Secured Information Retrieval using CIDS and Map Reducing in Cloud

J.Indra Mercy
Assistant Professor, CSE
Saveetha Engineering College
Chennai, India

M. Kanimozhi,
Assistant Professor, CSE,
Saveetha Engineering College,
Chennai, India.

L. Maria Michael Visuwasam
Assistant Professor, HOD/CSE,
Velammal Institute of Technology
Chennai, India

S. Previtha Shalomi Dinisha,
Assistant Professor, CSE,
Saveetha Engineering College,
Chennai, India

Abstract— Secured Information Retrieval using CIDS and Map reducing in Cloud is a method for information retrieval from cloud. In which fast access provided by, distributing data on multiple databases and parallel (SQL) queries are used to retrieve data in secure way. Query will be equally distributed to multiple peers for private information retrieval process. SQL to MapReduce translators emerge to translate SQL queries to MapReduce codes and it provides good performance in cloud systems. Map reduce is a parallel programming model for cloud computing platforms. It is an effective method for processing huge amount of data on the cluster of components. A Collaborative Intrusion Detection System (CIDS) provides security for the data's placed inside the cloud networks by preventing and detecting the attacks. Paxos Algorithm is used to maintain consistent in available data, automatic updates amongst peer nodes is maintained. Peer Node failures will updated to co-operative peers.

Index Terms— private information retrieval; Privacy; distributed systems; Security databases; MapReduce; Intrusion detection.

I. INTRODUCTION

SPI is the most popular cloud computing service model it includes, Software as a Service(SaaS), Platform as a Service(PaaS) and Infrastructure as a Service(IaaS). SaaS is a software distribution model. Here applications are hosted by vendor/service provider and then it is available to the customers over the Internet. PaaS is a concept for delivering operating systems and the associated services in the Internet with no downloads or installation. IaaS involves outsourcing the equipment, which is used to support operations, including networking, storage, servers and hardware components [4].

The increasing selection of services delivered over the Internet or everything/anything for service is sometimes referred to as XaaS. MapReduce is a parallel programming model for cloud computing platforms, and an effective method for processing huge amount of data by using a cluster of computers [2]. SQL-to-MapReduce translator is used to deploy an application to cloud systems through translating standard query language to MapReduce codes. Currently, SQL to MapReduce translators emerge to translate SQL-like queries to

MapReduce codes and have good performance in cloud systems. Moreover, SQL-to-MapReduce translators mostly focus on SQL-like queries, but not on numerical computation.

We use a system called YSmart[1], a correlation aware SQL-to-MapReduce translator. This method uses the collection of rules to implement a least number of MapReduce jobs to execute multiple correlated operations in a complex query. YSmart can drastically reduce redundant computations, network transfers and I/O operations compared to the existing translators. The results show that YSmart can provide better performance than Hive and Pig, by more than four times for query execution [6][7].

Additionally, it scales well with a growing number of peers, achieving a linear speedup [5]. Conventional standalone IDSs are susceptible to cooperative attacks, so they're unsuitable for collaborative environments (such as a cloud computing environment). To protect against this attack, CIDSs (Collaborative Intrusion Detection Systems) associate suspicious evidence between different IDSs to improve the intrusion detection efficiency. Unlike conventional standalone IDSs, CIDS shares traffic information with the IDSs located inside the local network's entry points.

In summary, our contributions are the following: We used YSmart translators [1], evaluate its cost in terms of computational time, performance challenges and bandwidth consumption when retrieving large data blocks.

II. SQL-TO-MAPREDUCE TRANSLATORS

Map reduce has become an efficient approach in large cluster systems.

2.1 YSmart System

The source of inefficiency comes from the basic approach for translating SQL queries into MapReduce jobs. Existing translators acquire one-operation-to-one-job approach. For a query plan tree, every operation in the tree is interchanged by a pre-implemented MapReduce program, and then the tree is finally translated into a chain of programs.

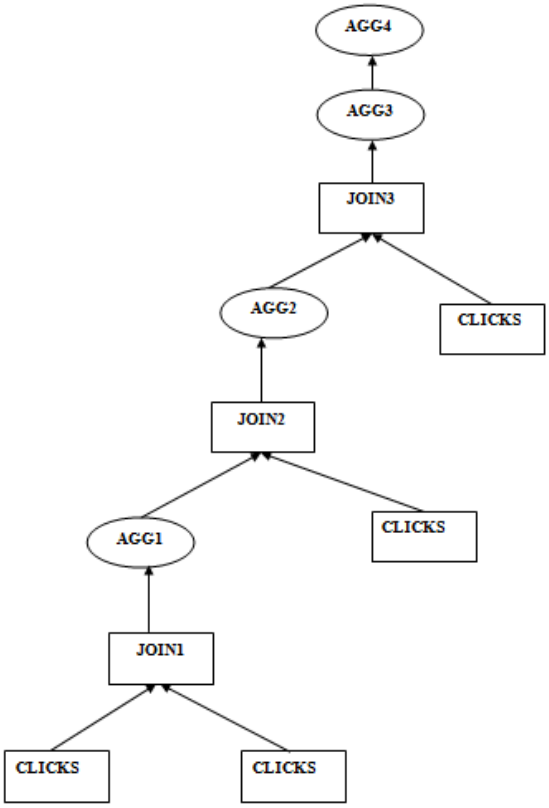

Fig.1. Q-CSA plan

For example, Hive generates six jobs to execute the six operations (JOIN1, AGG1, JOIN2, AGG2, JOIN3, AGG3, and AGG4) in the plan tree shown in Fig.1. It is an inefficient approach, since it can cause redundant table scans (e.g., JOIN1, JOIN2 and JOIN3 need to scan CLICKS) and it leads to unnecessary data transfers among multiple jobs.

This query Cloud Service Automation (Q-CSA) is used to find the average number of files in private cloud a user visits between a file in category A and a page in category B based on a single click-stream table CLICKS(user id int, category id int, file id int, ts timestamp). It requests self-joins and the multiple aggregations of same table, since it is a complex query. Its execution plan tree is shown in Fig.1 .

Thus, existing translators cannot generate high-performance MapReduce programs for following reasons. First, they cannot address the restrictions of the simple MapReduce structure for complex query. And Second, they cannot make use of the unique opportunities supplied by intra-query correlations in a complex query.

YSmart is built on top of the Hadoop. It supports three types of intraquery correlations. It is based on the Key/value pair model, of the MapReduce framework. After detecting such correlations in a query, YSmart uses a set of rules to compute optimized MapReduce jobs, which are governed by Common MapReduce Framework (CMF) in YSmart, so that minimum number of queries can executes multiple correlated operations in a cloud system. It provides a excellent query performance done by reducing redundant computations, unnecessary disk accesses and network overhead.

III. RELATEDWORK

IDSs (Intrusion detection System's) are very vital to securing cloud computing environments. Host-based IDSs (HIDSs) are used to detect malicious events on host machines. They can handle insider attacks and user-to-root attacks. Network–based IDSs (NIDSs) monitors and the flag traffic carries malicious contents or presenting the malicious patterns. DLPs also not provide complete Security. Firewalls can block unwanted network traffic packets but they can't detect attacks such as flooding and insider attacks.

3.1 Conventional IDSs

CIDSs are standalone systems residing on computer or host machines. Depending upon the detection mechanism, it can be categorized as misuse-based or anomaly-based IDSs. Misuse-based IDSs provides high detection accuracy but are vulnerable to zero day intrusions. It detection mechanism checks with the match with existing attack signatures. Since IDS can't generate signatures for an unidentified attack. Anomaly-based IDSs can detect zero-day intrusions, but it leads to high false positives.

Conventional standalone IDSs are vulnerable to cooperative attacks, so they are not suitable for cloud computing environment.

3.2 Collaborative Intrusion Detection (CID)

CIDs shares traffic information with the IDSs located in the local network entry points [3].

And also we can organize IDSs within a CIDS in a decentralized or hierarchical manner in large network. According to the organization, IDSs can communicate directly or with the central coordinator. Detection of malicious attempts is handled by each IDS.

Network traffic aggregation is performed on the central coordinator. When intrusive behaviors are detected, it raises an alert to the system administrator [7]. It combines both misuse-based and Anomaly-based detection mechanisms reduces the time required to detect and improve the detection accuracy of known and unknown attacks [7]. This scheme provides a strong, secure mechanism when nodes join in or leave the network.

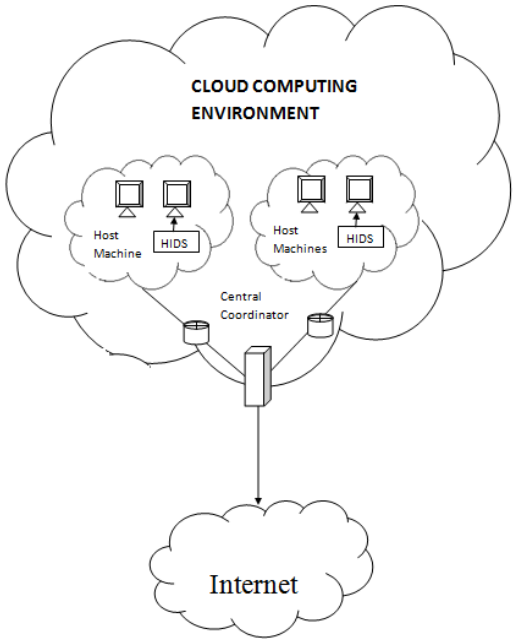

Fig2. Collaborative Intrusion Detection System Framework

IV. PAXOS Algorithm

Paxos is a an algorithm, which is used to solve the distributed consensus problem. And it is able to work on asynchronous network with the non-byzantine assumptions [9].

4.1 Paxos Algorithm Procedure

4.1.1 The Constraint Condition of Paxos Algorithm

The Paxos Algorithm has constraint conditions as follow:

(1) Algorithm executor (server) may operate at any speed, it may failure by downtime or restart. Because any single agent may shut down and then restarted after the resolution is selected, the solution requires the agent must be able to remember certain information, so that it can reload after the restart.

(2) The speed of messaging is unpredictable, they may be duplicated or lost, but the content will not be damaged [9].

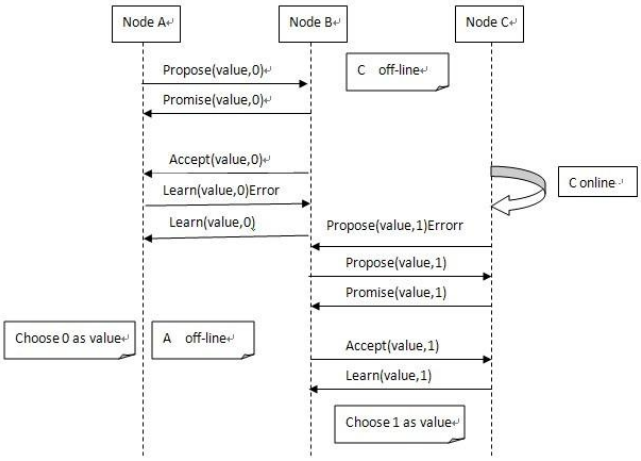

Fig.3 The framework model

V. CONCLUSIONS

Secured Information Retrieval using CIDS and Map reducing in Cloud is an information retrieval from cloud with fast and secured access supplied by, distributing the data on multiple databases and parallel queries are used to retrieve data in a secure way. A CIDS (Collaborative Intrusion Detection System) provides security for the data's placed inside the cloud networks and Mapreduce is used to process huge amount of data on the cluster of components. SQL to MapReduce translators emerge to translate SQL queries to MapReduce codes and have better performance in cloud systems. The only drawback is maintaining consistency in the cloud databases. In future we will use better consistency algorithm to improve consistency.

REFERENCES

[1] Rubao Lee, Tian Luo, Yin Huai, Fusheng Wang, Yongqiang He, Xiaodong Zhang, "YSmart: Yet Another SQL-to-MapReduce Translator," Distributed Computing Systems (ICDCS), 2011 31st International Conference on , 2011.

[2] Junbo Zhang, Dong Xiang, Tianrui Li, and Yi Pan, "M2M: A Simple Matlab-to-MapReduce Translator for Cloud Computing," Tsinghua Science And Technology, 2013.

[3] Zhiyuan Tan, Upasana T. Nagar, Xiangjian He, and Priyadarsi Nanda, Ren Ping Liu, Song Wang, Jiankun Hu, "Enhancing Big Data Security with Collaborative Intrusion Detection," Secure Big Data in the Cloud, 2014.

[4] Yang Lai and Shi ZhongZhi, "An Efficient Data Mining Framework on Hadoop using Java Persistence API," 10th IEEE International Conference on Computer and Information Technology, 2010.

[5] W. Gasarch, "A Survey on Private Information Retrieval (Column: Computational Complexity)," Bull. of the European Assoc. for Theoretical Computer Science (EATCS), vol. 82, pp. 72-107, 2004.

[6] Weikuan Yu, Yandong Wang, and Xinyu Que, "Design and Evaluation of Network-Levitated Merge for Hadoop Acceleration," IEEE Transactions On Parallel And Distributed Systems , Vol. 25, No. 3, March 2014.

[7] Wooyoung Kim_, Martin Diko, and Keith Rawson, "Network Motif Detection: Algorithms, Parallel and Cloud Computing, and Related Tools," Tsinghua Science And Technology, Volume 18, Number 5, October 2013.

[8] Wen-Cheng Shi, Jian-Ping Li, "Research on Consistency of Distributed System Based on Paxos Algorithm," 2007,

[9] Lamport L, "Paxos Made Simple". ACM SlGACT News, vo1.32, noA, pp.18-25, 2001.