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

Dr. K. Sekar, Professor, Dept. of CSE (AI&ML),
D. Shashikala, Billu Eswaraiah, Batchu Mahesh Kumar, P. Rakesh, PC Prakash, M.M Rao
PG Scholars,Department of CSE, Chadalawada Ramanamma Engineering College (Autonomous), Tirupati.

Abstract: The present Best Peer++, a system which delivers elastic data sharing services for corporate network applications in the cloud based on Best Peer—a peer-to-peer (P2P) based data management platform. By integrating cloud computing, database, and P2P technologies into one system, Best Peer++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.

Key words: System, Flexible, P2P, Scalable, Cloud computing.

1.Introduction:

The inter-company data sharing and processing poses unique challenges to such a data management system including scalability, performance, throughput, and security. In this paper, we present Best Peer++, a system which delivers elastic data sharing services for corporate network applications in the cloud based on Best Peer—a peer-to-peer (P2P) based data management platform. By integrating cloud computing, database, and P2P technologies into one system, Best Peer++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 Best Peer++ on Amazon EC2 Cloud platform. The benchmarking results show that Best Peer++ outperforms Hadoop DB, 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 Best Peer++ achieves near linear scalability for throughput with respect to the number of peer nodes. A peer-to-peer (P2P) network is created when two or more PCs are connected and share resources without going through a separate server computer. A P2P network can be an ad hoc connection—a couple of computers connected via a Universal Serial Bus to transfer files. A P2P network also can be a permanent infrastructure that links a half-dozen computers in a small office over copper wires. Or a P2P network can be a network on a much grander scale in which special protocols and applications set up direct relationships among users over the Internet. The corporate network is often used for sharing information among the participating companies and facilitating collaboration in a certain industry sector where companies share a common interest. It can effectively help the companies to reduce their operational costs and increase the revenues. Sharing Companies having common interest are always connected to a corporate network for sharing purposes. A Through a web console interface, companies can easily configure their access control policies and prevent undesired business partners to access their shared data. Best Peer++ employs P2P technology to retrieve data between business partners. Best Peer++ 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. Best Peer++ employs a hybrid design for achieving high performance query processing. The major workload of a corporate network is simple, low over head 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. For in frequent time consuming analytical tasks, we provide an interface for exporting the data from Best- Peer++ to Hadoop and allow users to analyse those data using Map Reduce. 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.

2. Existing System

Such a warehousing solution has some deficiencies in real deployment. First, the corporate network needs to scale up to support thousands of participants, while the installation of a large-scale centralized data warehouse system entails nontrivial costs including huge hardware/software investments (a.k.a total cost of ownership) and high maintenance cost (a.k.a total cost of operations) . In the real world, most companies are not keen to invest heavily on additional information systems until they can clearly see the potential return on investment (ROI). Second, companies want to fully customize the access control policy to determine which business partners can see which part of their shared data.

Disadvantages of Existing System

- Most of the data warehouse solutions fail to offer such flexibilities.
- Solution has not been designed to handle such dynamicity.

3.Proposed System

The main contribution of this paper is the design of Best Peer++ system that provides economical, flexible and scalable solutions for corporate network applications. We demonstrate the efficiency of Best Peer++ by benchmarking Best Peer++ against Hadoop DB, a recently proposed large-scale data processing system, over a set of queries designed for data sharing applications. The results show that for simple, low-overhead queries, the performance of Best Peer++ is significantly better than Hadoop DB. The unique challenges posed by sharing and processing data in an inter-businesses environment and proposed Best Peer++, a system which delivers elastic data sharing services, by integrating cloud computing, database, and peer-to-peer technologies.

Advantages

- 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.
- Best Peer++ adopts the pay-as-you-go business model popularized by cloud computing. 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 Best Peer++ instance's hours and storage capacity.
- Best Peer++ extends the role-based access control for the inherent distributed environment of corporate networks.
- Best Peer++ employs P2P technology to retrieve data between business partners.

4. Soft ware Description

The software requirements specification is produced at the culmination of the analysis task. The function and performance allocated to the software as a part of system engineering are refined by establishing a complete information description, a detailed functional and behavioural description, and indication of performance requirements and design constraints, appropriate validation criteria and other data pertinent to requirements.

Software Environment

Java Technology

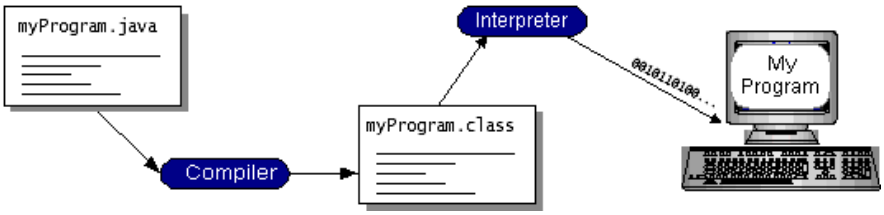
Java technology is both a programming language and a platform.

The Java Programming Language

The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

- Simple
- Architecture neutral
- Object oriented
- Portable
- Distributed
- High performance
- Interpreted
- Multithreaded
- Robust
- Dynamic
- Secure

With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called Java byte codes —the platform-independent codes interpreted by the interpreter on the Java platform. The interpreter parses and runs each Java byte code instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed. The following figure illustrates how this works.



An overview of the Software Development Process

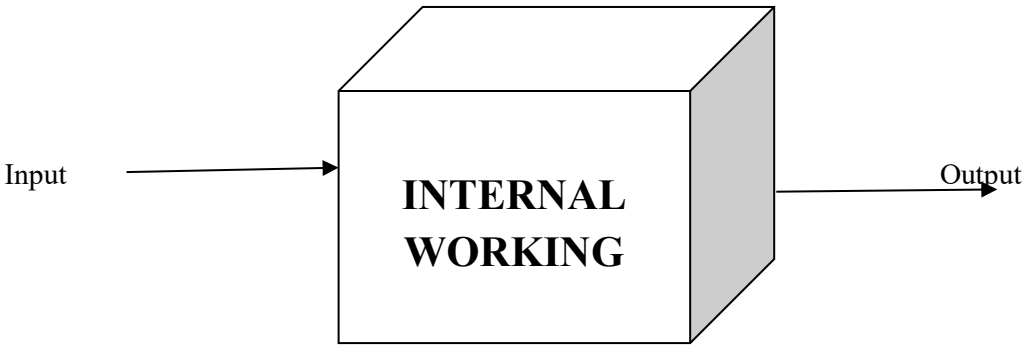
You can think of Java byte codes as the machine code instructions for the Java Virtual Machine (Java VM). Every Java interpreter, whether it's a development tool or a Web browser that can run applets, is an implementation of the Java VM. Java byte codes help make "write once, run anywhere" possible. You can compile your program into byte codes on any platform that has a Java compiler. The byte codes can then be run on any implementation of the Java VM. That means that as long as a computer has a Java VM, the same program written in the Java programming language can run on Windows 2000, a Solaris workstation, or on an iMac.

5. System Testing

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

(a) White box testing

White box testing is concerned with testing the implementation of the program. The intent of structural is not to exercise all the inputs or outputs but to exercise the different programming and data structure used in the program. Thus structural testing aims to achieve test cases that will force the desire coverage of different structures. Two types of path testing are statement testing coverage and branch testing coverage.



The White Box testing strategy, the internal workings

Unit testing focuses on the building blocks of the software system, that is, objects and sub system. There are three motivations behind focusing on components. First, unit testing reduces the complexity of the overall tests activities, allowing us to focus on smaller units of the system. Second, unit testing makes it easier to pinpoint and correct faults given that few components are involved in this test. Third, Unit testing allows parallelism in the testing activities, that s each component can be tested independently of one another. Hence the goal is to test the internal logic of the module.

(b) Negative Test Cases

Test case ID	Test cases	Procedure	Expected output	Actual output	Result
1	User Registration	User registration with insufficient details	Successfully Registered	Some information missing	Fail
2	Owner registration	Owner registration with insufficient details	Successfully Registered	Some information missing	Fail
3	User Login	User login with user id and wrong password	Successfully Login	Please enter valid password	Fail
4	Owner Login	Owner login with owner id and wrong password	Successfully Login	Please enter valid password	Fail
5	IP address	Entered wrong IP address	Connected to next page	Wrong IP address	Fail

6	Secret key	Entered wrong Secret key	Successfully file downloaded	unsuccessfully file downloaded	Fail

Negative Test Cases

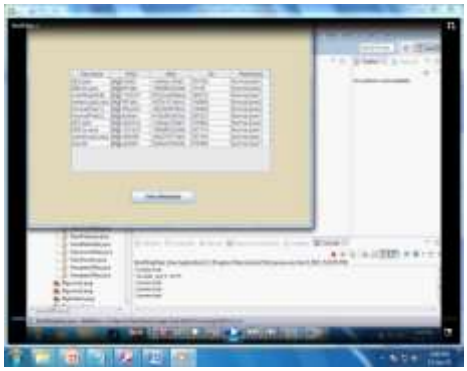
Positive Test cases

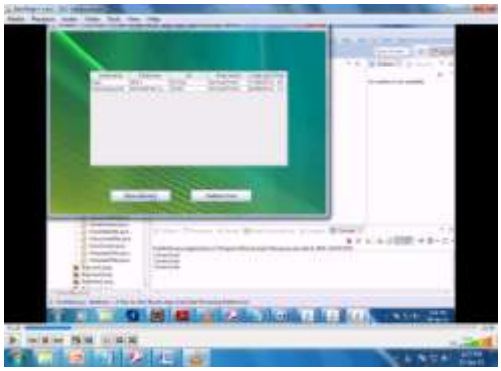
Test case ID	**Test cases**	**Procedure**	**Expected output**	**Actual output**	**Result**
1	User registration	Enter into the user register with details	Successfully Registered	Successfully Registered	Pass
2	Owner registration	Enter into the owner register with details	Successfully Registered	Successfully Registered	Pass
3	User Login	User login with user id and password	Successfully Login	Successfully Login	Pass
4	Owner Login	Owner login with owner id and password	Successfully Login	Successfully Login	Pass
5	Owner file Upload	Owner file upload with a Secret key	Successfully Uploaded	Successfully uploaded	Pass
6	User file Download	User file download with file name and Secret key	Successfully download	Successfully download	Pass

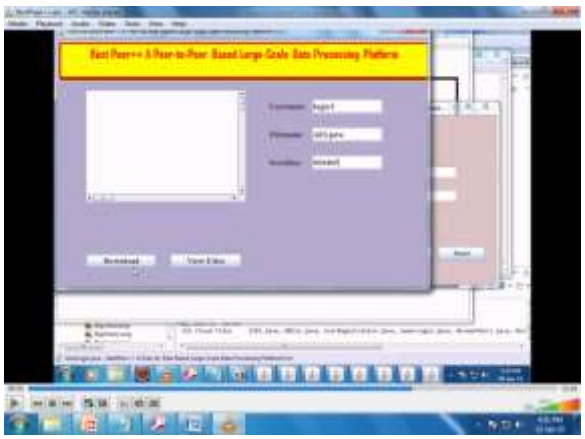
Table : Positive Test Cases

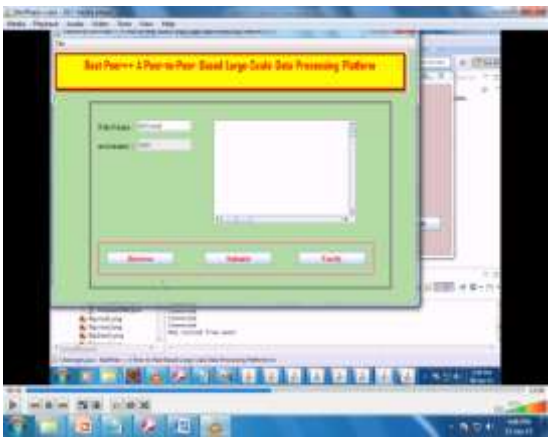
© Integration Testing

In the integration testing, many test modules are combined into sub systems, which are then tested. The goal here is to see if the modules can be integrated properly, the emphasis being on testing module interaction. After structural testing and functional testing we get error free modules. These modules are to be integrated to get the required results of the system. After checking a module, another module is tested and is integrated with the previous module. After the integration, the test cases are generated and the results are tested.









Conclusion

The unique challenges posed by sharing and processing data in an inter-businesses environment and proposed Best Peer++, 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, Best Peer++ is a promising solution for efficient data sharing within corporate networks.

In Best peer++, we are proposing the solution for large data processing platform. Here we integrated with cloud system for large scale data processing. But all the situations, Cloud environment is not a feasible one. Because of cost factor, The solution need to be address with open projects like hadoop. The data which is sharing in cloud we don't to put constraints. Because our system is integrated with peer to peer architecture.

References

[1] K. Aberer, A. Datta, and M. Hauswirth. Route Maintenance Overheads in DHT Overlays. In The 6th Workshop on Distributed Data and Structures, 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. PVLDB, 2(1):922–933 , 2009.

[3] 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. In SOSP, pages 205–220 , 2007.

[4] H. Garcia-Molina and W. J. Labio. Efficient Snapshot Differential Algorithms for Data Warehousing. Technical report, Stanford, CA, USA, 1996.

[5] Google Inc. Cloud Computing-What is its Potential Value for Your Company? White Paper, 2010.
[6] R. Huebsch, J. M. Hellerstein, N. Lanham, B. T. Loo, S. Shenker, and I. Stoica. Querying the Internet with PIER. In VLDB, pages 321–332 , 2003.
[7] H. V. Jagadish, B. C. Ooi, K.-L. Tan, Q. H. Vu, and R. Zhang. Speeding up Search in Peer-to-Peer Networks with a Multi-Way Tree Structure. In SIGMOD, 2006.
[8] H. V. Jagadish, B. C. Ooi, K.-L. Tan, C. Yu, and R. Zhang. iDistance: An adaptive b+-tree based indexing method for nearest neighbor search. ACM Trans. Database Syst., 30:364–397, June 2005.
[9] H. V. Jagadish, B. C. Ooi, and Q. H. Vu. BATON: A Balanced Tree Structure for Peer-to-Peer Networks. In VLDB, pages 661–672, 2005.
[10] A. Lakshman and P. Malik. Cassandra: structured storage system on a p2p network. In PODC, pages 5–5, 2009.