Analysis of Classical Web-Centric Query Optimization Techniques

Atul Garg
Associate Professor, MMICT&BMM, MMU, Mullana
atulgargmullana@yahoo.com

ABSTRACT

The growth of Internet and web is changing the life style of current generation. In its result, heterogeneous data is available on the web and to retrieve integrated data is a challenge for the researchers and developers. Query optimization techniques are used for information retrieval. In this paper, web centric classical queries processing and optimization techniques, classification of the different present techniques and a comprehensive framework is discussed.

Keywords: Mediator, Disco, Garlic, Hermes, Tsimmis, Infomaster, HiQIQ etc.

1. INTRODUCTION

Nowadays, Internet is playing an important role in the overall growth of the current generation. These days, people are engaging in more and more online activities ranging from reading news, banking, banking to conducting business, advertisement, online shopping, education, research etc.

This has been particularly enabled by the technical advances achieved in computer science. The various fields of computer science like high speed networking, communication protocols, mark-up languages, graphical interfaces, Java technologies, and communication middleware are growing.

Earlier (in evolution period of web) the text-based web pages were provided to the users. But, as the web become more powerful and more in demand the pressure from the user’s side is increased. The pressure was in the form of type of data, Internet speed and Information Retrieval methods. As a result, these days, a variety of data which can be multimedia-rich is available on the web. So, as the different type of data is available on the Internet its result different type of information. To retrieve different type of information the web developers are facing various challenges. The few challenges are discussed below:

1. Heterogeneous Data: The data on the web is not only text but it is a combination of text, images and multimedia. To retrieve this type of data the user sends only one query or request.
2. Security Issues: The people on the Internet are doing various secure transactions like online shopping, Internet banking etc. To retrieve or use the secure transaction the users are using the same Internet and tools.
3. Performance Issues: The good performance is the key of any web site. If it is poor in performance then the users select another site.
4. Remote Servers: With the global growth of Internet the data is stored in near or far severs. So, to retrieve the data from remote sever it increases the response time and security issues.
5. Decontrol: The traditional information retrieval systems were in control due to only one server and that was also in control and in security. But now, in web-centric information retrieval system there are various servers and to search the information the request information being moving from one server to another. So, the web-centric queries are decontrolled after the submitting the query.
6. Larger number of clients: With the revolution of electronic devices, the web users are growing day by day and become a challenge for the web developers. Because, the developer don’t know in advance with type of device is going to be use to retrieve the information.

Query optimization is a function of many relational database management systems in which multiple query plans for satisfying a query are examined and a good query plan is identified [11]. It takes a central stage in data integration systems on the web. The features of web make the task of enabling efficient querying even more difficult. The mediator approach [8] is adopted by most of the described systems for data integration. The related systems are those that match information requests from consumers, individuals or applications, to information providers [23].

Furthermore, the Web’s volatility and highly dynamic nature are a challenge when the expectation is that queries always return specific results. Also, all information sources are not expected to provide the same query capabilities. So, the working of query processor becomes more difficult and it needs to make sure that the generated query execution plan is feasible with respect to these limitations. In this paper, web centric classical queries processing and optimization techniques are discussed. A classification of the different present techniques and a comprehensive framework is also discussed. Mediator approach [5] is mostly used for data integration. The classical systems match the request from the user or users to the host. The classical query optimization techniques are classified based on optimization techniques. The numerous techniques like cost-based optimization techniques, adaptive query optimization, quality-based optimization, and query optimization are discussed.
2. BACKGROUND

In this section, the pre work done by the renowned researchers is discussed. In the conventional query optimization system a query for example in MySQL, there could be many execution plans to retrieve the results. The different methods respond the same results. Though, the different plans are different in their performance, response time and their parameters (resources). Different researchers are doing work on query optimization focusing on the performance, fast accessibility and security of the web-centric queries.

Unel et al. [7] focused on query optimization techniques for multimedia databases. Further, query processing strategies and query optimization techniques are different in multimedia and traditional databases. Researchers proposed query optimization approaches for processing spatio-temporal queries in video database systems and implemented on a specific system called BitVideo [7 and 19]. They applied reordering algorithms named internal node reordering and leaf node reordering on query execution tree. Rao [29] presented sampling techniques based on selecting points from the parameter space at random and optimizing them using a conventional optimizer. The set of plans returned was a subset of the parametric optimal set of plans.

Ouzzani and Bouguettaya [23] discussed the difficulties of query optimization for unstructured information on the web. The authors pointed out that querying abundant information scattered on the web raises several difficulties in achieving efficiency. The authors surveyed their research on fundamental problems to efficiently process queries over web data integration systems were evaluated and in fact four categories of query optimization techniques i.e. cost-based, quality-based, capability-based, and adaptive-based, in the context of web-based data integration were highlighted.

Rossi et al. [9] focused on personalized application. The researchers proposed Object-Oriented Hypermedia Design Method (OOHDM) [9] to attract scientists to design-oriented discussion of personalization for specifying and designing personalized behaviours in web applications. According to Liu and Ling [22] queries for the internal hierarchical structure of web documents is more important. The researchers [22] proposed conceptual model based Web Query Language (WebQL) [22]. WebQL is a rule-based language for querying the HTML documents for web documents. Further, WebQL provides a simple and very powerful way to query both the structure and contents of the HTML documents and to restructure the results. Cardellini et al. [32] considered different categories of web applications, and evaluated how static, dynamic and secure requests affect performance and quality of service of heterogeneous information on the web. Authors proposed a distributed web architecture that is able to guarantee the assessed service level agreement for all client requests.

Srivastava et al. have proposed the overall goal of a general-purpose Web Service Management System (WSMS) [31], enabling clients to query a collection of web services in a transparent and integrated approach. The authors focused on new query optimization issues that arose in a WSMS. Their execution model consists of pipelined query processing over web services and they derived the “bottleneck” cost metric to characterize the cost of a pipelined plan. For this cost metric, the new algorithms not only to decide the optimal arrangements of web services in a pipelined plan respecting precedence constraints but also to decide the optimal chunk size to use when sending data to each web service were devised. These algorithms formed the basis of a WSMS query optimizer.

Lam and Oszu [30] stated that high volatility of web, make querying of data difficult. A system called WebQA that provides a declarative query-based approach to web data retrieval that uses question-answering technology in extracting information from web sites that are retrieved by search engines was introduced. Hu and Bagga [14] focused on the idea that the web documents are increasingly accessed from wireless devices with small screens and low bandwidths. Web documents contain a large number of images and in fact, to identify the function of each image according to the device is an important issue in web content repurposing. The researchers proposed to explore image-recognition techniques to interpret images more effectively in the icon/logo category.

Cali and Calvanese [3] argued that the problem of optimizing conjunctive query answering in relational database. They showed an effective optimization technique for pruning unnecessary accesses from query plans; their approach is applicable in the case of full conjunctive queries that are different from literature. Meinecke et al. [13] introduced Web Composition Architecture Model (WAM) as an overall modeling approach tailored to aspects of highly distributed systems with federation as a related issue. Meinecke [13] and his co-researcher applied the technology used for the functional parts of the architecture and provided a web parts for querying and changing the model.

Gupta [4] et al. presented a system that matches user queries with web services operations. The research employed a self-learning mechanism and a set of algorithms and optimizations to match user queries with corresponding operations in web services. Their system uses lexical analysis, domain-independent matching techniques, domain-specific ontologies and a set of specialized algorithms and optimizations to match simple free-form queries. They proved that the proposed approach is better than traditional approach in terms of effectiveness and performance.
3. ANALYSIS AND COMPARISON

In order to make use of vast amount of data, proficient techniques to access web document information based on its content were needed to be developed. As a result, the role of Information Retrieval (IR) systems became most important. On the whole, the most significant and difficult operations in information retrieval was to generate queries that can succinctly identify relevant documents and reject irrelevant documents [25]. Therefore, in order to address the prominent i.e. moving from data-centric approach to web-centric approach, more known classical web centric query optimization techniques were discussed and are presented. A classification of the different existing techniques and a comparison among them is highlighted in Figure 1.

![Diagram of Query Optimization Classification]

The extended query optimization techniques was already analyzed and compared in [11]. Now in classical query optimization techniques a review of cost-based optimization techniques, adaptive query optimization, quality-based optimization, and query optimization in presence of sources with limited capabilities is being presented. Wiederhold [6] presented an architecture named as mediators. It was based on for some basic issues like maintenance, problem modelling, learning and knowledge attainment. Mediators represent a schema for multiple heterogeneous and independent information sources. Further, in this schema an interface is used for location, model and transparency. A wrapper is used for each source to show its participation in the mediator system. Wiederhold [6] apply this schema over a specific application domain. The working of mediator is explained in Figure 2.

![Diagram of Mediator Based Architecture]

3.1 Cost Based Optimization Techniques

Cost-based optimization techniques help to find or select the best efficient execution plan from the available network paths with the help of available schemas. The basic objective this technique or approach is best throughput with minimal resources. Information on web is in heterogeneous form. Now, to retrieve the information the basic challenge is to optimize the query and the response information. As discussed above the mediator keeps all the information with wrappers, so the estimated or provided information can be fetched from wrapper. Cost based optimization techniques are further divided in different types. Few of them are discussed below:
Disco (Distributed Information Search Component): Tomasic et al. [2], define the user of web as end users (Focus on data), application developers (focus on presentation), Database Administrators (focus on wrapping) and Database Implementers (focus on performance). Naacke[10], Disco is a cost based optimization technique. It combines a generic cost model with specific cost information exported by wrappers The Disco approach deals with the unavailable data. The Disco mediator contains its own internal database see figure 3. Disco returns a biased answer.

In the internal database on Disco it contains interfaces, views, data information, registration details etc. By [23], there are two basic algorithms are used named as cost formula integration and cost estimation. First algorithm is used for root to leaf, formulas are applied on nodes and in these rules are used to connect with the mediator cost model and 2nd algorithm is applied for its back traversal which is based on plans and operator’s cost is computed. In this approach, from the root to the leaves and back method is used to calculate the cost.

Garlic Architectures: Garlic [18, 20 and 21] approach is used for heterogeneous multimedia (text, videos, images etc.) information. An incorporated view of heterogeneous information sources shown in Figure 4 is presented by this approach. The query processor in the garlic architecture generates plans to apply accurate query on the heterogeneous repositories of information. This approach decomposes the queries and repositories in to pieces to distribute the load of repositories. In the Garlic approach wrappers plays an important role and with the help of wrappers this technique follows and improve the cost estimation techniques. In this framework, cost model, cost formulas, and statistics techniques are used for cost estimation.
3.3 Adaptive query optimization

Adaptive dynamic query optimization is most important in heterogeneous web-centric query optimization technique. These techniques remove the unpredictable events during query execution. Adaptive Query Optimization techniques are active at run time to change the execution plan, repeat the optimization query with some specific operators to increase the flexibility. The adaptive query, dynamic optimization finds its applications in various projects and approaches.

Telegraphy: Hellerstein [16] et al., proposed dataflow architecture based on rivers and eddies named as telegraphy query engine. Telegraphy is architectured for fine-grained adaptivity and interactive user control. To improve both the extent and the effects is the main goal of telegraphy. In this approach a metadata is used to store the information at end of each query and this information is used for the next queries, whereas to enhance the set of effects from rivers and eddies is more complicated. The working of eddy [27] is as in the following steps i) the eddy set order of tuples to operators, ii) the operators executes an independent thread, iii)and back it return tuples to the eddy. The eddy sends a tuple to the output only when it has been handled by all the operators.

Interleaving scheduling and optimization: Bouganim [17] proposed a new dynamic scheduling approach to deal with limited memory. The researchers basically focus on the performance problems due to random and wanted data release rates in data integration systems. This approach monitors the arrival rates at the information sources and memory availability. Further, the execution plan is revised as there is any significant change. The above statements proves that planning and execution phases are interleaved. In their query execution plan they proposed two types of edges names as blocking edge and pipelineable edge respectively. In the first edge the data is completely produced before consumed, it means the blocking input have to wait for the whole data before start. Whereas, in the another edge the consumer can start with a single tuple and no need to wait.
Further, Query engine divides the query execution into several components: dynamic query optimizer, dynamic query scheduler and processor, and communication manager [17]. The dynamic query optimizer implements dynamic re-optimization techniques. Each planning phase of the dynamic query optimizer can modify the query execution plan (QEP). Dynamic query scheduler takes the input from QEP and schedule the planning. The dynamic query processor implements the execution component of the system and processes concurrently. The Communication Manager (CM) receives data from the wrappers and makes it available to the dynamic query processor.

### 3.4 Optimizing queries over sources with limited capabilities

The web-centric queries optimization technique deals with different sources and queries. In the web queries the sources follow the specific requirements and limited constraints. For example, users can send only one query at a time. He/she can’t check the whole database with a single query. There are various reasons behind to send a single query at a time. The main reasons are: i) security ii) performance and iii) specific business. To overcome on these problems there are many query optimization techniques named as Tsimmis, Infomaster, Information Manifold, Annotated query plans and source sequence plan with limited capabilities. In analysis, Tsimmis and Infomaster are important so these are discussed in detail.

- **Tsimmis**: Molina et al. [12], proposed a prototype to integrate volatile information sources for mediation. The basic principles of Tsimmis (Figure 5 and 6) are: i) Object Exchange Model (OEM) is a lightweight due to less typing objects, it convey the information between components, ii) Mediator specification Language (MSL) is logic based which targets OEM, iii) the extension of MSL are wrappers or translators which are specific with Wrapper Specification Language (WSL) to permit for the description of source contents and querying capabilities iv) Wrapper- and mediator-generators. Tsimmis approach focuses on feasible query plans with limited capabilities of available sources.

![Figure 5: Tsimmis Prototype](image)

- **Infomaster**: By Duschka & Genesereth [24], Infomaster gives an illusion of centralized volatile information system,. It an information integration system for homogeneous data. For mediation, Infomaster [Figure 7] is source-centric, with different type of relations: i) for users to prepare queries interface-relations are available ii) for data information source-relations are available and iii) to generate a schema for the system and to describe the above relation a world-relation is used. By [23 and 24], the working of query processor in Infomaster follow the five basic steps i) Reduction: in this step the system rewrite each atom of query in terms of base relations ii) Abduction: base relations based expression is produced and with a set of definitions.

![Figure 6: Tsimmis Query Processor](image)

![Figure 7: Infomaster Architecture](image)
The set of all consistent and minimal conjunctions and retrievable atoms is produced iii) Conjunctive minimization: this step eliminates all the redundant conjunct iv) Disjunctive minimization: in this step Infomaster drops all disjuncts that can be shown in the union of the remaining disjuncts v) Grouping: the atoms are grouped within each conjunction for presentation. 

Table 1 lists the comparison of the optimization techniques presented so far. It is evident that none of technique can be called as the best and in fact all approaches are application specific rather than being generic.

4. CONCLUSIONS
In addition to difficulties encountered in optimizing queries various query optimization techniques are explored. To optimize the query and to control query execution is a difficult after submitted the query on the web. Further, with these problems the information source may show different behaviour from its expectations. So, it may reduce the expected results. It shows that, the traditional optimization techniques that are based on static information may not be appropriate for heterogeneous information. Further, expectations from web are increasing day by day and as its result the dynamic nature of web is also changing. Web centric query optimization could not work in the boundaries of classical query optimization techniques.

REFERENCES:
<table>
<thead>
<tr>
<th>Systems</th>
<th>Underlying Information sources</th>
<th>Source Description</th>
<th>Data Model</th>
<th>Query Language</th>
<th>Query Optimization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disco</td>
<td>Structure Data</td>
<td>Relations are represented by Information type</td>
<td>ODMG Data Model</td>
<td>Object or Relational SQL</td>
<td>Optimization through wrappers exporting statistics</td>
</tr>
<tr>
<td>Garlic</td>
<td>Database, Images, Archives etc.</td>
<td>Garlic objects are specified by GDL</td>
<td>Object Oriented</td>
<td>Extension of SQL</td>
<td>Optimization is achieved by means of wrapper co-operations</td>
</tr>
<tr>
<td>Ariadne</td>
<td>Semi-structured Data</td>
<td>Embedded Catalogs</td>
<td>Loom Knowledge Management system</td>
<td>SQL Query Language</td>
<td>Planning by rewriting and cost-based optimization</td>
</tr>
<tr>
<td>Hermes</td>
<td>Structure &amp; semistructure data</td>
<td>Domains</td>
<td>Relational</td>
<td>Domain Class</td>
<td>Optimization based on caching &amp; use of invariants</td>
</tr>
<tr>
<td>ObjectGlobe</td>
<td>Semi-structured &amp; Semi-structured data</td>
<td>Sources are described using RDF</td>
<td>XML based model</td>
<td>Extension of SQL</td>
<td>Distribution of query processing Capabilities &amp; QoS based optimization</td>
</tr>
<tr>
<td>HiQIQ</td>
<td>Structured data</td>
<td>Query Correspondence Assertions (QCAs)</td>
<td>Relational</td>
<td>Extension of SQL</td>
<td>Planning Based</td>
</tr>
<tr>
<td>Telegraph</td>
<td>Mainly relational streaming data sources</td>
<td>Relations</td>
<td>Relational</td>
<td>SQL</td>
<td>Based on the concept of eddy. Addictiveness is Implemented at the level of tuples.</td>
</tr>
<tr>
<td>Interleaving scheduling &amp; optimization</td>
<td>Relational databases</td>
<td>Relations</td>
<td>Relational</td>
<td>SQL</td>
<td>Adapt scheduling according to arrival rate changes</td>
</tr>
<tr>
<td>Tsimmis</td>
<td>Structured &amp; Semi-structured data</td>
<td>Templates</td>
<td>Object Exchange Model (OEM)</td>
<td>MSL &amp; LOREL</td>
<td>Capability based processing and standard optimization</td>
</tr>
<tr>
<td>Infomaster</td>
<td>Structured &amp; Semi-structured data</td>
<td>Relations &amp; built-in Predicates</td>
<td>KIF</td>
<td>Conjuncti-ve queries</td>
<td>Query optimization based on local completeness</td>
</tr>
</tbody>
</table>


