Towards a Grid Services-based Data Streams Resource in Support of LEAD

Beth Plale
Computer Science Dept.
Indiana University
Problem Statement

• Data streams are a growing information resource
• Grid support for data streams is incomplete, richer solutions are ad hoc
• We propose new model for stream support that is
  – General – can accommodate stream handling systems of different types
  – Leverages and extends existing, well accepted grid infrastructure
• We are building a prototype system that leverages earlier work on dQUOBO
Outline

- Linked Environments for Atmospheric Discovery (LEAD) project
- Characterization of data stream systems
- Stream model as a ‘data resource’
- Representation of stream model in Grid model of computing
- dQUOQB as prototype system for grid service access to stream data in LEAD
- Conclusion
Welcome to the home page for LEAD, a Large National Science Foundation Information Technology Research (ITR) proposal scheduled to be funded by the starting 1 September 2003. LEAD proposes to enable an integrated, scalable framework for use in accessing, preparing, assimilating, predicting, managing, mining/analyzing, and displaying a broad array of meteorological and related information, independent of format and physical location. Additional details may be found in the links shown to the left, and information about the eight collaborative partners is given below.
Motivation for LEAD

• Each year, mesoscale weather – floods, tornadoes, hail, strong winds, lightning, and winter storms – causes hundreds of deaths, routinely disrupts transportation and commerce, and results in annual economic losses > $13B.
The study of events responsible for these losses is stifled by rigid computing frameworks (middleware) that cannot accommodate the
– tremendous computational demands, which are among the greatest in all areas of science and engineering
– disparate, high volume data sets and streams; and
– real time, on-demand, and dynamically-adaptive needs of mesoscale weather research;

Some illustrative examples…
The Results Analysis Problem

• Specifically,
  – Under what conditions (vertical profiles of horizontal wind, temperature, and humidity) do supercell storms cycle, i.e., produce multiple mesocyclones/tornadoes?

• Critical implications for forecasters
• Clear guidelines exist for distinguishing among types of thunderstorms (supercells, single cells, lines)
• Use numerical model to span desired parameter space
  – 20 vertical profiles of temperature
  – 20 vertical profiles of humidity
  – 100 vertical profiles of the horizontal wind
  – Yields 4,000 simulations!
  – We have the computer time, but we don’t have the ability to analyze the results!
Q: When Do Supercell Storms Produce Multiple (Cyclic) Tornadoes?

How Will LEAD Help?

Create Continuous Climatology From Observations

Data Assimilation

Gridded Assimilated Data Sets

4000 Storm Simulations

Data and Metadata

My LEAD Virtual Public Space

300 Tbytes

Data Cloud

Data Mining

Cloud Data Mining Assimilation Data Sets

300 Tbytes

Create Continuous Climatology From Observations
Forecasts on Demand

- Requires a human to determine domain location, other parameters (no automated response to the weather)
- Requires known computational resources (not grid-enabled)
- Receives no special computational priority (no service guarantees)
Q: Why Do Some Severe Storms Produce Multiple (Cyclic) Tornadoes?

Create Climatology Based Upon All Observations

Data Assimilation

Gridded Assimilated Data Sets

500 Storm Simulations

Data Mining

300 Tbytes

Data and Metadata

My LEAD Virtual Public Space

Real-Time WRF Runs on Grid Only When Environment is Primed and Storms are Present

Streaming Radar Data

On-Demand Resource Scheduling

Current NEXRAD Sites Delivering Level II Data to NCDC in Near Real Time (64 WSR-88D Sites)
Q: Why Do Some Severe Storms Produce Multiple (Cyclic) Tornadoes?

Create Climatology Based Upon All Observations

Gridded Assimilated Data Sets

Data Assimilation

500 Storm Simulations

300 Tbytes

Data Mining

Real-Time WRF Runs on Grid Only When Environment is Primed and Storms are Present

Real Time Control of Radars

On-Demand Resource Scheduling

My LEAD Virtual Public Space

Data and Metadata
LEAD CS Research

- **Workflow orchestration** – the construction and scheduling of execution task graphs with data sources drawn from real-time sensor streams and outputs.
- **Data streaming** – to support robust, high bandwidth transmission of multi-sensor data.
- **Data management** – for storage and cataloging of observational data, model output and results from data mining.
- **Distributed monitoring and performance evaluation** -- to enable soft real-time performance guarantees by estimating resource behavior.
- **Data mining tools** – that detect faults, allow incremental processing (interrupt / resume), and estimate run time and memory requirements based on properties of the data (e.g., number of samples, dimensionality).
- **Semantic and data interchange technologies** – to enable use of heterogeneous data by diverse tools and applications.
Our focus: run time model adaptations in response to streaming data

Real-Time WRF runs executed on Grid when environment is primed and storms are present

Data Cloud: broad array of data stream sources
Outline

• Linked Environments for Atmospheric Discovery (LEAD) project
• Characterization of data stream systems
• Stream model as a ‘data resource’
• Representation of stream model in Grid model of computing
• dQUOB as prototype system for grid service access to stream data in LEAD
• Conclusion
Data streams:

- Indefinite sequence of events, messages, tuples
- Often time marked
  - Generation time, that is, timestamp, and
  - Logical time
- Events continuously generated
  - pushed or pulled from providers to remote consumers
- Because sequence is indefinite, decision-making takes form of requests that are long running, continuously executing
Types of Data Stream Systems

- **Data manipulation systems**
  - processing large amounts of data

- **Stream detection systems**
  - identifying unusual behavior

- **Stream routing systems**
  - delivery of events

Size and number of stream chunks analyzed vs. timeliness demands on response
Stream Routing Systems

- Known by various names
  - Publish/subscribe, selective data dissemination
  - Document filtering, message oriented middleware (MOM)
- Decisions made event-by-event
  - Set of queries (usually very large) managed over time duration, arriving event matched against set of queries.
- Projects
  - Xfilter (UMaryland), Xyleme (INRIA), XPushMachine (UWashington), NaradaBrokering (IndianaU), Bayou (XeroxParc), Echo (GeorgiaTech)
Stream Routing Example

- Timeliness requirement necessitates focus on efficient matching

queries added through user interface

Long-standing queries

Results multicast to owners of satisfied queries

Each arriving event matched against set of queries

stock quote stream
Data Manipulation Systems

- Event streams subject to transformation, filtering, aggregation.
- Looser timeliness requirements on results
- Long running queries, often periodic (based on assumption of synchronous streams)
- Results in generation of new streams
- Projects:
  - Antarctic Monitoring (UNottingham), sensor network query layer (Cornell), dQUOB (IndianaU), STREAM (Stanford), Fjords (Berkeley), NiagraCQ (UWisconsin)
Stream Detection Systems

- Event-oriented (versus periodic)
- Less predictable, asynchronous streams
- Intent is to detect anomalous behavior
- Timeliness is critical, time markers key to decision making
- Result is notification message
- Examples:
  - R-GMA (EU DataGrid), dQUOB (IndianaU), Conquer (GeorgiaTech), Gigascope (AT&T), Fjords (Berkeley)
Data resource has coherence and meaning

Circled systems qualify as ‘data resource’ because:

-- distributed global snapshot on stream behavior alone
-- distributed global snapshot has meaning and coherence
Outline

• Linked Environments for Atmospheric Discovery (LEAD) project
• Characterization of data stream systems
• Stream model as a ‘data resource’
• Representation of stream model in Grid model of computing
• dQUOB as prototype system for grid service access to stream data in LEAD
• Conclusion
OGSA Grid Data Service:
defined by standard extensible ports

Data description port
- database description

Data access port
- query/update access

Data management port
- monitor service

Data factory port
- create rowset instance
Database Access: OGSA-DAI

Grid data Service Registry

Grid data service Factory

Grid data Service (GDS)

Rowset Grid data service

DBMS

Grid service handle of factory

handle of GDS

create GDS

query

response (row-by-row)

service create
Grid Data Service Access to New Kind of Data Resource

Sub-queries pushed into external data resource

External data resource

Data manipulation and detection systems

Grid Data Service (GDS)

Rowset GDS

provider/consumer communication mechanism

request

response

Stream routing systems
Data Stream Store: access to distributed data streams

- Grid service
- Rowset GDS
- Grid Data Service (GDS)
- Data Stream Store: access to distributed data streams
- long running queries
- streamed response
- providers external
- query execution nodes
- streams
Goal: runtime model adaptation as access to a stream store

Real-Time WRF runs executed on Grid when environment is primed and storms present

On-Demand Resource Scheduling

SQL query

stream of responses
## Capabilities of Data Stream Store

<table>
<thead>
<tr>
<th>Stream capability</th>
<th>RDBMS equivalent</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create virtual stream store</td>
<td>Create database</td>
<td>Store has existence independent of any active streams</td>
</tr>
<tr>
<td>Register stream to store</td>
<td>Create table</td>
<td>Provider has control over which streams to publish</td>
</tr>
<tr>
<td>Publish events to stream</td>
<td>Update database</td>
<td>Selective, user-level event generation to store.</td>
</tr>
<tr>
<td>Query resource</td>
<td>Query database</td>
<td>Query is long running and distributed. Response is stream.</td>
</tr>
</tbody>
</table>
Outline

- Linked Environments for Atmospheric Discovery (LEAD) project
- Characterization of data stream systems
- Stream model as a ‘data resource’
- Representation of stream model in Grid model of computing
- dQUOB as prototype system for grid service access to stream data in LEAD
- Conclusion
dQUOB: data stream system for data manipulation and detection

- Relational database view of streams
  - Event == tuple, data stream == relation
- Query has rule-based syntax
  - Supports subset of SQL query
  - Coupled with user-defined functions (e.g., mathematical functions (FFT), compression).
- Time-based, two-way join
  - Two events satisfy join if they ‘happen at the same time’
  - Over logical time or timestamp
dQUOB – prototype for virtual stream store model

- dQUOB library
  - queries
  - quoblet runtime

- Grid Data Service
  - Rowset Grid Data Service
    - handles result set

- dQUOB SQL compiler
  - query script (Tcl)

- Converts script of SQL query into C++ DAG of selects, projects, joins.

Downside: user defined action must be dynamically linked.

Stream of events/tuples satisfying query
Queries arrive from Grid Data Service as scripts and are instantiated at quoblet as DAGs of C++ objects.
Research Issues

• Query/subquery distribution
  – Stream store is distributed resource
  – Location transparency enforced
  – Grid Data Service needs pluggable module for query distribution
  – Placement of query fragments done with consideration of existing queries, duplication, proximity to source, etc.
Issues in Data distribution

(a) Aggregation queries

(b) Transformation queries
Research Issues, cont.

- Management of long-running queries
  - Queries reside in virtual stream store, execute continuously
  - Query lifetime specified by extension to query language
  - Response to user should be stream of results
    - Result is generated each time query is run and evaluates to true
Research Issues, cont.

- Query placement - placement of query fragments into virtual stream store
  - Dynamic instantiation of query code at destination
- Updates – creation and publication to data stream
  - Publication of events should be point-to-point in data system
Our focus:
- virtual stream store model provides general framework for streaming systems to run in grid services environment.
- support for WRF adaptibility by means of query access to data streams
http://www.cs.indiana.edu/~plale/projects/dQUOB