ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Best Peer++: A Peer-to-Peer Based Large-Scale Data Processing Platform

N.Vijayalaxmi
M.Tech Student,
Dept of CSE,
KITS for Women's, Kodad, T.S, India.

Mr.B.Ramesh
Associate Professor,
Dept of CSE,
KITS for Women's, Kodad, T.S, India.

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. By integrating cloud computing, database, and P2P technologies into one system, BestPeer++ provides an economical, flexible and scalable platform for corporate network applications and delivers data sharing services to participants based on the widely accepted pay-as-you-go business model. We evaluate BestPeer++ on Amazon EC2 Cloud platform. The benchmarking results show that BestPeer++ outperforms HadoopDB, a recently proposed large-scale data processing system, in performance when both systems are employed to handle typical corporate network workloads. The benchmarking results also demonstrate that BestPeer++ achieves near linear scalability for throughput with respect to the number of peer nodes.

Index Terms:

Peer-to-peer systems, cloud computing, MapReduce, query processing, index.

1. INTRODUCTION:

Sharing Companies having common interest are always connected to a corporate network for sharing purposes [2].

A company creates its own website and shares a part of its business data with others which include supply chain networks such as supplier, manufacturer, and retailer who co-operate with each other to achieve their goals such as business planning, reducing production cost, developing business strategies and marketing solutions. Selecting right data sharing. platform is very important task for sharing network. Usually, centralized data such as Data warehouse is used for data sharing, which extracts data from the internal production systems (e.g., ERP) of each company for following querying. Actually this data warehouse having some deficiency Such as , First, The share data network wants to scope up to support thousands of participants. Second, companies want to fully modify the access control rule to determine which business partners can see which part of their shared data. Most of them failed to overcome such problem.

At last to increase the revenue; companies may change their business partners. Therefore, the participants may join and leave the share networks at resolve . This situation cannot be handled by physical data warehouse, to overcome such problem this designs the system for Shared Network for data sharing. 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. To address the aforementioned problems, this paper presents BestPeer++, a cloud enabled data sharing platform designed for corporate network applications. By integrating cloud computing, database, and peer-to-peer (P2P) technologies, BestPeer++ achieves its query processing efficiency and is a promising approach for corporate network applications, with the following distinguished features. _ BestPeer++ is deployed as a service in the cloud.

To form a corporate network, companies simply register their sites with the BestPeer++ service provider, launch BestPeer++ instances in the cloud and finally export data to those instances for sharing. BestPeer++ adopts the pay-as-you-go business model popularized by cloud computing [9]. The total cost of ownership is therefore substantially reduced since companies do not have to buy any hardware/software in advance. Instead, they pay for what they use in terms of BestPeer++ instance's hours and storage capacity. BestPeer++ extends the role-based access controlfor the inherent distributed environment of corporate networks. Through a web console interface, companies can easily configure their access control policies and prevent undesired business partners to access their shared data. _ BestPeer++ employs P2P technology to retrieve data between business partners. BestPeer++ instances are organized as a structured P2P overlay network named BATON .

The data are indexed by the table name, column name and data range for efficient retrieval. _ BestPeer++ employs a hybrid design for achieving high performance query processing. The major workload of a corporate network is simple, lowoverhead queries. Such queries typically only involve querying a very small number of business partners and can be processed in short time. Best-Peer++ is mainly optimized for these queries. Forinfrequent time-consuming analytical tasks, we provide an interface for exporting the data from Best- Peer++ to Hadoop and allow users to analyze those data using MapReduce.

2 OVERVIEW OF THE BESTPEER++ SYSTEM:

In this section, we first describe the evolution of Best-Peer platform from its early stage as an unstructured P2P query processing system to BestPeer++, an elastic data sharing services in the cloud. We then present the design and overall architecture of BestPeer++. BestPeer1 data management platform. While traditional P2P network has not been designed for enterprise applications, the ultimate goal of BestPeer is to bring the state-ofar database techniques into P2P systems. In its early stage, BestPeer employs unstructured network and information retrieval technique to match columns of different tables automatically [15]. After defining the mapping functions, queries can be sent to different nodes for processing. BestPeer++, a cloud enabled evolution of BestPeer.

Now in the last stage of its evolution, BestPeer++ is enhanced with distributed access control, multiple types of indexes, and pay-as-you-go query processing for delivering elastic data sharing services in the cloud. The software components of BestPeer++ are separated into two parts: core and adapter. The core contains all the data sharing functionalities and is designed to be platform independent.

The adapter contains one abstract adapter which defines the elastic infrastructure service interface and a set of concrete adapter components which implement such an interface through APIs provided by specific cloud service providers (e.g., Amazon). We adopt this "two-level" design to achieve portability.

With appropriate adapters, BestPeer++ can be ported to any cloud environments (public and private) or even non-cloud environment (e.g., on-premise data center).Currently, we have implemented an adapter for Amazon cloud platform. In what follows, we first present this adapter and then describe the core components.

2.1 Amazon Cloud Adapter:

The key idea of BestPeer++ is to use dedicated database servers to store data for each business and organize those database servers through P2P network for data sharing. The Amazon Cloud Adapter provides an elastic hardware infrastructure for BestPeer++ to operate on by using Amazon Cloud services. The infrastructure service that Amazon Cloud Adapter delivers includes launching/terminating dedicated MySQL database servers and monitoring/backup/auto-scaling those servers.

We use Amazon EC2 service to provision the database server. Each time a new business joins the BestPeer++ network, a dedicated EC2 virtual server is launched for that business. The newly launched virtual server (called a BestPeer++ instance) runs a dedicated MySQL database software and the BestPeer++ software.

The BestPeer++ instance is placed in a separate network security group (i. e., a VPN) to prevent invalid data access. Users can only use BestPeer++ software to submit queries to the network.

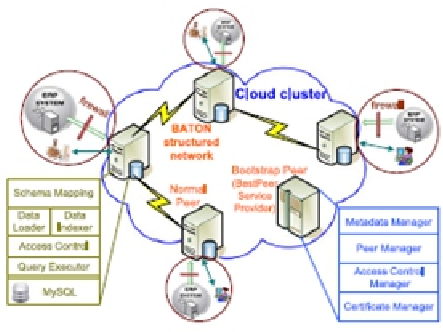


Fig. 1. The BestPeer++ network deployed on Amazon Cloud offering.

3 BOOTSTRAP PEER:

The bootstrap peer is run by the BestPeer++ service provider, and its main functionality is to manage the BestPeer+ + network. This section presents how bootstrap peer performs various administrative tasks.

3.1 Managing Normal Peer Join/Departure:

Each normal peer intends to join an existing corporate network must first connect to the bootstrap peer. If the join request is permitted by the service provider, the bootstrap peer will put the newly joined peer into the peer list of the corporate network. At the same time, the joined peer will receive the corporate network information including the current participants, global schema, role definitions, and an issued certificate.

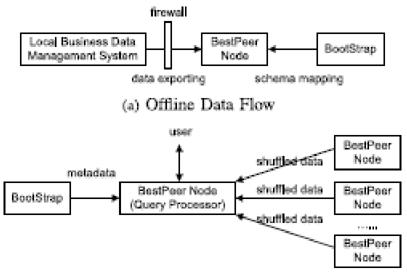
When a normal peer needs to leave the network, it also notifies the bootstrap peer first. The bootstrap peer will move the departure peer to the black list and mark the certificate of the departing peerinvalid. The bootstrap peer will the reclaim all resources allocated to the departing peer and finally remove the departing peer from the peer list.

3.2 Auto Fail-Over and Auto-Scaling:

In addition to managing peer join and peer departure, the bootstrap peer spends most of its running-time on monitoring the healthy of normal peers and scheduling failover and auto-scaling events. Algorithm 1 shows how the daemon service of the bootstrap works.

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
```



(b) Online Data Flow

The bootstrap periodically collects performance metrics of each normal peer (line 2). If some peers are malfunctionedor crashed, the bootstrap peer will trigger ana utomatic fail-over event for each failed normal peer (line 6-10). The automatic fail-over is performed by first launching a new instance from cloud. Then, the bootstrap peer asks the newly launched instance to perform database recovery from the latest database backup stored in Amazon EBS. Finally, the failed peer is put into the blacklist. Similarly, if any normal peer is overloaded (e. g., CPU is over-utilized or free storage space is low), the bootstrap peer triggers an auto-scaling event (line 12-17) to either upgrade the normal peer to a larger instance or allocate more storage spaces.

4 NORMAL PEER:

The normal peer software consists of five components: schema mapping, data loader, data indexer, access control, and query executor. We present the first four components in this section. Query processing in BestPeer++ will be presented in the next section. As shown in Fig. 2, there are two data flows inside the normal peer: an offline data flow and an online data flow.In the offline data flow, the data are extracted periodically by a data loader from the business production system to the normal peer instance. In particular, the data loader extracts the data from the business production system, transforms the data format from its local schema to the shared global schema of the corporate network according to the schema mapping, and finally stores the results in the MySQL databases hosted in the normal peer.

4.1 Schema Mapping:

Schema mapping [3] is a component that defines the mapping between the local schema of each production system and the global shared schema employed by the corporate network. Currently, BestPeer++ only supports relational schema mapping, namely both local schema and the global schema are relational. The mapping consists of metadata mappings (i.e., mapping local table definitions to global table definitions) and value mappings (i. e., mapping local terms to global terms). Besides schemalevel mapping, BestPeer++ can also support instancelevel mapping [19], which complements the mapping process when there is not sufficient schema information.

4.2 Data Loader:

Data Loader is a component that extracts data from production systems to normal peer instances according to the result of schema mapping. While the process of extracting and transforming data is straightforward, the main challenge comes from maintaining consistency between raw data stored in the production systems and extracted data stored in the normal peer instance (and subsequently data indices created from these extracted data) while the raw data being updated inside the production systems.

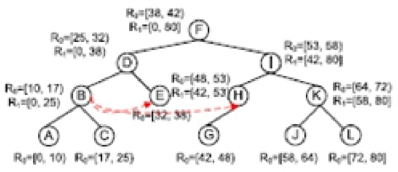


Fig. 3. BATON overlay.

5 PAY-AS-YOU-GO QUERY PROCESSING:

BestPeer++ provides two services for the participants: the storage service and search service, both of which are charged in a pay-as-you-go model. This section presents the pay-as-you-go query processing module which offers an optimal performance within the user's budget. We begin with the presentation of histogram generation, a building block for estimating intermediate result size.

Then, we present the query processing strategy. Before discussing the details of query processing, we first define the semantics of query processing in the Best- Peer++. After data are exported from the local business system into a BestPeer++ instance, we apply the schema mapping rules to transform them into the predefined formats. In this way, given a table T in the global schema, each peer essentially maintains a horizontal partition of it.

5.1 The Histogram:

In BestPeer++, histograms are used to maintain the statistics of column values for query optimization. Since attributes in a relation are correlated, single-dimensional histograms are not sufficient for maintaining the statistics. Instead, multi-dimensional histograms are employed. Best- Peer++ adopts MHIST [17] to build multi-dimensional histograms adaptively. Each normal peer invokes MHIST to iteratively split the attribute which is most valuable for building histograms until enough histogram buckets are generated. Then, the buckets (multi-dimensional hypercube) are mapped into one dimensional ranges using iDistance [12] and we index the buckets in BATON based on their ranges.

6 BENCHMARKING:

This section evaluates the performance and throughput of BestPeer++ on Amazon cloud platform. For the performance benchmark, we compare the query latency of Best-Peer++ with HadoopDB using five queries selected from typical corporate network applications workloads. For the throughput benchmark, we create a simple supply-chain network consisting of suppliers and retailers and study the query throughput of the system.6.1 Performance Benchmarking This benchmark compares the performance of BestPeer++ with HadoopDB. We choose HadoopDB as our benchmark target since it is an alternative promising solution for our problem and adopts an architecture similar to ours. Comparing the two systems (i.e., HadoopDB and BestPeer++) reveals the performance gap between a general data warehousing system and a data sharing system specially designed for corporate network applications. 6.1.1 Benchmark Environment We run our experiments on Amazon m1.small DB instances launched in the ap-southeast-1 region. Each DB small instance has 1.7 GB memory, 1 EC2 Compute Unit (1 CPU virtual core). We attach each instance with 50 GB storage space. We observe that the I/O performance of Amazon cloud is not stable. The hdparm reports that the buffered read performance of each instance ranges from 30 to 120 MB/sec. To produce a consistent benchmark result, we run our experiments at the weekend when most of the instances are idle. Overall, the buffered read performance of each small instance is about 90 MB/ sec during our benchmark. The end-to-end network bandwidth between DB small instances, measured by iperf, is approximately 100 MB/sec. We execute each benchmark query three times and report the average execution time.

The benchmark is performed on cluster sizes of 10, 20, 50 nodes. For the BestPeer++ system, these nodes are normal peers. We launch an additional dedicated node as the bootstrap peer. For HadoopDB system, each launched cluster node acts as a worker node which hosts a Hadoop task tracker node and a PostgreSQL database server instance. We also use a dedicated node as the Hadoop job tracker node and HDFS name node. 6.1.2 BestPeer++ Settings The configuration of a BestPeer++ normal peer consistsof two parts: the underlying MySQL database server and the BestPeer++ software.

7 RELATED WORK:

To enhance the usability of conventional P2P networks, database community have proposed a series of PDBMS (Peer-to-Peer Database Manage System) by integrating the state-of-art database techniques into the P2P systems. These PDBMS can be classified as the unstructured systems such as PIAZZA [22], Hyperion [20] and PeerDB [15], and the structured systems such as PIER [10]. The work on unstructured PDBMS focus on the problem of mapping heterogeneous schemas among nodes in the systems.

PIAZZA introduces two materialized view approaches, namely local as view (LAV) and global as view (GAV). PeerDB employs information retrieval technique to match columns of different tables. The main problem of unstructured PDBMS is that there is no guarantee for the data retrieval performance and result quality. The structured PDBMS can deliver search service with guaranteed performance. The main concern is the possibly high maintenance cost [1]. To address this problem, partial indexing scheme [26] is proposed to reduce the index size. Moreover, adaptive query processing [27] and online aggregation [25] techniques have also been introduced to improve query performance.

The techniques of PDBMS are also adopted in cloud systems. In Dynamo [6], Cassandra [14], and ecStore [24], a similar data dissemination and routing strategy is applied to manage the large-scale data.

BestPeer++ is different from the systems based on the MapReduce/Hadoop framework (e.g., HadoopDB [2], Hive [23] and Hadoop++ [7]). Hadoop-based systems are designed to process large-scale data sets in batch mode. They efficiently process aggregate queries by exploiting the parallelism. The SQL queries need to be translated binto multiple MapReduce jobs, which are processed sequentially.

BestPeer++, on the other hand, can handle both ad-hoc queries and costly analysis queries. It providesbuilt-in MapReduce support and adaptively switches between its distributed processing strategy and MapReduce strategy based on the cost model. BestPeer++ shares a similar design philosophy with HadoopDB. In both systems, each processing instance maintains a local DBMS. The local DBMS helps manage the local data and improve the query processing with the database techniques, such as index and optimizer.

8 CONCLUSION:

We have discussed the unique challenges posed by sharing and processing data in an inter-businesses environment and proposed BestPeer++, a system which delivers elastic data sharing services, by integrating cloud computing, database, and peer-to-peer technologies. The benchmark conducted on Amazon EC2 cloud platform shows that our system can efficiently handle typical workloads in a corporate network and can deliver near linear query throughput as the number of normal peers grows. Therefore, BestPeer++ is a promising solution for efficient data sharing within corporate networks.

9.ACKNOWLEDGMENTS:

I am N.Vijayalaxmi and would like to thank the publishers, researchers for making their resources material available. I am greatly thankful to Associate Prof: MR. Mr.B.Ramesh for their guidance. We also thank the college authorities, PG coordinator and Principal for providing the required infrastructure and support. Finally, we would like to extend a heartfelt gratitude to friends and family members.

10. 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] H. Garcia-Molina and W.J. Labio, "Efficient Snapshot Differential Algorithms for Data Warehousing," technical report, Stanford Univ., 1996.

[9] Google Inc., "Cloud Computing-What is its Potential Value for Your Company?" White Paper, 2010.

[10] R. Huebsch, J.M. Hellerstein, N. Lanham, B.T. Loo, S. Shenker, and I. Stoica, "Querying the Internet with PIER," Proc. 29th Int'lConf. Very Large Data Bases, pp. 321-332, 2003.

Author's Details:

Miss N.Vijayalaxmi. MTech student, in M.Tech Student, Dept of CSE in KITS for women's,kodad, T.S, India

MR. Mr.B.Ramesh working as a Associate at CSE in KITS for women's,kodad, T.S, IndiaJNTUH Hyderabad. He has 2 years of UG/PG Teaching Experience