Dynamic Information Retrieval Optimization Using Mobile Agents

Subrata Das, Kurt Shuster, Curt Wu
Charles River Analytics, Inc.
625 Mount Auburn St.
Cambridge, MA 02138
+1 617 491 3474
{sdas, kshuster, cwu}@cra.com

ABSTRACT

Mobile agents have the potential to substantially improve the speed and efficiency with which distributed and heterogeneous data is retrieved. By moving the computation to the data, retrieval times can be reduced by the elimination of unnecessary data transfer. One way to improve a mobile agent system's retrieval efficiency is to incorporate various query optimization techniques (Das et. al., 2002). These methods involve re-writing of the query execution graph so each mobile agent retrieves its requested data in an optimized order, thus minimizing total data transfer size. While these query re-writing methods can be highly effective in reducing both retrieval times and data transfer sizes, they are generally "static", in that the mobile agents retrieve data in a particular order based on an itinerary that is fixed at the time the plan is generated. We have developed a system by which the advantages of mobile agents are leveraged to optimize data retrieval by dynamically optimizing the retrieval strategy as it is carried out. This strategy equips each spawned agent with the full query execution graph and necessary code to execute the retrieval plan at any data site in the network. The spawned agents communicate and collaborate with each other to dynamically decide where to migrate, send data, and perform necessary computations. These decisions depend on retrieval factors such as network speed, data size, and the computational capabilities of the data servers involved in the retrieval. The feasibility of the approach has been demonstrated within a local area network environment using Earth Science data and we present some experimental results in this context.

Categories and Subject Descriptors

H.3.3 [Information Storage and Retrieval]: Information Storage and Retrieval – Retrieval Models.

General Terms:

Algorithms, Performance.

Keywords

Mobile Agents, Information Retrieval, Query Optimization, Retrieval Optimization.

1. INTRODUCTION

Mobile agents have the potential to substantially improve the speed and efficiency with which distributed and heterogeneous data is retrieved. By moving the computation to the data, retrieval times can be reduced by the elimination of unnecessary data transfer. One way to improve a mobile agent system's retrieval efficiency is to incorporate various query optimization techniques as in, where we developed an Agent-based Complex QUerying and Information Retrieval Engine (ACQUIRE) for large, heterogeneous, and distributed data sources. ACQUIRE acts as a softbot or interface agent by presenting users with the appearance of a single, unified, homogenous data source, against which users can pose high-level declarative queries.

ACQUIRE translates each such user query into a set of sub-queries by employing a combination of planning and traditional database query optimization techniques. For each sub-query, ACQUIRE then spawns a corresponding mobile agent, which retrieves data from the appropriate data source. These mobile agents carry with them data-processing code that can be executed at the remote site, thus reducing the size of data returned by the agent. When all mobile agents have returned, ACQUIRE filters and merges the retrieved data and presents the results to the user. Validation experiments on simulated NASA Distributed Active Archive Centers (DAACs) have demonstrated that complex queries can be effectively decomposed and retrieved by this approach, resulting in the twin benefits of improved ease of use and significantly reduced query retrieval times.

The static ACQUIRE query optimization method involves re-writing of the query execution graph so each mobile agent retrieves its requested data in an optimized order, thus minimizing total data transfer size. While this query re-writing method can be highly effective in reducing both retrieval times and data transfer sizes, they are generally "static", in that the mobile agents retrieve data in a particular order based on an itinerary that is fixed at the time the plan is generated. We have enhanced ACQUIRE to a system in which the advantages of mobile agents are leveraged to optimize data retrieval by dynamically optimizing the retrieval strategy as it is carried out. This strategy equips each spawned agent with the full query execution graph and necessary code to execute the retrieval plan at any data site in the network. The spawned agents communicate and collaborate with each other to dynamically decide where to migrate, send data, and perform necessary computations in order to complete the query execution based the graph. These decisions depend on retrieval factors such as network speed, data size, and the computational capabilities of the data servers involved in the retrieval. For example, an agent that finishes retrieval earlier than other agents producing a small size dataset may decide to migrate to another agent's site to facilitate the follow on computation that require the data produced by both the agents. The feasibility of this approach has been demonstrated within a local area network environment using Earth Science data and we present some experimental results in this context.

Copyright is held by the author/owner(s).
AAMAS'03, July 14–18, 2003, Melbourne, Australia.
ACM 1-58113-683-8/03/0007.

2. QUERY OPTIMIZATION

Once a query is posed to the system and mapped against the actual data sites accessible to the system, ACQUIRE's optimization engine attempts to rewrite the query in a manner that minimizes the amount of data that must be transferred in order to retrieve the query. Consider the following satellite photo query.

```
SELECT *
FROM SAT_PHOTO, CITY_DATA
WHERE SAT_PHOTO.name = CITY_DATA.name,
        AND CITY_DATA.pop > 50000
        AND CITY_DATA.pop < 80000
```

In our example, the CITY_DATA repository is located in one site A, and the SAT_PHOTO data is located in three different sites: B, C, and D. Figure 1 shows what the mobile-agent retrieval graph generated by ACQUIRE's Query Planning module looks like.

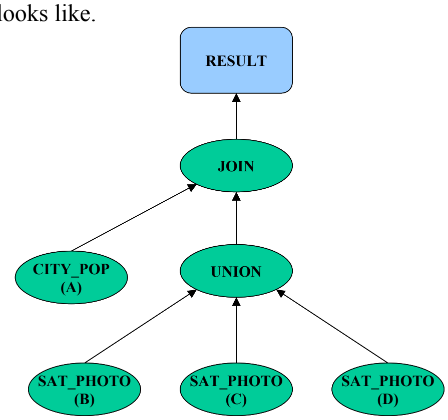

Figure 1: Unoptimized Retrieval Graph

Without any optimization, the retrieval would proceed as follows: Four mobile agents would be sent out, one to each of the four data sites. The agent sent to repository A would retrieve a list of all cities whose populations are between 50,000 and 80,000, and return this list to the home agency. The three other agents would travel to sites B, C, and D, respectively, and download all satellite photographs of all cities in California. With all required data, the system would then be able to answer the query.

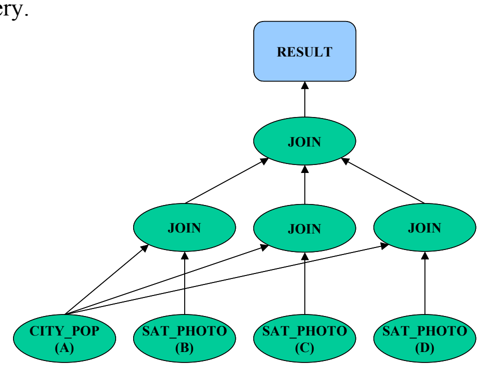

Figure 2: Optimized Retrieval Graph

Obviously, the query efficiency can be improved by first selecting the list of cities whose populations are within the desired range before retrieving their corresponding satellite images. ACQUIRE's query optimization does just this. By employing a series of commonplace query optimization heuristics, we can begin to reduce the size of data that must be

retrieved, and hence reduce the total time required to retrieve the query. In this case, the system takes advantage of the fact that the UNION operation can be accomplished *after* the join operation, as shown in

Figure 2.

Now, the retrieval is optimized such that not all of the SAT_PHOTO images need to be retrieved. In this optimized plan, an agent is first sent out to retrieve all of the required names from the CITY_DATA repository before any of the other agents are sent out (Figure 3). These agents are then loaded with the required population data and then sent to the photograph repositories to retrieve only the photos from towns that meet the population requirement (Figure 4, Figure 5). Finally, the system computes and returns to the user a list of all requested satellite photographs.

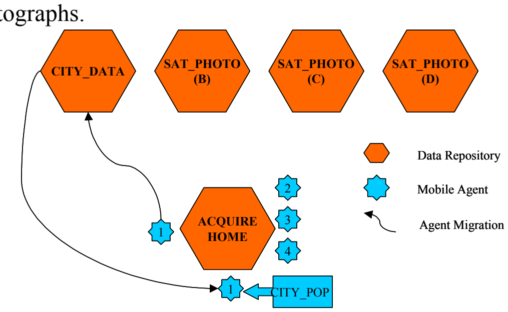

Figure 3: Static Retrieval Plan

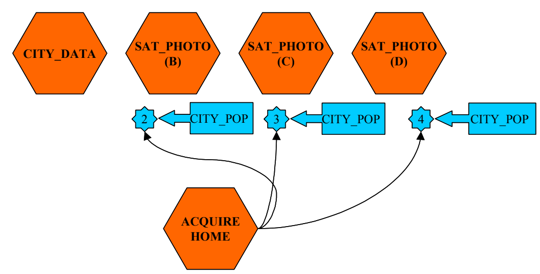

Figure 4: Static Retrieval Plan

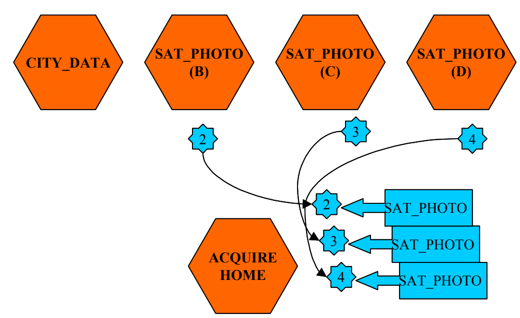

Figure 5: Static Retrieval Plan

3. REFERENCES

Das, S., Shuster, K. and Wu, C. Agent-based Complex Querying and Information Retrieval Engine. Proceedings of the First International Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS 2002), Bologna, Italy.