International Journal of Contemporary Research in Computer Science and Technology (IJCRCST) e-ISSN: 2395-5325
Volume 2, Issue 3 (March'2016)

ACHIEVING QUERY OPTIMIZATION OF ENHANCED PARALLEL PROCESSING IN CLOUD

T.Sneha,
B.Tech,
Information Technology,
Velammal Institute Of Technology,
Chennai, India.

E.Karthiyayeni,
B.Tech,
Information Technology,
Velammal Institute Of Technology,
Chennai, India.

P.Sailaja,
Assistant Professor,
Velammal Institute Of Technology,
Chennai, India.

Abstract: Transfer speed effective execution of online enormous information examination in telecom systems requests for customized arrangements. Existing gushing investigation frameworks are intended to work in extensive server farms, accepting boundless transmission capacity between server farm hubs. Applying these arrangements unmodified to disperse telecom mists, neglects the way that accessible transfer speed is a rare and expensive asset making the telecom system significant to end-clients. This article presents Continuous Hive ,a gushing investigation stage custom-made for appropriated telecom mists. The principal commitment of CHive is that it streamlines inquiry arrangements to minimize their general data transfer capacity utilization when sent in a circulated telecom cloud. Moreover, these enhanced inquiry arranges have a high level of parallelism implicit, profiting rate of execution. Early examinations on information from an extensive portable administrator demonstrate that CHive can yield data transmission.

I. INTRODUCTION

Big data companies like Facebook, Google, Twitter or LinkedIn continuously collect massive amounts of data from users and their devices. According to the business model of these companies, data has become a very valuable asset to extract knowledge about user interests, social contacts, intentions, etc.—for instance to drive targeted advertisement campaigns, to recommend news items or to present personalized offers. Traditional telecommunication companies, in contrast, adopt a different business model. These companies generate revenues by selling their premium communication services, including network bandwidth. However, similar to the web-scale companies listed above, telecommunication companies also have access to massive amounts of information, in particular data related to how their networks are being used. This includes both (anonymized) network traffic information and statistics about the operational status of the deployed telecommunication equipment. Processing and mining this data generates valuable insights that enable improving the operation of the affected network, including the ability to predict and prevent erroneous situations (like overloads and traffic congestion), to support dynamic network capacity planning, to perform user and user-device segmentation, and even to predict user behavior. Existing IT platforms and solutions for big data analytics are designed to operate on large clusters of processing nodes, located in the same data center. Additionally, these platforms assume the availability of virtually unlimited resources, such as compute power and network bandwidth. When executing big data analytics in telecommunication clouds, however, these assumptions cannot be taken for granted anymore. First, telecommunication clouds tend to be highly distributed in nature, being built up as a constellation of micro DCs in the edge and/or access network. These micro DCs have the unique benefit to be located much closer to the end-user, which enables e.g. hosting lower latency services and location-aware processes. Second, if the data generation velocity is high and/ or the size of the events is large, transporting this data over the network to a central DC may consume a significant portion of the available bandwidth, which overlooks that network bandwidth is a scarce and costly resource making the telecom network valuable to end-users. This article presents Continuous Hive (CHive), developed by Alcatel-Lucent Bell Labs, which offers a Hivelike solution to simplify and optimize streaming analytics in telecommunication clouds. Similar to Hive, CHive aims to facilitate the execution of SQL-like queries to process massive datasets. In contrast to Hive, however, CHive is not designed to execute ad-hoc queries on large datasets stored in Hadoop , but instead executes continuous queries on data collected in an online fashion. The fundamental contribution of CHive is that it optimizes query plans to minimize their overall bandwidth consumption when deployed in a distributed cloud. This is accomplished by rewriting query plans such that data events can be processed as close as possible to their source, hence limiting the amount of information that needs to be sent all the way down to the network core where the analytics applications are typically running.

II. EXISTING SYSTEM

Existing IT platforms and solutions for big data analytics are designed to operate on large clusters of processing nodes, located in the same data center (DC). Additionally, these platforms assume the availability of virtually unlimited resources, such as compute power and network bandwidth. When executing big data analytics in telecommunication clouds, however, these assumptions cannot be taken for granted anymore. First, telecommunication clouds tend to be highly distributed in nature, being built up as a constellation of micro DCs in the edge and/or access network. These micro DCs have the unique benefit to be located much closer to the end-user, which enables e.g. hosting lower latency services and location-aware processes. Second, if the data generation velocity is high and/or the size of the events is large, transporting this data over the network to a central DC may consume a significant portion of the available bandwidth, which overlooks that network bandwidth is a scarce and costly resource making the telecom network valuable to end-users.

DISADVANTAGES OF EXISTING SYSTEM

1. Less bandwidth
2. High cost
3. Slow process

III. PROPOSED SYSTEM

This article presents Continuous Hive (CHive), a streaming analytics platform tailored for distributed telecommunication clouds. The fundamental contribution of CHive is that it optimizes query plans to minimize their overall bandwidth consumption when deployed in a distributed telecommunication cloud. Additionally, these optimized query plans have a high degree of parallelism built-in, benefiting speed of execution. Early experiments on data from a large mobile operator indicate that CHive can yield bandwidth reductions upwards of 99 percent.

ADVANTAGES OF PROPOSED SYSTEM

1. More bandwidth
2. Low cost
3. Fast process

IV. SOFTWARE DEVELOPMENT

MODULES

1. Authentication and Authorization
2. File Encryption and Data store to Cloud.
3. Query Annotations
4. Analyzing Streaming Data

V. MODULE DESCRIPTION

A. Authentication and Authorization

In this module the User have to register first, then only he/she has to access the data base. After registration the user can login to the site. The authorization and authentication process facilitates the system to protect itself and besides it protects the whole mechanism from unauthorized usage. The Registration involves in getting the details of the users who wants to use this application.

B. File Encryption and Data store tocloud

In this module, User Upload the files which he wants to share. At first the uploaded files are stored in the Local System. Then the user upload the file to the real Cloud Storage (In this application, we use Dropbox). While uploading to the Cloud the file got encrypted by using IES (Identity Based Encryption Standard) Algorithm and generates Private key. Again the Encrypted Data is Converted as Binary Data for Data security and Stored in Cloud.

C. Query Annotations

CHiveQL includes a set of annotations that enable a data analyst and/or monitoring system to provide hints or context information helping the CHive query compiler to calculate the optimized query plan. When searching for the most optimal query plan, the compiler calculates the end-to-end stream data volume that each candidate plan is expected to generate.

D. Analyzing Streaming Data

Streaming data is an analytic computing platform that is focused on speed. This is because these applications require a continuous stream of often unstructured data to be processed. Therefore, data is continuously analyzed and transformed in memory before it is stored on a disk. We use Chive is that it optimizes query plans to minimize their overall bandwidth consumption when deployed in a distributed telecommunication cloud. Additionally, these optimized query plans have a high degree of parallelism built-in, benefiting speed of execution.

VI. ARCHITECTURE DIAGRAM

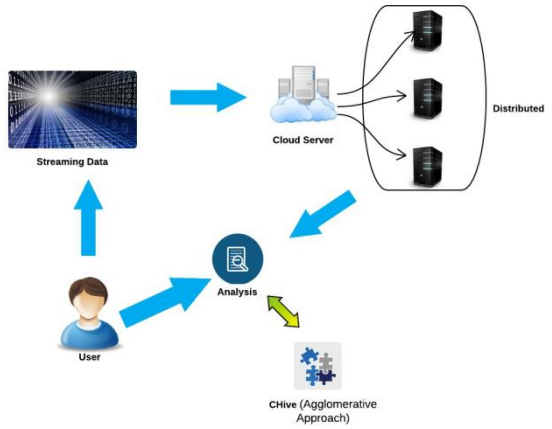

Figure1. Architecture Diagram

VII. FUTURE WORK

Future work focuses on various advancements of both the CHive inquiry compiler and the execution motor. Accordingly,we plan to incorporate checking segments tapping with different places in the inquiry execution work process to assemble estimations on information

appropriation, current occasion rate, outputto- data proportions, execution idleness and throughput, all together to consistently enhance inquiry arranges and question execution execution. Note that these estimations will make client gave insights outdated. Another streamlining is to reuse segments (handling parts, accumulation windows) over numerous comparable constant questions in request to fundamentally diminish the general memory foot shaped impression. We likewise plan to guarantee end-to-end adaptation to internal failure for circulated inquiry arranges conveyed over different Storm bunches. At long last, a definitive Big Data investigation device offers a brought together stage encouraging examination on both chronicled information and live streams. CHiveQL was deliberately intended to have highlights of both Hive and Esper so that information investigators natural with one of these advancements have a commonplace dialect to characterize questions crossing the cluster arranged and ongoing preparing universes. At the execution layer, we plan to manufacture the important parts connecting both universes to frame an widely inclusive examination device.

VIII. CONCLUSION

This article delineates the advantages of another innovation in the occasion stream handling tool compartment, empowering bandwidthefficient circulated stream preparing in Big Data systems facilitating thickly circulated occasion sources—telecom organizes and dispersed cloud situations. CHive looks to offer a Hive-like answer for rearrange and advance gushing examination in telecom mists. In that capacity, it offers a spilling question dialect improving Esper's occasion handling dialect to encourage circulated inquiry execution. inquiry organization arrange for that minimizes the normal general data transmission utilization. Minimizing transfer speed utilization is accomplished by diminishing occasion streams when conceivable, i.e. as close as could be expected under the circumstances to the occasion sources. Following this requires a lot of parallelism in the inquiry plans, the CHive inquiry compiler incorporates an arrangement of substitution rules for supplanting inquiry primitives with semantic reciprocals that expand the subsequent level of interstream parallelism. What's more, we built up an adaptable Storm-based execution environment including a homebuilt store of inquiry primitive usage to execute a CHive inquiry arrangement. A numerical assessment and early tests utilizing genuine administrator information .

REFERENCES

[1] D. Peng and F. Dabek, "Large-scale incremental processing using distributed transactions and notifications," in Proc. 9th USENIX Conf. Oper. Syst. Design Implementation, 2010, pp. 1–15.

[2] F. Frattini, K. S. Trivedi, F. Longo, S. Russo, and R. Ghosh, "Scalable analytics for IaasS cloud availability," IEEE Trans. Cloud Comput., vol. 2, no. 1, pp. 57–70, Jan.–Mar. 2014.

[3] S. Babu and J. Widom, "Continuous queries over data streams," SIGMOD Rec., vol. 30, no. 3, pp. 109–120, Sep. 2001.

[4] I. Satoh, "Mapreduce processing on IoT clouds," in Proc. IEEE 5th Int. Conf. Cloud Comput. Technol. Sci., 2013, vol. 1, pp. 323–330.

[5] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," Commun. ACM, vol. 51, no. 1, pp. 107–113,Jan. 2008.