International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 5, May - 2013

Cache Investment Strategies In Distributed Database

Sanju Gupta

The Research Scholar, The IIS University Jaipur

***Abstract*- Caching frequently asked queries is an effective way to improve the performance of both centralized and distributed database system. Modern distributed information systems depend on caching to dynamically place data close to where it is needed. Caching can have a dramatic effect on the performance of distributed queries. Likewise, query execution planes dictate the shipment of data and hence, help determine what can be cached where. Because of this interdependency, performing query optimization without concern for caching can have disastrous consequence for short term and long-term performance. Cache investment extends the query optimizer of a distributed database system to account for such concerns. .In this paper a study of Caching and Cache Investment Strategies is carried out and explains how these can be integrated into a relational query optimizer to influence decision of query optimizer. The study indicates that an integration of Cache investment in query optimization bring out animate results.**.

***Keywords*- Cache Investment, Dynamic data placement, Query Optimization, Physical Caching, Logical Caching**

I. INTRODUCTION

A distributed database (DDB) is a collection of multiple, logically interrelated database distributed over a computer network. This distribution improves performance, reliability, availability and modularity, which are inherent in distributed database system.

Distributed database system is physically distributed and logically centralized database system, is the product generated from the mutual penetration and organic integration of computer network technology and database technology. Physical dispersion refers to that the data composing the distributed database is distributed to the different computers in the network, and each site in the network has the ability to deal with and can implement local applications. Concentrated in logic refers to that each site is a logical whole which is managed by a distributed database management system, and each site implements the global application through the network communication subsystem. [5].

In distributed database, storage devices are not all attached to a common processing unit such as the C.P.U. It may be stored in multiple computers located in the same physical location, or may be dispersed over a network of interconnected computers.[11].

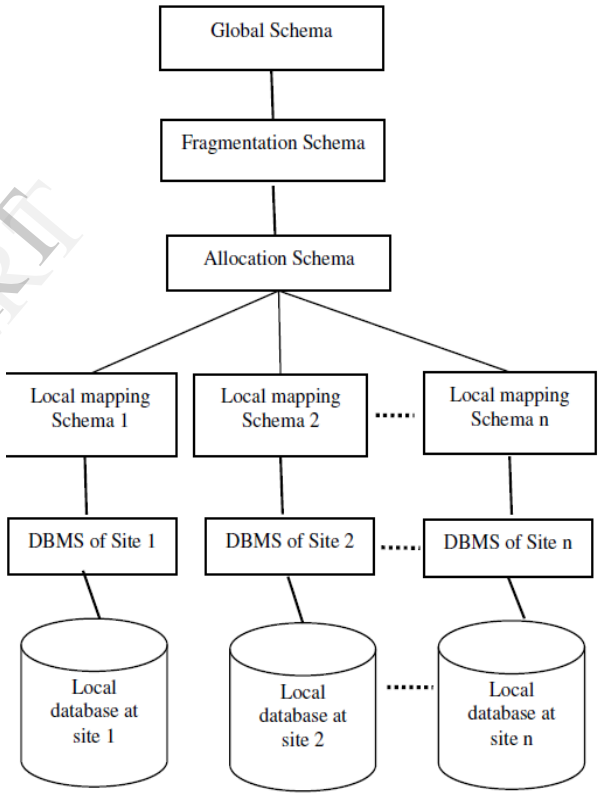

Figure1 Distributed database system architecture [1]

Figure 1 shows the location of data (e.g. in a database) that can be distributed across multiple physical locations. A distributed database can reside on network servers on the Internet, on corporate internals or extranets or on other company networks. The replication and distribution of database improves database performance at end-user worksite [1].

In recent years, with the development of computer network and database technology, distributed database is more and more widely used; with the expanding application, data queries are increasingly complex, the efficiency requests are increasingly high, so the performance enhancement of query processing is a key issue of the distributed database system.[3].There are different ways to improve the performance of distributed database such as caching, join ordering, cost analysis etc Caching has emerged a fundamental technique for ensuring high performance in distributed database system [4].

The rest of the paper is organized as follows. In section 2, we describe the caching and cache investment strategies, their types, and explain how cache investment module integrated into query optimizer. In section 3, we summarize the literature survey and present an analysis of the survey. Section 4 describes the conclusion of the study.

II Caching and Cache Investment Strategies

Caching has emerged as a fundamental technique for ensuring high performance in distributed system. It is an opportunistic form of data replication in which copies of data that are brought to a site by one query are retained at that site for possible use by subsequent queries. Caching is particularly important in large systems with many clients and servers because it reduces communication costs and off-loads shared sever machines. Caching has been successfully integrated into many commercial and research database systems, data warehouses, and database application systems .

There are two techniques for database caching. These are –

1) Physical Caching –Physical Caching is performed in terms of records, pages (blocks) or static partitions of base tables. Physical Caching is used by most distributed systems and database application.
2) Logical Caching-Logical Caching is performed in terms of query results or subsets of query results.

Cache Investment, is a novel technique for combining data placement and query optimization. Rather than requiring the creation of a new optimizer from scratch, Cache Investment is implemented as a module that sits outside the query optimizer. This module influence the optimizer to sometimes make suboptimal operator site selection for individual queries in order to effect a data placement that will be beneficial for subsequent queries. In other words, it causes the optimizer to invest resource during the execution of one query in order to benefit later queries [7].

Figure 2 and figure 3 illustrate how cache investment can be integrated into an existing client –server database systems. In first step, figure-2, we make the optimizer cache-aware. Such a cache-aware optimizer asks the client's cache manager about the current state of the cache and based in this information generates an execution plan including operator placement for a query. Making an optimizer cache aware is quite simple but unfortunately, making an optimizer cache aware is not enough. Initially, all clients' caches are empty. In this case, a cache aware optimizer would likely place all scan operators at server. Thus, client caches would remain empty , and caching would never be exploited. Cache investment must be integrated with the optimizer to avoid this problem. Using cache investment as shown in figure 3 , when the cache aware optimizer requests the current status of cache , the answer it receives is augmented to include what the cache investment policy determine should be cached. In effect, that all the data items that should be cached are already present in the cache. The optimizer is therefore likely to place scan operators at the client, which will result in the desired items being brought to the client and possibly cache there. In this way, cache investment can be integrated into the cache aware optimizer without having to change its search strategy, cost model etc.

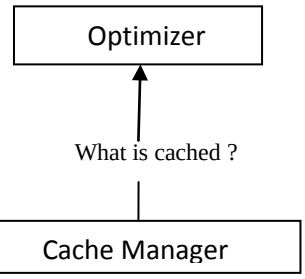

Figure2 Cache-aware optimizer [4]

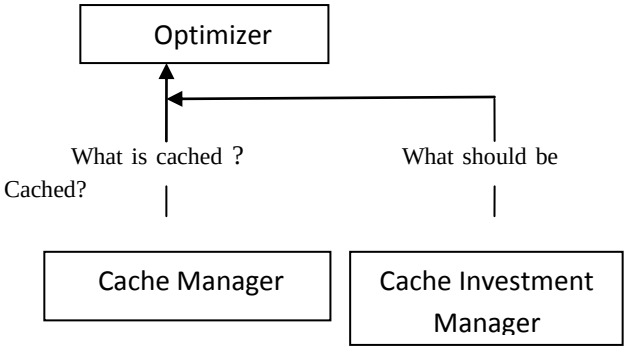

Figure3 Integrating cache Investment [4]

TYPES OF CACHE INVESTMENT POLICIES

Cache Investment Policies determine that when and for which fragments the investment required initiating caching. These policies are invoked for each query that is submitted at a client and can influence the way that operator site selection is done for that query. Types of these policies are depicted in figure 4.

Cache investment is based on the realization that there is a circular dependency between query optimization and caching. That is, the best execution plan for a distributed query will depend on the location of cache copies of relevant data, but the location of those copies is determined by the way that queries are executed .Cache investment breaks this circularity by taking a broader view of query optimization in the presence of caching in order to consider what data should be cache at a client. In contrast to traditional approaches cache investment will some time advise the optimizer to generate a suboptimal plan for a particular query in order to load the cache of a client with relevant data. The extra work for that query is an investment that will pay off in reduced response time for future queries initiated at that client.[8]

III LITERATURE SURVEY

Distributed database is an important field in database research and development. The field has experienced an increase of interest in the recent years, mainly due to new demands on the database to handle larger data volumes and more users.

A distributed database is a collection of independent cooperating centralized system. Query processing in a distributed database requires transfer of data from one computer to another through a communication networks. A distributed information system depends on caching to dynamically place data close to where it is needed. Caching can have a dramatic effect on the performance of distributed database queries.

In this paper [12], the authors show that the cost of evaluation can be reduced by exploiting common sub-expression and maintaining intermediate data structure. In this paper for efficient execution of the query in DDBMS the authors design a module in which they use two methods. 1) Cache investment and 2) Multi query optimization. The cache investment method is used to determine which data items should be cache at the client. This improves a stream of queries submitted at a client. While multi query optimization is used to exploiting common sub expression.

Fan Yuanyuan et. al.[2] and Reza Ghaemi et. al.[3]emphasized that the search complexity is constantly increasing due to new distributed database applications such as huge deductive database system and we need better algorithms to speedup traditional database queries .Advantage of using cache in query optimization is to reduce server load for frequently requested content ,reserve server capacity for other non-cacheable request of client/user and capital expenditure as well as operational savings may also be achieved by optimizing server utilization.

The study [6] of D.Kossmann and J. Franklin identifies the circular dependency between caching and query optimization. This dependency is explained in detail here. This paper presents four policies, which determine when and for which data (Table, Partition of table) caching initiates. In this paper the authors give the procedure for integration

of cache module with an optimizer without changing basic components of optimizer such as query optimizer's search strategies, the query engine or the buffer manager.
Relative performance of these four cache investment policies is studied in this paper . In the performance experiments reference-counting and profitable policies showed significantly better performance than static policies in many situations. It was not possible to indentify a clear winner between the Reference-Counting and profitable policies.

G.Drasch D.Kossmann and J. Franklin et.al. [7] address the new challenges that arise when integrating caching with advance query processing system. This paper starts with identifying the relationship between caching and query optimization. It shows how the query optimizer can be extended with cache investment, so that it produces good query execution plans and implicitly makes long term caching decisions at the same time. In this paper client server architecture with hybrid shipping was used. The authors only concentrate on implementation of physical caching at granularity of data table and index pages. For caching the data only client's main memory is used.

In [9] the author presents the state of art of query processing for distributed database and information system. He discussed various query processing techniques developed for different types of distributed database such as client server, peer to peer, multitier and homogenous and heterogeneous system. For each category, paper presents and discusses the set of query processing techniques; these techniques include special join techniques, techniques to reduce communication cost and techniques to exploit caching replication of data. Both query optimization and caching are extensively described in this paper. This Paper describes the difference between replication and caching.

[8] In this paper authors examine how cache investment can be used to make the right decision about the caching of index. They develop two alternative cache investment policies for indexes and implement in the SHORE database system. Index are key to reducing the amount of data that must be sent from servers to client and thus caching then can greatly improve the performance of a distributed database system.

From the review of the literature we are able to infer that there are different methods to improve the performance of a distributed database. These methods are replication of data, prefetching and cache investment. In this paper we concentrate on caching and cache investment strategies to improve the performance of distributed database. Here we reviewed what is caching and cache investment strategies, there types and how these can be implemented into query optimizer to improve the decision of query optimizer without change the internal structure of query optimizer. Cache investment takes a broader view of query optimization in the presence of caching in order to consider what should be cache at a client. We refer to such data items as candidates. During query optimization it is these candidate items for which the cache investment policy will patch the cache content information provided to the optimizer. The decision of whether or not a data item should be consider a candidate is a tradeoff between the cost of initiating the caching of that item that is investment cost and the expected gain to realize by caching the item i.e. return on investment (ROI) . In general, a cache investment policy should consider an item to be a candidate if it meets both of the following criteria: 1) The ROI is higher than the investment 2) The ROI minus the investment is higher than the ROI of the currently cached data item it would replace if parts of it were brought in.

IV CONCLUSION

A distributed database is a collection of independent cooperating centralized system. Here management of query processing becomes very complex and time taking process. So performance enhancement of distributed database queries is a key issue in distributed database system.. To improve the performance of distributed database; here we review the caching and different cache investment strategies. These strategies help to take decision that when and for which fragment the investment require initiating caching. These policies are invoked for each query that is submitted at a client and can influence the way that operator site selection is done for that query. Query optimization using cache based approach has proved to be a better option in distributed database.

REFERENCES

[1] Mantu kumar,Neera Batra and Hemant Aggarwal , "Cache Based Query Optimization Approach in Distributed Database",IJCSI, 2012.

[2] Fan Yuanyuan and Mi Xifeng,"Distributed Database system Query Optimization Algorithm Research,"IEEE,2010.

[3] Reza Ghaemi,Ain Milani,Fadr, hamid Tabatabee, Mahdi Sadaghizadeh "Evolutionary Query Optimization for Heterogenous Distributed Database System" , World Academy of Science ,Engineering and Technology , vol 45 ,2008.

[4I Ideh Azari, "Efficient Execution of Query in Distributed Database System", 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE),2010.

[5] XUE Lin, "Query Optimization Strategies and Implementation Based on Distributed Database" ,IEEE,2009.

[6] Donald Kossmann , Michael J. Franklin. "Cache Investment Strategies". Univ.of MD Technical CS-TR-3803 and UMIACS-TR -97-50,May 1997.

[7] Donald Kossmann , Michael J. Franklin,Gehard Drasch, "Cache Investment : Integrating Query Optimization and Distributed Data Placement," ACM Transaction on Database System (TODS), Dec. 2000.

[8]. Donald Kossmann , Michael J. Franklin. "Cache Investment for indexes".VLDB Conference,Feb,1998.

[9]] Donald Kossmann ," The State of the Art in Distributed Query Processing",ACM Computing Surveys,Dec.2000.

[10.] C.T. Yu and C.C. Chang, "Distributed Query Processing" ACM Computational Surveys, vol. 16, Dec. 1984 .

[11] Swati Gupta, Kunal Saroba, Bhawna,"Fundamental Research of Distributed Database",International Journal of Computer Science and Management Studies, vol 11,2011.

[12] R.M Monjurul Alom, Frans Henskens and Michael Hannaford, " Query processing and optimization in Distributed Database System," ,International Journal of Computer Science and Network Security(IJCSNS),Sep2009.

[13] Elmasri R. and Navathe S. B.,"Fundamentals of Database Systems", Reading, MA, Addison- Wesley, 2000.

[14] Ozsu M.T. and Valdureiz P: "Principles of Distributed Database System", 2nd Edition, Prentice Hall, 1999.

[15] Shahabi C, Zarkesh A M, Adibi J, "Introduction of distributed database". IEEE ,2001.

[16] Yan T,IacobesnM,Garcia-Mo Lina H,"Introduction of Query optimization of distributed database", WAM Press, I 999.

[17] Doshi P. and Raisinghani V., "Review of Dynamic Optimization Strategies in Distributed Database", Electronics Computer Technology (ICECT), 3rd International Conference, April 2011.

[18]. Konrad Stocker, Donald Kossmann, Reinhard Braumand and Alfons Kemper, "Integrating Semi-Join-Reducers into State-of-the-Art Query Processors", Proceedings of the 17th International Conference on Data Engineering, HYPERLINK IEEE Computer Society Washington, DC, USA 2001.

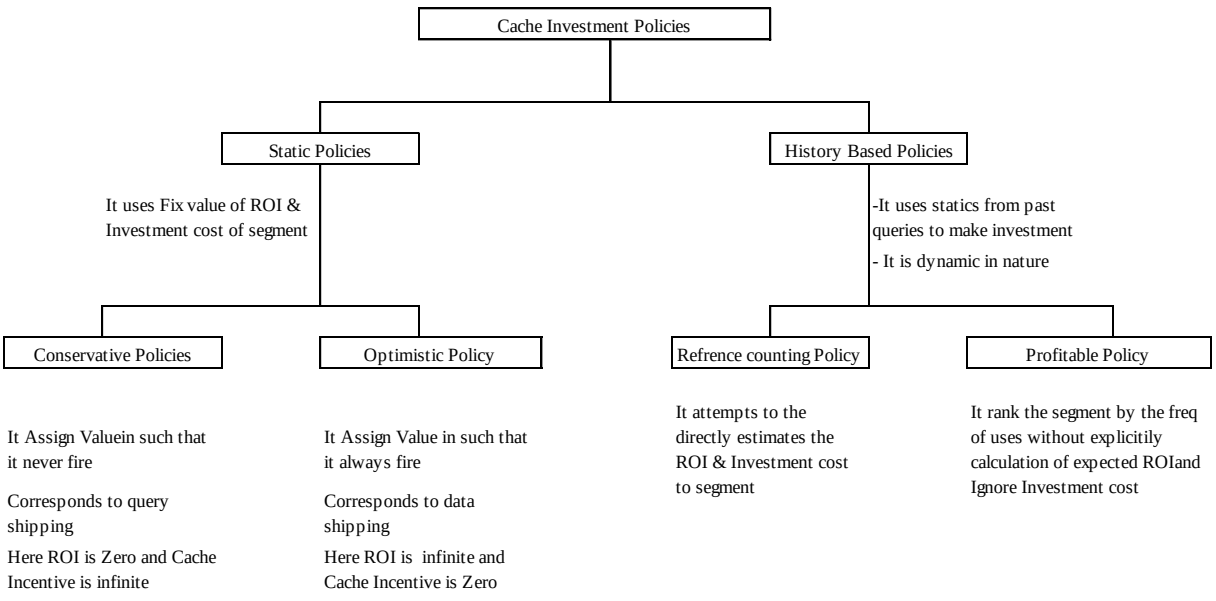

Figure 4: Cache Investment Policies