Scalable Data Collection and Retrieval Infrastructure for Digital Government Applications*

Hanan Samet
Department of Computer Science
Center for Automation Research
Institute for Advanced Computer Studies
University of Maryland at College Park
hjs@cs.umd.edu

Leana Golubchik
CS and EE-S Departments
Integrated Media Systems Center
Information Sciences Institute
University of Southern California
leana@cs.usc.edu

1. INTRODUCTION

In this paper we describe highlights of the project titled "Scalable data collection infrastructure for digital government applications" under the auspices of the Digital Government Research Program of the National Science Foundation. Our research is focused on taking advantage of the distributed nature of data and the interaction with it. Our efforts have been directed at both the systems/theoretical and applications levels. On the systems and theoretical levels, we have continued our development of the BISTRO system (Section 2). On the applications level, work has commenced on the development of a mechanism for spatially tagging text documents for retrieval by search engines based on both content and spatial proximity (Section 3).

2. BISTRO

Hotspots are a major obstacle to achieving scalability in the Internet; they are usually caused by either high demand for some data or high demand for a certain service. At the application layer, hotspot problems have traditionally been dealt with using some combination of increasing capacity, spreading the load over time and/or space, and changing the workload. Previous classes of solutions have been studied in the context of applications using one-to-many, many-to-many, and one-to-one communication. However, to the best of our knowledge there is no existing work, except ours on making applications using many-to-one communication scalable and efficient; existing solutions simply use many independent one-to-one transfers. This corresponds to an important class of applications, whose examples include digital government tasks such as submission of income tax forms to IRS. We proposed Bistro, a framework for building scalable and secure wide-area digital government upload applications.

Briefly, the Bistro upload architecture works as follows. Given a large number of clients that need to upload their data by a given deadline to a given destination server, the Bistro architecture breaks the upload problem into three steps. Step 1, which is the timestamp step, must be accomplished prior to the deadline for clients to submit their data to the destination server. In this step, each client sends to the server a message digest of their data and in return receives a secure timestamp ticket from the destination server as a receipt indicating that the client made the deadline for data submission. The purpose of this step is to ensure that the client makes the deadline without having to transfer their data which is significantly larger than a message digest and might take a long time to transfer during high loads which are bound to occur around the deadline time. It is also intended to ensure that the client (or an intermediate bistro used in Step 2) does not change their data after receiving the timestamp ticket. All other steps can occur before or after the deadline. Step 2 is the transfer of data from clients to intermediate hosts, termed bistros. This results in a low data transfer response time for clients. Since the bistros are not trusted entities (unlike the destination server), the data is encrypted by the client prior to the transfer. Step 3 is the collection of data by the destination server from the bistros. The destination server determines when and how the data is collected in order to avoid hotspots around the destination server. Once the destination server collects all the data, it can decrypt it, recompute message digests, and verify that no changes were made to a client's data (either by the client or by one of the intermediate bistros) after the timestamp ticket was issued. A summary of main advantages of this architecture is: (1) hotspots can be eliminated around the server because the transfer of data is decoupled from making of the deadline, (2) clients can receive good performance since they can be dispersed among many bistros and each one can be direct to the best bistro for that client, and (3) the destination server can minimize the amount of time it takes to collect all the data since now it is in control of when and how to do it (i.e., Bistro employs a server pull).

Our main research activities within the Bistro framework have been along the above described three steps. In addition to focusing on performance and security issues, our recent efforts have also included research directions on fault tol-

*This work was supported in part by the US National Science Foundation under Grant EIA-00-91474, as well as the Policy Development and Research Division of the Department of Housing and Urban Development.

erance issues related to the entire Bistro framework. That is, the security mechanisms in the Bistro upload protocols guarantee integrity and privacy of the data being upload. However, to improve the performance characteristics of our scheme, it is still desirable to provide mechanisms and policies for ensuring that data will not have to be retransmitted due to losses or temporary unavailable which could occur due to failures or malicious behavior of various system components.

To this end, our work focuses on augmenting our current Bistro architecture with appropriate fault tolerance and redundancy mechanisms and policies, where the amount of redundancy and degree of fault tolerance depends on the application and the reliability characteristics of the system components. Our goal in this work is to maintain comparable performance to that of a system without fault tolerance mechanisms and to reduce the overhead attributed to fault tolerance mechanisms (such as storage and network bandwidth overheads) as much as possible.

Lastly, this year, we have also focused on designing incentive schemes for encouraging (non-malicious and reliable) participation in the infrastructure. We are currently pursuing a reputation based approach to this problem. Reputation is a measure of how trustworthy a bistro has been in the past. It is also indicative of how much of its own resources a bistro had contributed to aiding others in the infrastructure. The higher the reputation of a bistro, the higher preference it would receive in the allocation of the infrastructure's resources. The incentive schemes are needed to encourage bistros to volunteer their resources as well as to incentivize nodes that are currently contributing resources to behave in a reliable and non-malicious manner. (Examples of malicious behavior include corruption of data or reluctance to forward data to an appropriate destination).

3. SPATIAL TAGGING

Spatial data can be found in a multitude of forms and variety. We are currently involved in building automated tools that can automatically identify and extract spatial information from web documents. Our first study aimed at converting structured documents, such as, EXCELL spreadsheets and semi-structured data, such as XML and GML documents, into spatial data using an interactive tool. Subsequently, we built tools for identifying postal addresses in documents and tools for geocoding these postal addresses to points on a road map. The real challenge is to automatically extract, and recognize references to geographical locations in text, pdf or word document which do not have any underlying structure.

Our goal is to build a search engine that retrieves documents where the similarity criterion is not based solely on exact match of elements of the query string but instead also based on spatial proximity. For example, the user could search for "Housing Projects" in the vicinity of "College Park, MD". Thus, the search has a content and location specifier associated with it. The results would only return such documents that qualify both the content and location specifier that was provided to the system by the user. Our testbed application domain is a set of documents on a website of the Department of Housing and Urban Development with whom we are collaborating on this project. Below, we report some of the progress made in this direction this year.

We started by investigating into algorithms that automatically identify spatial references in text, pdf, word and other unstructured documents. On identifying spatial references in a document, we associate the document with a set of spatial tags. For example, a document that relates to events in *College Park, Maryland*, is assigned a spatial tag corresponding to the latitude/longitude of College Park.

The document tagger makes use of the GNIS dataset which is a publicly available gazetteer containing the names of places in the world. Given a document, one strategy would be to compare every word in the document with the gazetteer to look for potential matches to records in the GNIS database. However, this process is inefficient. First of all, posing queries to the gazetteer is expensive and should be limited to a few sampled words in the documents. Secondly, a word in the document may match to multiple entries in the gazetteer. For example, a word "York" in the document may correspond to a dozen equally likely entries in the gazetteer. Thirdly, there is no mechanism to avoid *false hits*, *i.e.*, a word "nice" in a document may or may not be a spatial reference to "Nice, France". The tagger that we are building resolves these ambiguities by assigning a relevancy measure to each identified spatial location, the geographical distances between the matches, their offset position in the document, and the size of the document. This model has been shown to perform well in a sample test scenario.

Once the tagger has identified a relevant set of spatial descriptors for a document, we must decide the extent of the tag. In particular if the region has extent such as a county or a road then we must decide whether to tag it with the locations of its starting and ending locations or should we just tag it with its centroid? These issues arise for other types of spatial data as well such as counties, countries, states, etc.

Having developed a document tagger, we need to rank the various locations that are specified in the document. This is important in finding the documents most relevant to a given spatial search string. We are working on the development of a number of different spatial ranking algorithms and will evaluate their effectiveness. We will do this by weighting the spatial references. There are a number of options. One is by frequency. Another is by the extent of the distribution of the references to the spatial search string in the document.

To reinforce the importance of ranking we turn for an example to a search for documents related to Hurricanes. Suppose that we are scanning a news archive. It is not unusual to encounter articles in place A (e.g., Singapore) about a Hurricane in place B (e.g., New Orleans). Clearly, the important spatial location here is New Orleans and not the fact that the article appeared in the Singapore Strait Times newspaper.

Finally, we are working on the development of a method to present results to the user that possibly give an indication of the location of the documents as well as the range of the locations referenced by the relevant documents. Alternatively, we may want to rank a collection of documents by the most relevant spatial locations that they reference. For this we are investigating use of the SAND spatial browser developed by our research group. We also plan to try to show users the distribution of the locations referenced by a collection of documents. Until now the SAND spatial browser has been used primarily to respond to spatial queries involving nearest neighbors and ranges. So, this work represents a significant conceptual change in its structure.