SEBestPeer++: A Query Technique for High Speed Secure Data Processing

Suruchi Shrivastava  
Mtech Scholar, CSE,  
LNCTS, Bhopal, India

S. Sathepam  
Associate Professor  
CSE, LNCTS, Bhopal, India

Dr. Sadhna K. Mishra  
Head of Department  
CSE, LNCTS, Bhopal, India

Abstract: Data mining and Query processing technique over the large data is proposed by different authors, where the computational time, thus leads to cost for computation is a parameter to monitor which is always required to minimize and to obtain high end security for the communication between the input and output. Different technique such as Piazza, PeerDB, and Best Peer++ were proposed and claims at best with computation time and security. In this paper presented work is a new algorithm SEBestPeer++ which removes the existing system disadvantage of providing a traditional encryption technique PKI based cryptosystem and the new algorithm utilized a high performance ECC elliptic curve cryptosystem for the data exchange. Our work simulated on JavaAPI and executes the processing time and found as best while compare with the traditional security mechanism and Best Peer++ technique for the large query processing over large dataset.

Keywords- Query technique, Data Processing, ECC, Best peer++, TCP-H dataset.

I. INTRODUCTION

Data mining, query processing [6] [7] techniques and result performance over the data from the different resources is performed over the time. The efficient query technique is required as different e-commerce platform given the different execution unit for the multiple query techniques. In recent system the technique such as Hive [12], HadoopDB [2], Peer Based technique, Thread based mechanism, Single execution technique and Best Peer++ technique is introduced for large data processing and result generation scenario.

Peer to peer Network: Peer-to-Peer networks involve millions of machines connected in a network. It is a decentralized and distributed network architecture where the nodes in the networks (known as peers) serve as well as consume resources. It is one of the oldest distributed computing platforms in existence. Typically, Message Passing Interface (MPI) [10] is the communication scheme used in such a setup to communicate and exchange the data between peers. Each node can store the data instances and the scale out is practically unlimited (can be millions of nodes).

The major bottleneck in such a setup arises in the communication between different nodes. Broadcasting messages in a peer-to-peer network is cheaper but the aggregation of data/results is much more expensive. In addition, the messages are sent over the network in the form of a spanning tree with an arbitrary node as the root where the Broadcasting is initiated. MPI, which is the standard software communication paradigm used in this network, has been in use for several years and is well-established and thoroughly debugged. One of the main features of MPI includes the state preserving process i.e., processes can live as long as the system runs and there is no need to read the same data again and again as in the case of other frameworks such as Map Reduce (explained in section “Apache hadoop”).

II. RELATED WORK

As per the literature survey is performed with different techniques and different result from the algorithms were monitored such as Piazza, HadoopDB [2], Best Peer++ and other different technique for query data processing on large amount of structured RDBMS available dataset our monitoring is performed.

Upon verifying different scenario and the available technique different short comes with the Existing algorithm Best Peer++, cloud environment [4] data processing system which is taken as base for our research work.

The following are the monitored points which identified as problem and further analyzed and performed further with enhancements.

1. Previous technique such as HadoopDB [13] utilized clustering technique where a high configuration is required to prepare and then a highly configured hardware requires to process the data. The technique HadoopDB [13] work on the big data, which is having a dataset in structured manner with large data values and set for consideration under experiment.

2. HadoopDB perform and required High RAM and Complex architecture for the experimental setup, thus a high computation time is required while analyzing the program output, while making it data transferrable and communicable with the large data and queries.

3. Best Peer++ which is a technique based on peer not on Hadoop or big data processing framework perform better than HadoopDB but at the same time it uses some security consideration where it lowers the security constraints by using A
PKI based encryption system, which is not highly secure with the available market scenario, where different active attackers are available to steal the data and network. The existing technique computes high computation time, as the number of data processing an encryption rounds are more as compare to the proposed work given by our research.

III. PROPOSED WORK

As per the observed Best peer++ technique which utilizes PKI [1] encryption technique for the key exchange and secure data sharing technique. Our work propose a new algorithm SEBestPeer++ algorithm which is peer based high secure algorithm utilizes a highly proven symmetric key based encryption algorithm for the communication in between the Peer daemon process. The proposed algorithm utilizes Elliptic curve cryptography (ECC) [14] algorithm for the communication message exchange in between the normal peer and bootstrap peer.

The proposed algorithm is described below:

1. Listing and loading of the entire available normal peer in the network which is participating for the communication.
2. Creating an object of new normal peer.
   Normal Peer nap=new Normal Peer ();
3. Perform communication in between normal peer and bootstrap peer using a secure algorithm ECC.
4. Perform key generation for ECC.
5. Perform encrypted data transmission over the bootstrap peer and normal active peer in the scenario.
7. Observing the execution time and thus it affect computational cost for the complete transmission.
8. Exit.

Algorithm Pseudo Code(SEBestPeer++) :

Input: Query Qi, Dataset tables DS.
Output: Communication process, Metadata, Computation time.
Steps:
   normalpeer1=Inactive, normalpeer2=Inactive.
While (true) do {
   Peer listing {peer1, peer2.....peer};
   NewPeerCreationRequest ();
   NormalPeer1 normal1=new NormalPeer1 ();
   If(peer creation()>0)
   { Peercreationsuccess ();
   Perform Communication Bootstrap to Normal Peer using Q1;
   Apply ECC Encryption (Q1...Qn)
   { Perform authentication;
   Perform key generation using ECC points;
   Send Encrypted data to normal peer;
   } Set status=Active; generate Metadata ();
   } else
   { Peer Status=Inactive;
   Generating metadata for peer request;
   }
   Bootstrap- remove active peer ();
}

IV. EXPERIMENTAL SETUP

All the experiments were performed using an i5-2410M CPU @ 2.30 GHz processor and 4 GB of RAM running windows 8. The discussed feature selection algorithms were implemented using language Java. DatasetUsed: in order to execute the experiment and execution the dataset which is having large unit of table and data TCP-H is taken. TCP-H Windows – In order to perform experimental setup and result analysis part requirement is to access a large dataset, thus a file name DBGEN is downloaded and accessed by us name “dbgen.exe” which is the database generator file under GNU license and generated data for all the structured rdbms in .tbl extension.

DBGen: DBGEN is a database population program for use with the TPC-H benchmark. It is written in ANSI C for portability, and has been successfully ported to over a dozen different systems. The dataset obtained using the presented dbgen.exe in the form of .tbl extension and further in order to convert it in oracle RDBMS , we performed conversation from .tbl to .txt , then .txt to .csv and then further data structure table with specification given is created where 8 tables with different attribute and sizes is created.
Further three effective queries is takes Query1, query 2 and query 3 which is taken as input for processing with TCP-H dataset and further result monitoring, processing is performed by our research work which describes in next section.

V. RESULT ANALYSIS

Proposed as well as existing algorithms were applied one by one in same dataset. At last, comparative study was prepared for all algorithms.

The proposed framework is designed and the components presented to communicate in between different components and query is processed numbering query1, query2 and query 3. The work following is presented:

After loading complete dataset we have performed the data processing using different query and both technique on our dataset and got following table data of best.

Once the dataset is loaded and the query, technique is selected from the user, query is execute over the data and following output monitored using the dataset query execution and the meta data generation for the performed activity and query selection and its data information and other activity information as meta data is monitored performed by the bootstrap peer.

Computation time: it is defined as the interval time consumed in between the initialization of query processing and ending of query processing execution and it can be monitored as:

\[
\text{Longinit} = \text{System.currentTimeInMS}();
\]

\[
\text{Longfini} = \text{System.currentTimeInMS}();
\]

\[
\text{Longexecutiontime} = \text{fini} - \text{init};
\]

Upon based on different algorithm we have calculated four parameters:

1. Computation time.
2. Key Size.

And observed following 2 best algorithm results.

<table>
<thead>
<tr>
<th>Query Name</th>
<th>Best Peer++(in MS)</th>
<th>SEBestPeer++(in MS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1- Query 1</td>
<td>116695</td>
<td>99521</td>
</tr>
<tr>
<td>Q2 – Query2</td>
<td>18254</td>
<td>17023</td>
</tr>
<tr>
<td>Q3- Query3</td>
<td>183015</td>
<td>162032</td>
</tr>
</tbody>
</table>

Figure 5.1: Result Analysis comparison stats
Upon analyzing various results here, we can assure about the algorithm which is applied. Secure technique performs best among others in terms of computation time and key size, which is better than other competitive algorithms for data processing query technique over large data platforms.

A graphical analysis for the proposed technique is presented. This shows SEBestPeer++ provides better performance in query processing. This provides a better way to process data.

![Comparison analysis for propose technique](image)

Figure 5.3: Graphical analysis for the propose method.

### VI. CONCLUSION AND FUTURE WORK

In this paper, we have conducted various experiments of different algorithms and observed results, by considering all features in dataset TCP-H and query processing result execution is performed. We have analyzed the result from BestPeer++ and SEBestPeer++ both algorithms that select relevant features for the proposed frameworks. The experiment results show the usability of our technique, which is more secure and compact in time computed. A further work enhancement can be performed on reducing framework architecture which can utilize the CPU and perform other parameters such as CPU utilization and throughput.

### REFERENCE


