I.J. Information Technology and Computer Science, **2016, 10, 56-62**
Published Online October 2016 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2016.10.07

Graph Based Data Governance Model for Real Time Data Ingestion

Hiren Dutta
Associate Consultant, Tata Consultancy Services Limited
E-mail: hiren.dutta@gmail.com

Abstract**—**Data governance is one of the strongest pillars in Data management program which goes hand in hand with data quality. In industrial Data Lake huge amount of unstructured data is getting ingested at high velocity from different source systems. Similarly, through multiple channels of data are getting queried and transformed from Data Lake. Based on 3Vs of big data it's a real challenge to set up a rule based on traditional data governance system for an Enterprise. In today's world governance on semi structured or unstructured data on Industrial Data lake is a real issue to the Enterprise in terms of query, create, maintain and storage effectively and secured way. On the other hand different stakeholders i.e. Business, IT and Policy team want to visualize the same data in different view to analyze, imposes constraints, and to place effective workflow mechanism for approval to the policy makers. In this paper author proposed property graph based governance architecture and process model so that real time unstructured data can effectively govern, visualize, manage and queried from Industrial Data Lake.

Index Terms**—**Data Governance Architecture, Property Graph Process Model, Near Real time Data Governance, Data lake Governance.

I. Introduction

The era of real time Big Data [5] and enterprise government regulation are driving considerable changes in how Financial and other companies manage and use the varied types of data they acquire and store. The legacy Relational Database Management System (RDBMS), Enterprise Data Warehouse (EDW), and Storage Area Network (SAN) infrastructure used by companies today to create siloed data environments is too rigid to accommodate the demands for massive storage, high performant and analyses on a larger and wider variety of data. Forcing this legacy architecture into today's enterprise requirements is costly and risky. Enterprise needs today NoSQL based Data Lake to aggregate disparate data types from the myriad of systems that span their enterprises. Data Lake has massive data storage capacity and processing engine. It has ability to store any kind of data with concurrent access to it. Data ingestion is the process by which real time data stored into Data Lake from external source system. Data Lake quickly loads data "as-is" eliminating months of development work associated with relational EDWs and ETL. The agility enabled by Data Lake services are helping enterprise with cost pressures, customer needs and demands for greater transparency. This high volume data transmission within different source systems demands high performance data governance architecture to be placed so that incoming and outgoing data from Data Lake should be scrutinized and effectively governed and managed. The scope of this study to analyze Graph based architecture as a solution of high performance data governance architecture of Data Lake and real time data ingestion architecture. Also, different stakeholders (Business, IT, Policy) can view, monitor, setup and apply policies of governance metadata for different businesses that should be queried and applied on Data Lake effectively. Recent studies shows data quality [15] resolution on bigdata is based on data loading through ETL [12] and relational data and logic transformation [14]. In this paper we have paper I have proposed data governance model and architecture based on property graph.

This paper is organized as follows- In Section II we will discuss about goals and scope of real time data governance. In Section III we will introduce data lake concepts, and need o it. In Section IV we will introduce graph as a solution for real time data governance and its architecture. Section V is focused on implementation details with graph process model concept map, graph loading process, tools and technology used and execution strategy.

II. Data Governance

Data governance [10] is the science as well as art. It's about people taking responsibility for the information assets of their organization by looking at the processes they use to interact with information as well as how and why it's being used. Creating a governance framework and architecture to ensure the confidentiality, quality, and integrity of data, the core meaning of data governance, is essential to meet both internal and external requirements. Data governance roots out business and compliance risks. The goal is to ensure that data serves business purposes in a sustainable way.

A. Goal of Real time Data Governance

Following the real time / near real time data governance goals to be achieved. A. Data discovery and profiling to find hidden data quality problems, wherever they exist. B. Data lineage and proactive data quality monitoring to trace data quality issues across the enterprise and ensure data quality expectations are continuously met. C. Effective data management to establish an authoritative view of Business, IT and Policy persona. D. Metadata management [11] to provide the visibility and tools to manage change effectively in data integration.

B. Scope of Real time Data Governance

Following are major scope of real time data governance. A. Ensure data quality at the time of capture. B. Ensure immediate data compliance with corporate data standards. C. Allows prompt feedback on errors at the time of entering /leaving data. D. Eliminate proliferation of defective data. E. Reduces overall system maintenance by capturing defects early. The contingency model [8] defines governance model based on performance strategy, diversification breadth, organization structure, competitive strategy, degree of process harmonization, degree of market regulation, and decision-making style for traditional data quality management (DQM). It's having short coming on real time big data governance with respect to 3Vs. Big data governance strategies in [7] shows direction on how big data processing and governance are different than traditional information governance. It requires reliable analytic architecture, next gen IT processes and system architecture for analyzing insights and not just automation. This work is highly influenced by strategies defines in [9] where explosion of unstructured data modelling and collaboration between different stakeholders are key parameters addressed by Graph based data modelling, storage and processing.

III. Industrial Data Lake and Data Ingestion

In today's enterprise, data does not flows in only from internal systems but wide range of data sources including unstructured social media, sensory devices, external systems and other 3rd party cloud providers to data lake. So there is definite need to automated data lake governance process in place to overcome data quality and data security related issues.

A. Need of Data Lake governance

Nature of data in today's world is very dynamic. Multiple systems are talking to each other 24x 7. Large volume of data getting transferred between systems. Now, on the other side in any big data scenario humongous size of data are getting stored, transferred and analyzed every day. With social media, web crawling and crowdsourcing data quality is also questionable to the industry. In this big and fast data environment, manual automated rule based data quality check engine are no more viable solution now due to 3V's of big data [13]. So, in the heterogeneous data communication environment data security is always be a concern. Another angle to this problem is cloud storage. Industries are shifting from in-house data management to cloud based environment. Cloud providers are not also responsible to provide optimum security to the data based on industry, organization, project, business and geography. And these attributes are dynamic in nature. As a result industry needs an effective automated, fast reliable data governance engine that can be scalable, updatable with optimum ease.

IV. Graph based Data Governance

NoSQL databases deals with homogeneous data in terms of data size and data quality. They provide particular data model (key value store, document based etc.) to address these specified dimensions. Using compound aggregate values NoSQL databases resolves issues like scaling out and high data values. Every type of NoSQL databases addresses some specific problems. Maintenance of governance metadata and visualization of same data to different stakeholders are classic problem and application of property graph database. Instead of de-normalizing the data for performance graph normalizes attributes into the nodes and edges which makes it much easier to perform operations like move, filter, project and aggregate. It has high performance real time query capabilities. It processes high volume of raw data in map reduce in Hadoop or real time event processing (Apache Strom, Apache Spark, Esper etc.) data and project computation results into graph. Graph queries are most effective when starts with a node and traverse though possible directions to find out interesting relationships among dataset.

A. Architecture

Below is the architecture of property graph based real time data governance model. Real time data is getting ingested from un-structured, semi structured and structured data sources into the data lake with different time interval and velocity. The respective data consumer components (Spout) reads the data and put it into data processor queue. One consumer can be connected with multiple data processor based on incoming data velocity. Data processor (Bolt) reads the queue message and transform in to canonical form and push it to data manager component with uniform velocity. Data Manager invoke high speed non-blocking graph database to validate incoming data into defined policies which is highly unstructured in nature. Post policy apply transformed data is stored into data lake. In business data lake storage in mainly HDFS/GFS or any other high volume unstructured data storage engine. On the other side when 3rd party applications wants to consume data from data lake request and response goes through same data governance filter to confirm business policies and restrictions.

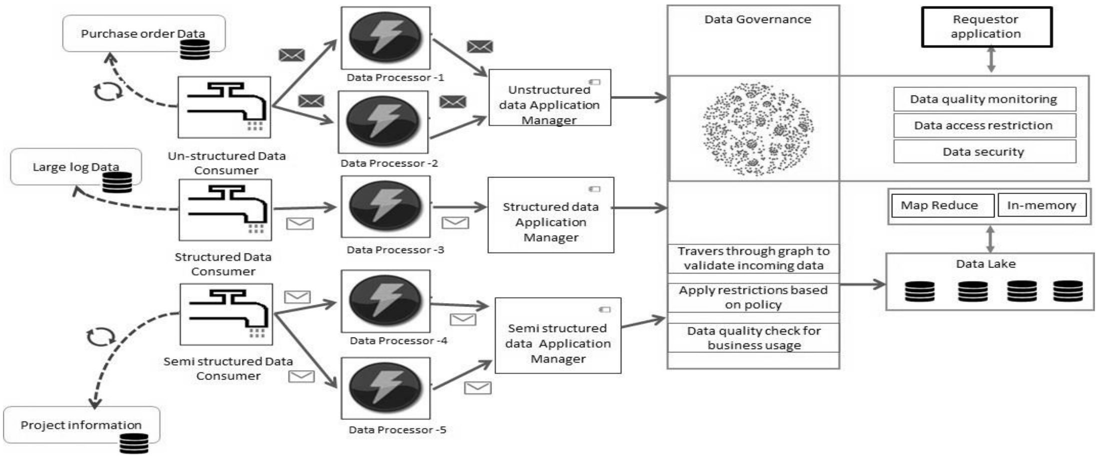

Purchase order Data
Large log Data
Project information
Un-structured Data Consumer
Structured Data Consumer
Semi structured Data Consumer
Data Processor -1
Data Processor -2
Data Processor -3
Data Processor -4
Data Processor -5
Unstructured data Application Manager
Structured data Application Manager
Semi structured data Application Manager
Data Governance
Travers through graph to validate incoming data
Apply restrictions based on policy
Data quality check for business usage
Requestor application
Data quality monitoring
Data access restriction
Data security
Map Reduce
In-memory
Data Lake

Fig.1. Real time data governance architecture

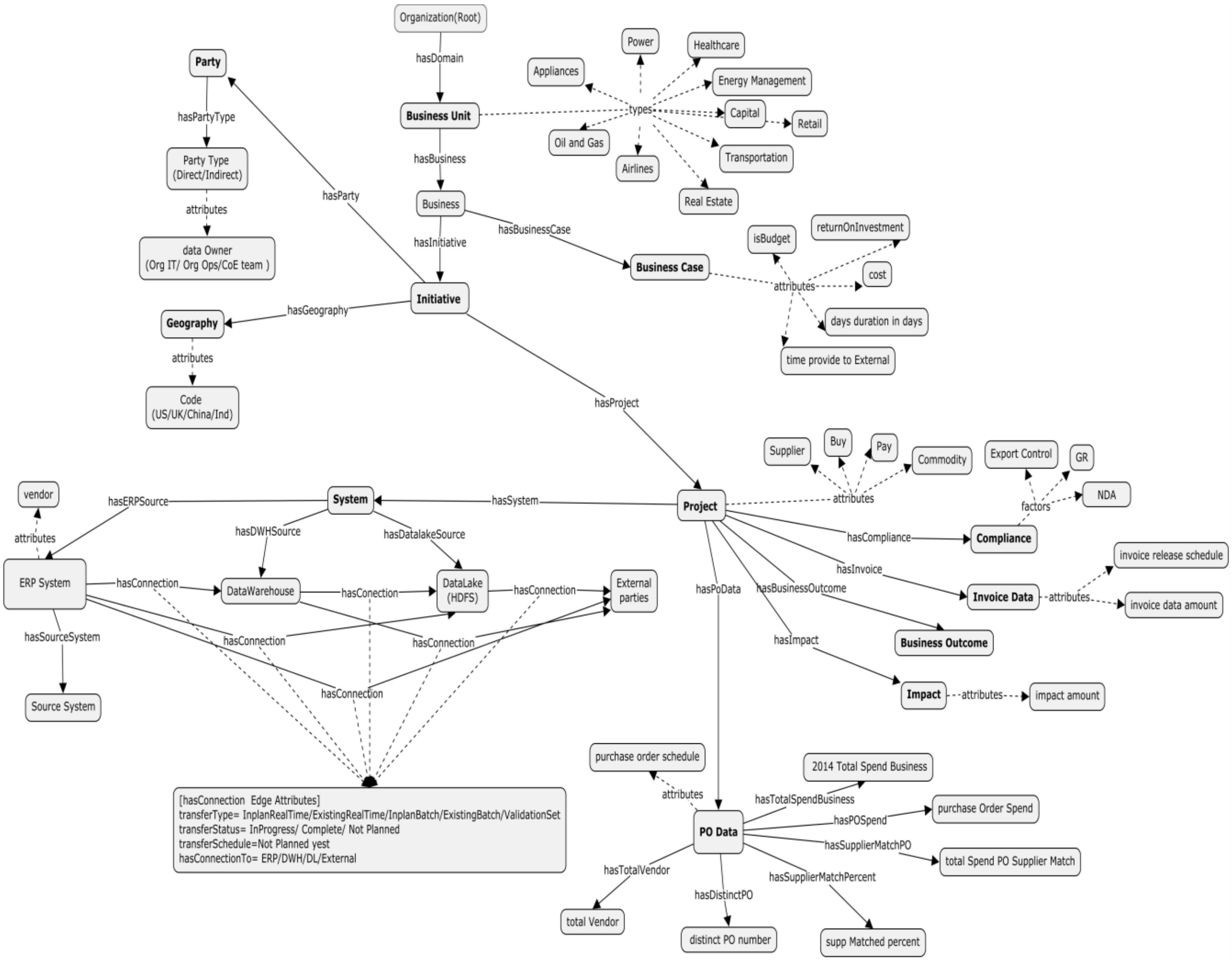


Fig.2. Graph based data governance process map

V. Implementation

To evaluate the proposed concept this section comes up with Process concept map definition, graph loading process and execution information in specific environment.

A. Process model

Creation of process model is the step performed to capture all required data governance elements into the graph and its relationships. Process model is used for organizing and representing knowledge in organized fashion. This empowers users to construct, navigate, share and criticized on the knowledge model. It’s treated as the blueprint of graph data governance to analyses the basic structure, graph complexities, cycles and graph data (node and attributes) access patterns.

Graph based process model supports extreme data wrangling in terms of visual analytic of same graph data to different stakeholders or entities. Projects are connected to business, systems are connected to projects. Each project can have set of compliance. Through technical persona, IT team would be more interested to visualize the systems which are connected to inbound and out bound data injection. Business team member can be assigned as internal party role who wants to visualize the same system information through projects with different representations. Policy maker would be more interested to play with policy rules as a part of project compliances.

B. Technology used

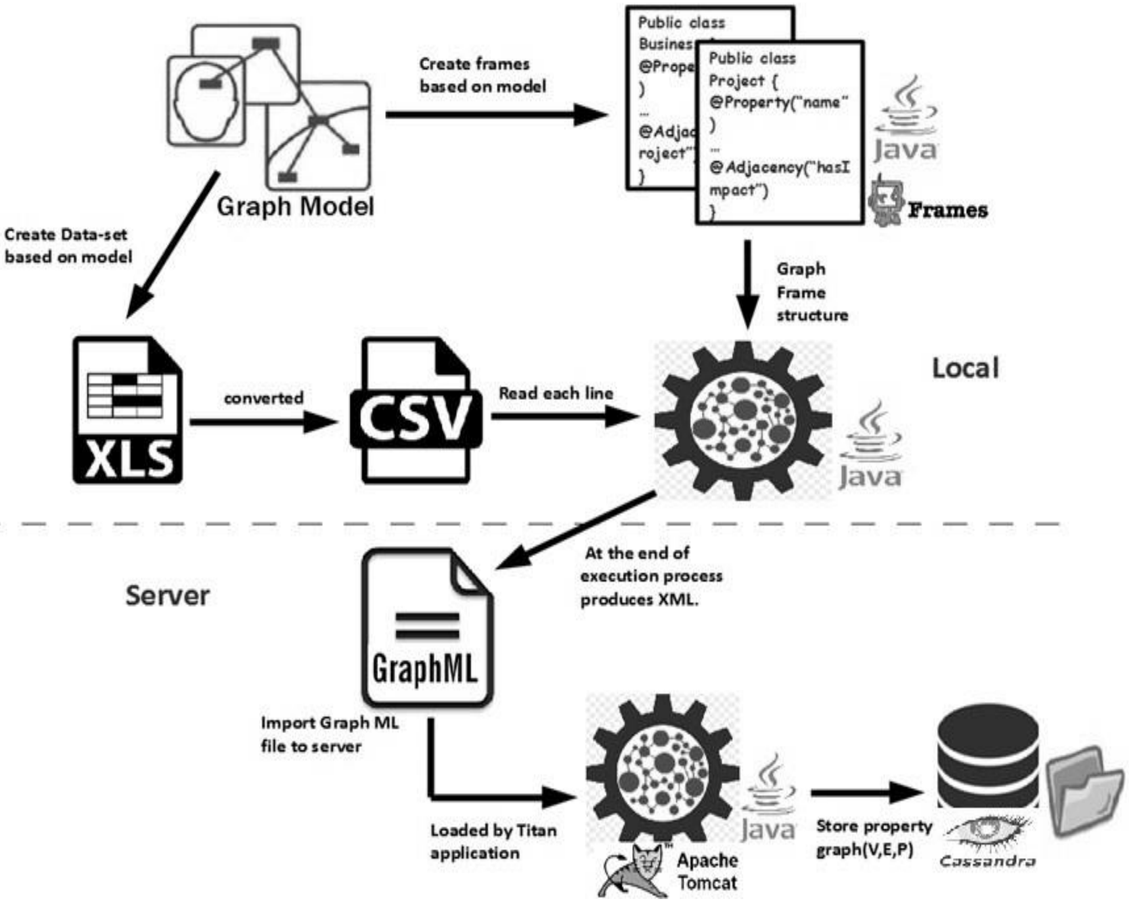
We have implemented java based framework to create, load, query property graph database as a backbone of real time data governance process. We have setup standalone infrastructure based on Titan graph database processing engine with Cassandra as columnar back end storage. Front end visualization application to manage the graph metadata is deployed on Apache Tomcat. D3.js, JQuery, are to build user interface. The server side framework has been implemented using Tinkerpop blueprint stack to overcome vendor locking of the underlying graph database. Process model has been realized using Tinkerpop Frames. Below is the tabular representation of software and technologies used.

Table 1. Used software and their usage

Software	Version	Usage
JDK7	1.7.72	JDK and JVM installation
Apache Cassandra	2.1.2	Act as a backend graph data storage.
Rexster-Server	2.6.0	Graph server with embedded titan exposes graph over REST protocol, Web gremlin console.
Apache Tomcat	8.x	Web server for UI development, Java app development.
Blueprint-Tinkerpop	2.6.0	Blueprint specification for graph database., Frames definition
IHMC CMap tool	6.0.0	Used for Process modelling.

C. Graph loading process

Below is the architecture of graph process model to load, update, port, and maintain governance metadata into graph.


Graph Model
Create frames based on model
Java
Frames
Create Data-set based on model
Graph Frame structure
Local
XLS
converted
CSV
Read each line
Java
Server
At the end of execution process produces XML.
GraphML
Import Graph ML file to server
Loaded by Titan application
Java
Apache Tomcat
Store property graph(V,E,P)
Cassandra

```
Public class
Busines
@Prope
)
...
@Adjac
roject")
}
```

```
Public class
Project {
@Property("name"
)
...
@Adjacency("hasI
mpact")
}
```

Fig.3. Graph loading process

The process model defines the contract between incoming data structure and tinkerpop frames. Governance metadata information can be directly inserted from user interface or through spreadsheet in csv format. Local java processing engine scans csv metadata files and transforms flat data into object oriented frames. Then frames are converted into portable graphML [1] [2] format that is open standards for any graph processing engine to load into graph server. Graph data is stored in columnar apache Cassandra backend storage.

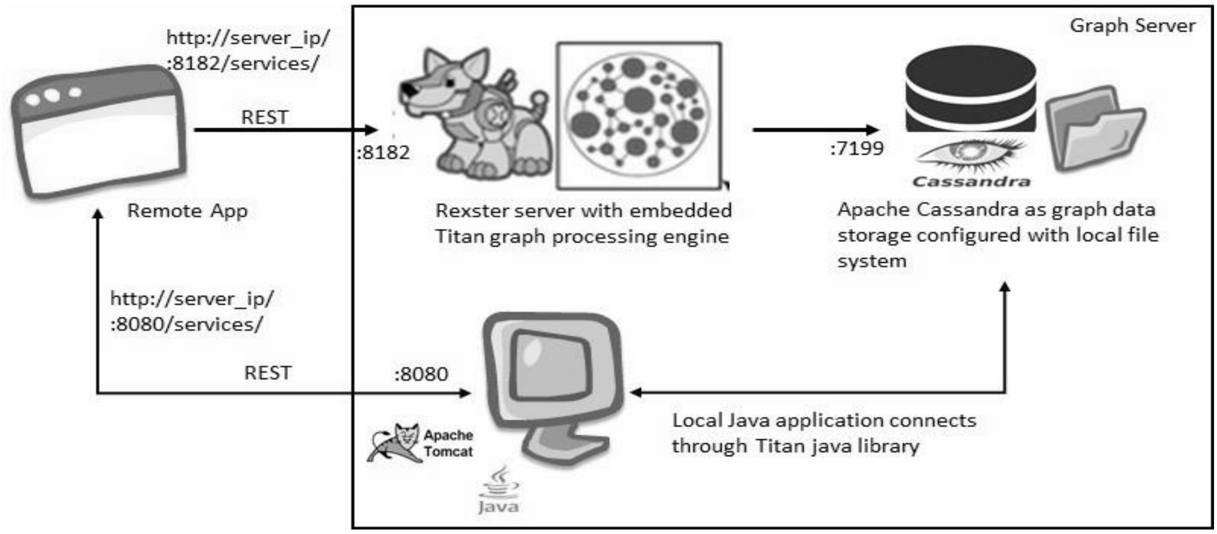
D. *Execution Environment*

Rexster-Titan graph processing engine is installed with Apache Cassandra to provide out of box support for REST API. Custom implemented services are deployed in tomcat server. Table shows the server configuration of graph metadata execution engine

Table 2. Execution environment

Operating System	Windows 8.1
Memory	8 GB
Disk space	1TB
Processor	Intel family i3 generation
Number of execution core	4

Below is the standalone graph server architecture for graph Meta data management and query engine.



Graph Server
http://server_ip/
:8182/services/
REST
:8182
Remote App
Rexster server with embedded
Titan graph processing engine
:7199
Cassandra
Apache Cassandra as graph data
storage configured with local file
system
http://server_ip/
:8080/services/
REST
:8080
Apache
Tomcat
Java
Local Java application connects
through Titan java library

Fig.4. Graph server architecture

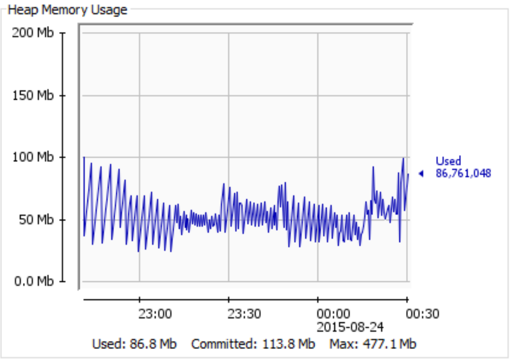
E. *Execution Analysis*

Prototype data contains metadata information of each business. In first experiment we have loaded metadata for five businesses which are converted into property graph model by local custom processing engine. Graph is mostly balanced and each node is having three attributes on an average. Automatic indexing is done on each vertex, edge id and name. Below table shows the performance results based on single thread and multi thread execution.

Number of Nodes – 384, Number of edges – 493
Load test – simultaneous 10 threads, Strategy – Simple

Table 3. Execution results

Test step	Single request (ms)	Load test avg (ms)
Query Vertices	1455	1726
Query Edges	1257	1895
Query Vertices Id	9	12.76



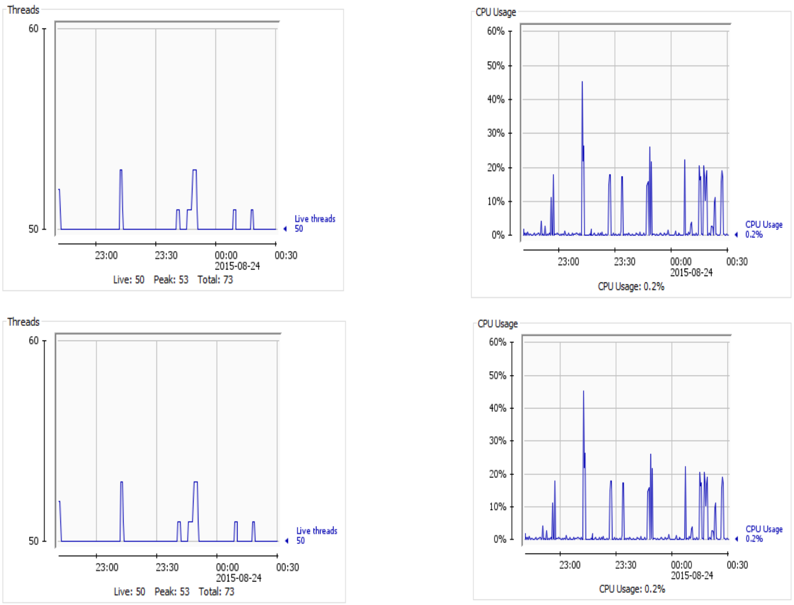


Fig.5. JVM memory and CPU usage

We also noticed that with increased node and the execution time did not differ much. Indexes place a key role here. Actual node data resides into the back end storage. Indexes are loaded into memory once the system up. Bases on indexing actual graph information fetched from storage in no time. Performance will mostly be the same as long as memory is capable of holding whole index structure into the RAM. Experimentation with increasing number of nodes an edges shows processing time does not differs much. Practically the searching time is O (1). This provides biggest advantage over relational databases where search time linearly increases over with increase in volume of data. Saw tooth graph of heap usage confirms no memory leakage in object loading and garbage is getting collected properly. CPU usage is stable. CPU usage will not matter much when implementation is running on cloud infrastructure. In cloud infrastructure processing power is considered as unlimited.

F. Scalable graph processing-Future Direction

This work could further extended into distributed graph environment where scalable governance graph metadata will be stored and queried into distributed environment. Faunus [4] is a Hadoop based graph analytics engine for analyzing graph in multi machine cluster. It processes infinite size graph using functional and MapReduce computing model.

VI. CONCLUSIONS

In this work we have analyzed issues related to real time data governance on unstructured data and proposed graph based architecture to govern data effectively which are either coming into data lake of going out from data lake. We have shown how different stakeholders can bale to interact with the same system to access same set of metadata with different degree of views. In experiment we have implemented the graph model and tested the application using real time data set. As a future extension it has been discussed about distributed graph processing using Faunus platform.

REFERENCES

[1] Graph Markup Language (GraphML) , Chaper-16 Ulrik Brandes -University of Konstanz, Ulrik Brandes, J ̈urgen Lerner- University of Konstanz, Christian Pich-Swiss Re.

[2] http://graphml.graphdrawing.org/specification.html-GraphML specification

[3] Simplifying Data Governance and Accelerating Real-time Big Data Analysis in Financial Services with MarkLogic Server and Intel. White paper 2014

[4] https://github.com/thinkaurelius/faunus/wiki

[5] Manuika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., and Hung Byers, A. (2011): "Big data: The next frontier for innovation, competition, and productivity". McKinsey Global Institute (MGI).

[6] McAfee, A. and Brynjolfsson, E. (2012): "Big data: the management revolution". Harvard business review, 90(10), pp. 59-68.

[7] T.H. Davenport, P. Barth, and R. Bean, "How 'Big Data' Is Different," *Sloan Management Rev.*, vol. 54, no. 1, 2012, pp. 43-46.

[8] Weber, K., Otto, B., and ̈Osterle, H. 2009. One size does not fit all—a contingency approach to data governance. ACM J. Data Inform. Quality 1, 1, Article 4 (June 2009), 27 pages.

[9] C. Beath et al., "Finding Value in the Information Explosion," *Sloan Management Rev.*, vol. 53, no. 4, 2012, pp. 18-20.

[10] Alves de Freitas, P.; Andrade dos Reis, E.; Senra Michel, W.; Gronovicz, M.E.; De Macedo Rodrigues, M.A., "Information Governance, Big Data and Data Quality," in Computational Science and Engineering (CSE), 2013 IEEE 16th International Conference on , vol., no., pp.1142-1143, 3-5 Dec. 2013, doi: 10.1109/CSE.2013.168

[11] Batra, V.; J. Bhattacharya; H. Chauhan; A. Gupta; M.Mohania; U. Sharma. 2002. —Policy Driven Data

[12] Saha, B.; Srivastava, D., "Data quality: The other face of Big Data," in Data Engineering (ICDE), 2014 IEEE 30th International Conference on , vol., no., pp.1294-1297, March 31 2014-April 4 2014 doi: 10.1109/ICDE.2014.6816764

[13] J. Tee: Handling the four 'V's of big data: volume, velocity, variety, and veracity. TheServerSide.com 2013.

[14] M. Zhang, M. Hadjieleftheriou, B. Ooi, C. M. Procopiuc and D. Srivastava: On multi-column foreign key discovery. PVLDB 3(1): 805-814 (2010).

[15] Mengjie Chen; Meina Song; Jing Han; Haihong, E., "Survey on data quality," in Information and Communication Technologies (WICT), 2012 World Congress on , vol., no., pp.1009-1013, Oct. 30 2012-Nov. 2 2012 doi: 10.1109/WICT.2012.6409222

Authors' Profiles

Hiren Dutta received the B.E. degree in Computer Technology from the Nagpur University, Maharashtra, India, in 2004 and received M.E. degrees in Software Engineering (Information Technology) from the Jadavpur University, Kolkata, WB, India, in 2013. He completed PGD in supply chain management from St. Xavier's college, Kolkata in 2009 and awarded certificate on analytics from Indian Statistical Institute in 2014.He is currently designated as an Associate Consultant in Tata Consultancy Services Ltd. He is an expert in providing solution architecture of business applications based on IIoT, brokered SOA, JEE architecture and middleware Integration. He worked in a number of open source technologies, design methodologies, patterns and developing algorithms in system integration and communication. His research interests are IIoT, molecular communication, swarm intelligence, real time data processing and analytics.

How to cite this paper: Hiren Dutta, "Graph Based Data Governance Model for Real Time Data Ingestion", International Journal of Information Technology and Computer Science (IJITCS), Vol.8, No.10, pp.56-62, 2016. DOI: 10.5815/ijitcs.2016.10.07