Award Number: DE-FG02-04ER25600, DOE Early Career
Report Type: Final Report
Report Period: 07/01/2004 to 06/30/2007
PI Name: Beth Plale
PI E-mail: plale@cs.indiana.edu

DOE Early Career: Time-based Data Streams: Fundamental Concepts for Streaming Data Resource

Participant Individual: Beth Plale, Ph.D., PI, Department of Computer Science, School of Informatics, Indiana University

1. Overview

The time-based data streams research investigated concepts for understanding and acting on real time streaming data. Real time data, which we call data streams, are readings from instruments, environmental, bodily or building sensors that are generated at regular intervals and often, due to their volume, need to be processed in real time. Often a single pass is all that can be made on the data, and a decision to discard or keep the instance is made on the spot. Too, the stream is for all practical purposes indefinite, so decisions must be made on incomplete knowledge. The data generated by the Large Hadron Collider is of this nature, the large majority of the information contained in the 1 billion atomic collisions per second are discarded immediately. This notion of data streams has a different set of issues from a file, for instance, that is byte streamed to a reader. The file is finite, so the byte stream is becomes a processing convenience more than a fundamentally different kind of data.

Through the duration of the project we examined three aspects of streaming data: the first, techniques to handle streaming data in a distributed system organized as a collection of web services, the second, the notion of the dashboard and real time controllable analysis constructs in the context of the Fermi Tevatron Beam Position Monitor, and third and finally, we examined provenance collection of stream processing such as might occur as raw observational data flows from the source and undergoes correction, cleaning, and quality control. The impact of this work is evidenced by the publications and presentations. We were one of the first to advocate that streams had little value unless aggregated, and that notion is now gaining general acceptance. We were one of the first groups to grapple with the notion of provenance of stream data also.

2. Contributions

2.1 Data Streams in Web Service (Grid) Architecture

Bringing indefinite streams into a distributed Grid-based computational framework first required framing the problem as one of understanding the critical operations. We developed a service framework for stream processing based on the Global Grid Forum Data Access and Integration (DAIS) specifications for grid service access to a database. Since a set of widely distributed data streams cannot be collected into a single database without incurring additional latencies in moving already large volumes of data, we built a custom grid service and query engine on top of DAIS reference implementation OGSA-DAI (University of Edinburgh) to set up, manage, and query a stream processing network. The interface is given in Figure 1. Through the SQL logical interface the user adds and removes long running queries from the system. The Rowset logical interface provides rowset to the data stream returned as a result of a long running query. The

Stream channel interface creates and removes channels. The administrative interface is used to manage the stream processing resource. Additional details are given in [AxGrids04].

Subservice	Operation
SQL logical interface	SQLAccess:sqlQuery
	SQLAccess:dropQuery
	SQLFactory:createService
	SQLMgt:listQueryStatus
Rowset logical interface	RowsetAccess:getTupleTi
	RowsetFactory:createService
Stream Channel Interface	PublishAccess:createChannel
	PublishAccess:removeChannel
Admin logical interface	AdminAccess:createVSS
	AdminAccess:dropVSS
	AdminAccess:addComputationalElement
	AdminAccess:dropComputationalElement
	AdminAccess:connectStreamSystem
	AdminAccess:describeVSS

Figure 1. Grid service interface to stream processing system. Each interface defines a separate functionality, for adding a query to the stream network, retrieving the results, adding and removing data streams from the stream network, and managing the logical collection of streams.

Deploying queries to a network of computation nodes can be costly, so we developed and analyzed a query reuse strategy and threshold to determine when reuse can reduce deployment costs. ***We argue in a couple of our papers that the right model for bringing data streams to the grid is in the aggregate, that a single readout stream from a small sensor has maximum value when available with spatial and temporal neighbors.*** To gain a better understanding of the performance properties of the system, we experimentally evaluated several aspects of system behavior: the overhead of deploying a new query to the computation network, the performance under increasing query workload, and performance under increasing number of users in the system. Calder extends the dQUOB system[TPDS03].

2.2 Fermi Beam Position Monitor (BPM)

Tevatron is a proton anti-proton accelerator collider operating at the Fermi National Accelerator Laboratory. The machine delivers beam for the CDF and D0 experiments, which expect increasing luminosity until the conclusion of Run II, planned for 2009. The Fermi lab Beam Position Monitor (**BPM**) is an instrumentation device that takes measurements inside the Fermi Tevatron accelerator. BPM measures beam intensity, orbit closure, orbit smoothing and Beta function. We constructed a BPM stream simulator that simulates a configurable number of BPM instruments and configured it to model 324 simultaneous BPM monitors.

The simulator generates data with meaningful physics. Specifically, BPM system works under different data acquisition modes: Idle, Closed Orbit, Turn-by-turn, First turn, Asynchronous Injection, Calibration, Diagnostic. The data types used in each mode are different. We model Closed Orbit, Turn-by-turn, and First Turn modes. In applying the Calder system to BPM, we viewed the BPM monitor data collected by each BPM monitor as a data stream. Users access the data streams through the web service enabled Calder front-end. As shown in Figure 1, the BPM data simulator communicates with Calder stream processing system.

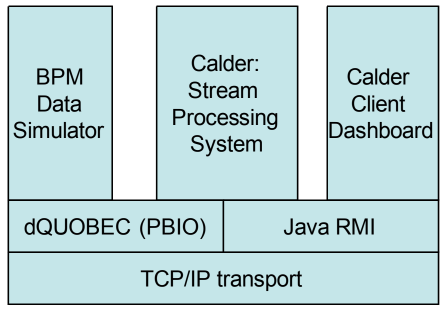

Figure 2. Communication stack of Customizable BPM Dashboard

The ***Dashboard***, see Figure 2, is a real-time graphical interface to the streaming system. It is unique in that in addition to providing a visual depiction of default behavior of the BPM system in real time, it also supports visual depiction of the results of a user's own queries.

Two kinds of queries are supported: horizontal position monitoring query, and amplitude function query. The *horizontal position monitoring* query monitors horizontal BPM data from a specified BPM instrument. The query executes on every turn (i.e., turn-by-turn) of the instrument. At each turn (each particle bunch), the vertical and horizontal positions are read. With the *horizontal/vertical position monitoring query*, a user can monitor the particle's horizontal/vertical position from the stream in which the user is interested. The *Amplitude function query* describes the maximum extent of transverse oscillations and slopes of trajectories at a point in an accelerator. With this query, the user can check the amplitude function in the Interaction Regions. In real physical experiment, Turn-by-turn BPM data totally have 8192 turns. Hence, our graph refreshes for each 8192 turns. It is shown in Figure 3.

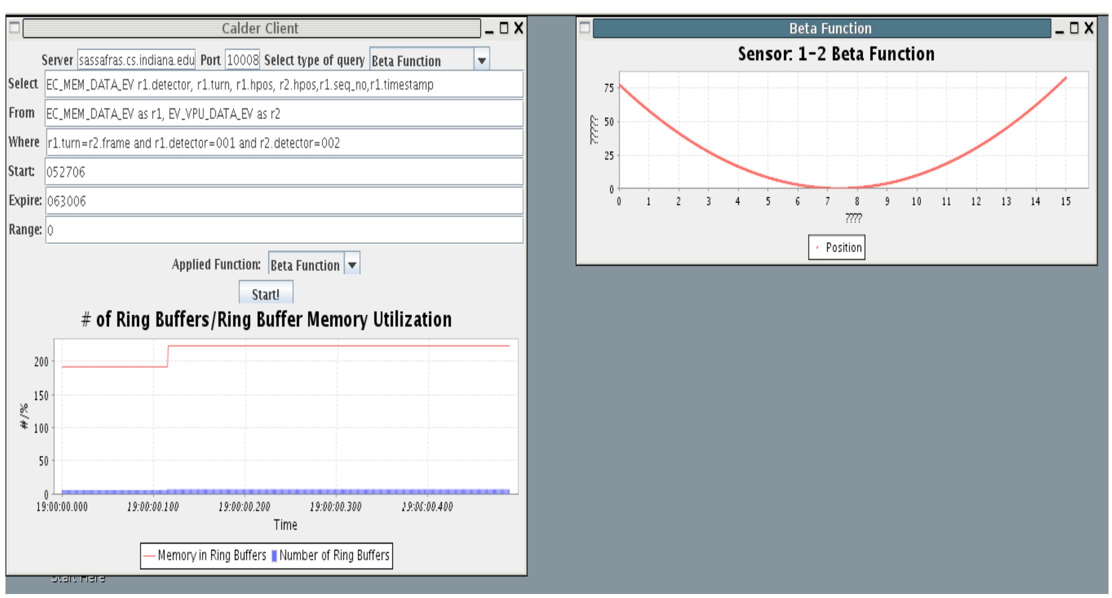

Figure 3. Snapshot of Amplitude Function Study

2.3 Provenance of streaming data

Provenance of a data object is the trace or lineage of the product. It carries information about the processes and data objects that have influenced the creation of the data object in question. Provenance is used in asserting the quality of a data object, so is instrumental in preservation and sharing of data. A data stream is an indefinite sequence of time ordered events. Our research in provenance of data streams began by asking the fundamental question of what is the equivalent of a data object in a data stream? That is, identify the smallest unit for which provenance is collected in a stream filtering system. A dataset, the data object that has first class citizenship in Grid computing, corresponds to an event in a stream. An implication to granularity is that tracking provenance of stream datasets has to be done without burdening the system.

Further, we studied a number of issues related to access and understanding of provenance. The source of a stream must be traceable long after the filtering process has completed. The environment in which a particular set of events needs to be identifiable. Since stream filtering systems adapt themselves to changes in underlying resources, this involves changes in query execution plans and approximations when streams are not available. Finally, the provenance model needs to enable tracing the accuracy of a subset of the stream to a specific time period. Deducing an accuracy value for a derived event based on the accuracy of the input streams and stream filtering environment is a challenge in itself.

We defined a model for provenance tracking consisting of three atomic units of provenance collection in streams: base streams, adaptive filters and derived streams. Base streams are streams that are generated outside the stream filtering system. The generation source may be an instrument, experiment, or any process. Adaptive filters are declarative queries or application code that are associated with a life time and continuously execute on the data streams; Derived streams are streams that are produced by executing adaptive filters on base streams or other derived streams. We propose a timestamp based append only stack approach for collecting provenance of streams and filters, and a bottom-up provenance tree to associate the base streams and derived streams. By append only stack we mean a data structure in which information can only be added not removed; and also that the latest information identified by the timestamp represents the current status. The model is implemented and evaluated in [xx].

3 Training

This project involved intense engagement and effort by two graduate students Ying Liu who graduated with her PhD, and Nithya Vijayakumar who also graduated with her PhD. Both took positions at Cisco, having been recruited when attending a Grace Hopper Celebration of Women in Computing conference sponsored by the Women in Computing group at Indiana University that PI Beth Plale co-founded. The project also involved an undergraduate student A.J. Ragusa, who went on to switch his major from music to computer science. At the time of graduation he was seriously considering graduate school.

4 Publications and Talks

A number of publications have resulted from this work and this collaboration.

[1] Ying Liu, Nithya N. Vijayakumar, and Beth Plale, Stream Processing in Data-driven Computational Science, *7th IEEE/ACM International Conference on Grid Computing*, April 2006.

[2] Ying Liu and Beth Plale, Multi-model Based Optimization for Stream Query Processing, *KSI*

Eighteenth International Conference on Software Engineering and Knowledge Engineering (SEKE 06), San Francisco, July 2006.

[3] Ying Liu and Beth Plale, Query Optimization for Distributed Data Streams, *ISCA 15th International Conference on Software Engineering and Data Engineering (SEDE 06)*, Los Angeles, July 2006.

[4] Beth Plale, Framework for Bringing Data Streams to the Grid, *Scientific Programming*, IOS Press, Amsterdam, Vol. 12, No. 4, 2004.

[5] Beth Plale, Using Global Snapshots to Access Data Streams on the Grid *2nd EUROPEAN ACROSS GRIDS CONFERENCE (AxGrids 2004)*, *Lecture Notes in Computer Science Series* Springer Verlag, Vol. 3165, 2004.

[6] Beth Plale, Dennis Gannon, Daniel A. Reed, Sara J. Graves, Kelvin Droegemeier, Bob Wilhelmson, Mohan Ramamurthy, "Towards Dynamically Adaptive Weather Analysis and Forecasting in LEAD". *International Conference on Computational Science (ICCS)*, 2005.

[7] Beth Plale and Nithya Vijayakumar, Evaluation of Rate-based Adaptivity in Joining Asynchronous Data Streams, *ACM/IEEE 19th International Parallel and Distributed Processing Symposium (IPDPS),* Denver Colorado, April 2005.

[8] Nithya Vijayakumar, Ying Liu, and Beth Plale, Poster: Calder: Enabling Grid Access to Data Streams IEEE High Performance Distributed Computing (HPDC), Raleigh North Carolina, July 2005.

[9] Nithya Vijayakumar, Ying Liu, and Beth Plale, Short Paper: Calder Query Grid Service: Insights and Experimental Evaluation, *IEEE Cluster Computing and Grid (CCGrid)*, May 2006.

[10] Nithya Vijayakumar and Beth Plale, Towards Low Overhead Provenance Tracking in Near Real-Time Stream Filtering*, International Provenance and Annotation Workshop (IPAW'06)*, May 2006.

[11] Nithya Vijayakumar, Beth Plale, Rahul Ramachandran, and Xiang Li, Dynamic Filtering and Mining Triggers in Mesoscale Meteorology Forecasting,*IEEE/IGARS 2006 International Geoscience and Remote Sensing Symposium,* Denver, CO, July 2006

Presentations related to this project include the following:

• Beth Plale, "Transforming the Sensing and Prediction of Intense Local Weather Through Dynamic Adaptation", NSF Dynamic Data Driven Application Systems (DDDAS) Workshop, January 19-20, 2006.

• Beth Plale, "Wringing Kilobytes of Knowledge from Petabytes of Data: Something Has to Change", School of Informatics, Indiana University, September 2005.

• Beth Plale, "Calder Continuous Query Grid Service", Argonne National Labs, August, 2005.

Software and documentation for the software described here can be found at http://www.cs.indiana.edu/dde/projects/Calder.html