International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 12, December 2015
Available at http://internationaljournalofresearch.org

A Novel Approach for Processing Large-Scale Data by Using Peer-To-Peer Techniques

Nallajarla Yesubabu[1]& K .Naresh Babu [2]
[1]PG Scholar,Dept Of CSE, Nova's Institute Of Technology, Eluru ,Andhra Pradesh
[2]Associate Professor,Dept Of CSE Nova's Institute Of Technology, Eluru,Andhra Pradesh

ABSTRACT:
The Companies used sharing data where they need to contribute or they share common interest. As per increasing business trends and maximum used of cloud computing, the new system evolved in new stage of growth towards cloud enabled system. In this system based on peer to peer system develop data sharing service in shared network. This system is the combination of cloud computing, databases and peer to peer based technologies. This system gives the efficiency as pay as you go manner. This paper used benchmarking method for analysis and evolution by comparing this method with Hadoop DB which is Large Scale data processing platform. This system having security by providing private key and admin authorized to provide access to other user. Using Cloud Computing they allowed to user to store their data into cloud and access when required.

KEYWORDS: Cloud computing; Map Reduce; Network Security; Peer-to-peer systems; Query processing

I.INTRODUCTION

Peer-to-peer (P2P) has become a new wave of innovative Internet-based computing technologies, and is expected to be an alternative of the traditional Client/Server model in many applications. In some domains, P2P-based applications have †The author is partially supported by Microsoft Research Fellowship been proved to be successful and are attracting more and more attention. These include file sharing [1], [2], [6], message exchanging [4], [5], scientific computing [7], bandwidth sharing and Web search [3], etc. In this paper, our interest is P2P systems for file sharing, information retrieval, and data management applications. It is well-known that current file sharing systems support only or mainly key-based exact matching and keyword-based searching (e.g., Napster [6], Gnutella [2]) for files discovery and location, which is not enough to meet the requirements of more advanced applications such as information retrieval and data management. A natural advancement of key-based exact matching and keyword-based searching is similarity-based query (or simply similarity query). Similarity query is to retrieve the data objects that have similarity to the query not less than a prespecified similarity threshold. Obviously, the number of returned query answers relies on the similarity threshold value: the larger the similarity threshold value is set, the more query answers can be obtained. Considering the decentralization nature (autonomy of peers, dynamism and ad hoc of network, and lack of global metadata) of P2P environment, it is inefficient to conduct similarity query in P2P systems (without special declaration, in this paper we refer to P2P systems as unstructured P2P systems such as Gnutella).

Current solutions to this problem are to develop efficient routing algorithms [10] or build up distributed indices [8]. Generally, data distribution implies a certain cluster structure. The clustering task is to expose the underlying structure by using different algorithms. And data objects in the same cluster inherently share more similarity that these in different clusters. With these in mind, in this paper we propose a new query model for P2P

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 12, December 2015
Available at http://internationaljournalofresearch.org

application. We term the new query model Clustering-Based Query Answering (CBQA). Given a query (it can be an example, or a set of keywords for document retrieval), we define CBQA as to retrieve the data objects that are in the same cluster of the query from the global dataset distributed over peers of a P2P system. Comparing with similarity query, 1) CBQA can obtain results that are more conform to the nature of queried data because all query answers it retrieves are in the same cluster of the query. However, similarity query may retrieve some objects that do not belong to the same of the query if the similarity threshold is too small; on the contrary, it may excludes some objects that belong to the same cluster of the query if the similarity threshold is too large. In Section 2, we will depict the cases above more clearly by example. 2) CBQA is more general. Considering that data clusters may in arbitrary shapes, which means query results of CBQA may distribute in the interested cluster region of possibly arbitrary shape; while results of similarity query are limited to the hypershperical region with the query as center and the similarity threshold as radius.

II.LITERATURE SURVEY

This section explains the existing System and some basic concept used in this system

A. Existing System

The original system attempt to develop peer-to-peer (P2P) technologies for shared Network. Existing system was designed to work as a scalable, sharable, and secure P2P-based Data Management system for building corporate networks in which a part of association controlled by different executive domains work together in order to reduce operation cost and pick up efficiency. Supply chain management and national healthcare network are consider as shared network application. Existing system provides an effective and efficient way to share data among to different association and also provide enterprise quality query facility, without the requirement to set up a big centralized server [5]. Data warehouse used in existing system have some problem. First, the shared network environment, most companies are not dedicated to invest deeply on additional information systems until they can clearly see the potential return on investment (ROI) [11]. Second, companies want to completely modify the access control policy to determine which business partners can see which part of their shared data.

B. What is Peer to Peer Distributed Data Management :

Practically all existing P2P systems are designed to support sharing of data at a coarse granularity (e.g., files, documents) [3]. In this section, it first distinguishes between P2P systems and distributed database systems. Then define P2P distributed data management by looking at three examples (due to space constraints) of how P2P technology can be employed for distributed database applications. This will also serve to motivate the need for database technology in P2P systems [6].

C. P2P Distributed Data Management Systems Applications

Instead of formalizing the concept of P2P distributed data management systems, It show with sample applications on what such systems may be like [4].

1. Health Care

In a hospital, each specialist has a group of patients that are solely under his care. While some patient data are stored in a centralized server of the hospital (e.g., name, address, etc), other data (e.g., X-rays, prescription, allergy to drugs, history, Reaction to drugs, etc) are typically managed by the specialist on his personal PC [5]. For most of these patients, the specialist is willing to share their data, but there are always some cases that he is unwilling to share for

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 12, December 2015
Available at http://internationaljournalofresearch.org

different reasons (e.g., part of his research program on a new drug, etc). By making the shareholder patient data available to other specialists, it allows them to look for other patients who may have similar symptoms as their own
patients, and hence can help them in making better decisions on the treatment (e.g., drugs to prescribe, reactions to look out for, etc).

2. *Genomic Data*

The discovery of new proteins necessitates complex analysis in order to determine their functions and classifications[5,7]. The main technique that scientists use in determining this information has two phases. The first phase involves searching Known protein databases for proteins that ”match” the unknown protein. The second phase involves analysing the functions and classifications of the similar proteins in an attempt to infer commonalities with the new protein. While there are several known servers on genomic data (e.g., GenBank, SWISS-PROT and EMBL) [8], there are many more data that are produced each day in the many lab oratories all over the world. These scientists create their own local databases of their newly discovered proteins and results, and are willing to share their findings to the world! Clearly, this is an application for P2P distributed data management systems for the same reasons as the health care application.

3.Data Caching

In the above two examples, each participant is actively involved in the process of consuming and supplying data. P2P distributed data management can also be deployed in passive nodes: nodes that are used to share resources (storage or computational power) on data that they may or may not be interested. Caching results from earlier queries is one such example - a node may have issued a query to some server (e.g., a data warehouse), the results of the query can be cached on the node (or some other neighbouring nodes). In this way, another node that requests for data that overlap the query result can potentially obtain partial answers quickly from this node, and the remainder from the original server. This also lightens the load on the original server. Indeed, Kalnis et. al. [5,8,9] have shown how distributed caching can be deployed in P2P environments to speed up OLAP queries.

D.Query Processing

This system gives the two mode processing approach i.e parallel processing approach and adaptive processing approach [1]. Query is submitted over normal peer using fetching and processing. The normal peer remote the subquery and the results are shuffled to the query submitting peer P [11]**.**

III. SYSTEM ARCHITECTURE

A.System Architecture

This system uses the pay-as-you-go business model popularized by cloud computing. By combining cloud computing, database, and peer-to-peer (P2P) technologies [6] containing a cloud development of Best Peer. At the last stage of its development, this System is improved with distributed access control, multiple types of indexes and pay-as you- go query processing for deliver elastic data sharing services in the cloud [5]. The software components of the system are separated into two parts: Core and Adapter. The Architecture is shown in Fig. 2. The core having all function of data sharing and it shows that it is platform independent.

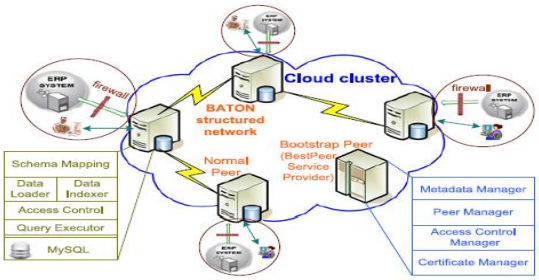

Fig. 1 The System Architecture

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 12, December 2015
Available at http://internationaljournalofresearch.org

The adapter contains one abstract adapter for getting portable service edge and a set of real adapter components which by specific cloud service providers (e.g., Amazon). To achieve portability it build two level design, With appropriate adapters, this System can be portable to any cloud environments (public and private) or even non-cloud
environment (e.g., on-premise data centre) [7]. The System can be implemented an adapter for Amazon cloud platform.

B. Components of the System

1.Amazon Cloud Adapter*:* The main approach of the system is to use committed database servers to store data for each business and arrange those database servers through P2P network for data sharing [3, 10]. The Amazon Cloud Adapter provides portable hardware transportation for system by using Amazon Cloud services.

2) The Core*:* This system having core contains all portable logic, containing query processing and P2P overlay [3]. The core consists of two parts of software i.e Bootstrap Peer, Normal Peer. In Bootstrap peer which is containing only single instances as service provider and having set of normal peer instances. It is considered as entry point of whole network [10]. In normal peer works under two mode i.e on-line mode and off-line mode. The data flow using normal peer in the system that id off line data flow and on-line data flow shown as Fig. 2: (a) and Fig.2: (b)

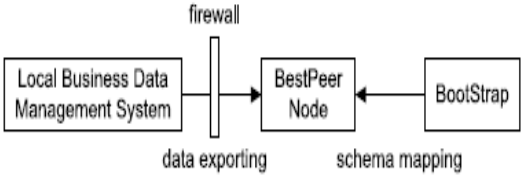

Fig.2: (a) Off-line Data Flow

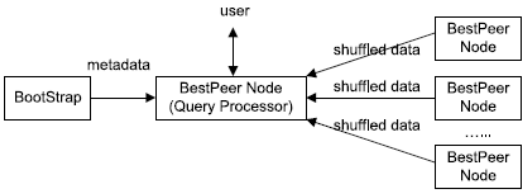

Fig.2: (b) On-line Data Flow

Algorithm 1: BootStrapDaemon()

```
1:  while true do
2:      Status S = invokeCloudWatch()
3:      ArrayList peerList = BootStrap.getAllPeer()
4:      ArrayList newPeer= new ArrayList()
5:      for i=0 to peerList.size() do
6:          if peerList.get(i).fails() then
7:              Peer peer = new Peer()
8:              peer.loadMySQLBackUpFromRDS(peerList.get(i))
9:              newPeer.add(peer)
10:             BootStrap.setBlackList(peerList.get(i))
11:         else
12:             if peerList.get(i).overloaded() then
13:                 Peer peer = new Peer()
14:                 peer.upScale(peerList.get(i))
15:                 peer.clone(peerList.get(i).getDB())
16:                 BootStrap.setBlackList(peerList.get(i))
17:                 newPeer.add(peer)
18:     BootStrap.removeAllPeersInBlackList()
19:     BootStrap.addAllNewPeer(newPeer)
20:     BootStrap.broadcastNetworkStatus()
21:     sleep T seconds
```

3) Query Processor*:* This system is design for achieve high query processing. These queries occupy querying a very small number of business partners and can be processed in small moment [3]. This system is mainly developed for these queries. For less time-consuming analytical tasks, they provide an interface for exporting the data from the system to Hadoop and allow users to analyse those data using Map Reduce [10].

IV. CONCLUSION

A novel query model, namely, clustering-based query answering, is studied intensively in this paper. The research is driven by the applictions, e.g. file sharing and information retrieval, in peer-to-peer environments. CBQA meets the requests of these applications in that, 1) the model generalizes several traditional clustering criteria, so that it can be used to solve more analysis problems than tranditional similarity queries. 2) Only a small portion of dataset need to be transferred over network under certain conditions, while the query

result is proved to be consistent with the global clustering. In other words, the query result is guaranteed, while the cost is low. 3The query model can be supported by current P2P platforms easily.

V.REFERENCES

[1] K. Aberer, A. Datta, and M. Hauswirth, ―Route Maintenance Overheads in DHT Overlays,‖ in 6th Workshop Distrib. Data Struct., 2004.

[2] A. Abouzeid, K. Bajda-Pawlikowski, D.J. Abadi, A. Rasin, and A.Silberschatz, ―HadoopDB: An Architectural Hybrid of MapReduce and DBMS Technologies for Analytical Workloads,‖ Proc. VLDB Endowment, vol. 2, no. 1, pp. 922-933, 2009.

[3] C. Batini, M. Lenzerini, and S. Navathe, ―A Comparative Analysis of Methodologies for Database Schema Integration,‖ ACM Computing Surveys, vol. 18, no. 4, pp. 323-364, 1986.

[4] D. Bermbach and S. Tai, ―Eventual Consistency: How Soon is Eventual? An Evaluation of Amazon s3's Consistency Behavior,‖ in Proc. 6th Workshop Middleware Serv. Oriented Comput. (MW4SOC '11), pp. 1:1-1:6, NY, USA, 2011.

[5] B. Cooper, A. Silberstein, E. Tam, R. Ramakrishnan, and R. Sears, ―Benchmarking Cloud Serving Systems with YCSB,‖ Proc. First ACM Symp. Cloud Computing, pp. 143-154, 2010.

[6] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P.Vosshall, and W. Vogels, ―Dynamo: Amazon's Highly Available Key-Value Store,‖ Proc. 21st ACM SIGOPS Symp. Operating Systems Principles (SOSP '07), pp. 205-220, 2007.

[7] J. Dittrich, J. Quian_e-Ruiz, A. Jindal, Y. Kargin, V. Setty, and J. Schad, ―Hadoop++: Making a Yellow Elephant Run Like a Cheetah (without it Even Noticing),‖ Proc. VLDB Endowment, vol. 3, no. 1/2, pp. 515-529, 2010.

[8] M. M. Breunig, H.-P. Kriegel, P. Kroger, and J. Sander. Data bubbles: Quality preserving performance boosting for hierarchical clustering. In Proceedings of ACM SIGMOD 2001 International Conference on Management of Data (SIGMOD'2001), 2001.

[9] R. O. Duda, P. E. Hart, and D. G. Stork. Pattern Classification. Wiley-Interscience, 2nd edition, 2000. [10] M. Ester, H.-P. Kriegel, J. Sander, M. Wimmer, and X. Xu. Incremental clustering for mining in a data warehousing environment. Conference on Very Large Data Bases (VLDB'98), pages 323–3.