JASEM ISSN 1119-8362
All rights reserved

Full-text Available Online at
www.ajol.info and
www.bioline.org.br/ja

J. Appl. Sci. Environ. Manage. *June 2014*
Vol. 18 (2) 249-253

A Database Query Processing Model in Peer-To-Peer Network

***[1]NDUESO, EE; ASAGBA, PO**

Department of Computer Science,
University of Port Harcourt, Port Harcourt Nigeria
ekendumo@yahoo.com, pasagba@yahoo.com

KEYWORDS: Peer-To-Peer Network, Database Query, Query Processing, Object-Oriented Systems

ABSTRACT: Peer-to-peer databases are becoming more prevalent on the internet for sharing and distributing applications, documents, files, and other digital media. The problem associated with answering large-scale ad hoc analysis queries, aggregation queries, on these databases poses unique challenges. This paper presents an extensive evaluation of database query processing model in peer-to-peer networks using top-*k* query processing technique and implemented by Java,and MySql. ©JASEM

http://dx.doi.org/10.4314/jasem.v18 i2.15

Introduction: The researches in the area of query processing in peer-to-peer networks have started to increase the efficiency of retrieving and exchanging information between peers and file sharing (Gupta and Abbadi, 2003). They have developed a peer-to-peer architecture for computing approximate answers for complex queries. However, it can only return approximate results. (Dongsheng et.al, 2009) have proposed a delay-bounded general range query scheme called Armanda. An efficient range query processing scheme to support delay bounded single and multiple attribute range queries .The maximum delay is less than 2log N which is efficient but can be improved. Since DHT is a structured topology, when peers join and leave the network. The whole structure should again be modified. (Boyd et.al, 2000) proposed a super peer search algorithm to discover the domains on the fly, which is required for the domain direct search. It reduces the response time by 35%. The advantage of this technique is that they provide robustness in queries routing. The advantage of this process is that, it avoids to making semantic mapping, between heterogeneous data sources owned by super peers, each time the system decides to route query to other super peers. It improves the robustness in queries routing mechanism and scalability in Peer-to-Peer Network. (Ogheneovo and Asagba, 2012) proposed a model based on grammatical structure of an SQL statement using parse tree to test a query by dynamically generating a parse tree and comparing their structures at runtime. They were able to determine if their structures match or not. If they match, the query is parsed signifying that it is legitimate; otherwise it is suspicious and possibly malicious. Their approach was targeted at improving security in database vulnerabilities. They moved to solve the problem of SQL injection attack which is a class of command injection attacks in which specially crafted input string result in illegal queries to a database (Ogheneovo and Asagba, 2013).

A database query is a piece of code (a query) or command that is sent to a database in order to get information back from the database (Stephen, 2013); its processing time in retrieving the information from database is the focus of this paper. In this paper, a database query processing scheme is proposed to limit the search space of the given query by a domain based query routing mechanism which limits the search to related peers. A database query processing model in peer-to-peer networks was designed using top-*k* query processing technique and implemented using Java, and MySql.

Supporting top-*k* processing in database systems is a relatively recent and active line of research. Top-*k* processing has been addressed from different perspectives in the current literature (George et.al, 2008). We consider a problem on peer-to-peer system that is different from the search and retrieval application. The vast amount of data within peer-to-peer database poses a different challenge that has not been adequately researched. Although data integration systems and parallel database systems can scale up to hundreds of database partitions, they still rely on centralized global schema and strong assumptions about the network (Arturo et.al, 2002). Since peer-to-peer systems have emerged as a powerful paradigm for file and data sharing over the Internet, a natural question arises if the power of

**Corresponding authors Email: ekendumo@yahoo.com, pasagba@yahoo.com* peer-to-peer systems can be harnessed to support database functionality over peer-to-peer systems. Indeed, several research initiatives are underway to answer this question. This paper has discusses goals in supporting database query processing over peer-to-peer systems. Peer-to-peer systems are mainly used for object sharing and file sharing, but they can provide the infrastructure for many other applications.

This system uses an algorithm to evaluate the queries efficiently so that the best of the results can be achieved in data sharing and message passing (Boyd et.al, 2000). The delay in the query can be caused by scale of the system and the size of the query space. (Ghada et.al, 2011) proposed a query search algorithm which keeps the search costs low. The peers are divided based on their content present in the different peers. All peers having similar data come under one domain. This minimizes the search of the requested query. (Yamada et.al, 2003) have proposed an indexing method for query processing in Peer-to-Peer systems. The proposed index stores the usefulness of peers and prevents peers from forwarding messages to less informative peers. To perform the query routing, this mechanism provides the required methods for specifying the interaction between the peers in the same or different groups (Ratnasamy and Stoica, 2002). It is very necessary to see that many factors like file sharing, file exchange and various search algorithms involved in peer to peer networks can be made more efficient by proper query processing.

MATERIAL AND METHODOLOGY

The platform used to implement our program is java and we have used single-attribute tuples. SQL Server 2005 is used for database creation in all peers. In an effort to set an independent database standard API for Java; Sun Microsystems developed Java Database Connectivity, or JDBC. JDBC offers a generic SQL database access mechanism that provides a consistent interface to a variety of Relational Database Management System (RDBMS). This consistent interface is achieved through the use of “plug-in” database connectivity modules, or drivers. If a database vendor wishes to have JDBC support, he or she must provide the driver for each platform that the database and Java run on it. To gain a wider acceptance of JDBC, Sun based JDBC’s framework on ODBC.

In this paper, we introduce the object oriented model and the principle of query processing. Object-oriented development offers a different model from the traditional software development approach, which is based on functions and procedures. In simplified terms, object-oriented systems development is a way to develop software by building self-contained modules or objects that can be easily replaced, modified, and reused. In an object-oriented environment:

Software is a collection of discrete objects that encapsulate their data as well as the functionality to model real-world "objects." An object orientation yields important benefits to the practice of software construction. Each object has attributes (data) and methods (functions). Objects are grouped into classes; in object-oriented terms, we discover and describe the classes involved in the problem domain. Everything is an object and each object is responsible for itself.

Object-oriented methods enable us to create sets of objects that works together synergistically to produce software that better model their problem domains than similar systems produced by traditional techniques. The systems are easier to adapt to changing requirements, easier to maintain, more robust, and promote greater design and code reuse. Object-oriented development allows us to create modules of functionality. Once objects are defined, it can be taken for granted that they will perform their desired functions and you can seal them off in your mind like black boxes.

The Architectural Design of the Existing System: The client/server architecture is cross-platform compatible, which allows you to run the client and server software in mixed operating system environments. So, for example, you can run the IPM server on a Solaris workstation and access it from an IPM client running on a Windows 95 workstation. Figure 1 shows the architectural design of the existing client server model.

The Proposed System Architecture: This is a type of network in which each workstation has equivalent capabilities and responsibilities. This differs from client/server architectures, in which some computers are dedicated to serving the others. Peer-to-peer networks are generally simpler but the usually do not offer the same performance under heavy loads. Figure 2 shows the system architecture of the proposed system.

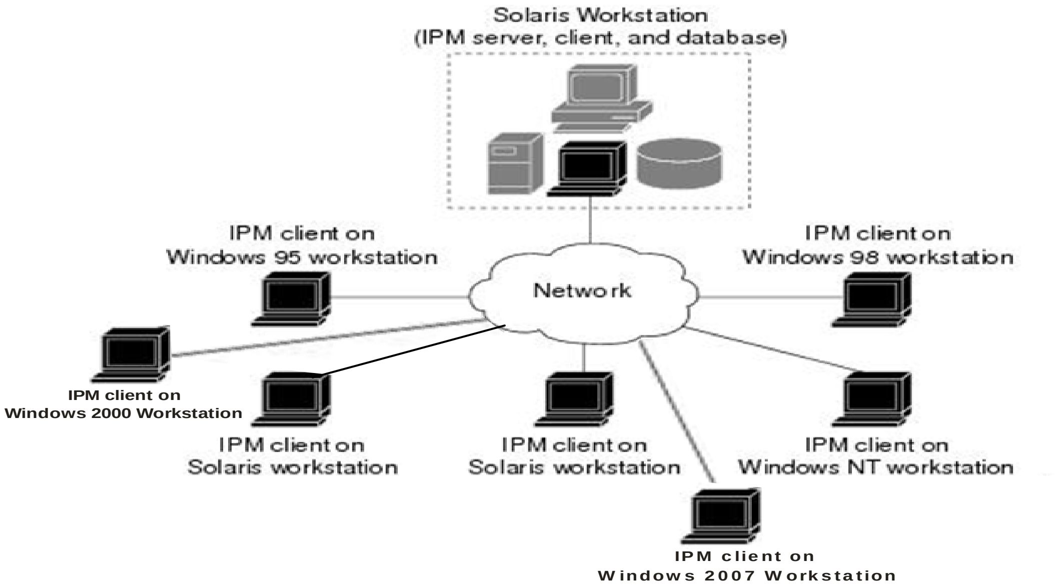


Fig. 1: Architectural design of client server

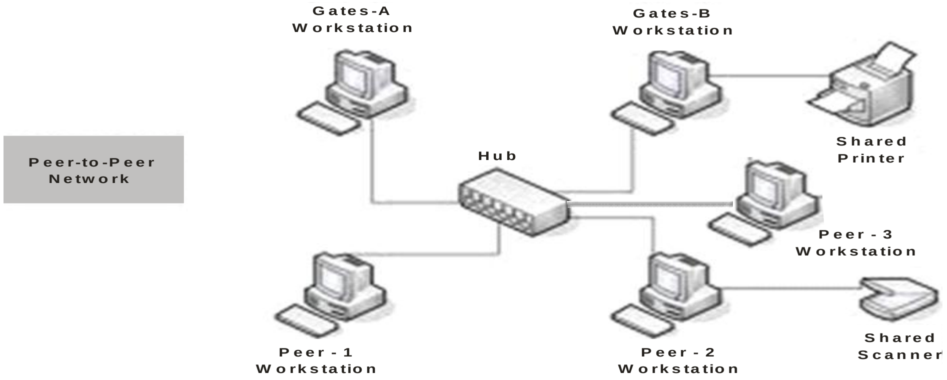


Fig. 2: The proposed system architecture

Algorithm of the Proposed System: The following algorithm shows the complete procedure for random walk in Peer-to-Peer network and efficient query processing. This algorithm is implemented in Java framework and used for selection of active peers in network randomly and execution of query on selected peers predefined values: M: total number of peers in a network, m: number of peers to visit, j: jump size for random walk, t: max tuples to be sub sampled per peer Inputs: Q: Query with selection condition; Sink: Peer where query is initiated Output: Query result to Sink (Peer where query is initiated) Start , Check number of active nodes , If number of peers = 1, .1. Execute query on that peer , Else perform Algorithm Curr = Sink; Hops = 1; While (Hops < j * m) { If (Hops % j) Visit (Curr); Hops ++; Curr = random adjacent peer , }, Visit (Curr){ If (# tuples of Curr) <= t){ Execute Q on all tuples , Else, Execute Q on t randomly sampled tuples , } Return result to sink , Compute Error rate, Exact Processing time, Approximate Processing time Return this result to Sink , End

A Proposed Database Query Processing Scheme: This paper proposes the use of top-k querying techniques for processing data in the database. In processing of top-*k*, queries are a crucial requirement in many interactive environments that involve massive amounts of data. In particular, top-*k* processing in domains such as the Web, multimedia search and distributed systems has shown a great impact on performance. In this paper, we describe and classify top-*k* processing techniques in relational databases. We discuss different design dimensions in the current techniques including query models; data access methods, implementation levels, data and query certainty, and supported scoring functions. We show the implications of each dimension on the design of the underlying techniques. Figure 3 shows a top-k Processing technique.

Top-*k* processing connects too many database research areas including query optimization, indexing methods and query languages. As a consequence, the impact of top-k includes: Processing is becoming evident in an increasing number of applications. The impact of efficient top-*k* processing is becoming evident in an increasing number of applications. Top-*k* similarity queries are traditionally answered using high-dimensional indexes built on individual video features, and a nearest neighbor scan operator on top of these indexes. Unlike the existing system, this approach does not suffer from scalability problems with respect to database size and the number of features.

It maps the query into a join query that joins the output of multiple single-feature queries, and then sorts the joined results based on combined score. This approach also does not scale with respect to both number of features and database size since all join results have to be computed then sorted. It enables the user to extract relevant information from a database. So that instead of scanning the whole table, they can pre-define the categories of information which will be sought. It also allows the user to join multiple tables. For example, if you have a table called "customers" and a table called "invoices", you can use a query to join the two tables. When you run a query that joins these two tables, you can get results showing the name of the customer (from the customer table) and all associated invoices (from the invoices table). It is used as the basis for reports in MySql. All databases can be called up from any peer on the network.

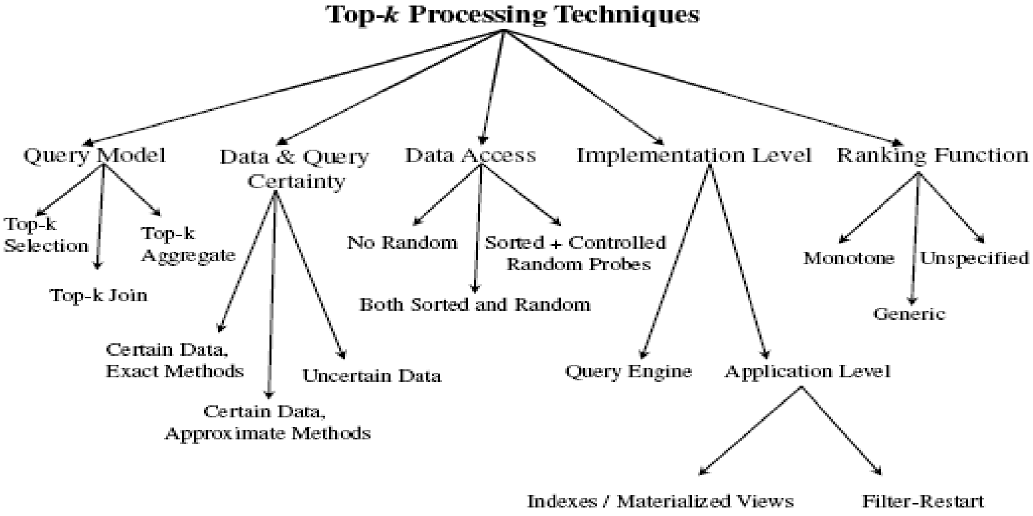
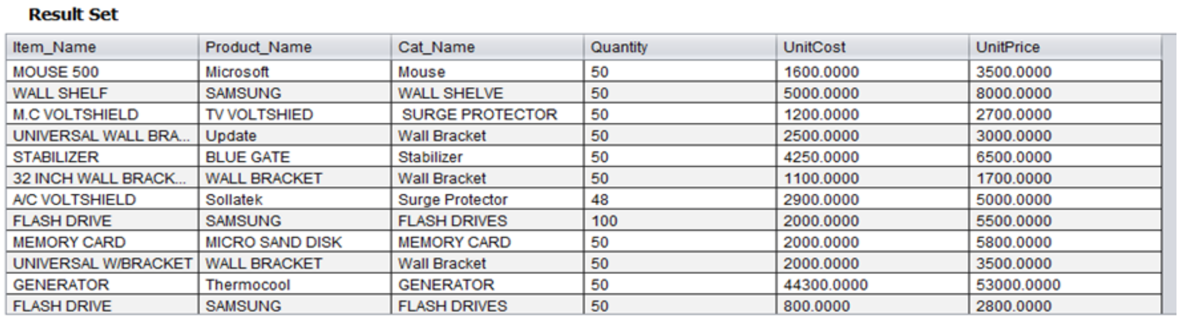


Fig. 3: Top-k Processing Techniques

Design of the Proposed System: This is the stage in the project where the theoretical design is turned into a working system and is giving confidence on the new system for the users that it will work efficiently and effectively. It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the change over, an evaluation of change over methods. Apart from planning major task of preparing the implementation are education and training of users. The implementation process begins with preparing a plan for the implementation of the system. According to this plan, the activities are to be carried out, discussions made regarding the equipment and resources and the additional equipment has to be acquired to implement the new system. In network backup system no additional resources are needed. Implementation is the final and the most important phase. The most critical stage in achieving a successful new system is giving the users confidence that the new system will work and be effective. The system can be implemented only after thorough testing is done and if it is found to be working according to the specification.

The login Class takes connects the user to the login section, and then in the login session helps the user in Querying the Processor. The user can as well access the database directly from the login section. The query processor will always query the database to get the required information.

Table 4. the Result Set



Result Set

Item_Name	Product_Name	Cat_Name	Quantity	UnitCost	UnitPrice
MOUSE 500	Microsoft	Mouse	50	1600.0000	3500.0000
WALL SHELF	SAMSUNG	WALL SHELVE	50	5000.0000	8000.0000
M.C VOLTSHIELD	TV VOLTSHIED	SURGE PROTECTOR	50	1200.0000	2700.0000
UNIVERSAL WALL BRA...	Update	Wall Bracket	50	2500.0000	3000.0000
STABILIZER	BLUE GATE	Stabilizer	50	4250.0000	6500.0000
32 INCH WALL BRACK...	WALL BRACKET	Wall Bracket	50	1100.0000	1700.0000
A/C VOLTSHIELD	Sollatek	Surge Protector	48	2900.0000	5000.0000
FLASH DRIVE	SAMSUNG	FLASH DRIVES	100	2000.0000	5500.0000
MEMORY CARD	MICRO SAND DISK	MEMORY CARD	50	2000.0000	5800.0000
UNIVERSAL W/BRACKET	WALL BRACKET	Wall Bracket	50	2000.0000	3500.0000
GENERATOR	Thermocool	GENERATOR	50	44300.0000	53000.0000
FLASH DRIVE	SAMSUNG	FLASH DRIVES	50	800.0000	2800.0000

Once the output requirements are determined, the system designer can decide what to include in the system and how to structure it so that they require output can be produced. For the proposed software, it is necessary that the output reports be compatible in format with the existing reports. The output must be concerned to the overall performance and the system's working, as it should. It consists of developing specifications and procedures for data preparation, those steps necessary to put the inputs and the desired output, that is, maximum user friendly. Proper messages and appropriate directions can control errors committed by users.

Conclusion: A database query processing model in peer-to-peer networks was designed using top-*k* query processing technique and implemented using Java, and MySql. The object-oriented methodology was adopted. Peer-to-peer systems are becoming more useful as they distribute and connect users and data that are distributed across the system. So far, most peer-to-peer attempts have been restricted to exact match lookups and therefore are only suitable for file-based or object-based applications. This paper represents a first step toward the support of a more diverse and richer set of queries, and presents the techniques for routing queries to relevant peers

.

REFERENCES

Arturo, C. and Gracia-Molina, H. (2002), Routing Indices For Peer-to-Peer Systems, In Proceedings of the Internation Conference on Distributed Computing Systems (ICDCS), July Edition.vol 11, pp.122,132.

Boyd, S., Ghosh, A., Prabhakar, A. and Shah, D. (2000), Analysis and Optimization of Randomized Gossip Algorithms, In Proceeding 43rd IEEE Conf. Decision and Control (CDC ',04).

Dongsheng, L, Jiannong C., Xicheng L, and Keith C. C. C. (2009), Efficient Range Query Processing in Peer-to-Peer Systems, IEEE transactions on knowledge and data engineering, vol. 21, no. 1, pp. 78–90 , January Edition.

George, B., Ihab, F. and Mohamed A. (2008), A survey of top-k query processing techniques in relational database systems

Ghada, H., Hamidah I., Md.Nasir S. and Rasali Y., (2011), A Domain-based query routing mechanism for peer-to-peer Networks.

Gupta, A. and Abbadi, E. (2003), Approximate Range Selection Queries in Peer-to-Peer Systems, Proc. First Biennial Conf., Innovative Data Systems Research (CIDR '03), January Edition.

International Journal of Database Theory and Application Vol. 5, No. 1-3, September, 2012, http://www.serc.org/journals/IJDTA.

Ogheneovo, E. E. and Asagba, P. O, (2013), A Parse Tree Model For Analyzing And Detecting SQL Injection Vulnerabilities, West African Journal of Industrial and Academic Research, Vol 6, No. 1, March 31, 2013, Pp 36 - 52.

Ogheneovo, E. E. and Asagba, P. O. (2012), SQLDefend: An Automated Detection and Prevention Technique for SQL Injection Vulnerabilities in Web Applications, SCIENTIA AFRICANA, Int'l Journal of Pure & Applied Sciences, Vol. 11, No. 2, pp. 41-58.

Ratnasamy, S., Shenker, S. and Stoica, I. (2002), Routing algorithms for DHTs: Some open questions. In Proceedigns of the International Peer-to-Peer Workshop,

Stephen C. (2013), Http://.wordpress.com/2013/07/23/peer-to-peer-the-past-and-future-of-social/html.

Yamada,T., Aiharat, K.,Takasut, A.and Adachit, J. (2003), A Distributed Index System for Efficient Query Processing in Peer-to-Peer Networks, IEEE transaction on data engineering, pp.139-142.