International Journal of Innovative Research in Education, Technology & Social Strategies
p-ISSN: 2465-7298 | e-ISSN: 2467-8163
Volume 11, Number 1 August, 2024

IJIRETSS

A Framework for Improving the Location-Based Service Using Casandra Technology

[1]B. Temuujin, [2]Jaewon Park, & [3]Eui-In Choi
[1,2&3]Department of Computer Engineering,
Hannam University, Daejeon, Korea

Article DOI: 10.48028/iiprds/ijiretss.v11.i1.02

Abstract

Recently, many researches on positioning technology using LBS (Location-Based Services) have been conducted with the development of wearable devices. In addition, the data used in these devices is helping to perform LBS using Big data technology. And the existing method of finding a specific location is not suitable for collecting and processing all the data. Therefore, in order to process all streaming data in real time, it is necessary to use Big data processing technology. In this paper, we use NoSQL technology to solve this problem. We propose a framework for improving the performance of LBS using NoSQL.

Keywords: *Big Data, NoSQL, Cassandra, LBS*

Corresponding Author: B. Temuujin

First Published: https://www.scirp.org/pdf/jcc_2019122615282676.pdf

Background to the Study

For LBS, most companies use wireless communication-based technology to provide related services. The basic method is technology such as LDT (Location Determination Technology). The GPS (Global Positioning System) and IPS (In-door Positioning System) information used here is calculated and provided in real time. GPS is often used to cover large areas of satellite or mobile base stations as a whole. IPS is mainly used in confined spaces (local) (Sun 2005).

For example, in parks or inside buildings that use ZigBee, RFID, CSS, UWB, Bluetooth and Wi-Fi environments. In particular, IPS uses Wi-Fi and sensors built into devices such as smartphones to improve accuracy. For example, it indicates where your current location is. This information must be sorted and stored through the information collector. In addition, the commands must be performed quickly while the information is sent to the location server. However, it is difficult to implement a smooth LBS due to complicated operations Therefore, big data processing technology is required for the processing of data collected from indoor and outdoor communication infrastructures. NoSQL can solve this problem (Choi 2003).

Between real-time data analysis and big data processing, the NoSQL database provides a storage and processing mechanism for data streaming. Cassandra, in particular, supports complex stream processing with scalability and high availa-bility without compromising performance on the open source approach. In this paper, we propose a framework for improving the performance of LBS with NoSQL, in particular Cassandra, using NoSQL system architecture.

Related Works

Positioning Services

Now we are living in the one of the great flows of technological change, Internet of Things. These techniques have been developed for identification of objects and its location. And it is being used to remote control and product marketing.

Beacon

Beacon is a short-range transmission technology that sends signals to smart-phones or wearable devices, using low energy Bluetooth. In particular, beacons can tell all the details of the local area to indoor GPS less than 5 cm from indoor or outdoor locations.

IZat

IZat is Qualcomm's IPS like a child with a Bluetooth beacon-based service has almost similar features and services. IZat is kind of IPS, has Qualcomm's platform called Gimbal.

NoSQL

NoSQL uses a method other than the tabular relationships used in relational databases. Use it to store and retrieve modeled data. This method includes control over design simplicity, horizontal scaling and availability. NoSQL data structures (such as key-values, graphs, or documents) are different from RDBMSs. Therefore, some tasks are processed faster in

NoSQL than in some RDBMSs [11]. NoSQL satisfies two of the CAP theorems. First, consistency means that all nodes see the same data at the same time. Secondly, availability ensures that every request gets a response, whether it is a success or a failure. Finally, partition tolerance means that the system will continue to operate in case of any message loss or error. NoSQL is suitable for tasks that require high scalability to store and process the generated stream data.

Key-Value Store

Horizontally scalable key-value storage is the simplest form of NoSQL. And because it provides only API, it is easy to use. The disadvantage is that you cannot query using the contents of the key. Therefore, the user has to read the value using a key and then deal with it at the application level. Key-value storage is where data is stored in key and value pairs. Here keys are used to access values, and values are used to store some form of data (Zhao 2002). Key-value can be prior for data search by a single key and it provides the availability of information regardless of any platform.

Column Store

Column stores also as a kind of key-value store, each record is independent. When the query pattern is subset of the whole record, it is possible to form column families in the same way. Cassandra can be applied. It made based on dis-tributed architecture called Dynamo. It is to solve this problem that when the existing head node is down, the entire system also down. It can be stored more data no matter what data format. In addition, Distributed processing make it loss SPOF (Single Point of Failure) and prevent data loss. (Lee 2002).

Document Store

The document store is a special type of key-value store. In document-oriented databases, it is used to store, retrieve, and manage document-oriented information. Therefore, this requires semi-structured data. Document-oriented databases can be thought of as a category of NoSQL databases. All of the document repositories contain documents that encapsulate and encode in standard format or encoded data. Common coding uses binary formats such as XML, YAML, JSON, and BSON. Documents in a document repository are almost like object-oriented programming concepts. Therefore, there is no need to comply with the standard schema.

Cassandra

Moving object location's data stream information is really useful in variety applications. The existing process of position information streaming data has been collecting, pre-processing and finally analyzing. However, additional techniques are required to handle collected data from the GPS and indoor wireless communication infrastructure for improve the performance of the outdoor and indoor positioning techniques using streaming processing. These processes of massive amounts of unstructured data can be solved by big data processing techniques. Cassandra starts as one of the Apache incubators and became a top-level Apache project after two years. The basic design concept is each node is using gossip-based algorithms which can find out failed node just by state replacement without master node in pure-p2p protocol. Of course, a new node also can be joining to new group. When system fails, it can prevent single point of failure in advance as master nodes were not existed. Additionally, it can control its data eventually consistent level. For example, Cassandra's processing speed of the data is improving linearly while minimizing the hard disk and memory cost in p2p networks by spreading position data which coming from a variety of sources process on a large scale. This environment provides superior LBS performance consequentially. Another benefit provided by Cassandra is the authority to decide what data will be copied or how much will be. The entire row in Cassandra is recognized by the unique keys and there is no size limit on those keys. Secondly, Cassandra instance is taking a table, defined column family by users. It may be a super column by configurable infinite column. The Cassandra's basic philosophy of the error handling, data distribution and replication as follows. Data distribution is based on order preserving hash and consistent hashing through nodes in the cluster basically. Cluster membership can be maintained by gossip algorithms and monitor the error of each node using actual style failure detector.

Proposed Framework

Figure 1 shows a framework using Cassandra to process LBS streaming data. Stream data processing components are provided to ease the development of various LBSs. It shows the overall system structure, consisting of key spaces, column families, and columns. Cassandra's SSTable (Sorted String Table) is a file of key/value string pairs sorted by key. Each SSTable has a bloom filter that Cassandra needs to check before disk scanning. You can also freely query for keys that rarely exist. Cassandra uses bloom filters to save IO when performing a key lookup. Compact facility is also useful. Since data files accumulate over time periodically data files are merged sorted into a new file and creates new index (Maria 2013). This frame-work based on Cassandra configures clustering structure using numerous nodes. Each node is operating independently without a separate master server. Further, Cassandra support the column compressed store file format called ORC file. It is processing streaming data, one of kind complex event processing (CEP). It can be applied to stream data analysis quickly without additional infrastructure. In the process of raw data stream converting for real time analysis, we can com-press its size in order to reduce the processing load.

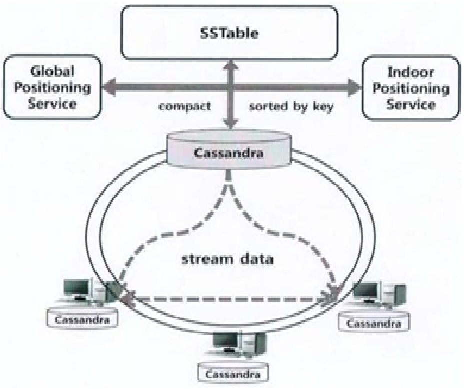

Figure 1: Cassandra framework for location-based services (LBS).

Conclusion

Efficient LBS service requires processing of large data streams. In addition, processing delays and high computational complexity made it difficult to implement a variety of existing positioning techniques. This makes LBS less flexible. As a result, NoSQL environments can handle large amounts of unstructured data. In particular, Cassandra is based on a larger framework such as Dynamo, which enables consistent event handling. Larger frameworks based on big table column for column-based key-value systems are also possible. Apply flexible da-ta consistency to focus on stream data processing availability. In addition, a ring structure other than the master-slave architecture is chosen to ensure reliable system operation with fault tolerance. So, we expect that it will support monitoring stream and processing LBS for across in-and-out of space. By distributing positioning data in many paths and processing a large-scale, it can provide LBS with great performance. And as a result, we expect that is able to contribute to improving the in-and-outdoor positioning accuracy of the mobile communication network. In order to compensate for the shortcomings of the current study, the following research will be developed a prototype as a data repository for monitoring, and that will be showing the proposed model to evaluate the performance of a streaming data service model.

References

Choi, H. (2003). *LBS, Location-based Services Trend Computer Software Research*, Korea Electronics and Telecommunications Research Institute, No.86.

Lee, G. (2002). Development of location data Stream Professor for RTLS, *Korea GIS Conference*, No. 6, 15-20.

Maria, C. (2013) Managing Service Performance in the Cassandra Distributed Storage System. *IEEE 5th International Conference on Cloud Computing Technology and Science*, 64-71.

Sun, G., et al. (2005) Signal Processing Techniques in Network Aided Positioning, *IEEE Signal Processing Mag.*, **22**, 12-23.

Zhao, Y. (2002) Standardization of Mobile Phone Positioning for 3G Systems, *IEEE Communication. Mag.*, 40, 108-116.