

## Query Processing on OLAP System with Cloud Computing Environment

Nam Hun Park<sup>1</sup> and Kil Hong Joo<sup>2\*</sup>

*Dept. of Computer Science, Anyang University, 102 Samsungli, Buleunmyun,  
Ganghwagun, Incheon, Korea, 417-833  
nmhnpark@anyang.ac.kr*

*Dept. of Computer Education, Gyeongin National University of Education, San 6-8  
Seokudong Manangu Anyangsi, Gyeonggi Korea, 430-040  
khjoo@ginue.ac.kr*

### Abstract

*The conventional researches on a distributed On-Line Analytical Processing (OLAP) system have been in hardship to be adapted to real business environment. However, the recent spread of Cloud PaaS (Platform as a Services) provides new chances in the field of a distributed OLAP. OLAP query execution costs many minutes by its enormous data and OLAP query properties. On the other hand, MOLAP has fast responses. But it has a physical space limit to materialize all cells in possible combinations. Therefore, MOLAP is unsuited to analyze large data. In this paper, to provide not only flexibility and expandability of ROLAP but also the speediness of MOLAP, the cloud server architecture is proposed which shares clients' cube cache by P2P and manages central index on cube data on P2P nodes. In this paper, each node acts as not only a P2P duplicator of aggregated data cubes but also a hybrid server to process queries on sub-cubes. Also, the requests on server are confined to non-aggregated areas and multiple client nodes exchange data simultaneously and asynchronously in the proposed Cloud P2P OLAP. In particular, data are retrieved from physically or logically adjacent nodes. With time series properties, the volume of requested data is minimized and the reuse of past cache data is focused. While central managed P2P has blocked the extendibility, Cloud P2P OLAP guarantees the performance and the extendibility. While Grid OLAP should keep self-distributed system, Cloud use services as much as used by public services. Therefore, Cloud P2P OLAP solves a lot of theoretical limits of conventional distributed OLAP.*

**Keywords:** *Cloud Computing, OLAP, Query process*

### 1. Introduction

The conventional researches on a distributed On-Line Analytical Processing (OLAP) system have been in hardship to be adapted to real business environment. However, the recent spread of Cloud PaaS (Platform as a Services) provides new chances in the field of a distributed OLAP. OLAP query execution costs many minutes by its enormous data and OLAP query properties. On the other hand, MOLAP has fast responses. But it has a physical space limit to materialize all cells in possible combinations. Therefore, MOLAP is unsuited to analyze large data. In this paper, to provide not only flexibility and expandability of ROLAP but also the speediness of MOLAP, the cloud server

---

\* Corresponding Author

architecture is proposed which shares clients' cube cache by P2P and manages central index on cube data on P2P nodes.

Google's Bigtable [1] extended the index concept like B+ tree into multi-level distributed system. This paper focuses on such a distributed central managed index. As a result, the multi-level hybrid P2P is grafted on OLAP environment. Although OLAP researches on P2P or Grid with limitedly managed indexes have already progressed, this paper proposes improved algorithms of range queries and time series management which are passed over by existing works.

- (1) The central P2P is superior to the conventional DHT P2P in terms of the performance and the range query.
- (2) Because the materialized cube has phased relationship, subset inclusion relational index should be considered.
- (3) With P2P natures, logical and physical adjacency on indexes should be considered.
- (4) Considering time-series properties of data warehouse, the index mechanism to reuse past cubes.
- (5) Client nodes act as a query server on included cubes

Besides, the size of a cube is minimized by lifting the cube granularity. When the detailed view is requested on ROLAP, the detailed view inquiry without tuple-level cube transmission is available.

In this paper, each node acts as not only a P2P duplicator of aggregated data cubes but also a hybrid server to process queries on sub-cubes. Also, the requests on server are confined to non-aggregated areas and multiple client nodes exchange data simultaneously and asynchronously in the proposed Cloud P2P OLAP. In particular, data are retrieved from physically or logically adjacent nodes. With time series properties, the volume of requested data is minimized and the reuse of past cache data is focused. While central managed P2P has blocked the extendibility, Cloud P2P OLAP guarantees the performance and the extendibility. While Grid OLAP should keep self-distributed system, Cloud use services as much as used by public services. Therefore, Cloud P2P OLAP solves a lot of theoretical limits of conventional distributed OLAP.

This paper is composed of 5 sections. Section 1 gives outlines and index considerations. Section 2 describes theoretical backgrounds and related works on the similar topics. In Section 3, Cloud P2P OLAP architecture and index algorithms are proposed. In Section 4, the performance of the proposed method is verified by several experiments. Finally, Section 5 concludes this paper.

## 2. Related Works

Most commercial OLAP systems provide MOLAP and ROLAP simultaneously and also provide hybrid OLAP which has MOLAP and ROLAP properties together. But, MOLAP requires too large storage space to materialize all combinations with millions of records, ROLAP is universally used by major companies. Recently, the extension of OLAP to a distributed, grid and P2P environment is a challenging issue. Also, to make ROLAP with quick responses like MOLAP and to solve astronomical space allocation problems of MOLAP, a new concept of storing result cells from ROLAP into clients or distributed nodes of grid systems receives attentions. Recently, Cloud Computing grafted to OLAP [2] is proposed. Because Cloud system includes OLAP data storage, [2] doesn't have advantages in performance and cost efficiency. Focusing on use of distributed file-system without using public cloud, only companies with own private cloud center make the best use of it.

However, except large portals like google, yahoo, naver, etc., with private cloud, the advantages from economies of scale is hard to be utilized. This is not a proper approach for OLAP on companies. Recent researches of Hive and Thusoo [3] propose data warehouse solutions to manage time series data of petabytes with hadoop. Also, infrastructure only for analysis on specified items of mass data has been proposed.

The definition of cloud system can be varied. Generally, platform, infra or service system with large scale of server groups is cloud system. Since personal computer can be included, proposing cloud p2p OLAP system is categorized into hybrid cloud system which public cloud is combined with private cloud. The reason which cloud system gets the limelight is that it uses resources with flexibility and extendibility. However, to adapt cloud systems to database systems, lots of problems have to be solved[4]. The database and mass storage service on cloud under virtual systems have a drawback of using large network costs when input-output cost passes over available zones.

P2P approaches are categorized into structured P2P and unstructured P2P. The unstructured P2P is classified into the distributed P2P based flooding, centralized P2P and hybrid P2P. Structured P2P has been researched recently. Such as Chord, Pastry and CAN, P2P based the distributed hash table is the mainstream. Although the structured P2P is efficient to manage file locations with block units, the execution of range queries or multi-dimensional queries of complicated combinations is limited [5]. Kalnis [6] introduced OLAP with P2P concept for the first time. The cache of OLAP inquiry results are stored at P2P nodes and without central index, queries are broadcasted node to node until the query result is found. Also, genetic and fuzzy is used to prevent message flooding and improve query efficiency[7].

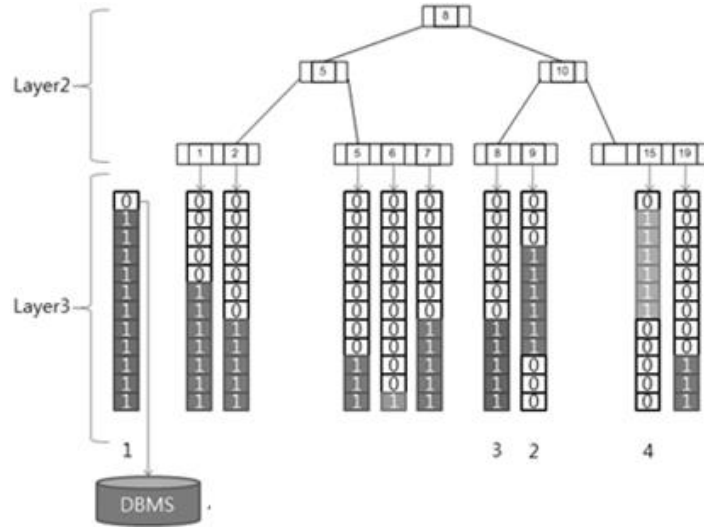
This paper proposed the multi-dimensional range query execution by quad tree grafted with interval tree additionally on time-series index. Furthermore, the advanced index with multi-level hybrid P2P is used to work on Cloud P2P OLAP which includes hierarchical nodes and time-series concepts.

### 3. Cloud OLAP Architecture

This paper proposes three index layers with Cloud P2P OLAP. The central cloud index server is composed by two layers and the client-server which works as a client and also a server is composed by one index layer. By using multiple layers together, difference performance improvement factors are considered together and the depth of layers on the hybrid P2P is expanded when further costs arise. In this paper, the multi-layer (N levels) hybrid P2P is proposed. N means three level layers are able to be expanded by the size of indexes. When loads are on index servers, the first layer and the second layer can be expanded physically like a tree structure. The second layer is much larger than the first layer, one instance on the first layer is connected to M instances just like a tree. After the depth of the index is increased, the horizontal expansion to the maximum number of available star schemes is feasible. The infinite expandability of memory index on cloud system increases the expandability and flexibility of the system and the linear improvement on performance are brought.

In P2P environment, the physical and logical adjacency of receiving data is a major consideration on the performance and the economy of resources. Apart from the adjacency, the time-series range of a cached cube should be considered. When the recent one year data is queried, the access to the cache data in a long distance is more efficient than the drill-down data in an adjacent server. The best strategy would be the set of multiple nodes with satisfying given a time-series range. These nodes can be used asynchronously and simultaneously. However, for the time-series range which is not

included to the existing nodes, queries should be requested directly to RDBMS servers. Most companies maintain physical location information in one-dimension information such as zip-code. B+tree can be considered for sorting. However, the physical adjacency is not expressed because the actual physical location would be two-dimensional information.



**Figure 1. Adjacency index and time-series index**

In this paper, two-dimensional adjacency information is proposed to represent division similarity information together with one-dimensional physical information. The physical adjacency is important considering network transfer efficiency, and the logical adjacency is important to find existing data in a similar department and to reuse data afterward. For an index structure to show two-dimensional information, R-tree, KD-tree, etc. are used. However, to adapt time-series indexes together, a quadtree index is proposed in this paper. After determining the adjacency, nodes within the queried time-series are examined from the most adjacent nodes by expanded recursive calls. If the balance mechanism works, the time-series information at each depth are summarized and used to reconstruct the entire time-series tree. In this case, the quadtree is very advantageous. However, the balance is not guaranteed, so the inquiry performance can be a drawback. In this paper, to overcome the unbalanced quadtree drawbacks, the preprocessing algorithm is additionally proposed to reconstruct the index. When OLAP updates the materialized view at nighttime periodically, Cloud P2P OLAP also reconstructs the index to maintain day-unit balance. When the adjacent index is considered, the entire bit of the time-series should be examined. Figure 1 shows this case. The time complexity in Figure 1 does not guarantee the good performance. To make up for the weak point, the summary of the entire time-series index are centrally managed by bit or sum indexes. The detailed index with each bit is managed like linked-lists. Even in the worst case, the search complexity would be  $\log N$  because fragmented time-series are managed by sorted order.

#### 4. Experiments

The proposed system is implemented on .NET Framework 3.5 with C#. The SOA service communication protocols in P2P and Cloud are developed with WCF and SOAP

protocols. The 4 dimensional data set with 2,930,459 row is used to analyze the performance. A star scheme in Figure 2 is used.

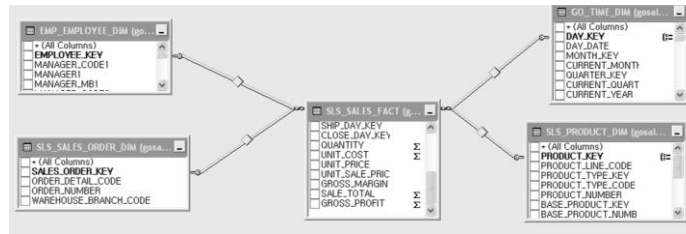


Figure 2. A star scheme model used in experiments

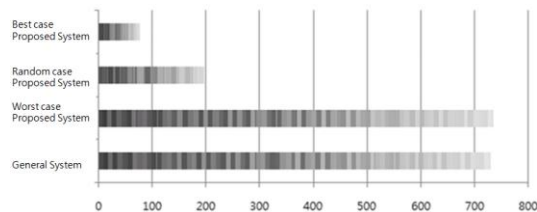


Figure 3. Performance Comparison

Figure 3 shows the performance comparison between the proposed method and the general OLAP system. When queried results are asked directly to a server, the response time is over 8sec. Query scenarios are executed 100 times. In the worst case, the response time is similar to the general OLAP. In the best case, every query has responses within 1sec. The Cloud P2P OLAP is proven to have much faster responses and much less costs.

## 5. Conclusions

This paper proposed Cloud P2P OLAP, which dynamically adds/removes instances to maximize expandability and usability. To make use of ROLAP’s flexibility and MOLAP’s expandability, lots of user-clients are used as distributed nodes, and cube caches are shared through P2P. The architecture including cloud systems and P2P together are newly proposed and proved to have better performance than general OLAP systems by experiments.

## ACKNOWLEDGEMENTS

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science Technology (NRF-2011-0025300).

## References

- [1] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes and R. E. Gruber, “Bigtable: A distributed storage system for structured data”, Journal ACM Transaction on Computer Systems, vol. 26, no. 2, (2008).
- [2] S. Russell, V. Yoon and G. Forgionne, “Cloud-based Decision Support Systems and Availability Context: The Probability of Successful Decision Outcomes”, Information Systems and E-Business, vol. 8, no. 3, (2010), pp. 189-205.

- [3] A. Thusoo, Z. Shao, S. Anthony, D. Borthakur, N. Jain, J. Sen Sarma, R. Murthy and H. Liu, "Data warehousing and analytics infrastructure at facebook", Proceeding of the international conference on Management of data, (2010) June 6-10, Indianapolis, Indiana, USA.
- [4] C. Loboz, S. Smyl and S. Nath, research.microsoft.com, "DataGarage: Warehousing Massive Amounts of Performance Data on Commodity Servers", Proceedings of the VLDB Endowment, vol. 3, no. 2, (2010).
- [5] M. Arnedo, M. del, P. Villamil and R. Villanueva, "Improving Performance of Declarative Query Execution in DHT-Based Systems", Fifth International Conference on Internet and Web Applications and Services, (2010), May 9-15, Barcelona, Spain.
- [6] P. Kalnis, W. S. Ng, B. C. Ooi, D. Papadias and K. Tan, "An adaptive peer-to-peer network for distributed caching of olap results", ACM SIGMOD, (2002), June 4-6, Madison, Wisconsin, USA.
- [7] P. Kalnis, W. Ng, B. Ooi and K. Tan, "Answering similarity queries in peer-to-peer networks", Information Systems, vol. 31, no. 1, (2006), pp. 57-72.

## Authors



**Nam Hun Park**, he received the B.S., M.S. and Ph.D. degree in Computer Science from Yonsei University, Seoul, Korea, in 2000, 2002 and 2007. He was a post-Ph.D. at the Department of Computer Science, Worcester Polytech Institute, Worcester, MA. He is currently a professor of Department of Computer Science at Anyang University, Korea. His current interests include mining data streams.



**Kil Hong Joo**, he received the M.S. and Ph.D. degree in Computer Science from Yonsei University, Seoul, Korea, in 2000 and 2004. He is currently a professor of Department of Computer Education at Gyeongin National University of Education, Korea. His current interests include mining data streams, data analysis and smart learning.