

Dynamic Filtering and Mining Triggers in Mesoscale Meteorology Forecasting†

Nithya N. Vijayakumar and Beth Plale
Department of Computer Science,
Indiana University,
Bloomington, IN, USA
{nvijayak, plale}@cs.indiana.edu

Rahul Ramachandran and Xiang Li
ITSC
University of Alabama in Huntsville,
Huntsville, Alabama, USA
{rramachandran, xli}@itsc.uah.edu

Abstract— Mesoscale meteorology forecasting as a data driven application is capable of reacting to events in real-time. We explore a framework for dynamic filtering and mining of data products to generate timely triggers for invoking forecasting applications. In this paper, we present our framework, which couples the Calder stream processing system developed at Indiana University for filter processing and trigger generation, and data mining algorithms developed as part of the ADaM data mining tool kit developed at ITSC, UAH, which detect events for trigger generation.

Keywords- *Dynamic Triggers, Data Mining, Stream Processing, Meteorology Forecasting*

I. INTRODUCTION

Mesoscale meteorologists investigate regional weather phenomenon by running computational weather forecast models using regional observational and model generated weather data products such as observational readings taken by the NEXRAD WSR-88D Doppler radars and the ADAS model output. Meteorologists today are limited in their efforts to understand severe storm phenomena by a rather rigid forecast framework. That is, forecast models are started at fixed time intervals and involve human interaction [13]. A time schedule, however, has little bearing to what is happening in the weather since severe weather may occur at any time. LEAD [6], in which the authors are involved, is developing cyberinfrastructure to enable weather forecasting to be responsive to regional scale weather phenomena such as tornadoes, severe storms and flash floods.

We address this problem by treating a forecast application as a data-driven application, and as such, one that is capable of responding to events in the physical environment in real time. Scientists set up the desired dynamic behavior by specifying rules that define a relationship between an event and the specific forecast model to be invoked. For instance, a data-mining agent could monitor a Doppler radar data feed to identify and count the number of vortices detected within a short period of time. When the count exceeds a threshold, the data-mining agent issues a message to a specific forecast model.

The scientist sets up the behavior by specifying dynamic triggers. Dynamic triggers, a term adapted from active databases, is an event-action rule that enables automatic response to events that are taking place within or external to a database system [10]. A *rule*, specified by a user and run for a specified period of time, is a combination of filtering tasks and data mining actions triggered by the occurrence of certain events. *Tasks* can range from simple logical expressions (filters), to simple or mathematically complex data transformations, aggregation operators, analysis operators, or cleansing operators. The *action* portion of the rule is an invocation that kicks off a latent forecast model.

Stream processing engines are designed specifically for processing data flows on the fly. In many systems described in literature and available commercially, engines execute queries continuously over arriving streams of data [3, 5, 16]. Clients describe their filtering and processing needs through a declarative query language or through a graphical user interface (GUI) [1, 4] that is converted. Events are processed on the fly, without necessarily storing them. Queries can be deployed dynamically [16], and can have their operators reordered on the fly [3].

In this paper we describe a framework for dynamic sensing of the physical environment that couples general filtering and triggering operations with domain specific data mining algorithms. The framework uses Calder [16], a stream processing system that gives users SQL query access to collections of live data streams. Because much data in meteorology is in a binary format, the framework handles metadata generation on the incoming data streams. The details of metadata generation are discussed in Section III. The Mesoscale Detection Algorithm (MDA [8]) and Algorithm Development and Mining (ADaM [14]) algorithms mine observational data, assimilated data sets, and model output for events to generate triggers.

The remainder of the paper is organized as follows: Section II provides an overview of the Calder stream processing system and discusses its usage scenario in LEAD. Section III delves into the details of dynamic filtering and mining triggers. In Section IV, we summarize our conclusions and discuss future work.

†This work funded by National Science Foundation ATM-0331480, ATM-0331579 and CNS-0202048; and Department of Energy DE-FG02-04ER25600.

II. ARCHITECTURE

Dynamic stream processing can be viewed as deploying triggers that watch the observational weather data, and take action when an interesting phenomenon is detected. The framework for this capability is provided by Calder [16], a stream processing system that provides access to stream data for scientific applications. It follows the service-oriented architecture (SOA) principles, so can be a middleware tool in a larger SOA suite. Calder views a collection of streams as an unbounded data repository, a virtual stream store [11] and provides SQL query access to the resource. We believe that the value of triggers streams increases dramatically when streams are aggregated and global behavior can be interrogated.

A view of the architecture of Calder is given in Fig. 1. Calder provides a grid service interface, meaning that users submit continuous queries using a grid data service (GDS). A grid data service provides a well-defined service interface to interact with a data resource. We used the OGSA-DAI v6 OGSi grid data service and extended it to support a data stream resource [9]. Calder has an built-in query planner service that chooses an execution plan for the query and distributes it to the query execution engines on different computational nodes.

The data streams from data sources enter Calder through a pub-sub system. The resulting stream is stored in an in-memory ring buffer at the rowset service. The user or application can then retrieve the results from the rowset service asynchronously by issuing data access requests or synchronously by subscribing as a receiver. The architectural details of the Calder system are described in [16]. The following scenario depicted in Fig. 2, shows the usage of Calder in a dynamic data-driven meteorology application.

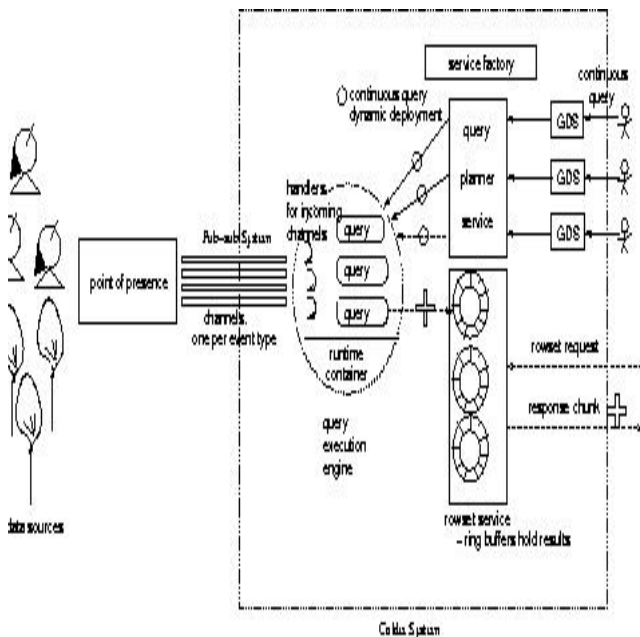


Figure 1. Calder Architecture

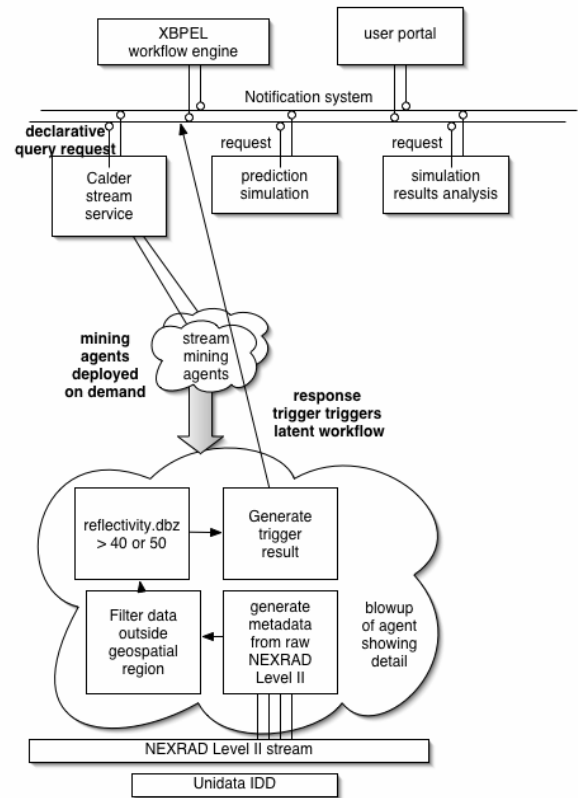


Figure 2. Dynamic Filtering and Mining Triggers in LEAD.

Suppose a storm front is moving across the U.S. Midwest, threatening to spawn tornados. A user wants to deploy a data mining agent that can detect precursor conditions for a tornado, and when detected, spawn a weather prediction model. A scientist creates an experiment by interacting with an experiment builder [12] accessed through a science gateway. The specification is handed off to a workflow engine, which interacts with component pieces through a notification system. The workflow engine interacts with Calder by passing it a declarative query, similar to how it would interact with a database management system.

Calder optimizes the query and deploys it (which includes the data mining classification components [14]) at a computational node located, for instance, on the Teragrid [15]. The query when instantiated at the computational node executes the filtering/data mining loop depicted at the bottom of Fig. 2 for every incoming NexRad Level II Doppler radar volume scan. When the classification algorithm detects a vortex pattern whose intensity exceeds a pre-defined threshold, a response trigger is issued to the response channel. Workflow engine is reading the response channel, and acts on the message to wake the dormant prediction simulation.

An example query for the above mentioned scenario is given below. The START and EXPIRE clauses are used for specifying the lifetime of the query and the EXEC_FUNC clause for specifying a user-defined function to be executed on the resulting events.

```

SELECT *
FROM NexRad Level II
WHERE southBound >= "28.00"
      and eastBound <= "-89.00"
      and northBound <= "31.00"
      and westBound >= "-91.00"
EXEC_FUNC MDA_Algorithm
START "2006-03-24T00:00:00.000-05:00"
EXPIRE "2006-03-25T00:00:00.000-05:00";

```

Query 1. Filter and Mining query on NexRad II data.

III. DYNAMIC INVOCATION OF FORECAST MODELS IN LEAD

In this section, we discuss in detail the architecture and design details of the scenario provided in Section II. Fig. 3 shows the interaction of Calder with the metadata translation service and the workflow engine with listeners and also the input and output data elements. As shown in Fig. 3, arriving NexRad data is mined on-the-fly by Calder to create metadata that is then passed to a filter operation which filters out all geospatially irrelevant data.

In our application, the metadata of the incoming data products is first extracted into XML events (Fig. 3). Calder supports execution of continuous queries and triggers on XML events. The XML events are converted into the internal format of Calder and queries are processed on this internal format. The results are converted back to XML before being sent out. Currently Calder supports SQL-like queries on C structures. We have implemented serializing and de-serializing operators that transform the XML data into the internal C format and back in memory and with less overhead. In future we hope to directly support query processing on XML streams [2].

Remaining data is passed to a feature detection algorithm (labeled “Mining: MDA Algorithm”) to identify candidate mesocyclone features. These features are classified by a ADaM classifier to determine a true mesocyclone feature and if so, Calder generates a WS-Notification [7] message that is passed to an event broker and workflow handler, two core components to the LEAD model execution framework [6].

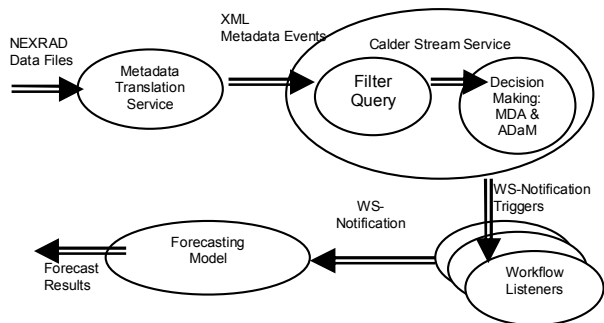


Figure 3. Dynamic invocation of computational forecast model begins with arrival of streaming NexRad data. The data is mined according to a query specified by the user. Data mining classification and feature detection algorithms are used to detect the event and the underlying stream system generates an appropriate response.

The workflow engine with listeners, is a persistent web service and can manage up to hundreds of workflow instances concurrently. It is capable of identifying the appropriate workflow instance (to which the message is addressed) and the workflow instance invokes the forecast model.

To illustrate the query processing and data mining components working together to do real-time weather monitoring, we carried out a small proof of concept experiment. The experiment is executed on one node of a 128-node cluster where each node runs RHEL WS release 4 and has dual AMD 2.0 GHz Opteron 64 bit processors with 4GB memory. The Opteron nodes are interconnected by a 1 Gbps LAN. A workload simulator process executing on another node of the cluster was used to continuously stream the NexRad II data to the Calder system. The Calder system executed the continuous query specified as Query 1 in Section II on the incoming NexRad II data.

Table 1 shows the total service time taken for executing the query and MDA algorithm on NexRad II datasets. We can see that query execution consumes a small fraction of total service time. More complex queries may have a longer execution time, but from similar experiments conducted using different datasets, we observed that query execution time is fairly minimal (in ms) and dependant on the rates of the input streams when joins are involved [16]. The total service time is negligible compared to the data collection time for a radar volume, which is about 5 min.

TABLE 1. SERVICE TIME FOR EXECUTING FILTER QUERY AND DATA MINING ALGORITHM ON NEXRAD LEVEL II DATA.

Description	Average	Std. Deviation
Query Execution Time	0.343712(ms)	0.042686
MDA Execution Time	624.807(ms)	9.37785
Total Service Time	626.407(ms)	9.38881

IV. CONCLUSION AND FUTURE WORK

In this paper, we presented a framework for dynamic invocation of forecasting models in LEAD. The Calder stream processing system serves as the underlying framework and handles filtering operations, trigger generation, and communication. It invokes data mining algorithms like MDA and ADaM classifier on NexRad data for feature detection and classification. Interesting detections are conveyed in the form of WS-Notifications [7] to the dormant forecast simulations.

Data formats and data types play an important role in scientific data processing. The primary focus of the ongoing work is to support continuous query processing on XML streams. XML has been widely used in scientific applications due to its highly extensible nature. Calder, currently supports various data formats, lacks the ability to dynamically add new data formats and user-defined functions. XML based

language support will allow users to dynamically define new data formats. Future plans include support for additional mining and filtering operations and additional data stream types.

We are also working on developing a provenance tracking system [17] for Calder which tracks changes in its distributed execution environment as well as changes in input stream characteristics. Our goal is to build a full-fledged context management system that enables Calder to dynamically adapt to domain specific changes in streams.

REFERENCES

- [1] D. J. Abadi, et al., "The Design of the Borealis Stream Processing Engine", in Second Biennial Conference on Innovative Data Systems Research (CIDR) Conference, 2005.
- [2] M. Altinel and M. J. Franklin, "Efficient filtering of XML documents for selective dissemination of information", in the Very Large Database (VLDB) Conference, 2000.
- [3] R. Avnur and J. M. Hellerstein, "Eddies: continuously adaptive query processing", in ACM SIGMOD International Conference on Management of Data, 2000.
- [4] U. V. Catalyurek, "Supporting large scale data driven science in distributed environments", in Minisymposium on Distributed Data Management Infrastructures for Scalable Computational Science and Engineering Applications, SIAM Conference on Computational Science and Engineering (SIAM CSE '05), 2005.
- [5] S. Chandrasekaran, et al., "TelegraphCQ: Continuous dataflow processing for an uncertain world", in Conference on Innovative Database systems Research (CIDR), 2003.
- [6] K. Droegeleier, K. Brewster, M. Xue, D. Weber, D. Gannon, B. Plale, D. Reed, L. Ramakrishnan, J. Alameda, R. Wilhelmson, T. Baltzer, B. Domenico, D. Murray, A. Wilson, R. Clark, S. Yalda, S. Graves, R. Ramachandran, J. Rushing and E. Joseph, "Service-oriented environments for dynamically interacting with mesoscale weather", *Computing in Science and Engineering*, IEEE Computer Society Press and American Institute of Physics, Vol. 7, No. 6, pp. 12-29, 2005.
- [7] Y. Huang, A. Slominski, C. Herath, and D. Gannon, "WS-Messenger: A Web Services based Messaging System for Service-Oriented Grid Computing", in press, CCGrid 2006.
- [8] X. Li, R. Ramachandran, J. Rushing, S. Graves, K. Kelleher, S. Lakshminarayanan, K. Douglas, and L. Jason, "Mining NEXRAD Radar Data: An Investigative Study", Interactive Information and Processing Systems (IIPS), American Meteorological Society, 2004.
- [9] Y. Liu, B. Plale and N. Vijayakumar, "Realization of GGF DAIS Data Service Interface for Grid Access to Data Streams", IU-CS, TR613, May 2005.
- [10] N. W. Paton, and O. Diaz, "Active database systems", *ACM Computing Surveys*. 31, 1, 63-103, Mar. 1999.
- [11] B. Plale, "Framework for Bringing Data Streams to the Grid, Scientific Programming", IOS Press, Amsterdam, Vol. 12, No. 4, 2004.
- [12] B. Plale, D. Gannon, Y. Huang, G. Kandaswamy, S. Pallickara, and A. Slominski, "Cooperating services for data-driven computational experimentation", in *Computing in Science and Engineering (CiSE)*, IEEE Computer Society Press and American Institute of Physics, Vol. 7, No. 5, pp. 34-43, 2005.
- [13] B. Plale, R. Ramachandran, and S. Tanner, "Data Management Support for Adaptive Analysis and Prediction of the Atmosphere in LEAD", 22nd Conference on Interactive Information Processing Systems for Meteorology, Oceanography, and Hydrology (IIPS), January 2006.
- [14] J. Rushing, R. Ramachandran, U. Nair, S. Graves, R. Welch, and A. Lin, "ADaM: A Data Mining Toolkit for Scientists and Engineers", *Computers & Geosciences*, 31, 607-618, 2005.
- [15] TeraGrid, 2005. <http://www.teragrid.org>.
- [16] N. Vijayakumar, Y. Liu, and B. Plale, "Calder query grid service: Insights and experimental evaluations", in press, CCGrid Conference, 2006.
- [17] N. N. Vijayakumar, B. Plale, "Towards Low Overhead Provenance Tracking in Near Real-Time Stream Filtering", in press, IPAW 2006.