International Journal of Trend in Scientific Research and Development (IJTSRD)
International Open Access Journal
ISSN No: 2456 - 6470 | www.ijtsrd.com | Volume - 2 | Issue – 4

CarStream: An Industrial System of Big Data Processing for Internet of Vehicles

Rakshitha K. S[1], Radhika K. R[2]
[1]Masters in Technology, [2]Assistant Professor
Department of Computer Science & Engineering, BMSIT & M Yelahanka,
Bengaluru, Karnataka, India

ABSTRACT:
As the Internet-of-Vehicles (IoV) technology becomes an increasingly important trend for future transportation, de-signing large-scale IoV systems has become a critical task that aims to process big data uploaded by fleet vehicles and to provide data-driven services. The IoV data, especially high-frequency vehicle statuses (e.g., location, engine parameters), are characterized as large volume with a low density of value and low data quality. Such characteristics pose challenges for developing real-time applications based on such data. In this paper, we address the challenges in de-signing a scalable IoV system by describing CarStream, an industrial system of big data processing for chauffeured car services.

Photon is deployed within Google Advertising System to join data streams such as web search queries and user clicks on advertisements. It produces joined logs that are used to derive key business metrics, including billing for advertisers. Our production deployment processes millions of events per minute at peak with an average end-to-end latency of less than 10 seconds. We also present challenges and solutions in maintaining large persistent state across geographically distant locations, and highlight the design principles that emerged from our experience.

Keywords: *Internet of vehicles, Carstream, Fleet Management, Three layer monitoring, Streaming.*

1. INTRODUCTION
The cloud-based IoV has bene ted from the fast development of mobile networking and big data technologies.Different technologies traditional vehicle networking technologies, which focus on vehicle-to-vehicle (V2V) communication and vehicular net-works in a typical cloud-based IoV scenario, the vehicles are connected to the cloud data center and upload vehicle statuses to the center through wireless communications. The cloud collects and analyzes the uploaded data and sends the value-added information back to the vehicles.

To build these services, it is necessary to collect tremendous amounts of data about users' activities and movement patterns in different locations. In the context of our CarTel telematics infrastructure (see http://cartel.csail.mit.edu/) we have been focusing on road-usage, with a goal of reporting on traffic data as well as allowing individual users to browse and mine their driving patterns to detect inefficiencies, recommend alternative routes, find carpools, or detect ailing cars. For the past four years we have been continuously collecting speed, position, altitude, as well as a variety of sensor data, including accelerometer traces and data from the on board diagnostic system (called OBD-II, standard in all US cars since 1996) from a collection of 30 taxi cabs and 15 individual users' cars in the Boston metropolitan area. At this point, our database consists of about 200 million GPS readings, representing tens of thousands of drives and more than 68,000 hours of driving.

2. BACKGROUND AND RELATED WORK
To address the concern with very large trajectories, several systems have proposed segmenting trajectories to reduce the sizes of bounding boxes and group portions of trajectories that are near each other in space together on disk. Hadoop and batch analytics. A number of large Hadoop clusters comprise the core

of Twitter's data warehouse. Data are imported from a variety of sources, including structured databases (e.g., user pro les and the interest graph), un-structured text (e.g., Tweets), and semi-structured inter-action logs For analytics and building data products, data scientists typically use a higher-level data ow language such as Pig or Scalding (Twitter's Scala API to Cascading). This is a mature production system.

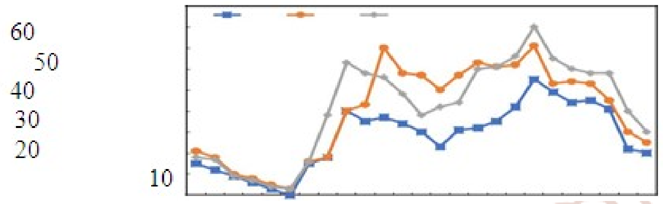
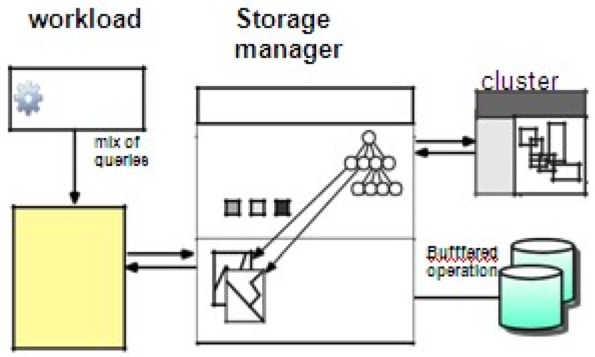
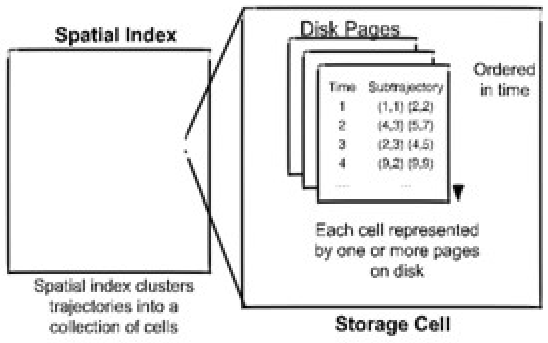


Fig 1:The data amount changes with time

A special characteristic of the data stream in CarStream is that the data stream is subjected to the track pattern: the data amount uploaded in peak hours can be as much as 80 times larger than that are in o -peak hours. For this reason, CarStream also needs to deal with burst streams. Figure 1 illustrates the data amount of 24 hours in three days, and clearly shows a tracklike pattern with morning and evening peak hours.

Based on the data, CarStream provides multiple functionalities and applications to facilitate the fleet management for chauffeured car service.





Disk query processor

Fig 2:Traj store architecture

Fig 3:Basic storage structure

3. ARCHITECTURE

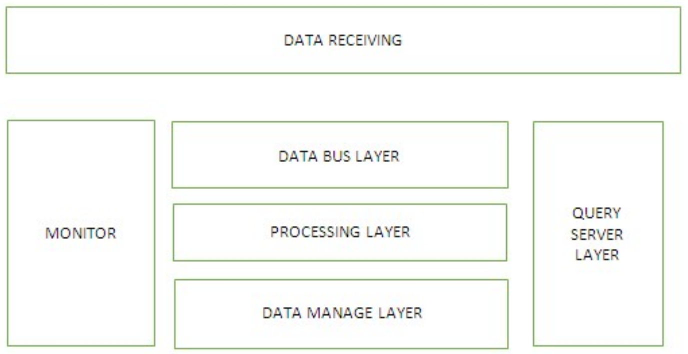
Based on the aforementioned challenges and the requirements, we design CarStream, a high performance big data analytics system for IoV. In this section, we present the overview of CarStream and its comparison with other possible solutions. As illustrated in Figure 4, CarStream is an integrated system with multiple layers involved. The components that are marked by the dotted line are the core parts of CarStream and therefore are discussed in detail in this paper.

3. 1 Data Bus Layer

The data bus layer provides a data-exchange hub for system modules to connect different computing tasks. This layer is implemented with a distributed messaging platform with fault-tolerant, high performance, and horizontally scalable ability.

3. 2 Processing Layer

The processing layer includes two parts: the online stream-processing subsystem and the online batch-processing sub-system. The stream processing subsystem is designed to provide a preprocessing functionality to the streams in a way that the query pressure of the database can be reduced. Without preprocessing, the applications based on the raw data would bring huge read pressure to the database.



1. Figure 5:The system architecture of carstream.

3.3 Data Management Layer

The data management layer adopts a heterogeneous structure constructed with multiple databases. In CarStream, both NOSQL and SQL databases are adopted to achieve high performance. The design decision is based on a comprehensive analysis of the IoV applications. We store the data in different storage systems per the access requirements of the applications.

4. EXISTING SYSTEM

The underlying system supporting such services can be regarded as a simplified version of IoV where smart phones, instead of dedicated sensors, are used to connect passengers, drivers, and vehicles. This technology has given rise to a huge market targeted by many similar companies established in recent years. To implement an IoV platform, the basic functional requirements should at least include vehicle-data collection, storage, analysis, and services. Vehicles Uber, Lyft can upload driving data, including instant locations and vehicle-engine statuses, to the backend servers. The data can be used in both real-time and online vehicle-management applications.

In addition to the major pain of, essentially, writing everything twice (once for batch processing and once for online processing), there was no standard online processing frame-work at Twitter until recently. The systems for counting events in real-time were responsible only for gathering signals and ordered little support in helping a client manipulate and process them. Over the past several years, the result has been a proliferation of custom one-o processing engines for various specialized tasks.

A good example that illustrates all these issues is described in a previous paper about Twitter's real-time related query suggestion architecture.

5. PROPOSED SYSTEM

1: Application-level monitoring is necessary for achieving high reliability.

Infrastructure monitoring and computing-platform[12] monitoring are the most commonly adopted echniques for assuring system reliability. Doing so can allow us to take timely reactions when a problem occurs. We can also use the monitoring information to help to utilize system re-sources by adjusting the deployment. However, we learn through the maintaining of CarStream that infrastructure-level and platform-level monitoring are insufficient to pro-vide highly dependable services, especially in safety-critical scenarios.

2: Low data quality widely exists in IoV. A fixing model extracted from historical data patterns can be helpful.

According to our experience, IoV applications usually face a severe issue of low data quality, such as data loss, data disorder, data delay, insufficient client data, and wrong data, etc. There are multiple causes of those problems. For exam-plea, the data disorder can be attributed to the distributed processing mode of the system. Because the data are pro-cessed concurrently so that later data are possible to be processed earlier than the earlier-arrived data, resulting in a disordered sequence.

Category	Problem	Consequences	Cause	Solution
Lack of data	1.Data loss 2.Insufficient Data	1.No result 2.Inaccurate result	1.software failure 2.Physical limitations	1.Redundant deployment 2.interpolation by data patterns
Wrong data	1.Disorder 2.outlier data	1.Wrong result 2.Wrong result	1.Distributed nature of processing platform. 2.Hardware malfunction	1.Delay and wait. Fixing the prediction. 2.outlier detection and data cleaning

Table 1 :Common data quality issues in iov scenarios

3: In large-scale scenarios, the linked-queue-based sliding-window processing may cause a performance issue.

In stream-processing applications, the sliding window is one of the most commonly used functionalities. Typically, the sliding window is implemented with a linked queue. A new data is appended to the queue when the data comes. Meanwhile, when the queue has reached the maximum length, the oldest data will be removed.

4: A single storage is usually insufficient; a heterogeneous storage architecture is necessary for managing large scale vehicle data.

As a typical big data processing scenario, IoV needs to manage a huge quantity of vehicle data. In this scenario, data management faces severe challenges: the data volume is huge, and the applications are of variety such that different data-access requirements need to be satisfied. For ex-ample, decision-making-related applications require a large throughput of the data access, while time-critical applications require higher data-access performance. Based on our experience, there might not be a one-for-all storage platform that satisfies the requirements of all the applications, and a heterogeneous storage architecture is usually necessary for maximizing the system performance.

5: Single platform may not be sufficient for multiple requirements in IoV.

Simplicity is an important principle in system design: the bene t of a simple design includes ease of maintenance, up-grade, and use. For general computing system, high performance often comes with simplicity of design.

6.CONCLUSION

In this paper, we have described our experience on ad-dressing the challenges in designing CarStream, an industrial system of big data processing for IoV, and the experience of constructing multiple data-driven applications for chauffeured car services based on this system. CarStream provides high-dependability assurance for safety-critical services TrajStore, a new scheme for indexing, clustering, and storing trajectory data. TrajStore is optimized for relatively large spatio-temporal queries retrieving data about many trajectories passing through a particular location. Our approach works by using an adaptive gridding scheme that partitions space into a number of grid cells, and adaptively splits or merges cells as new data is added or the query size changes.

7. REFERENCES

1) Lyft engineering blog. https://eng.lyft.com/. Accessed: 2017-02-28.

2) Uber engineering blog. https://eng.uber.com/. Accessed: 2017-02-28.

3) A. Aji, F. Wang, H. Vo, R. Lee, Q. Liu, X. Zhang, and Saltz. Hadoop GIS: A high performance spatial data warehousing system over mapreduce. In Proc. VLDB Endow., 6(11):1009{1020, 2013}.

4) A. Arasu, B. Babcock, S. Babu, J. Cieslewicz, Datar, K. Ito, R. Motwani, U. Srivastava, and Widom. STREAM: The Stanford data stream management system. In Data Stream Management, pages 317{336. 2016.

5) L. Neumeyer, B. Robbins, A. Nair, and A. Kesari. S4: Distributed stream computing platform. In Proc. IEEE International Conference on Data Mining Workshops (ICDMW), pages 170{177, 2010.

6) S. Rasetic, J. Sander, J. Elding, and M. A. Nascimento, “A Trajectory Splitting Model for Efficient Spatio-Temporal Indexing,” in VLDB, 2010.

7) M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, “OPTICS : Ordering Points to Identify the Clustering Structure,” in SIGMOD, 1999.

8) Y. Tao and D. Papadias, “The mv3r-tree: A spatio-temporal access method for times-tamp and interval queries,” in VLDB, 2001

9) B.-U. Pagel, H.-W. Six, H. Toben, and P. Widmayer, “Towards an Analysis of Range Query Performance in Spatial Data Structures,” in PODS, 1993.

10) J. Lin and D. Ryaboy. Scaling big data mining infrastructure: The Twitter experience. SIGKDD Explorations, 14(2):6{19, 2012.}

11) E. Meijer and G. Bierman. A co-relational model of data for large shared data banks. Communications of the ACM, 54(4):49{58, 2011}.

12) W. Lam, L. Liu, S. Prasad, A. Rajaraman, Z. Vacheri, and A. Doan. Muppet: MapReduce-style processing of fast data. VLDB, 2012.

13) H. Cao, O. Wolfson, and G. Trajcevski, “Spatio-temporal Data Reduction with Deterministic Error Bounds,” in DIALM-POMC, 2003.

14) N. Meratnia and R. By, “Spatiotemporal Compression Techniques for Moving Point Objects,” in EDBT, 2004.

15) B.-U. Pagel, H.-W. Six, H. Toben, and P. Widmayer, “Towards an Analysis of Range Query Performance in Spatial Data Structures,” in PODS, 1993.