Distributed Query Processing on the Web

Nalin Gupta * Jayant R. Haritsa * Maya Ramanath *

1. Introduction

Current Web querying systems are based on a "data shipping" mode wherein data is downloaded from remote sites to the user-site, queries are processed locally against these documents, and then further data is downloaded from the network based on these results. A data shipping approach suffers from several disadvantages, including the transfer of large amounts of unnecessary data resulting in network congestion and poor bandwidth utilization, the client-site becoming a processing bottleneck, and extended user response times due to sequential processing.

In this paper, we present an alternative "query shipping" approach wherein queries emanating from the user-site are *forwarded* from one site to another on the Web, the query is processed at each recipient site, and the associated results are returned to the user. Our design does not require co-ordination from any "master site", making it a *truly distributed* scheme. It has been implemented as part of **DIASPORA** (DIstributed Answering System for Processing of Remote Agents), a new Java-based Web database system that is currently operational and is undergoing field trials on our campus network [2].

2. Query Processing System

Our design supports queries that have predicates on both *structure* and *content* – for example, "Find all the RealAudio song files that are within two local hyperlink traversals from the Microsoft homepage". Each user-query $\mathcal{Q}$ is converted into an equivalent alternating sequence of hyperlink traversal patterns and remote site-queries:

$$\mathcal{Q} = \mathsf{S}\ p_1\ \ q_1\ \ p_2\ \ q_2 \cdots p_n\ \ q_n$$

where S is the set of starting sites for processing the query, p_i is the i^{th} traversal pattern, and q_i is the i^{th} sub-query.

The query processing operates as follows: From the user-site, $\mathcal{Q}$ is sent to the sites enumerated in S. Each of these sites then sends it to the sites that can be reached either directly or transitively by a p_1 traversal. These sites evaluate q_1 against the local Web contents and send results, if any, to the user-site. Further, they *rewrite* the query to reflect the processed part and *forward* it, either directly or transitively, to the set of sites reachable by p_2. At the recipient sites, the same operations are performed, and this process continues until all the query-evaluation paths have been fully explored.

A variety of interesting issues arise in implementing the above scheme, including the following:

Recognizing Query Completion: Since queries migrate from site to site without centralized coordination, users cannot determine for sure when their queries have been *fully* executed and all the results have been received. We have therefore designed a protocol wherein each site at which the query is processed sends, to the user-site, simple state information about the query that it received and the set of sites to which it forwarded the query. This data is sufficient to unambiguously detect query completion.

Eliminating Query Recomputation: Due to the highly interconnected structure of the Web, a site may receive the same user-query *multiple* times through different traversal paths. Recognizing and preventing the recomputation of these duplicate queries is important not only to save on local processing effort but also because of the *cascading effect* such recomputations could have because of query forwarding. This issue is addressed by maintaining, at each site, a log of the queries (and their associated states) that have been previously processed at the site.

More details of the issues involved in distributed Web-query processing and our system design and implementation are available in the full version of this paper [1].

References

[1] N. Gupta, J. Haritsa, and M. Ramanath. Distributed Query Processing on the Web. Tech. Report, Indian Institute of Science (http://dsl.serc.iisc.ernet.in/pub/TR/TR-99-01.ps).

[2] M. Ramanath. DIASPORA: A Fully Distributed Web-Query Processing System. Master's Thesis, Indian Institute of Science (http://dsl.serc.iisc.ernet.in/thesis/maya.ps).

*Database Systems Lab, Supercomputer Education and Research Centre, Indian Institute of Science, Bangalore 560012, INDIA. Email: {nalin,haritsa,maya}@dsl.serc.iisc.ernet.in