Comparative Analysis of Centralized Vs. Distributed Locality-based Repository over IoT-Enabled Big Data in Smart Grid Environment

Isma Farah Siddiqui[O], Asad Abbas, Scott Uk-Jin Lee
Dept. of Computer Science and Engineering, Hanyang University ERICA, South Korea
e-mail:{isma2012[O], asadabbas, scottlee}@hanyang.ac.kr

● 요 약 ●

This paper compares operational and network analysis of centralized and distributed repository for big data solutions in the IoT enabled Smart Grid environment. The comparative analysis clearly depicts that centralize repository consumes less memory consumption while distributed locality-based repository reduce network complexity issues than centralize repository in state-of-the-art Big Data Solution.

키워드: Big Data, IoT, Smart Grid.

I. Introduction

Big Data has evolved with new dimensions and analytical approaches for large repository solutions. Traditional data repositories had two limitations i.e. (i) flexible increment of increasing data elements to unlimited scope, and (ii) management of unmeasured scope data and their indices.

The Hadoop [1] emerged as a state-of-the-art solution to the discussed problems and gave a wide scope to resolve unmeasured data scope analysis. The Apache Hadoop is an ecosystem, that manage large size of dataset with unmeasured scope of processing capacity.

Internet of Things (IoT) [2][3] is a new paradigm that interconnect devices to the functional, operational and storage norms. IoT replaced many traditional paradigms i.e. traditional storage of data over limited scope repositories, inter-storage between multiple repositories and transformation of data to store between multiple repositories.

The smart grid is a new phenomena that has transformed traditional data processing to automated data processing. The grid has replaced traditional devices with IoT smart devices and enabled data management over Big Data profile.

Against this background, we have analyzed that grid repository works over two types i.e. (i) Centralized repository Jena [4], and (ii) Distributed locality-based Hbase [5][6] repository. We compared the two types and analysed that distributed locality based repository reduces network complexity while central repository consumes less memory. The main contributions of the papers are as follows:

Operational analysis of central and distributed locality-based repository.

Network complexity analysis of central and distributed locality-based repository.

The remaining paper is written as follows. Section II explain centralize repository. Section III depicts distributed locality based repository. Section IV discuss comparative analysis, and finally Section V presents the conclusion.

II. Centralized Repository

The centralized repositories include Jena TDB, SESAME, AllegroGraph and Oracle11g. The Jena repository is most suitable for our scenario because it stores dataset format of tuple and provide accessibility API of Hadoop which is not available in SESAME, AllegroGraph and Oracle11g.

1. Jena TDB

The Jena TDB is the most suitable processing repository to store smart grid dataset. It is categorized intro two sections i.e., (i) Dataset repository and (ii) TDB API set. The Dataset repository stores native RDF, non-transactional RDF and OWL data. It also supports extraction of data through SPARQL queries. The second section TDB API is a set of functional methods that provide functionality of storing and accessing data tuple from the repository and allow security protocols to provide secure transactions of million triples dataset.

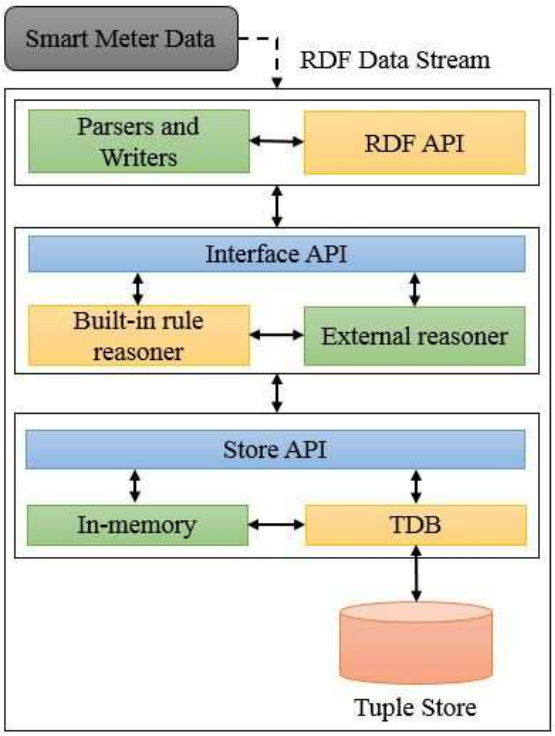

Fig. 1. Smart Grid Central Repository

When a sample of smart grid dataset is stored over Jena TDB, it is passed through multiple layers. The smart meter RDF data is parsed through RDF API.

The parsing prototype divide RDF tuple into ‘subject’, ‘object’ and ‘predicate’ format. The parsed dataset is processed over Interface API that place the above three elements over fields of ‘business rules’. The term ‘business rule’ relates a tuple to be given a complete set of reasoning rules to store in the repository.

The connecting tuple are given ‘external reasoning’ rules because of the outer bound communication arguments. The store API provides storage of rules-based dataset in two forms i.e. (i) In-memory, and (ii) TDB. The in-memory uses system memory to process tuple and store over tuple store and TDB provides traditional storage mechanism to store dataset over tuple store as seen from Fig. 1.

III. Distributed Locality-based Repository

The distributed locality-based repositories includes MapR, Cloudera and Hadoop Hbase. The Hadoop Hbase is most suitable because it is free of cost with open source functionality and provides module level independence to manipulate programs as per user modules.

Moreover, it is customized for IoT dataset management and works over key-value to RDF tuple transformation and independent accessibility rather than MapR and Cloudera systems.

1. Hadoop HBase

The Hadoop is an ecosystem that works over distributed processing environment. The Hadoop stores key-value data over HBase repository. The HBase repository stores dataset in tabular format. The table includes a hierarchy of storage dataset as seen from Fig. 2

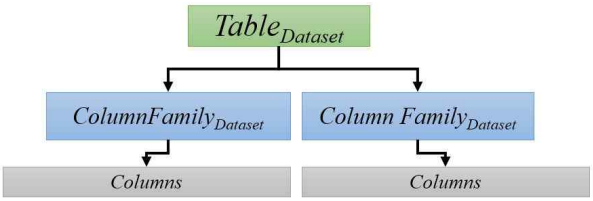

Fig. 2. HBase Repository Structure

As we know that, Hadoop is a distributed processing system that works over master and slave architecture. The HBase slave repository nodes are placed over distributed locality terminals at distributed end units. The locality-aware repository provides extra support to access large scale datasets through nearest accessing point. Therefore, when Hbase repository collects data information, it stores over locality-aware end units and accessibility becomes fast and convenient.

IV. Comparative Analysis of Centralized and Distributed locality-based Repository

The comparative analysis includes operational and network complexity features comparison.

1. Operational Analysis

The operational analysis includes factors involved while accessing repository API. The method of generating template for input dataset requires RDF instance. The Jena TDB consume '4' GB of memory while HBase consume '5' GB of memory as seen from Picture-3. The main factor of HBase consuming more memory than Jena TDB is locating mirror templates to locality ends.

2. Network complexity Analysis

The operational analysis includes factors involved while accessing repository API. The method of generating template for input dataset requires RDF instance. The Jena TDB consume '4' GB of memory while HBase consume '5' GB of memory as seen from Fig. 3. The main factor of HBase consuming more memory than Jena TDB is locating mirror templates to locality ends.

We perform NS2 simulations of forwarding a dataset worth '8' GB to the Jena TDB and HBase and calculate that HBase distribute the dataset in '2' equal locality-based end units and share the bandwidth while Jena TBD process dataset over single socket consuming twice the bandwidth, as seen from Fig. 4.

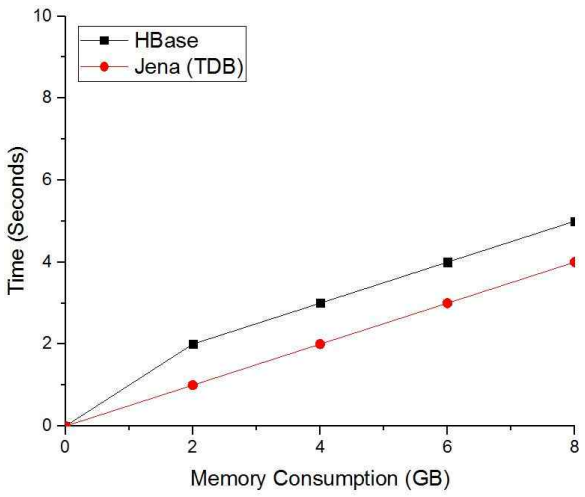

Fig. 3. Operational Consumption of Memory

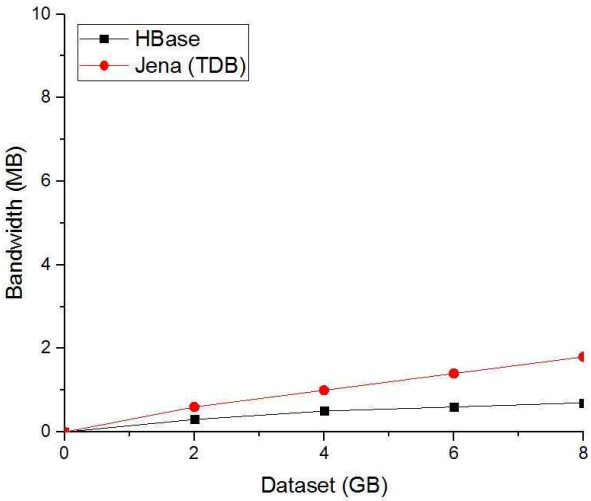

Fig. 4. Bandwidth consumption among Repositories

V. Conclusions

The comparative analysis among two repositories i.e. centralize and distributed locality-based, depicts a prominent difference of operational and network complexity paradigm. We evaluate that HBase is far better than Jena TDB in terms of network complexity reduction, while Jena TDB consumes less memory than HBase to process IoT-based dataset.

Acknowledgment

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korean government (MSIP) (No. NRF-2016R1C1B2008624).

References

[1] "Welcome to Apache Hadoop", 2014. [Online]. Available: [Online]. Available: http://hadoop.apache.org/. Accessed: Jan. 01, 2017.

[2] Whitmore, Andrew, Anurag Agarwal, and Li Da Xu. "The Internet of Things—A survey of topics and trends." Information Systems Frontiers 17.2 (2015): 261-274.

[3] I. F. SIDDIQUI, A. ABBAS, S. U.-J. LEE, "A hidden markov model to predict hot socket issue in smart grid", Journal of Theoretical and Applied Information Technology 31st December 2016, Vol. 94. No. 2 2016

[4] "Apache Jena,". [Online]. Available: http://jena.apache.org/. Accessed: Jan. 01, 2017.

[5] “HBase”. [Online]. Available: [Online]. Available: http://hbase.apache.org/. Accessed: Jan. 01, 2017.

[6] A. Peña and Y. K. Penya, "Distributed semantic repositories in smart grids," 2011 9th IEEE International Conference on Industrial Informatics.