Dissemination of Dynamic Data:

Semantics, Algorithms, and Performance

Krithi Ramamritham

Indian Institute of Technology Bombay
krithi@iitb.ac.in

Extended Abstract

The Internet and the Web are increasingly used to disseminate fast changing data such as sensor data, traffic and weather information, stock prices, sports scores, and even health monitoring information. These data items are highly dynamic, i.e., the data changes continuously and rapidly, streamed in real-time, i.e., new data can be viewed as being appended to the old or historical data, and aperiodic, i.e., the time between the updates and the value of the updates are not known a priori. Increasingly, users are interested in monitoring such data for online decision making. To provide users with dynamic, interactive, and personalized experiences, websites are relying on dynamic content generation applications, which build Web pages on the fly based on the run-time state of the website and the user session on the site. These applications make use of database backends. But, these benefits come at a cost, each request for a dynamic page requires computation as well as communication across multiple components inside the data dissemination and information processing infrastructure.

Consider the following scenario.

A company involved in developing IT enabled services responds to Request For proposals (RFPs). Often RFPs are brought to its attention by customers, sometimes through word of mouth. Won't it be convenient if the posting of a relevant RFP at a (potential) customer's website is automatically brought to the attention of the appropriate business unit or group within company? Our work is motivated by such needs -- the need to constantly track and monitor the dynamics of information sources -- some of which are identified through historical access patterns, others by monitoring potentially useful sites judiciously. Also, often a company responding to RFPs is looking to bolster its case by citing completed projects where the relevant skillsets have been demonstrated. The needed information can be retrieved by maintaining a knowledge repository and setting the following query that continuously sends up-to-date information, as the knowledge base gets updated, to the proposal writer(s).

```
CQ RFP_tracker:

   SELECT project_name, contact_info

   FROM RFP_DB

   WHERE skill_set_required  ⋐ available_skills
```

Such a knowledge repository can be seen as an aggregator of data from specific dynamic sources. As another example, consider a user who wants to track a portfolio of stocks, in different (brokerage) accounts. He or she might be using a third party data aggregator which provides a unified view of financial information of interest by periodically obtaining information from multiple independent sources.

These examples reflect applications which make use of information that experience rapid and unpredictable changes for on-line decision making in time critical or value critical environments. The growth of the Internet as well as Intranets has made the problem of managing and disseminating such dynamic data both interesting and challenging. Resource limitations at a source of dynamic data or within the dissemination infrastructure will limit the number of users that can be served directly by the source. As user load on a site increases, the computation and communication costs can result in significant delays, leading to poor scalability and availability of dynamic data. Solutions needed to mitigate these problems involve techniques from multiple domains:

- *WWW and the internet* -- caching, replication, dynamic page generation techniques, edge servers;
- *Distributed systems* -- replication, load balancing, distribution of data;
- *Networking* -- content distribution networks, application level multicasting, peer-to-peer networks; and
- *Databases* -- active, real-time databases, caching.

There is a lot of excitement about this topic if the papers in conferences related to all the above areas are any indication. As part of our work, we have contributed to this excitement, but many questions remain. In this keynote talk, we will discuss the following issues, focusing on the open problems (see reference for details):

Specification of user QoS needs:

The focus in many applications such as traffic monitoring, network fault management, etc., has been on the dissemination of important events as opposed to data, and on the execution of continuous queries. There are several alternative ways in which these can be expressed; Event-condition-action rules have been used for situation monitoring, profiles have been proposed for retrieving data or changes from the web and other sources, and continuous SQL-like queries have been used for processing dynamic data. In spite of the communication and computation overheads being non-negligible, the system should provide temporally coherent responses to queries over distributed data. So, in addition to specifying the queries, users' QoS should also be formulated to quantify the required coherency in the responses.

Caching-based approaches:
Caching and replicating are widely used approaches to mitigate the performance degradations due to content distribution and delivery. But, unless updates to the data are carefully disseminated from sources to caches (to keep them coherent with the sources), the communication and computation overheads involved can lead to further losses of coherence in the results of queries executed over dynamic data.

Content Distribution Networks (CDNs) for dynamic data:
Resource limitations at a source of dynamic data will limit the number of users that can be served directly by the source. A natural solution to this is to have CDNs for Dynamic Data, formed by a set of repositories which replicate the source data and serve it to geographically closer users. Services like Akamai and IBM's edge server technology are exemplars of such networks of repositories, which aim to provide better services by shifting most of the work to the edge of the network (closer to the end users). Although such systems scale quite well, when the data is changing rapidly, the quality of service at a repository farther from the data source will deteriorate. In general, replication can reduce the load on the sources, but replication of time-varying data introduces new challenges. Unless updates to the data are carefully disseminated from sources to repositories (to keep them coherent with the sources), the communication and computation overheads involved can result in delays as well as scalability, further contributing to loss of data coherence.

Change detection and monitoring:
This is a critical requirement for many dynamic data intensive applications. Timely dissemination of changes to interested information sources is especially critical as periodic pull by humans (current usage) is a waste of resources. Algorithms for detecting changes to the contents of HTML and XML pages have been developed and used in many systems. In general, it is important for the change tracking procedure to be adaptive. Rather than having a periodic fetching of pages, the time of next fetch needs to be determined depending on the observed trend of changes in fetched pages. This would further reduce the amount of resources consumed for tracking and monitoring.

While discussing solutions to the above topics, we will make connections to those from peer-to-peer systems, stream processing, as well as sensor networks.

References

http://www.cse.iitb.ac.in/~krithi/ddd.html