Continuous Queries: Fishing for Research Directions in Data Streams

Michael J. Franklin
Computer Science Division
UC Berkeley
franklin@cs.berkeley.edu

January 2002

Emerging pervasive networked computing environments present challenges that cannot be met by existing data management technology. These challenges stem from their large scale, their highly-distributed nature, their need to monitor and adapt to changes in the physical environment, and their need to actively support users in wading through vast quantities of data in a near-real time manner. A key requirement for data management in these environments is *adaptability*. That is, the data management components of the infrastructure must be able to quickly evolve and adapt to radical changes in data availability and content, systems and network characteristics, and user needs and context.

The *Telegraph* project at UC Berkeley [SMFH01] is developing an Adaptive Dataflow Architecture that is aimed at supporting a wide variety of data-intensive, networked applications. Motivating applications for Telegraph include event-based, "real-time" business processing, profile-based data dissemination, and query processing over data feeds from *ad hoc* sensor networks and other streaming data sources, such as mobile devices.

Traditional database query processing systems are inappropriate in such applications for a number of reasons: First, they are based on *static* approaches to query optimization and planning. Database systems produce query plans using simple cost models and statistics about the data to estimate the cost of running particular plans. In a dynamic dataflow environment, this approach simply does not work because there are typically no reliable statistics about the data and because the arrival rates, order, and behavior of the data streams are too unpredictable [HFCD+00].

Second, the existing approaches cannot adequately cope with failures that arise during the processing of a query. In current database systems, if the failure of a data source goes undetected, the query processor simply blocks, waiting for the data to arrive. If a failure is detected, then a query is simply aborted and restarted. Neither of these situations is appropriate in an environment in which sources and streams behave unpredictably, and queries can be extremely long-running (e.g., "continuous queries").

Third, existing approaches are optimized for a *batch* style of processing in which the goal is to deliver an entire answer (i.e., they are optimized for the delivery of the *last* result). In a pervasive computing environment, where users will be interacting with the system in a fine-grained fashion, such approaches are unacceptable. Processed data (e.g., query results, event notifications, etc.) must be passed on to the user as soon as they are available. Furthermore, because the system is interactive, a user may choose to modify her queries on the basis of previously returned information or other factors. Thus, the system must be able to gracefully adjust to changes in the needs of the users [HACO+99].

The fundamental concept underlying Telegraph is the continuously adaptive processing of (possibly infinite) data streams. In Telegraph, all queries are viewed as continuous queries that have varying lifetimes. That is, when a query is submitted to the system, it executes over a specified amount of previously received data as well as all data that arrives at system during the specified lifetime of the query. In addition, queries can be invoked to return results as they are computed, periodically, or on-demand.

Some of the data stream and continuous query research topics we are addressing in Telegraph and related projects are the following:

Adaptive Data Flow Processing — Telegraph uses a novel approach to query execution based on "eddies", which are dataflow control structures that route data to query operators on an item-by-item basis [AH00]. Telegraph does not rely upon a traditional query plan, but rather, allows the "plan" to develop and adapt during the execution. For queries over continuous streams of data, the system can continually adapt to changes in the data arrival rates, data characteristics, and the availability of processing, storage, and communication resources. An initial prototype of Telegraph has been built, but much remains to be done. The challenges to be addressed include: 1) the development of cluster-based and wide-area implementations of the processing engine, 2) the design of fault-tolerance mechanisms, particularly for long-running queries, 3) support for continuous queries over streaming data from sensors and web-based sources, and 4) the development of appropriate user interfaces for manipulating data flows.

Sensor Query Processing — Much of the data to be processed in pervasive computing applications will be continually streaming in from tiny, low-power sensors. Techniques for querying these sensor data streams will be crucial. These techniques must not only be efficient, but must also be tolerant of the power limitations and error characteristics of the sensors. We have extended the data flow query processing architecture with two techniques for dealing with sensors: 1) the "Fjords" operator architecture, and 2) power-sensitive "sensor proxy" operators [MF02]. The Fjords architecture provides the functionality and interfaces necessary to integrate erratic, streaming dataflows into query plans. It allows streaming data to be pushed through operators that pull from traditional data sources, efficiently merging streams and local data as samples flow past. Fjords also allow processing from multiple queries to share the same data stream, thereby providing huge scalability improvements. Sensor proxies are specialized query operators that serve as mediators between sensors and query plans, using sensors to facilitate query processing while adapting to their power, processor, and communications limitations. In addition, we are also investigating the movement of query processing onto the sensor devices themselves [MCFC01].

Context Aware Data and Event Dissemination — A data management system for pervasive computing applications must also provide special support for the targeted and timely delivery of relevant data and notifications to users based on their interests, roles, and context at a particular time. Such dissemination must be driven by *user profiles*, which contain information about user requirements, priorities, and information needs. In this regard, we are investigating filtering of XML documents using interest specifications expressed in XPath and XQuery . An initial approach was described in [AF00]. More recently, we have developed a much more efficient approach based on Nondeterministic Finite Automata (NFA) [DFFT02].

The "Data Centers" project has been investigating more expressive profile languages [CFZ01]. As described in [CFZ01] user profiles contain three types of information: 1) *Domain specification:* a declarative specification of the kinds of data that are of interest to the user. 2)*Utility specification:* expresses the user's preferences in terms of priorities among data items, desired resolutions of multi-resolution items, consistency requirements, and other properties, in order to deal with limited bandwidth and connectivity. 3) *Context specifictation:* user context can be dynamically incorporated into the data magagement process by parameterizing the user profile with user context information. While not currently part of the Telegraph project, such advanced profile functionality will be needed in many pervasive computing applications.

To summarize, emerging networked applications ranging from event-based business processing to more "pervasive computing" environments such as sensor networks and mobile computing raise the need for new query processing and data management solutions. These solutions will necessarily be highly-adaptive, allow for closer and more seamless interaction with users, and will be based on a continuous model of queries and streaming data sources. These issues raise a number of important challenges, and present a tremendous opportunity for database researchers. In this short paper, I have outlined some of those challenges, focusing on those we are addressing in the context of the Telegraph project at UC Berkeley.

References

[AF00] Mehmet Altinel, Michael J. Franklin. Efficient Filtering of XML Documents for Selective Dissemination of Information, *Proceedings of the International Conference on Very Large Data Bases*, Cairo, September 2000.

[AH00] Ron Avnur, Joseph M. Hellerstein. Eddies: Continuously Adaptive Query Processing, *Procedings of the ACM SIGMOD Conference,* Philadelphia, PA, June 2000.

[CFZ01] Mitch Cherniack, Michael J. Franklin, Stan Zdonik. Expressing User Profiles for Data Recharging, *, IEEE Personal Communications,* August 2001.

[DFFT02] Yanlei Diao, Peter Fischer, Michael Franklin, Raymond To. YFilter: Efficient and Scalable Filtering of XML Documents (Demo description), *ICDE Conference*, February, 2002, *to appear.*

[HACO+99] Joseph M. Hellerstein Ron Avnur, Andy Chou, Chris Olston, Vijayshankar Raman, Tali Roth, Christian Hidber, Peter J.Haas. Interactive Data Analysis with CONTROL, *IEEE Computer 32(8),* August, 1999, pp. 51-59.

[HFCD+00] Joseph M. Hellerstein, Michael J. Franklin, Sirish Chandrasekaran, Amol Deshpande, Kris Hildrum, Sam Madden, Vijayshankar Raman, Mehul Shah. Adaptive Query Processing: Technology in Evolution, *IEEE Data Engineering Bulletin,* June 2000, pp 7-18.

[MCFC01] Samuel Madden, Robert Szewczyk, Michael Franklin, David Culler. Supporting Aggregate Queries Over Ad-Hoc Wireless Sensor Networks, *submitted for publication,* December, 2001.

[MF02] Samuel Madden, Michael Franklin. Fjording the Stream: An Architecture for Queries Over Streaming Sensor Data, *ICDE Conference*, February, 2002, *to appear.*

[SMFH01] Mehul Shah, Samuel Madden, Michael Franklin, Joseph M. Hellerstein. Java Support for Data-Intensive Systems: Experiences Building the Telegraph Dataflow System, *ACM SIGMOD Record,* 30(4), December, 2001, pp 103-114.