

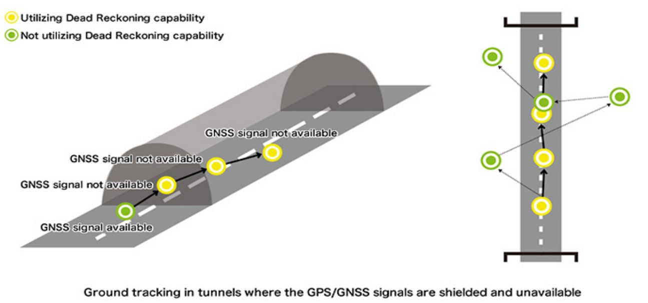

Figure 2. Positioning in tunnel using DRS

High cost & complexity. DRS hardware is expensive due to its nature of complexity of design and processing.

Limited available processing power. A DRS application is, by nature, rigid and runs in real-time due to nature of RTOS.

Limited commercial usage. DRS is designed and maintained for high end applications. So its availability for commercial usage is limited.

Limited mobile DRS usage. DRS for mobile platforms (e.g., Android) is still not available for general public usage.

In this paper, we have addressed the aforementioned limitations by proposing a cost-effective, efficient, Android-based mobile DRS. We named our system Big Data analytics based DRS (BDA-DRS). The hardware requirement is fulfilled by employing an Android phone. Since Android phones are unsuitable for real time processing, we rely on big data technologies to solve this problem. Specifically, the data generated from Android sensors is big data with respect to streaming and has a high velocity in nature.

In BDA-DRS, we used Apache Spark for real time processing of sensory and prediction engines. Spark is a state-of-the-art, in-memory, and scalable cluster computing technology [9] with extendable big data processing system on various successful applications in diverse domains [10]. To meet Spark requirement of fast and agile storage & transaction we employed Aerospike, a NoSQL big data store which stores data as key-value pairs. Like Spark, Aerospike is also in-memory, scalable and proven efficient [2] and hence ensures smooth data transactions for real time processing. Following are the two research questions which we tried to answer in this research paper:

RQ1. Can Big Data technologies be used in predicting real time position in a GNSS denied environment with good accuracy?

RQ2. Can commodity gadgets like Android and IPhones be used for providing robust, precise and accurate GNSS positioning information with the support of Big Data technologies.

2 Relevant Background

In this section, the relevant background is discussed.

2.1 Kalman Filters

A Kalman filter is a type of linear quadratic estimator whose primary function is to, estimate an unknown variable by utilizing the previous experience and other statistical inputs, recursively. The algorithm is a simple two-step process. In step one, values of current state variables are estimated, and in second step, these values are updated using weighted averaging. Since the output is predicted using only the last experience and current sensory, it tends to be more effective than a single measurement or simple averaging. Kalman filters are widely used in navigation, controls, and guidance systems [12, 17].

The example given in Figure 3, illustrates a mouse movement inside a tunnel whose position is desired on time Tn.

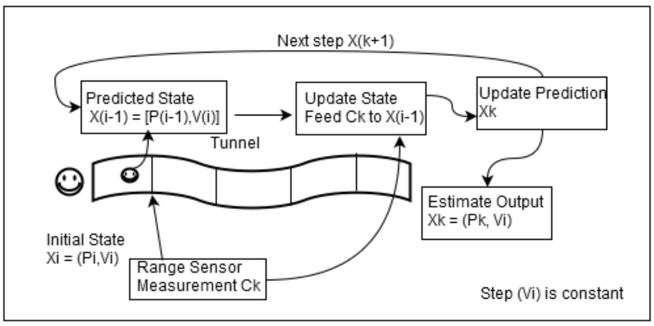

Figure 3. A mouse in tunnel (constant time step)

For this we have:

- X_i: state of the system (Pi, Vi)
- C_k: estimated measurement (approximates x)
- σ^2: variance of a measurement
- X: vector
- P: co-variance matrix of position
- X_k: best estimate of state

- $X_{(k-1)}$: best estimate before measurement was taken
- $X_{(k+1)}$: best next estimate of state

Taking assumption of scalar measurements, we can use (6) to estimate the state of mouse on certain time.

$$x^k = C_1 + K.(C_2 - C_1) \quad \textbf{(6)}$$

While K is Kalman gain and denoted by (7).

$$K = r_1^2 / r_1^2 + r_2^2 \quad \textbf{(7)}$$

With knowledge of initial point Pi, the stepping value Vi and amount of time elapsed Td after its travel, and by utilizing Kalman filter equation given in (8), we can find out the possible position of mouse inside the tunnel at time Tn.

$$x^k(t_n) = C(t_i) + K.[C(t_n) - C(t_i)] \quad \textbf{(8)}$$
$$r^2 = [1-k].r_z^2$$

2.2 Apache Spark

Apache Spark is an open-source and efficient cluster computing platform with the capability of data parallelism and fault tolerance [21]. Spark provides a new type of data set called Resilient Distributed Data-set or RDDs, which are read-only multisets of data elements. Due to the nature of RDD iterative algorithms can be easily applied on Spark with a high responsiveness. The architecture of Spark consists of cluster management engine and a distributed storage system (abstract view shown in Figure 4). For small size development, Spark provides a pseudo-distributed mode which runs on local machine with single instance per core of CPU [1, 3, 6, 13]. The selection of Spark is due to its adoptable nature, simplicity in learning, fast processing, ease of applicability and configuration [21].

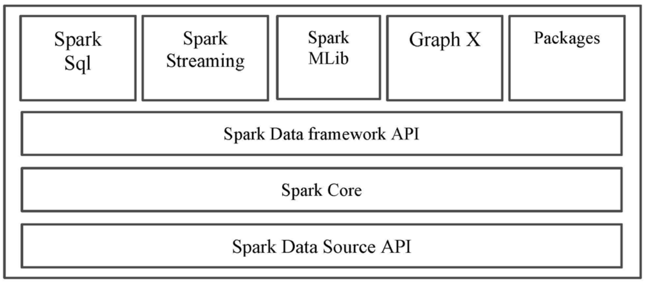

Figure 4. A generic spark framework architecture

2.3 Aerospike

For real time processing we require real time data storage and acquisition. To the best of our knowledge, Aerospike serves the purpose with its NoSQL in-memory data storage and retrieval capability [22-23]. Aerospike is a 3-layer architecture:

- flash optimized data layer
- self-managed distribution layer and a cluster-aware client layer
- distribution layer which is replicated across data centers to ensure consistency.

Figure 5 shows layered architecture of Aerospike. The in-memory computing with flash optimization has made real time computing over clusters a reality to meet the need of today's computing. The selection of Aerospike is done as it is open-source, it is Key-Value pair based and it has successful use cases [13].

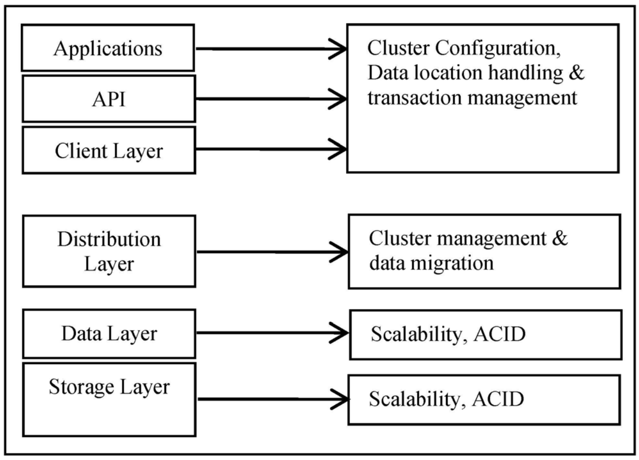

Figure 5. Working architecture of the Aerospike NoSQL Database

3 Problem Statement

We now formally mention our problem statement. The new era of positioning and data analytics on user movement is totally dominated by accurate and efficient GNSS based position. However, acquiring accurate and on demand position in a GDE poses a difficult problem. The devices which can provide positioning solution in GDE are not only expensive but also require specific level of knowledge of GNSS. This preventing widespread of these solutions, e.g., for commercial or public usage. So, there is need to bring a more robust, accurate and efficient positioning solution inside GDE on general purpose devices like Android cell phones. The availability of this technology on commodity gadgets which can provide precise and accurate position and timing information inside GDE will enable a new era location-based computing and enhance the e-commerce market.

4 Proposed Solution

In this section we proposed and worked out a solution for predicting GNSS position on Android phone using Big Data technologies. As data is being continuously generated by the moving object on real time rate, so we need to employ Big Data related technologies to overcome the challenge of real time computing. The proposed solution which we called BDA-DRS can be distributed in the following manner:

- Android application which is capturing sensory (Accelerometer, Gyro, Magnetometer, Orientation) from android sensor hub, GNSS information from

GNSS hardware & GSM information from Communication Module. The captured information is processed (ETL) and sent to Spark based processing system using raw async sockets (spark custom receivers). Figure 6 shows the above mentioned model.

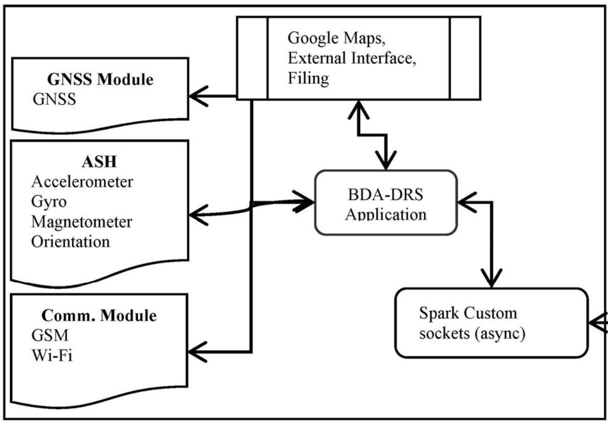

Figure 6. BDA-DRS, Android apk working architecture

- Spark based processing engine which is basically the heart of this research work employees stream processing on incoming data from cell phone via custom sockets. The data is parallelized and broadcast to workers for processing. Figure 7 shows the above mentioned model.

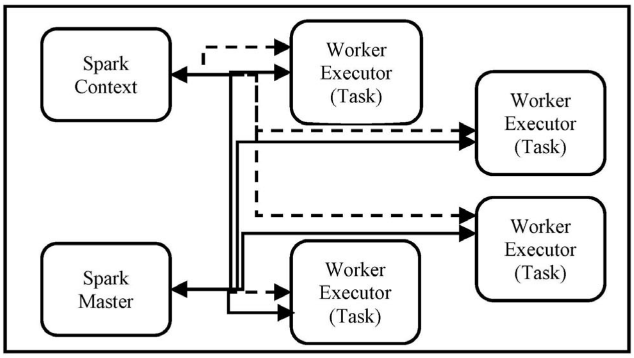

Figure 7. BDA-DRS, Master and worker Model Apache Spark

- Each worker has its own copy of Kalman prediction logic and implements it on the incoming stream of data from master.
- The accumulated result from workers is stored into NoSQL storage unit and a copy is sent to Android device via raw async sockets for displaying position on Google map which is running in the application. The application also reflects an interface which can be used by other applications for utilizing the DRS result.

The over all system can be distributed into two major components, one is the hardware component and other is the software component.

4.1 Hardware Component

Hardware devices capable of providing position in GDE in real time are expensive and sophisticated to acquire and utilize [14-15] for general usage. To ensure low cost and easy to use hardware component of BDA-DRS we select Android device as its hardware which can generate sensory, GNSS and GSM related information [4].

4.2 Software Component

Android OS is not RTOS due to nature of non-deterministic hardware [18-20]. Therefore, to meet the requirement of real-time processing of BDA-DRS we employ lightening fast cluster computing of Spark and we separate the software component into three layers. Figure 8, displays the above documented layering.

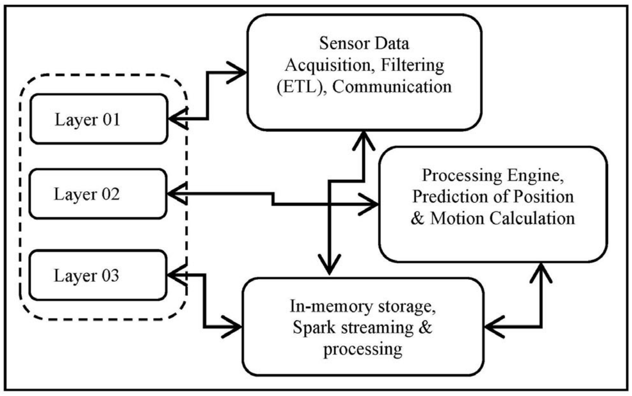

Figure 8. Processing and communication layers for BDA-DRS

- Layer 01, Android application which is doing sensor data acquisition and communication. module provides the necessary ingredient for prediction.
- Layer 02, Spark stream based processing engine which is executing on personal Laptop.
- Layer 03, Data storage & transaction engine of Aerospike which is also executing on laptop.

4.3 Working of BDA-DRS

In BDA-DRS, the sensor data is filtered on android application using software based low & high pass filters in ETL phase. The ETL phase removes null values, non-uniformity and outliers etc. The filtered data is then fed to Spark processing engine via custom receiver using async sockets.

The streaming engine which is running on Spark in laptop captures data from spark custom receiver and parallelize it to distribute to the workers via broadcasting.

The workers then execute the Kalman predictive logic which is available on their end to estimate the position from sensory data on the basis of last known experience. The processing on worker side is done with in the given quota of time to ensure the real time behavior of BDA-DRS. The accuracy of data is maintained using motion prediction engine. The motion prediction engine utilizes sensor fusion algorithm to predict the movement of device in a 3 dimensional space and the output of this engine serves as the argument of position prediction engine. The argument is used to correlate the difference of movement from last known position to current.

The accumulated result is stored on Aerospike NoSql database (which serves as the experience of Kalman for next iteration) and a copy is sent to device for display on Google map. The whole process is executed in a loop and predicted position is estimated on movement made by the user (carrier of android device). The major hurdle in position prediction is the initial experience (last known position). To acquire this BDA-DRS takes coarse position from Wi-Fi, 3G and GSM as its initial position and feeds it to prediction engine. The complete architecture of BDA-DRS is shown in Figure 9.

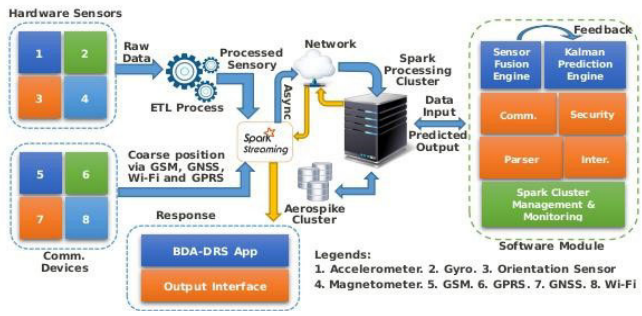

Figure 9. Architecture of BDA-DRS showing sensor to processing work

4.4 Proof of Concept

We initially conducted two experiments as Proof of Concept (POC) of BDA-DRS. The first was in static mode, where BDA-DRS acquire the position inside a closed room which is a GDE. Then, the position of the device was taken on the roof of that room which is a perfect GNSS environment. These two positions were compared in terms of accuracy, acquisition time and stability. The position taken from BDA-DRS is very swift in acquisition and stability while positioning from GNSS receiver requires more time & continuously hops in nearby locations. The prediction engine in BDA-DRS gives an accuracy of less than 3.0m CEP. Figure 10 and Figure 11 provide a comparative view on Google Earth of both scenarios.

Figure 12 shows predicted latitude, longitude by BDA-DRS in a GDE. The precision in the values of latitude and longitude details the smoothness & accuracy of position while clustering of value reflects promising result of BDA-DRS for further dynamic tests and scenario.

Figure 10. Positioning in GDE via BDA-DRS

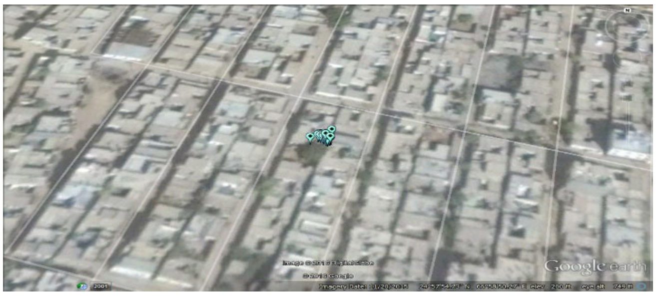

Figure 11. Positioning on Roof via GPS

Figure 12. Scatter plot of longitude vs latitude in static mode

In second POC experiment, BDA-DRS was tested in a running car on Android device. The motion was analyzed with respect to time and predicted latitude and longitude was recorded. Since the device is moving continuously so samples acquired from device and information processed on that sample by motion prediction engine and position prediction engine must agree on the line of synchronization. We overcome this synchronization issue through in-memory processing, the motion processed information is stored in RAM in Aerospike & fed to prediction engine for effective synchronization.

After achieving synchronization, the prediction was found to be correct by more than 80% in various test scenarios as compare to GNSS only positioning. Figure 13 and Figure 14 show the Google Earth image of positions taken during movement via BDA-DRS & GPS while Figure 15 shows the plot of BDA-DRS points. The above two experiments of PoC show promising results of BDA-DRS which led us to conduct our original real time experiments in GDE. These experiments are discussed in Section 4.5 as experiments.

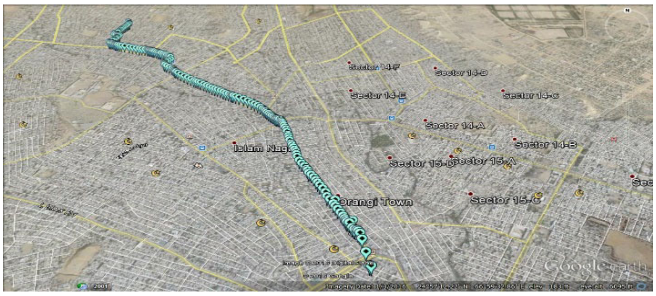

Figure 13. Position via BDA-DRS in dynamic mode

Figure 14. Position via GPS in dynamic mode

Figure 15. Scatter plot of lon vs lat in dynamic mode

4.5 Experiments

We tested BDA-DRS in the following three GDEs:

- Inside the Dolmen Shopping Mall, which is well-known mall and located on Tariq Road, Karachi
- Inside UBIT, which is the building of the Department of Computer Science in University of Karachi
- Inside PAF-KIET city campus, which is situated on Shahra-e-Faisal, Karachi.

4.5.1 Experiment in Dolmen Mall

Dolmen Mall is one of the largest shopping malls of Karachi. We did several runs and verified the result again & again to ensure the behavior of BDA-DRS. Figure 16 and Figure 17 show the camera and Google image of Dolmen Mall.

The result of BDA-DRS experiment conducted in Dolmen Mall is shown in Figure 18. The dotted blue lines are the movement of person carrying android phone inside the mall. Since the gate of dolmen mall provides good reception of GNSS signal so initial position is taken via GNSS at gate. The initial position is used to estimate the next possible position using Kalman estimation engine which is running in Spark on laptop. The positioning information is not so smooth due to the delay in communication between android and laptop. The data network in Dolmen mall was not working well due to huge crowed and dense architecture. But still BDA-DRS managed to provide effective and precise positions in real time to android phone.

Figure 16. Camera image of Dolmen Mall

Figure 17. Google image of Dolmen Mall

Figure 18. Positioning via BDA-DRS in Dolmen Mall

4.5.2 Experiment in UBIT

Department of computer science of University of Karachi is also known as UBIT (Umer Bhasha institute of technology). The Camera image and Google image of UBIT is given in Figure 19 and Figure 20 respectively.

Figure 19. Camera image of UBIT

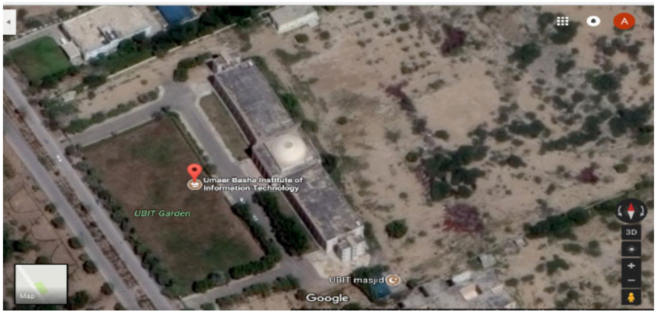

Figure 20. Google map image of UBIT

In the experiment, one of the authors carrying an Android device walked in a straight line in the central corridor of UBIT. The initial position was taken using GSM and Wi-Fi as the there is no reception of GNSS signal on either inner sides of UBIT building. Once the initial position was taken the author starts to move and the position of author is estimated using BDA-DRS. Figure 21 shows the output of BDA-DRS test in UBIT. The red circle in figure shows the starting point of movement where initial position was taken.

Figure 21. Positioning via BDA-DRS in UBIT

4.5.3 Experiment in PAF-KIET (City Campus)

PAF-KIET City Campus is engaged in providing grad. & under graduate level education to its students in the field of engineering, computer science and management. It's camera image and Google map image is shown in Figure 22 and Figure 23 respectively.

In this experiment the starting point was taken from GNSS as on the gate of PAF-KIET the GNSS signal are healthy and effective. After obtaining initial position on of the author start to moving in the corridor of PAF-KIET. The corridor of PAF-KIET is a semi-GDE, there are patches where 2 to 3 satellites are visible but the signal strength is not enough strong to compute effective position. While BDA-DRS provided effective, smooth and turn-by-turn position in real time which is shown in Figure 24. Since the network was so strong in PAF-KIET so obtained position via BDA-DRS is dense and precise as compare to other two experiments mentioned above.

Figure 22. Camera image of PAF-KIET

Figure 23. Google map image of PAF-KIET

Figure 24. Positioning via BDA-DRS in PAF-KIET City Campus

5 Conclusion

The idea of BDA-DRS for predicting real time position via Big Data technologies (Spark & Aerospike) in GDE was quite successful. As there is always a room of improvement which can lead things to best than better. So as part of the future work the data communication framework of BDA-DRS will be shifted to Apache Kafka from current async. Sockets to ensure more resilient and smooth communication. Based on results we answer our research questions as follows:

RQ1. Can Big Data technologies be used in predicting real time position in a GNSS denied environment with good accuracy?

Answer. BDA-DRS is based on Big Data technologies (Apache Spark and AeroSpike) and has given promising result in the domain of position prediction in GDEs with good accuracy and precision in real time. Several tests and experiments were done to ensure the output and working.prediction in real

RQ2. Can commodity gadgets like Android and Iphones be used for providing robust, precise and accurate GNSS positioning information with the support of Big Data technologies.

Answer. Since commodity gadgets are capable of high processing technology today. The only lacking feature is the availability of real time operating system which is essential for any DRS based systems. This lacking feature was compensated via Big Data technologies. After the novel combination of Big Data technology with commodity cell phone it was possible to obtain precise and accurate positioning information in GDEs using commodity cell phone.

The positioning solution of BDA-DRS is flexible that it can be easily integrated with various application of android or computers & future industry can use it to make more positioning-oriented solutions.

References

[1] S. Gopalani, R. Arora, Comparing Apache SPARK and Map Reduce with Performance Analysis Using K-Means, *International Journal of Computer Applications*, Vol. 113, No. 1, pp. 8-11, March, 2015.

[2] J. R. Lourenço, B. Cabral, P. Carreiro, M. Vieira, J. Bernardino, Choosing the Right NoSQL Database for the Job: A Quality Attribute Evaluation, *Journal of Big Data*, Vol. 2. No. 1, pp. 2-18, August, 2015.

[3] N. Nupairoj, P. Tangsatjatham, Hybrid Big Data Architecture for High-Speed Log Anomaly Detection. *Journal of Internet Technology*, Vol. 18, No. 7, pp. 1681-1688, December, 2017.

[4] L. Chou, D. C. Li, W. Chen, Y. Chang, Design and Implementation of a Novel Location-Aware Wearable Mobile Advertising System, *Journal of Internet Technology*, Vol. 18, No. 5, pp. 963-974, September, 2017.

[5] S. Wang, H. Yi, L. Wu, F. Zhou, N. Xiong, Mining Probabilistic Representative Gathering Patterns for Mobile Sensor Data, *Journal of Internet Technology*, Vol. 18, No. 2, pp. 321-332, March, 2017.

[6] E. Carlini, P. Dazzi, A. Esposito, A. Lulli, L. Ricci, Balanced Graph Partitioning with Apache Spark, in: *Euro-Par 2014: Parallel Processing Workshops: Euro-Par 2014 International Workshops, Porto, Portugal, August 25-26, 2014, Revised Selected Papers, Part I*, Springer International Publishing, 2014, pp. 129-140.

[7] J. Rosello, P. Silvestrin, R. Weigand, S. d'Addio, A. G. Rodríguez, G. L. Risueño, Next Generation of EPA's GNSS Receivers for Earth Observation Satellites, *30th AIAA International Communications Satellite System Conference (ICSSC)*, Ottawa, Canada, September, 2012.

[8] L. Pedalo, J. Borenstein, Personal Dead-reckoning System for GPS-denied Environments, *IEEE International Workshop on Safety, Security and Rescue Robotics (SSRR2007)*, Rome, Italy, 2007, pp. 1-6.

[9] H. Mustache, Z. Al-Ars, Cluster-based Apache SPARK implementation of the GATK DNA Analysis Pipeline, Bioinformatics and Biomedicine (BIBM), *2015 IEEE International Conference*, Washington, DC, 2015, pp. 1471-1477.

[10] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, I. Stoica, Spark: Cluster Computing with Working Sets, *Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing HotCloud'10*, Boston, MA, 2010, pp. 10-10.

[11] T. Bo, C. Pingyuan, C. Yangzhou, Sigma-Point Kalman Filters for GPS Based Position Estimation, *5th International Conference on Information Communications Signal Processing*, Bangkok, Thailand, 2005, pp. 213-217.

[12] F. Golnaraghi, S. Won, P. Won, W. Melek, A Kalman/Particle Filter-Based Position and Orientation Estimation Method Using a Position Sensor/Inertial Measurement Unit Hybrid System, *IEEE Transactions on Industrial Electronics*, Vol. 57, No. 5, pp. 1787-1798, May, 2010.

[13] Y. Yan, L. Huang, L. Yi, Is Apache SPARK Scalable to Seismic Data Analytics and Computations, *IEEE International Conference on Big Data (Big Data)*, Santa Clara, CA, 2015, pp. 2036-2045.

[14] L. Perneel, H. Fayyad-Kazan, M. Timmerman, Can Android be Used for Real-time Purposes? *International Conference on Computer Systems and Industrial Informatics*, Sharjah, United Arab Emirates, 2012, pp. 1-6.

[15] H. Tomiyama, S. Chikada, S. Honda, H. Takada, An RTOS-based Approach to Design and Validation of Embedded Systems, *IEEE VLSI-TSA International Symposium on VLSI Design, Automation and Test*, Hsinchu, Taiwan, 2005, pp. 185-187.

[16] A. A. Deshmukh, S. A. Kulkarni, GPS Signal Distribution System, *International Conference on Advances in Computing, Control, and Telecommunication Technologies*, Trivandrum, Kerala, India, 2009, pp. 672-674.

[17] S. H. Tang, T. Kojima, T. Namerikawa, C. F. Yeong, E. L. M. Su, Unscented Kalman Filter for Position Estimation of UAV by Using Image Information, *54th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, Hangzhou, China, 2015, pp. 695-700.

[18] P. Hambarde, R. Varma, S. Jha, The Survey of Real Time Operating System: RTOS, *International Conference on Electronic Systems, Signal Processing and Computing Technologies*, Nagpur, India, 2014, pp. 34-39.

[19] Z. He, A. Mok, C. Peng, Timed RTOS Modeling for Embedded System Design, *IEEE Real Time and Embedded Technology and Applications Symposium*, San Francisco, CA, 2005, pp. 448-457.

[20] M. Vetromille, L. Ost, C. A. M. Marcon, C. Reif, F. Hessel, RTOS Scheduler Implementation in Hardware and Software

for Real Time Applications, *Seventeenth IEEE International Workshop on Rapid System Prototyping (RSP'06)*, Chania, Crete, Greece, 2006, pp. 163-168.

[21] K. Wang, M. M. H. Khan, Performance Prediction for Apache Spark Platform, *17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems*, New York, NY, 2015, pp. 166-173.

[22] A. Woodie, *Aerospike Adds Spatial Data to Speedy Nosql*, January, 2016.

[23] J. Kunigk, *How-to: Detect and Report Web-Traffic Anomalies in Near Real-Time*, June, 2016.

Biographies

Anwar ul Haque is a Ph.D. Scholar in Faculty of Computer Science, IBA Karachi. He has 10 years of research & development experience of SUPARCO & 1 year experience of DHA SUFFA University. He has expertise in GNSS, Big Data technologies, NoSql technologies, and Sensor networks. He has 1 journal publication and 1 conference publication.

Tariq Mahmood is Assistant Professor at IBA, Karachi. He has 13 journals & 30+ conference publications. He has 10 years of professional experience in the domains of BI, Predictive Analytics, Warehousing, Big Data, and Advanced Analytics. He has conducted numerous workshops on Big Data. Dr. Tariq heads the BDA-LAB at IBA Karachi.

Muhammad Saeed is associated with the Department of Computer Science, UoK since 2004 as full time faculty member. He has 10+ Conference publications at reputed venues like ASONAM, WWW, and ACL, and 15+ recognized journal publications. His area of interests are BDA, Parallel & Distributed Computing. Currently he is working in the domains of SNA, cognitive sciences, LBS & SDN.