

Enabling Location-Based Services 2.0: Challenges and Opportunities

Saket Sathe*, Roie Melamed†, Peter Bak†, and Shivkumar Kalyanaraman*

*IBM Melbourne Research Laboratory, Australia.

Email: {ssathe, skalyana}@au.ibm.com

†IBM Haifa Research Laboratory, Israel.

Email: {roiem, peter.bak}@il.ibm.com

Abstract—The next-generation mobile devices include smart watches, wristbands, wearables (e.g., Google Glass), etc. In the future such devices will constitute a large fraction of the total devices available in the market [1]. Latest studies confirm that location-based services are the most requested feature by developers with a market share of \$13B in 2013 and have expected exponential growth [2]. Future location-based applications/services will use the data generated by the new mobile devices for providing enhanced user experience. This paper presents a vision of such next-generation location-based services, which we call LBS 2.0. We present the challenges and opportunities that LBS 2.0 will pose for mobile data management.

I. INTRODUCTION

Mobile devices have virtually exploded in the recent years. Surveys have predicted that this trend will continue in the future [1]. In addition to the smartphones, the newer mobile devices include smart watches¹, wristbands², and wearables like Google Glass. These devices not only have the ability to record location (using GPS) but also have the ability to *sense* and *communicate* various other attributes (e.g., video, images, orientation, linear/angular acceleration, ambient temperature, luminosity, and many more) about their surrounding [3], [4]. We refer to such data as multivariate spatio-temporal data³.

The LBS 2.0 applications will use all such sensed information for providing new features and services to the users. These applications will consume the data generated by the mobile devices for providing users an enhanced experience and awareness about their behavior or surrounding. Below we present a broad categorization of several sample LBS 2.0 applications:

- *Activity recognition*: detecting user activities like running, walking, standing, sitting, etc. [5].
- *Healthcare*: sleep pattern change detection, EEG/ECG/EMG measurements using smart shirts [6], breathing measurements, posture tracking, etc.
- *Smart spaces*: switching on appliances when a user enters his/her home or car, smart smoke alarms and thermostats⁴, indoor atmosphere sensing⁵.
- *Fitness/outdoor*: monitoring heart rate, pedometer, pace, calories burnt, etc.

- *Crowdsensing*: monitoring traffic density⁶, road condition, ambient pollution levels [7], weather conditions, etc.

Thus, it is clear that LBS 2.0 have huge, unexplored potential and companies/organizations that will leverage from it will have a greater chance of emerging as key players in the future. An important aspect of such services is the data these new devices will generate using a combination of GPS data and associated sensors. Furthermore, from a purely data management perspective there are two fundamental questions that should be addressed: **(Q1)** *How much data will be generated by the millions of new gadgets that will be used for providing LBS 2.0?* and **(Q2)** *What are the important challenges that will be posed for handling such a volume of data?* Below we provide arguments that will help us motivate these questions or at least give us intuitions of the scale of these problems.

Q1: The data volume (and velocity) can be approximated by the following back-of-the-envelope calculation. Lets consider a modern smartphone like the Samsung Galaxy S3. We installed an Android application called Androsensor⁷ to estimate the volume of data collected in 1 hour from all the sensors on board the Galaxy S3. This phone has about 10 sensors [4] including the GPS. Our estimate shows that in 1 hour approximately 1.6 MB of data (0.5 MB when GZip compressed) is generated.

Next, consider a large social networking website like Facebook. In Q3 2013 Facebook reported that it has 507 million daily active users [8]. Furthermore, it is natural that if LBS 2.0 are introduced, then users will adopt them at a slower pace in the first few years [9]. Let us assume a reasonable adoption rate of 4 %/year. This rate is typical for the adoption of new technologies. It indicates that when LBS 2.0 are introduced only 4% of the current daily active users will initially (i.e., in the first few years) use them. In addition, we assume that these users will use these services for an average 1 hour/day.

With these assumptions, LBS 2.0 will generate about 32 TB/day of uncompressed data or 10 TB/day of compressed data. In 2009, Facebook had reported that its Haystack photo storage system adds 25 TB/week of storage [10]. Comparing 32 TB/day to 25 TB/week (or 10 TB/day GZip compressed), we can obtain a clear idea of the significantly large volume

¹ Pebble Smartwatch, <https://getpebble.com/> ² Wearable Technologies, <http://bit.ly/waveWBT> ³ For brevity, we often drop the adjective *multivariate*. ⁴ Nest Labs, <https://nest.com/> ⁵ Cubesensors, <http://cubesensors.com/>

⁶ Waze, <https://www.waze.com/> ⁷ Androsensor, <http://bit.ly/AndroSensor>

(and velocity) of data that will be generated.

Q2: We believe this large volume of data will pose several substantial challenges in the future. Here we succinctly describe these challenges and in the rest of the paper we will deep dive into them individually and discuss each of them in sufficient detail. Following are the challenges that we believe are key for enabling LBS 2.0:

Data Acquisition: In many circumstances it is unnecessary to transmit all the sensed data to the backend (e.g., cloud or data center). For reducing bandwidth consumption, we can only send the required data. Designing intelligent algorithms that decide which data should be transmitted is a challenging task. We further elaborate on this topic in Section II.

Data Storage: We have already discussed that data volume and velocity are the main roadblocks in supporting LBS 2.0. Fast and reliable storage of such data is another challenging task. In Section III we provide a detailed discussion on the latest technological developments in this domain.

Data Management: The applications created in the context of LBS 2.0 services will generate novel query types that should be addressed by the underlying data management system. Processing these queries fast and in real-time is a difficult task. In Section IV, we discuss the issues and the required innovations in query processing.

Data Visualization: Considering the range of different attributes that the mobile sensors can sense, it is natural to think of methods of seamlessly visualizing the data in interesting dimensions. Challenges related to data visualization are discussed in Section V.

In the remaining sections we provide an in-depth discussion on each of these challenges. We begin with the topic of acquiring data generated in the context of LBS 2.0.

II. DATA ACQUISITION

Let us consider a straightforward strategy: acquire all the data sensed by the mobile devices and send it to the backend. This is an infeasible solution for the following reasons:

- *Power:* Battery consumption by the Wi-Fi/3G antenna is one of the highest components of the battery usage on a smartphone [1], and transmitting all the data will simply worsen the problem.
- *Storage:* As pointed out in Section I, even if we are able to acquire all the data, storing this data will be significantly costly due to high hardware costs.
- *Bandwidth:* It is predicted that the bandwidth consumption of a smartphone will increase on an average [1]. If we add more bandwidth utilization to this already inflated average utilization then LBS 2.0 could become a turn off for many users.
- *Number of devices:* Our back-of-the-envelope calculation in Section I was based on a conservative assumption that each user will continue to have a single device. This assumption is, however, not true [1]. In fact the number of smart devices per capita are going to increase leading to higher data storage requirements.

Thus, it is clearly not a sound approach to acquire all the sensed data given our aim of supporting millions of users. Naturally, we are forced to think how can we acquire all the information that interests us keeping in mind the reasons stated above. We briefly look at the strategies that will be invaluable in the future for effective data acquisition:

Query to Data: Instead of acquiring all the data from the mobile devices, which is a bandwidth-intensive task, we can send the query to the mobile device. The device can then process the query locally and send us the results. This is similar to the model followed by the parallel database systems or map-reduce systems [11], where the query is sent to the data for conserving network bandwidth and to obtain scalability.

Sensor Selection: In many use-cases we will not require data from all the sensors for providing a certain service or data from the same sensor could be used for providing many services. These observations should be exploited in the future to optimize the amount of data acquired from the sensors.

Semantic Summarization: In many instances applications could be interested only in acquiring the semantic states rather than actually sensor values. For example, in the activity recognition use-case, typically we are only interested to know whether a person is standing or sitting, we are not interested in the sensor data that has lead to this inference [5].

Model-Based Acquisition: Here, the data that is required for maintaining a pre-defined model (like, regression model, probabilistic graphical models, etc.) is acquired. The model is selected in such a way that all the supported queries can be answered using the model, without acquiring additional data [12]. In the future there will be a high demand for model-based data acquisition especially for the crowdsensing applications.

III. DATA STORAGE

There are various aspects to this challenge starting from hardware (flash storage), distributed file systems, and higher level techniques like data compression. Investigations focused on all these aspects are necessary for supporting LBS 2.0. In this section we will take a deeper look at each of these aspects and obtain a systematic understanding of the state-of-the-art and the research directions proposed for the future:

Flash Arrays and Tiered Storage: The most important parameters in regards to storage capacity and performance are: (a) read/write speed, (b) capacity per unit dollar and (c) physical area per unit dollar. It is obvious that we will potentially need large amount of storage for storing data generated by the LBS 2.0. Flash storage has been gathering momentum as cost of flash arrays has been decreasing over the past few years.

Flash arrays are available in two types MLC (multi-level cell) or SLC (single-level cell), where MLC is more economical than SLC but significantly slower. Many vendors (Dell, IBM, EMC, HP, NetApp, Hitachi) combine these two flash types (or tiers) and slower SATA HDDs in a single box. Then automated tiering technology that decides which data should be moved from the MLC and which should reside in SLC depending on

access frequency is used for gaining performance.

Distributed File Systems: The next data storage layer is the distributed file system. The most popular open-source distributed file system is the Hadoop Distributed File System⁸ (HDFS). It has shown great scalability and has been successfully deployed on thousands of nodes. However, recently it has been observed that HDFS is unable to scale beyond a point, due to the manner in which it stores meta information [13]. HDFS stores all the meta information in the memory of a single server called the *NameNode*. NameNodes in large HDFS deployments have already reached their peak capacity and are unable to handle additional data without compromising performance. Thus, the future DFS should have hierarchical storage of meta information [14].

Compression: A key technique used for saving space is to compress data. On the one hand compression results in lower file sizes and therefore lower disk space consumption, on the other hand it adds additional decompression overhead when the data is needed for tasks like query processing or data mining. In the literature there have been various techniques used for compressing trajectories and sensor data. Below we briefly summarize these approaches:

- L_∞ -norm based compression of multi-sensor data guarantees a user-defined worst-case maximum error [15].
- *Clustered* compression clusters multiple trajectories to obtain a representative trajectory and then proposes to store only the representative trajectory [16].
- *Semantic* compression first derives a much smaller number of semantic states from location data and proposes to only store the semantic states [5].
- *Amnesic* compression is an online technique that samples a trajectory non-uniformly. It stores more samples about the recent past and less samples about ancient past [17].

The above techniques are not designed to handle spatio-temporal data from multiple sensors, which is a key contributor to LBS 2.0. Techniques supporting such data and operating in real-time will be important in the future.

IV. DATA MANAGEMENT

In this section we discuss the data management challenges for enabling LBS 2.0. As discussed in Section I, the data generated in the context of LBS 2.0 will be of high volume and velocity. This requires that the *spatio-temporal data store* (STDS) supports both *real-time* queries and *analytical* queries (on large historical data). Current systems do not support both these queries simultaneously due to their complimentary requirements: real-time queries require in-memory processing and analytical queries require persistent spatio-temporal storage.

In addition to queries, a major challenge in managing such large quantity of data is supporting enhanced capabilities like *geo-fence* and *geo-trigger*, as it requires, upon each location update checking if any of the geo-trigger or geo-fence condition have been satisfied. Motivated by these observations, in

the following paragraphs we discuss technological challenges that should be addressed for supporting these capabilities.

Scalability and Performance: The STDS supporting LBS 2.0 should exhibit high scalability. Scalability demands will require *partitioning* or *sharding* of data across servers in order to achieve near-uniform load distribution. Known 1-dimensional sharding techniques used in many NoSQL systems support *locality*; here, each operation is performed on a small subset of servers even if the data could be partitioned across many servers. Hence, such techniques have *linear scalability*: system capacity is a linear function of the number of servers.

Unfortunately, for high-dimensional data (3-4 dimensional) the 1-D sharding techniques do not preserve locality in all dimensions. A spatio-temporal data query can be potentially sent to a large number of servers, thereby limiting scalability. Therefore, designing locality-preserving sharding techniques is one of the key research challenges for achieving scalability in managing spatio-temporal data.

Load-Balancing and Elasticity: Designing a sharding technique involves designing load-balancing algorithms for evenly balancing data. Load-balancing in a STDS is unavoidable due to a) the inherent nature of user movements: users move through space crowding some locations (or shards) at some times and emptying them at others, and b) the memory consumption, request load, throughput and latency requirements change constantly. In addition, similar to high-dimensional sharding, high-dimensional load-balancing is a challenging task as multiple dimensions of the data have to be balanced simultaneously. We also note that any load balancing technique requires data transfer between shards that can impact system performance and increase network costs. This is especially true for spatio-temporal data, since typically it contains terabytes of historical data (refer Section I). Therefore, any load-balancing technique for spatio-temporal data needs to achieve a reasonable tradeoff between load balancing and data movements.

Spatio-Temporal Indexing: While a sharding technique performs server-level data partitions, data on each server have to be indexed for further improving query latency. Spatial indexes (e.g., R-Tree) for indexing 2-D (spatial) data are effective, but indexing higher dimensional data is challenging and typically requires the usage of dimensionality reduction techniques like *space-filling curves*. Although these techniques can index high-dimensional data their effectiveness exponentially decreases due to the curse of dimensionality, thereby leading to further challenges.

In-memory Processing: Real-time queries require in-memory spatio-temporal data processing. However, in-memory (cache-based) data stores typically do not support spatio-temporal data workloads due to the high volume of such data and lack of in-memory indexing techniques. Obviously, for many LBS 2.0, a reasonable cache size cannot support high volume of spatio-temporal data and requires regular data eviction to persistent storage. Naïve strategies for data eviction from the cache to a persistent storage can result in high cache miss for spatio-temporal queries, and hence can lead to undesirably

⁸ <http://hadoop.apache.org/>

poor query latency. Therefore, designing algorithms that can potentially *learn* dynamic eviction policies from load and usage type is an interesting and challenging area of research.

V. DATA VISUALIZATION

Information visualization concerns the use of interactive computer graphics to get insight into large amounts of data like multivariate data, hierarchical data, network data, etc. Conventional methods to depict data, such as bar-charts, pie charts, and line-charts, are useful to show aggregate information but fail to depict large and complex data, such that patterns, trends, and outliers can be discovered. Therefore, many new and useful information visualization methods that enable people to understand large amounts of data have been developed (extensive overviews can be found in [18]–[20]).

Visualizing Spatio-Temporal Data: LBS 2.0 data is related to spatial properties and events. The visualization of such data in a geographic context is an important discipline with a rich heritage from cartography and geovisualization. Here, the depiction of multivariate spatio-temporal data is a topic of immense value and intense research, especially because mobile data are generated in high pace by modern GPS-devices [21]. In this context, spatio-temporal complexity poses significant challenges for the visualization, but it also enables the use of the data for many purposes. Analysts can study the properties of space and places, understand the temporal dynamics of events and investigate moving objects' behavior. Recently a great variety of possible visualization methods for spatio-temporal data have been developed. However, several research challenges still need to be addressed [20].

Scalable Visual Analytics: Visual Analytics is defined as the smart combination of automatic algorithms for information extraction combined with interactive visualization. Scalability of visual analytics is typically limited to thousands of elements, and the development of specific solutions is still a lengthy and tedious process. Despite of the frameworks that are becoming available, there is a lack of insight on how to select and combine automated and visual methods.

Data is usually (and wrongly) assumed to be homogeneous; often just one or two types of data are supported. Data are assumed to come from a single source and to be clean, reliable, exact, and present; noisy, polluted, uncertain, and missing data are rarely dealt with. If one or more of these assumptions is violated, standard methods from information visualization fall short, and a need for new representations arises. The techniques of the future are expected to cope with *heterogeneous data* types having multiple modalities.

Screen Space Utilization: Another key-issue in visualization for mobile device relates to the efficient use of the available screen space. *Coordinated Multiple View* systems allow showing heterogeneous data sets by distributing data over different views, but consume high screen real estate and require increased user interaction. Therefore, new visual mappings need to be devised that visually integrate key aspects of the data in a single display.

The rapidly growing popularity of mobile devices for every aspect of work and pleasure, as well as the growing number of open data sources fuel the need for new forms of visualization. In addition, the role of aesthetics gains attention with the increasingly important role of mobile devices for a general audience. Applying competencies from traditional graphic design, typography, or perception theory to the presentation of complex data visualizations could additionally increase their usability and hence their success in the real world.

Mobile Devices and Visualization: Mobile devices present an additional challenge for visualization in terms of security and consumability at all stages of the knowledge discovery process. It is not just about how we store and transmit data, mobile devices also change the way customers think about data. Mobile devices have increased the number of user types and tasks, and also users' expectations to learn fast and accurately from information. As a result, customers expect engaging and aesthetic displays with intuitive interaction techniques to explore their data and discover new insights.

VI. CONCLUSION

We presented a vision of the next-generation location-based services (LBS 2.0) and discussed in detail the challenges pertaining to acquiring, storing, managing and visualizing multivariate spatio-temporal data. We believe this will open exciting research areas and opportunities for building engaging applications and services.

REFERENCES

- [1] "Cisco Visual Networking Index 2013-2018," <http://bit.ly/CiscoVNI2013>.
- [2] "Gartner Press Release," <http://www.gartner.com/newsroom/id/2194115>.
- [3] "Glass Development Kit," <http://bit.ly/GlassSDKSensors>.
- [4] "Samsung Galaxy S3 Specifications," <http://bit.ly/sg3specs>.
- [5] Z. Yan, *et al.*, "Energy-efficient continuous activity recognition on mobile phones: An activity-adaptive approach," in *International Symposium on Wearable Computers*, 2012, pp. 17–24.
- [6] "Gizmag Article," <http://bit.ly/GizMagSmartClothing>.
- [7] S. Sathe, *et al.*, "EnviroMeter: A platform for querying community-sensed data," *PVLDB*, pp. 1294–1297, 2013.
- [8] "Facebook Quarterly Earnings Report," <http://bit.ly/FBQ313Earnings>.
- [9] "Technology adoption lifecycle," <http://bit.ly/TeAdLc>.
- [10] D. Beaver, *et al.*, "Finding a needle in haystack: Facebook's photo storage." in *OSDI*, 2010, pp. 1–8.
- [11] J. Dean and S. Ghemawat, "MapReduce : Simplified data processing on large clusters," in *CACM*, 2008, pp. 107–113.
- [12] A. Deshpande and S. Madden, "MauveDB: Supporting model-based user views in database systems," in *SIGMOD*, 2006, pp. 73–84.
- [13] K. Shvachko, "HDFS scalability : The limits to growth," *USENIX login*, pp. 6–16, 2010.
- [14] J. Corbett, *et al.*, "Spanner: Google's Globally-Distributed Database," *TOCS*, vol. 31, no. 3, pp. 1–14, 2012.
- [15] S. Gandhi, *et al.*, "GAMPS: Compressing multi sensor data by grouping and amplitude scaling," in *SIGMOD*, 2009, pp. 771–783.
- [16] J.-g. Lee, *et al.*, "Trajectory Clustering: A partition-and-group framework," in *SIGMOD*, 2007, pp. 593–604.
- [17] M. Potamias, *et al.*, "Online amnesic summarization of streaming locations," in *Advances in Spatial and Temporal Databases*, ser. LNCS. Springer, 2007, pp. 148–166.
- [18] M. Ward, *et al.*, *Interactive data visualization: Foundations, techniques, and applications*. AK Peters Ltd., 2010.
- [19] W. Aigner, *et al.*, *Visualization of time-oriented data*. Springer, 2011.
- [20] D. A. Keim, *et al.*, *Mastering the information age: Solving problems with visual analytics*. Florian Mansmann, 2010.
- [21] G. Andrienko, *et al.*, *Visual analytics of movement*. Springer, 2013.