Journal of Engineering and Applied Sciences 11 (8): 1855-1858, 2016
ISSN: 1816-949X

An Efficient Distributed Data Processing Method for Smooth Environment

[1]E. Seshatheriand and [2]T. Bhuvaneswari
[1]Manonmaniam Sundaranar University, Tirunelvel, Tamil Nadu, India
[2]Department of Computer Applications,
Queen Mary's College (Autonomous), Chennai, Tamil Nadu, India

Abstract: In current times, huge volume of data at a very high velocity gets produced through social media and various sensors in embedded systems that are associated to the internet which causes a very big data problem. These challenging big data's need to beprocessed and stored by traditional Relational Database Management Systems (RDBMS). Due to this motive, the need for new software solutions has occurred for managing the big data in an efficient, scalable and cool way. In this study, an approach to combine the concept of batch processing and stream processing to an end where it can query the data set which also supports adhoc querying with less latency that can be run on any large scale machine learning algorithms for recognizing any interest pattern in the streaming data set was employed. The functionalities of Hadoop ecosystem 's tool HIVE can also be used to produce the results to ad hoc queries, User Defined Functions (UDF) similar to writing a SQL stored procedures in the spark system. An interface with serdes which is serialization and de-serialization that helps us to talk to the standard stream where it can exactly query the dataset are employed. By proposing a new software solution AllJoyn Lambda in which AllJoyn is integrated in the lambda architecture and the prototype implementation of the architecture is done using Apache Hadoop Yarn over Apache Spark Streaming are presented. This study light up the high velocity streaming data set on a database without losing any data from the streaming domain, to support adhoc querying from the data set and to provide a mechanism for fast data processing and analytics using large scale machine learning. This research study highlights the analysis of large scale dataset processing, handling challenges and its comprehensive systematic review. From this study, here it conclude that building a smart environment by using the big data setup platform improves and enhances the results for the smart environment.

Key words: AllJoyn lambda architecture, big data analytics, internet of things, smart environment, spark streaming

INTRODUCTION

In engineering sector, database management is gaining interest and achieving importance day by day becoming tremendously important. So many researchers have worked on this subject and published number of research works and reports. Therefore, benefits of this field need to beutilizing fully by knowing its efficiency and state-of-the-art. Relational Database Management Systems (RDBMS) which are ideally suited for storing structured data have been used main in the past decade. However in the present day world, users and devices often generate a lot ofsemi-structured and unstructured data in various formats that as well as at a very fast rate. In this context, it is essential to provide a distributed fault tolerant data architecture that satisfies the following requirements: To store the high velocity streaming data set on a database without losing any data from the streaming domain, to support Adhoc querying from the data set and to provide a mechanism for fast data processing and analytics using large-scale machine learning. In recent past many researchers have studied and worked on big data analytics by using various technologies on Hadoop platform. For example, Losup, Ghit and Epema reported processing of big data more efficiently by optimizing the MapReduce processing further to process huge amount of data with MapReduce programming model (Chen *et al*., 2012). Suzumura studied and worked on studies related to processing big data with standard sets of data by various applications (Jabez and Muthakumar, 2014). Semi structured cloud storage, non-structural storage and distributed file system was employed for big data processing in cloud computing environments were explored by Ji (Manyika *et al*., 2011).

Corresponding Author: E. Seshatheriand, Manonmaniam Sundaranar University, Tirunelvel, Tamil Nadu, India

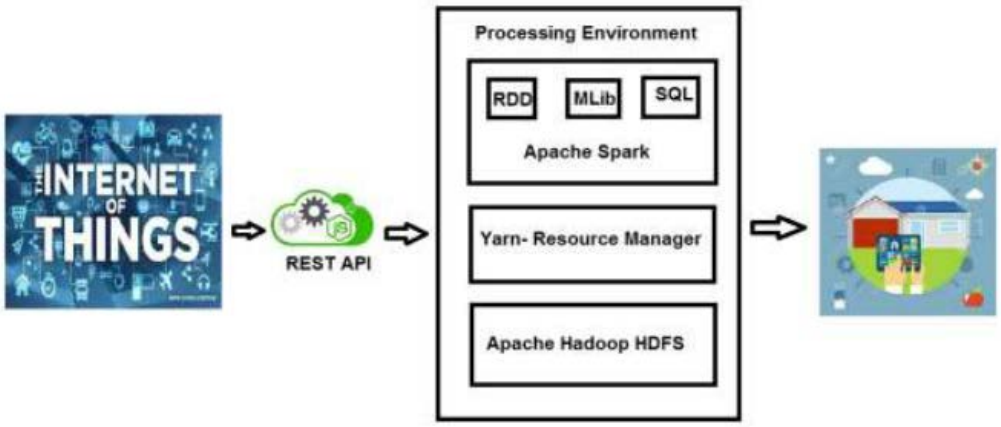

Fig. 1: System architecture

The researchers Abhinandan and Kumar (2016) worked on "Shared disk big data analytics with Apache Hadoop " in which they retrieved valid information for huge data and uncover the hidden patterns by MapReduce framework. The researchers of (Zaharia *et al*., 2012) reported "Addressing Big Data Problem Using Hadoop and MapReduce" in which optimal solutions using MapReduce programming architecture for processing volumes of data and Hadoop cluster for storage were experimentally worked. To meet the requirements like processing large scale dataset, dataset handling challenges, a new software solution, AllJoyn lambda which is integration of AllJoyn in the lambda architecture has emerged. AllJoyn technology being used for executing various applications across heterogenous embedded devices, providing fascinating solutions for mobility, networking, security and dynamic configuration issues, it is not sufficient to handle complex smart environment and also communication among devices belonging to different broadcast domains is not being supported and also it does not provide a feature for Bigdata analytics and storage.

In order to overcome the shortcomings of the AllJoyn, a new scalable software solution has emerged by integrating AllJoyn in the lambda architecture which is presented in this study. Hence, it enables Bigdata storage, processing and real time analytics.

This research project addresses the designed/ proposed software architecture which can be adapted to the evolution of a smart environment by its activities. Also, it can be well fitted into the mutiple IoT smart environments enabling the production of different context-aware applications and services.

MATERIALS AND METHODS

Relational data processing in spark: spark SQL: Spark SQL can act as a distributed SQL query engine and provides a programming abstraction called dataframes (Nirmalrani and Sakthivel, 2015a, b). A collection of distributed data which are organized into named columns is called data frames. With higher optimizations it can be relevant to a data frame in R/Python or table in a relational database. Huge number of sources like structured data files hive tables, external databases or existing RDDs are used to construct data frames. Optimization rules and data sources acts as a catalyst to speed up the evaluation of data frame API performing relational operations on both built-in collections of spark and external data sources. High memory-efficient can be achieved by Spark SQL providing a columnar store for many aggregates than naive Spark code in computations expressible in SQL. This architecture shows that the database can be queried with the functionalities of hadoop ecosystems tool. Hive can also be used to produce the results to ad hoc queries, User Defined Functions (UDF) similar to writing a SQL stored procedures in the Spark system. The interface with SerDes which is serialization and de-serialization helps to talk to the standard stream where the dataset can be exactly queried. Figure 1 shows the system architecture.

Large scale smart environments management by Alljoyn lambda architecture: AllJoyn framework plays a significant role in enhancing and developing IoT systems which aimed at the interoperability among heterogeneous devices, creation of dynamic proximal network and execution of distributed applications. AllJoyn is an open source project which enables the development of multiple

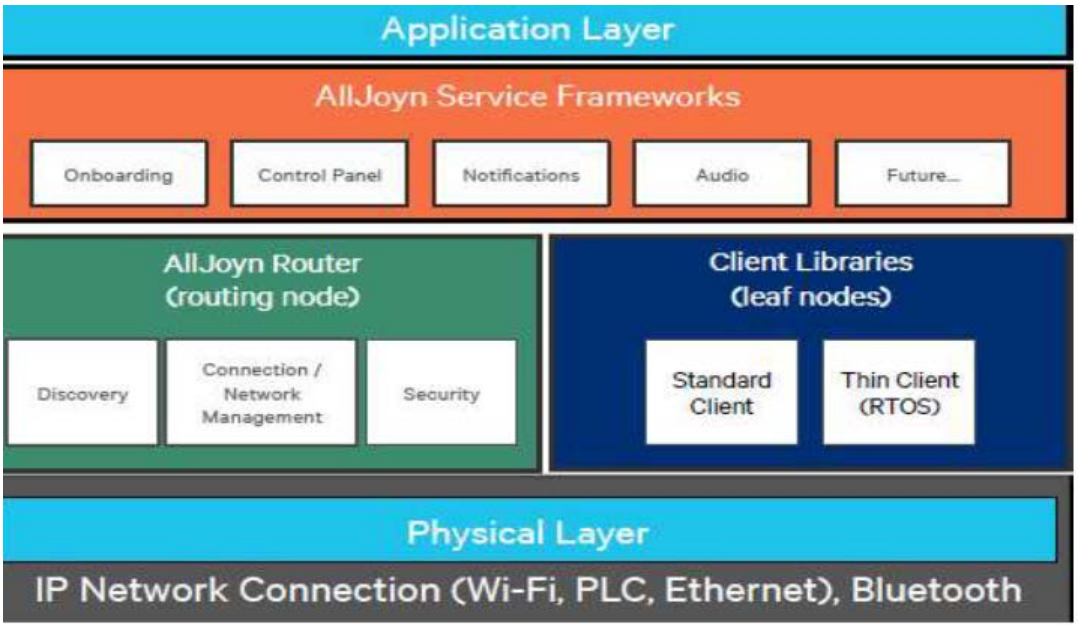

Fig. 2: AllJoyn architecture

applications like sharing media, proximal applications, chat rooms, multi-player games, domestic and social applications on looking forward at the evolution of IoT by providing a common interface towards smart devices. Important functions offered by AllJoyn are listed below:

Ability to adapt the framework to specific devices, Transferring data between devices through Bluetooth, Wi-Fi and other communication technologies. Efficient and secure exchange of data through D-Bus: Interoperability between different operating systems. AllJoyn architecture is presented in Fig. 2.

Though, the physical layer of AllJoyn architecture shows that it is responsible to manage IP connections, using different communication technologies like Bluetooth, Wi-Fi, it is not enough to manage complex smart IoT environments. The number of devices and information acquisition frequency increase, data management becomes quite tedious because AllJoyn does not support communications among devices belonging to different broadcast domains and does not provide any feature for storage of large amount of data and real time analytics which is called as big data problem. Due to these limitations of AllJoyn, it is not able to address two main challenges of the IoT which includes large-scale smart environment management and big data storage and analytics (Saravanan and Sailakshmi, 2015). To overcome the shortcomings of AllJoyn, this study propose a new scalable solution by integrating AllJoyn system in the Lamda architecture, thus enabling big data storage, processing and real-time analytics9.

The lambda architecture which is said to be distributed and scalable architecture is based on three principles: Fault tolerance because of hardware failures and error caused by humans the system does not loose its data. Data immutability Immutable data are stored permanently by the system, deleting and updating tasks on data are not allowed. Recomputation. For each query data is recomputed due to fault-tolerance and immutability. There are two levels of latency in three level lambda architecture: speed layer (corresponding to online); batch layer (corresponding to offline).

RESULTS AND DISCUSSION

Batch layer: Batch layer is responsible to store the immutable datasets, to produce data views and in constant growth. The whole master datasets are processed as on the arrival of new data.

Speed layer: It is responsible to produce the views of real time data uses an incremental model and data views are temporary, infact as each data are forwarded to batch layer, it gets deleted. Developing architecture to the smart environment in IoT, according to the lambda architectural model integration of AllJoyn system with storm is done for processing real time data. This architecture supports MongoDB for bulk storage and processing of data by three different types of data patterns which are given as.

Regular patterns: Many embedded devices running AJTC applications which needs to be repeated at intervals by set of actions comes under regular patterns.

Event based patterns: It has a set of action performed by several devices according to particular events, e.g., keep windows open on detection of smoke by sensors.

Automated patterns: Set of actions that are triggered by the complex algorithms and not configured by users. In this, priority is given to each pattern (Villari *et al*., 2014), e.g., movement of curtain when it is exposed to light.

Discretized streams: Stream processing by structuring computations as a set of short, challenging tasks instead of continuous operators can be avoided by D-streams. Later, the state in memory across tasks as fault-tolerant data structures that can be recomputed deterministically and stored by them (Nimralrani and Sakhtivel, 2015a, b). Degradation of computations into short tasks results in exposing possibilities at a fine granularity and allows; Powerful recovery techniques like parallel recovery and speculation. Other than fall tolerance the D-stream model gives other benefits like powerful unification with batch processing (Litwin *et al*., 1996).

CONCLUSION

In this research project, two main challenges are discussed namely management of large-scale smart environment and big data storage/ analytics by proposing a new software solution, AllJoyn lambda, mainly to overcome the limitations of AllJoyn. Thus, the data generated at a very large scale can be stored and processed using the machine learning approach to manage the large scale environment efficiently.

REFERENCES

Abhinandan, B. and S.B. Kumar, 2016. Big data-a review on analyzing 3vs. J. Sci. Eng. Res., 3: 21-24.

Chen, Y., S. Alspaugh and R. Katz, 2012. Interactive analytical processing in big data systems: A cross-industry study of mapreduce workloads. Proc. VLDB. Endowment, 5: 1802-1813.

Jabez, J. and B. Muthukumar, 2014. Intrusion detection system: Time probability method and hyperbolic hopfield neural network. J. Theor. Appl. Inf. Technol., 67: 65-77.

Litwin, W., M.A. Neimat and D.A. Schneider, 1996. LH-a scalable, distributed data structure. ACM. Trans. Database Syst. (TODS.), 21: 480-525.

Manyika, J., M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh and A.H. Byers, 2011. Big Data: The Next Frontier for Innovation, Competition and Productivity. McKinsey Global Inst., USA., pp: 1-137.

Nirmalrani, V. and P. Sakthivel, 2015a. A hybrid access control model with multilevel authentication and delegation to protect the distributed resources. J. Pure Appl. Microbiol. (JPAM.), 9: 595-609.

Nirmalrani, V. and P. Sakthivel, 2015b. Framework for providing access to the web databases using budget aware role based access control. J. Theor. Appl. Inf. Technol. (JATIT), 76: 296-308.

Saravanan, P. and P. Sailakshmi, 2015. Missing value imputation using fuzzy possibilistic C means optimized with support vector regression and genetic algorithm. J. Theor. Appl. Inf. Technol. (JATIT.), 72: 34-39.

Villari, M., A. Celesti, M. Fazio and A. Puliafito, 2014. Alljoyn lambda: An architecture for the management of smart environments in iot. Proceedings of the 2014 International Conference on Smart Computing Workshops (SMARTCOMP Workshops), November 5-5, 2014, IEEE, New York, USA., ISBN: 978-1-4799-6447-5, pp: 9-14.

Zaharia, M., M. Chowdhury, T. Das, A. Dave and J. Ma *et al*., 2012. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation, April 25-27, 2012, USENIX Association, Berkeley, California, pp: 2-2.