International Journal of Engineering Applied Sciences and Technology, 2016
Vol. 1, Issue 3, ISSN No. 2455-2143, Pages 110-113
Published Online December - January 2016 in IJEAST (http://www.ijeast.com)

DATA LAKE FOR SOCIO-KNOWLEDGE PLATFORMS

Ankit Khanna
G.L. Bajaj Institute of Technology and Management

Hardik Bhatt
G.L. Bajaj Institute of Technology and Management

Hardik Marya
G.L. Bajaj Institute of Technology and Management

Rakshit Sakhuja
G.L. Bajaj Institute of Technology and Management

ABSTRACT - In the past decade, the evolution and rapid uptake of information technology, sensing, big data and information-based products and services has shifted the way in which people exchange information. However the rapid influx of new data presents overwhelming challenges to their data centers.
It explains that the increasing accessibility of information will enable us to develop a platform that is capable of helping and improving, data handling centers to see more deeply into how people use the application, how they feel about it, where the application faces problems, and what kinds of remediation can be applied.
This paper describes steps towards a platform that can boost the efficiency of data utilities and realize a social knowledge platform for applications like e-commerce, social networking websites and many more.
This paper discusses the building of such platform and addresses the looming concerns opening the pathway to a new era for data handling efficiency and reliability.

I. INTRODUCTION

All organizations have data, and in order to understand performance it is important to analyze that data. Many organizations rely on traditional data warehouse and business intelligence solutions to build a one-stop-shop for decision makers to access their reports and data [1].
But in a traditional data warehouse solution, we would probably ignore most of these external data sources because they are either too voluminous or in a format that is not easy to manipulate and store. In recent years, this data explosion has spawned a new set of technologies and techniques.
The Hadoop Data Lake and Apache Hadoop are at the center of the Big Data movement.
A Data Lake [2] is a storage repository that holds a vast amount of raw data in its native format until it is needed. While a hierarchical data warehouse stores data in files or folders, a data lake uses a flat architecture to store data. Each data element in a lake is assigned a unique identifier and tagged with a set of extended metadata tags. When a business question arises, the data lake can be queried for relevant data, and that smaller set of data can then be analyzed to help answer the question. The term Data Lake is often associated with Hadoop-oriented object storage. In such a scenario, an organization's data is first loaded into the Hadoop platform, and then business analytics and data mining tools are applied to the data where it resides on Hadoop's cluster nodes of commodity computers. Like big data, the term Data Lake is sometimes disparaged as being simply a marketing label for a product that supports Hadoop. Increasingly, however, the term is being accepted as a way to describe any large data pool in which the schema and data requirements are not defined until the data is queried.
A data lake should provide a number of fundamental capabilities, referred to as Lake Rules [3]. These are:
1. Host a centralized index of the inventory of data (and metadata) that is available, including sources, versioning, veracity and accuracy.
2. Securely authorize, audit and grant access to subsets of data.
3. Enable IT governance of what is in the data lake and assist enforcing policies for retention and disposition (and importantly tracking PII and PII pre-cursors).
4. Ensure data protection at scale, for operational availability and BC/DR requirements.
5. Provide agile analytics into and from the data lake using multiple analytical approaches (i.e., not just Hadoop) and data workflows. Some of the key characteristics of the Data Lake are:

1. **Use of multiple tools and products**. Extracting maximum value out of the Data Lake requires customized management and integration that are currently unavailable from any single open-source platform or commercial product vendor. The cross-engine integration necessary for a successful Data Lake requires multiple technology stacks that natively support structured, semi-structured, and unstructured data types.
2. **Domain specification**. The Data Lake must be tailored to the specific industry. A Data Lake customized for biomedical research would be significantly different from one tailored to financial

services. The Data Lake requires a business-aware data-locating capability that enables business users to find, explore, understand, and trust the data.

3. **Automated metadata management**. The Data Lake concept relies on capturing a robust set of attributes for every piece of content within the lake. Attributes like data lineage, data quality, and usage history are vital to usability. Maintaining this metadata requires a highly-automated metadata extraction, capture, and tracking facility. Without a high-degree of automated and mandatory metadata management, a Data Lake will rapidly become a Data Swamp.
4. **Integrate with the existing environment**. The Data Lake needs to meld into and support the existing enterprise data management paradigms, tools, and methods. It needs a supervisor that integrates and manages, when required, existing data management tools, such as data profiling, data mastering and cleansing, and data masking technologies.

The Data Lake can be an effective data management solution for advanced analytics experts and business users alike. A Data Lake allows users to analyze a large variety and volume when and how they want. Following a Data and Analytics as a Service (DAaaS) model provides users with on-demand, self-serve data. The main risk [4] of using data lakes is the absence of descriptive metadata and an underlying mechanism to maintain it, the lack of which can turn a data lake into a "data swamp".

II. HISTORY

Earlier, Databases were meant for structured data storage but with time and advancement in technology, most of the available today is unstructured or semi structured which needs to be addressed and based upon the insights of this unstructured data one can build what is needed by the customers increasing the value and expectations of the product. It's but obvious that Storage of Exabyte of unstructured data in schema based architecture is not feasible. Therefore, the interest in Big data is trending from several years, where main objective was to handle high volume data streams and when it comes to handle it ,then Big data Engineers approves the customer solutions which turns into conversation to Data Lakes. The data is growing at very high velocity where the essential data can be in any form and structure. Big data analysis is one of the proven methods to find the key insights from the unstructured data at scale, at low computational cost even at real time. It is useful as an initial place to process all kind of data. Hadoop Environment is developing capabilities for analyzing structured data and real time processing of streaming data.

The data lakes were earlier misunderstood as data warehouse since primary purpose is simply a storage repository. Data lakes are designed particularly for low cost storage of structured, semi-structured and unstructured data. These are agile as well as configured and reconfigured as needed. The Data Lake applies "schema-on-read" processing mechanism which refers as data is stored as it is and transformed when it is ready to use.

Banking is one of the crucial but excluded sector for handling through NoSQL databases since this sector have to work on strict schema based relation where high availability and consistency has to be maintained which can be satisfied by relational databases.

III. METHODOLOGY

In the spheres of progress and development, with the urge to automate the data in socio knowledge applications involves many challenges to take this initiative. With the availability of digital data growing in zeta bytes proves to have the power to overcome the challenges possible.

Hence with the real time insights available, there is great possibility to generate things smarter and better with the complete utilization of resources. This transition of databases to smart platforms is definitely for better efficiency and reliability.

3.1. Implementation

We are working to address data handling through a platform that will provide diversification of data to be handled through various modules for efficient use.

Many different modules are possible and can be implemented. These may include:

3.1.1 2-Step Authentication

Considering the security risks, now every socio-Knowledge platform is using Two-Step Authentication which adds an additional security layer to your account. Mongo dB[4] in NoSQL databases is considered better for authentication which uses SCRAM-SHA-1 as the default authentication mechanism with is an IETF standard, RFC 5802 from the version 3.0 and above along various other mechanisms like MONGODB-CR. It verifies user credentials against the user's name, password and authentication database, which serves to identify the user's identity. Twilio, a text messaging API adds onto the OTP mechanism when used along Mongo dB Authentication creating a better framework to maintain security and authentication.

3.1.2 Basic Profile Information

This module can be rather seen as the most constant and root for all other modules. This module would contain the basic information of the user such as Name, Email, Contact Information, Birthday, Education, Organization working for, etc. For such organized and structured information, this module can be modeled using RDBMS and the key to this module would be the e-mail id of the user which could act as the primary key for all relations and interaction with other modules.

3.1.3 User Connections

A connection is a contact that user has a 1st-degree connection to. User can connect with someone by accepting an invitation from them or when they accept an invitation that user sent. The basic type of connection is a contact known personally and who trusted on a professional level. Once user has "connected" to others, he/she is considered a 1st-degree connection. User can also have an extended network of connections made up of people that his/her connections know. Connections can also be extended to Communities or Groups that are formed by a user usually referred to as the 'admin'.

A graph data model [5] representing a graph database can used to build such a module. A Graph database has no set structure or schema for the data, much like a NoSQL database. Each node represents an entity – a User in our case; each node contains property values, in this case it's the User's name; and each line represents a relationship between the nodes. And to complement the scenario described in the traditional SQL example, if we did decide to add additional properties to a subset of users, we could easily perform this action on a per-node level instead of a table-wide transaction. This is possible because of the path-finding algorithms involved are easy to implement by traversing through the graph.

3.1.4 Recommendations & Intelligent Search

The Recommendations feed displays the most recommended content on your site, using actions by your connections and other people. It is different from the Activity which displays recent actions on your site. The recommendations is based upon the connections, past searches, activities etc.

A recommender engine helps a user find items within a pool of resources. Numerous types of recommendation algorithms and a graph database can serve as a general-purpose substrate for evaluating such algorithms. Graphs can help you make connections with your users, increasing accuracy and engagement. Whether it's using declared social connections to make recommendations, or connecting the dots between seemingly unrelated facts to infer interests, graphs offer a world of fresh possibility when it comes to making better recommendations. GraphDB [6] and Neo4j [7] can be easily used to solve very basic recommendation engine requirement.

3.1.5 Latest Activities & Posts

Latest Activities and posts describe the latest happening which may be related to the user's connections, his search based results or his joined communities. These activities are well sorted by their occurrence time and place.

There are basically two methods to publish out the activities and posts related to user. The first being the "Push" Model, or Fan-out-on-write which involves de-normalizing the user's activity data and pushing the metadata to all the user's friends at the time it occurs. You store only one copy of the data as in the schema above, and then push pointers to friends with the metadata.

Second method is the "Pull" Model or Fan-out-on-load which involves keeping all recent activity data in memory and pulling in (or fanning out) that data at the time a user loads their home page. Data doesn't need to be pushed out to all subscribers as soon as it happens, so no back-log and no disk seeks are required.

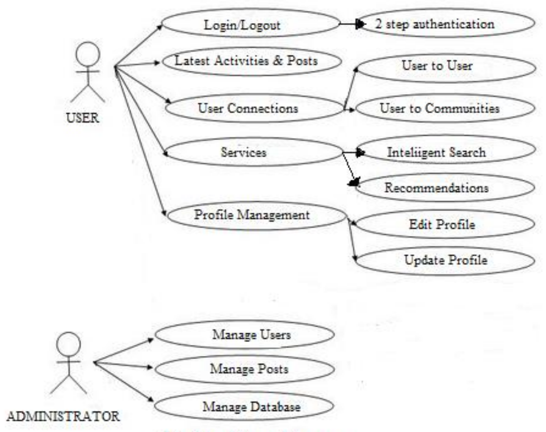

Fig: Use Case Diagram

3.2. Fault Tolerance

Since above implementation is designed to help process of very large amounts of data using hundreds or thousands of machines, and customers, the model must tolerate failures gracefully. To effectively improve functioning and tolerate failures high availability using multi-masters and master-slave architectures can be achieved.

3.2.1Model Failure

The above platform would fail to work for banks, due to the fact that banking operations would include only structured data for which consistency and atomicity are the two most important functionalities, which can be achieved even through RDBMS only. Thus, working with the modules for large amount of unstructured data would be of no use.

The above platform works well only for socio knowledge platforms such as e-commerce, social networking sites, etc.

IV. CONCLUSIONS

The model can be successfully used for many different applications. We attribute this success to several reasons. First, the model is easy to use for any

analytical searches, since it includes fault tolerance with high availability. Second, a large variety of problems are easily expressible as it compensates for big and complex data.
There is no doubt that the future belongs to the Data Lake, and this model can be significantly optimized by the time it becomes a reality. Ultimately, we will use the experience and insights from this model to explore how to support other applications of the web by scaling up a viable model.

V. REFERENCES

[1]http://www.blue-granite.com/blog/embracing-hadoop-data-lakes-modern-data-platform
[2]http://searchaws.techtarget.com/definition/data-lake
[3]http://searchdatacenter.techtarget.com/feature/Data-lakes-swim-with-golden-information-for-analytics
[4] The Data Lake Fallacy, by David Ramel, July 2014

[5]https://docs.mongodb.org/manual/core/authentication/
[6]https://www.facebook.com/notes/facebook-engineering/tao-the-power-of-the-graph/10151525983993920
[7]http://www.reco4j.org/
[8]http://neo4j.com/resources/wp-recommendations-bus/
[9] A Modern Data Architecture, with Apache Hadoop A Hortonworks White Paper, March 2014