Earth Sci Inform (2008) 1:49–57
DOI 10.1007/s12145-008-0010-7

RESEARCH ARTICLE

Real-time storm detection and weather forecast activation through data mining and events processing

Xiang Li · Beth Plale · Nithya Vijayakumar · Rahul Ramachandran · Sara Graves · Helen Conover

Received: 5 November 2007 / Accepted: 27 March 2008 / Published online: 28 May 2008

Abstract Each year across the USA, destructive weather events disrupt transportation and commerce, resulting in both loss of lives and property. Mitigating the impacts of such severe events requires innovative new software tools and cyberinfrastructure through which scientists can monitor data for specific severe weather events such as thunderstorms and launch focused modeling computations for prediction and forecasts of these evolving weather events. Bringing about a paradigm shift in meteorology research and education through advances in cyberinfrastructure is one of the key research objectives of the Linked Environments for Atmospheric Discovery (LEAD) project, a large-scale, interdisciplinary NSF funded project spanning ten institutions. In this paper we address the challenges of making cyberinfrastructure frameworks responsive to real-time conditions in the physical environment driven by the use cases in mesoscale meteorology. The contribution of the research is two-fold: on the cyberinfrastructure side, we propose a model for bridging between the physical environment and e-Science[1] workflow systems, specifically through events processing systems, and provide a proof of concept implementation of that model in the context of the LEAD cyberinfrastructure. On the algorithmic side, we propose efficient stream mining algorithms that can be carried out on a continuous basis in real time over large volumes of observational data.

Keywords Cyberinfrastructure · e-Science · Weather forecast · Data mining · Workflow-driven analysis

Communicated by: H. A. Babaie

[1] e-Science is used to describe computationally intensive science that is typically carried out in highly distributed network

X. Li · R. Ramachandran (✉) · S. Graves · H. Conover
University of Alabama Huntsville,
Huntsville, AL, USA
e-mail: rramachandran@itsc.uah.edu

B. Plale · N. Vijayakumar
Indiana University,
Bloomington, IN, USA

Introduction

Information technology research has made significant advances over the past several years in cyberinfrastructure architectures for computational science investigation. These infrastructures, often assembled as loose collections of services, enable new forms of scientific investigation; of particular interest for this paper is adaptive investigation that is responsive to sensors, instruments, and sensor networks in the physical environment. The meteorology community is on the leading edge in motivating adaptive cyberinfrastructure research by means of use cases that involve the sensing and recognition of severe storms, and associated complex parallel and distributed analysis sequences that are invoked in response. Until recently meteorologists conducted research in severe storm modeling through either executing weather forecast runs on a static schedule or through conducting an analysis post-mortem, long after the weather conditions had moved on. The models and codes used in meteorology research were often stitched together by complex and brittle scripts. The cyberinfrastructure developed as part of the Linked Environments for Atmospheric Discovery (LEAD) project (LEAD 2007) was funded to address this challenge. The LEAD cyberinfrastructure (LEAD-CI) gives a scientist tools with which they can automatically spawn weather forecast models in response to real time weather events for a desired region of interest. The interaction modes and data-

driven analysis patterns supported in the LEAD-CI are applicable to a broad class of domain science problems generally known as *e-Science*. Cyberinfrastructure solutions developed here and elsewhere are bringing advanced research and education capabilities to all e-Science applications, often first by domain-specific solutions that are then generalized further.

The LEAD-CI has emerged as an outcome of the LEAD project, an interdisciplinary research project between computer scientists, meteorology community data providers, and meteorology researchers and educators. The purpose of the project, which began in 2001, has been to address the fundamental IT and meteorology research challenges needed to create an integrated, scalable framework for identifying, accessing, decoding, assimilating, predicting, managing, analyzing, mining, and visualizing a broad array of meteorological data and model output, independent of format and physical location (Droegemeier et al. 2005). A cyberinfrastructure framework engineered to address this need must account for multiple dimensions of complexity. The framework has to handle users specifications, computational requirements, data access and integration in real time, and complex workflow execution. The latter complex workflow execution is accomplished by deploying as part of the infrastructure a workflow engine; these engines are capable of executing arbitrarily complex sequences of tasks on behalf of a user.

A couple of challenges exist in making cyberinfrastructure frameworks responsive to real-time conditions in the physical environment: first, workflow engines are often not well suited to the continuous and indefinite nature of data arriving from sensor networks and instruments. Instead, the workflow engines are more likely to execute a directed graph of analysis tasks once, then exit. Second, detection of physical or environmental phenomena in streaming data requires data mining algorithms, often embedding domain specific knowledge, that are not designed for continuous processing over large amounts of high volume data.

Our solution to the problem of bridging between the physical environment and workflow-driven analyses derives from the observation that events processing systems (Gartner 2007) are well suited to continuous stream mining as demonstrated by their application to network processing (Cranor et al. 2003), Interstate traffic patterns (Arasu et al. 2004), and ecology monitoring (Mainwaring et al. 2002). Specifically, anomalous behavior in the physical environment (such as severe storms) can be recognized by an events processing model capable of filtering through the vast volumes of environmental data, carrying out sophisticated data mining algorithms for classifying weather patterns, and interacting with workflow systems thus creating a linkage in real time between observational data arriving in real time from sensor networks and instruments and the complex workflow-driven analysis sequences that ingest and act upon the data (Vijayakumar et al. 2006).

The contribution of this research is two-fold: on the cyberinfrastructure side, we propose a model for bridging between the physical environment and e-Science workflow systems, specifically through events processing systems, and provide a proof of concept implementation of that model in the context of the LEAD-CI cyberinfrastructure for mesoscale meteorology research. On the algorithmic side, we propose efficient stream mining algorithms that can be carried out on a continuous basis in real time over large volumes of observational data.

Science motivation

Each year across the USA, floods, tornadoes, hail, strong winds, lightning, and winter storms—so-called mesoscale weather events—cause hundreds of deaths, routinely disrupt transportation and commerce, and result in annual economic losses on the order of $12B (Pielke and Carbone 2002). Mitigating the impacts of such events requires frameworks to accommodate the real time, on-demand, and dynamically-adaptive needs of mesoscale weather operational forecasting. Most current operational weather prediction systems throughout the world run on fixed time schedules, and in fixed configurations, regardless of user need or the actual weather. Software tools are needed to provide scientists with the ability to monitor data for specific weather events and simulative modeling computations for these specific evolving weather events. These tools should be able to launch weather models at finer-scale features in response to specific weather event detections to deal most effectively with weather that is driven by local physical influences (e.g., terrain) and has local impact. Tools with such capability can have profound implications for operational weather forecasting in the USA. The impacts of such tool will not just be limited to forecasting operations but will also be vital for education. It will allow students to analyze weather events as they occur and then study the model forecasts and its associated societal impacts, particularly for the local area.

These tools must be based on dynamically adaptive on-demand frameworks that can monitor and be steered by data continuously; change configuration rapidly and automatically in response to the weather; initiate other processes such as spawning weather models or other mining algorithms automatically; and be easily configurable by users. For instance, the Weather Research Forecast model (WRF) is initialized with observational data and external model forecast data (such as data from the North American

Meso (NAM) model) and lateral boundary conditions are set by coarser forecast data (such as an external model or WRF run at coarser resolution). Key components of the LEAD-CI architecture that support the automatic response is shown in Fig. 1. The LEAD-CI is organized as a set of cooperating web services. The user interacts with the cyberinfrastructure through a portal, often called a "science gateway". The user will bring up a graphical interface and link together components to form a workflow graph. This is shown in the top center of the diagram. The workflow graph is submitted to the Workflow Engine, which submits it as a distributed and parallel job to the back end computational resources. The web services that work on the user's behalf communicate with one another by publishing messages to the event notification bus. For instance, when the WRF model has finished executing on the parallel computer, a message is returned to the Workflow Engine, which can then take action to start up the next analysis task in the sequence of tasks indicated by the graphical picture the scientist drew. The LEAD-CI is richer in functionality than what is depicted in Fig. 1. It provides meteorology researchers and students secure X.509 certificate access to services, access to a large number of indexed data products that are available in Unidata's THREDDS Data Server and OPeNDAP servers, complex weather forecasting models, assimilation and visualization codes such as Unidata's IDV.

Simplifying assumptions

Severe weather events such as severe storms, tornadoes have both a geospatial and a temporal component to them. Even though both these components are important in characterizing these events, for the initial research the temporal component was ignored. This assumption allows the detection algorithms to view the streaming data as a single independent snapshot of the state of the environment. This simplifies the mining algorithms as it is no longer required to track these weather events as they grow, evolve, move and die over time.

The justification of launching a finer resolution model forecasts run after severe weather events have been detected in real-time is based on the supposition that the storms may still grow and move. It is also assumed that the storm growth and movement is not too rapid, in other words the detection and the forecast model run results are produced in reasonable time to be useful. This assumption applies to most storm systems but may not be applicable under few scenarios such as squall lines.

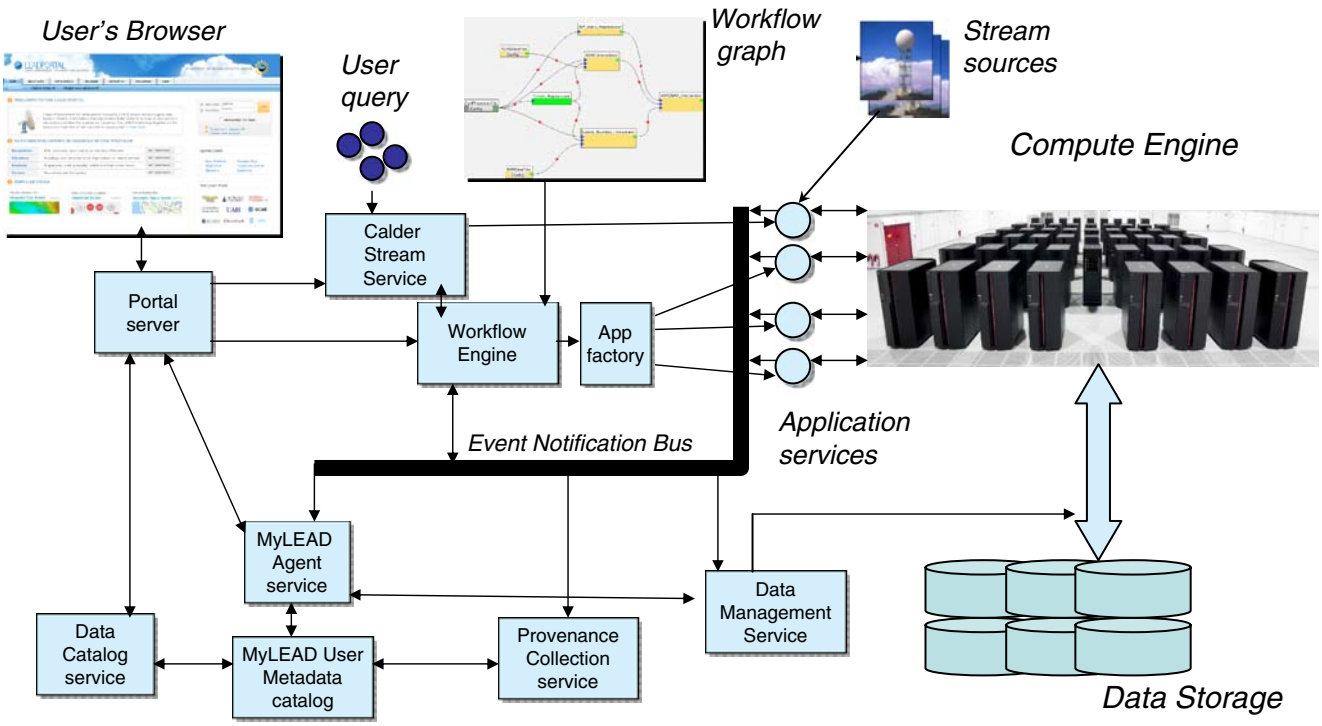

Fig. 1 LEAD-CI cyberinfrastructure support for stream mining. The workflow engine submits workflows through the application factory to the back end compute engine. The events processing component (called Calder Stream Service) is selectively mining stream sources (*upper right*), and communicating its results to the workflow engine through trigger messages

Model for environment driven action in Service Oriented Architecture (SOA)

Computational science cyberinfrastructures often include a workflow engine used to automatically execute sequences of tasks on behalf of a user. The scientist specifies a desired sequence of tasks through some manner of task graph construction, either by writing the XML-based script directly, through a visual interface (Gannon et al. 2007) wherein which the scientist connects tasks together, or through a declarative language or other simplification means. A workflow graph's ability to respond to events in the external environment is built into the formal specification of most workflow languages (through support for a specific workflow patterns called a "persistent trigger pattern" (Russell et al. 2006) where a trigger can initiate a task (or the beginning of a thread of execution) that is not contingent on the completion of any preceding tasks. In the BPEL workflow language this behavior is achieved through the <pick> construct waiting on specific message type; in UML 2.0 Activity Diagrams, support is provided through signals. We extend the persistent trigger pattern (Russell et al. 2006) with event sequences to capture the behavior of an interaction between a workflow and its external environment through a bounded period of time.

The manner in which a workflow responds to the environment is as follows. The physical environment (sensor networks and instruments), as shown in Fig. 2, causes the generation of some number, i_m, of timestamped events over a bounded time range, $[t_0 - t_k]$. The events are not necessarily of the same type or contained in a single "stream", and no ordering of events can be assumed. Events are received at an activity, a rule or SQL-based events processing activity, which carries out an arbitrarily complex decision process to determine if an anomalous event (severe weather) has occurred. An anomalous event causes the generation of an event e_i that is sent to the trigger production activity. Note that since the activity is looking for anomalous behaviors only, the number of events generated in response to events processing, n, is significantly smaller than the number of events generated by the physical environment, m, that is $(n << m)$. The persistent trigger workflow pattern is defined such that a "start thread" listener can respond to a sequence of triggers, and in response can execute a new activity that receives parameters in the form of the event or message contents at startup. An execution of a workflow results in the generation of one or more result products that are labeled $<o_1, \ldots o_p>$ where p is not necessarily equal to n. The scientist's request to monitor the physical environment is bounded in time. When the end of the time interval is reached, a different kind of trigger, a "stop" trigger, must be sent to signal the workflow that there are no further triggers.

To the scientist and cyberinfrastructure programmer alike, the bridging of real time observational data into a Service Oriented Architecture (SOA) such as LEAD is transparent. This is accomplished by making events processing just another web service that interacts with the user through the portal and responds to requests the workflow engine. As illustrated in Fig. 1, the events processing tools are wrapped as a web service as shown in the figure as the Calder Stream Service. The Calder Stream takes requests in the form of "User query" through the Portal Server. The stream service deploys requests (as

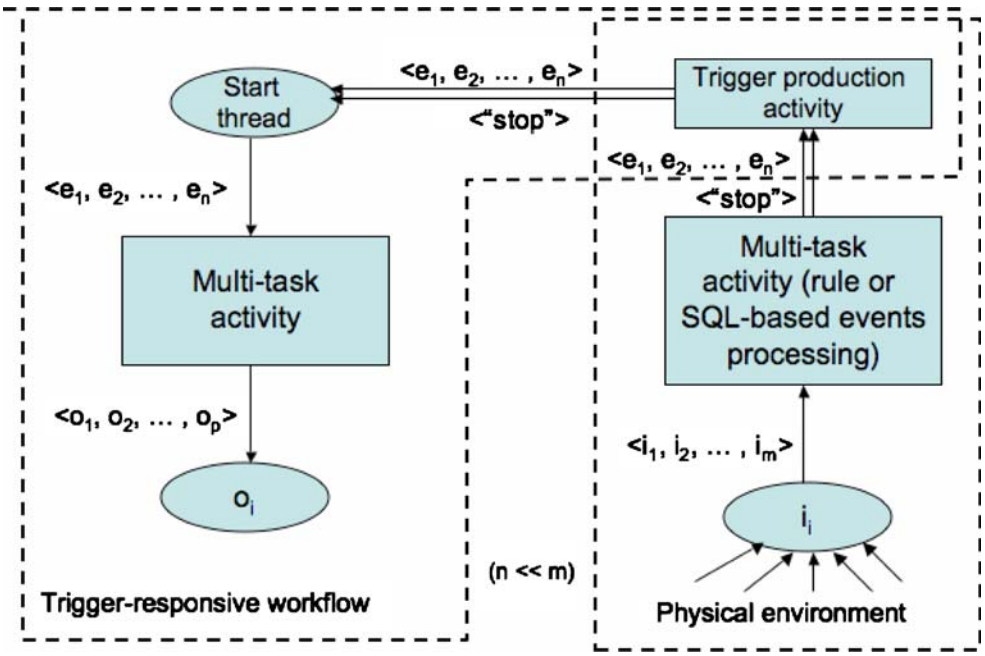

Fig. 2 The trigger-responsive workflow on the left receives a stream of notification events from the trigger production activity. The stream mining system on the right detects and responds to changes in the physical environment. Many workflow languages support responses to external triggers. Extended from (Russell et al. 2006)

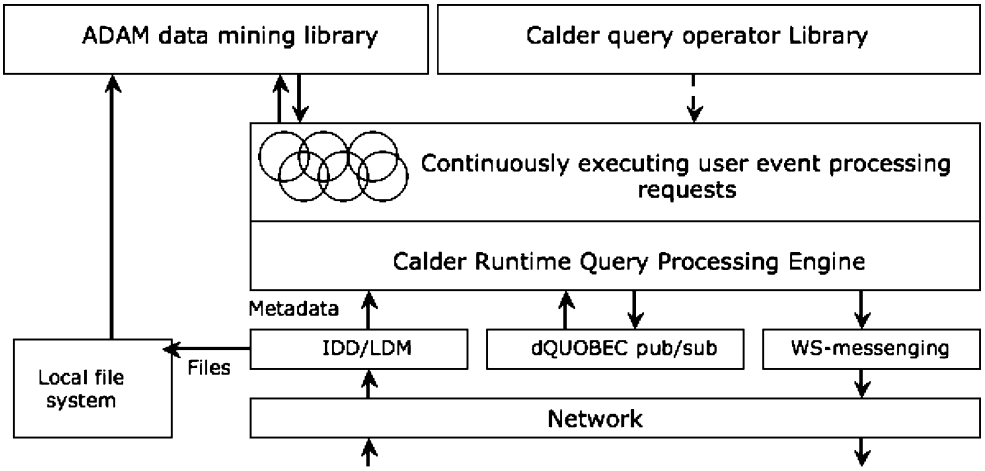

Fig. 3 Architecture of stream processing system. The three transport systems (IDD/LDM, dQUOBEC, and WS-messenger) transport events from outside observational data sources, from within the continuous query system, and to the trigger activity respectively. The Calder library provides SQL-like operators and the ADAM data mining library provides the classification algorithms. Together the libraries support filter, fuse, transform, and mining of data streams

long-running queries) to the compute engine. Queries ingest data in real time from the stream sources, shown in the upper right of the diagram as radar icons.

The architecture of one stream processing system is shown in Fig. 3. Above the network layer are three transport systems, each serving a unique role. The Unidata Internet Data Dissemination/Local Data Manager (IDD/LDM) system ingests and distributes observational data. The data products, most often in the binary netCDF format, are cracked open and the metadata passed as an XML document to the Calder Runtime query processing engine (Liu et al. 2006). The XML documents are converted into the internal format of Calder and queries are processed on this internal format. The resulting derived events are converted back to XML before being sent out. Calder supports SQL-like queries on C structures. We have implemented serializing and de-serializing operators that transform the XML documents into the internal C format and back in memory.

The netCDF data file is written to the local file system for use by the Algorithm Development and Mining (Rushing et al. 2005) library (described in next section). The dQUOBEC publish-subscribe system transports events within the system. The WS-messenger (Huang et al. 2006) notification system, a content based publish-subscribe system, is used to ferry trigger messages to the workflow engine in the LEAD-CI. Above the communication layer is the Calder runtime which invokes queries upon arrival of relevant events. The events are instantiated from operators in the Calder query operator library, which provides SQL-like query operators and functionality, and the ADAM data mining library which provides functions for mining meteorology and atmospheric data.

Mining components

The Algorithm Development and Mining (ADaM) is an extensive image processing and data mining toolkit (Rushing et al. 2005). This toolkit contains almost all of the typical image processing and data mining algorithms used by researchers in their analysis as well as numerous data preprocessing algorithms such as feature reduction, subsetting, subsampling, etc. This toolkit was augmented with new specialized mining algorithms to address the use case selected for this proof of concept implementation. These additional specialized mining algorithms are presented in detail in this section.

Detection algorithms

Mesocyclones

Mesocyclones are rotating updraft/downdraft structures inside severe thunderstorms. Detection of mesocyclones is important for severe weather forecast because over 90% of mesocyclones are accompanied by severe weather such as tornadoes or large hail (Burgess 1976). Mesocyclone signatures can be identified from the Weather Surveillance Radar —1988 Doppler (WSR-88D, (Crum et al. 1993)) and appear as couplets of incoming and outgoing radial velocities. Experienced radar analysts can identify mesocyclone signatures in WSR-88D data. However, this manual identification process is tedious and time-consuming, and can be overwhelming during a severe weather outbreak. Consequently, several algorithms have been developed (Stumpf et al. 1998; Zrnic et al. 1985; Desrochers and Donaldson 1992) for automated mesocyclone signature detection. The National Severe Storm Laboratory (NSSL) MDA (Stumpf et al. 1998) is one such algorithm and is an enhancement to the Build 9 WSR-88D Mesocyclone Algorithm (B9MA) (Zrnic et al. 1985). Compared to B9MA, NSSL MDA identifies a broader spectrum of mesocyclones and has an improved probability of mesocyclone feature detection. The latest version of the NSSL MDA includes a neural network classifier to filter out false mesocyclone signatures (Marzban and Stumpf 1996).

Mesocyclones have a velocity signature known as a Rankine Vortex (Donaldson 1970), representing incoming and outgoing radial velocity couplets in the radar velocity field. We designed a Mesocyclone Detection Algorithm (UAH MDA) that uses this velocity signature for detection. The detection algorithm has three steps: (1) identify 1-dimensional (1D) shear segments based on Rankine Vortex, (2) identify two-dimensional (2D) shear regions in a scan sweep, and (3) Aggregate collocated shear regions (along the elevation scans) into 3-dimensional (3D) mesocyclone signatures. A known criteria is used to determine the strength of the shear segments (Stumpf et al. 1998). The algorithm builds 2D shear regions using a region-growing technique (Gonzalez and Woods 1992), i.e., grouping those 1D shear segments adjacent to each other into a 2D region. A feature vector is calculated for each 2D shear region and its features include central azimuth angle, central range, central height, diameter, shear, maximum Gate-to-Gate Velocity Difference (GTGVD), rotational velocity difference (Vrot), and mesocyclone strength index, as defined in Stumpf et al. (1998). For a 3D mesocyclone signature, a number of features such as base and top heights, core base and top heights are calculated in addition to the mean values of the corresponding 2D features. Consequently, each 3D mesocyclone signature contains twenty features. The UAH MDA can be used in conjunction with a classifier to accurately detect mesocyclones and filter out false signatures. Details on this algorithm and comparison of its performance against other similar algorithms can be found in Li et al. (2004).

Storm detection algorithm (SDA)

A flexible Storm Detection Algorithm (SDA) based on user defined thresholds that would work on multiple types of data sets such as the WSR88-D and numerical models outputs was developed to meet the requirements of LEAD project scientists and students. The algorithm is an extension of a basic image thresholding algorithm where the user provides the threshold value and the data points with intensities higher than the specified threshold are retained. Since storms must have a minimum size and spatial volume, SDA uses region growing technique to build 3D volumes using the retained pixels. Volumes that meet the minimum size criteria are kept. SDA output contains spatial location in latitude and longitude, distance from radar sites, the sizes and depths of the storm, the base height of the storm, and mean and maximum reflectivity in the storm for each of the storms detected in a data volume. The SDA complements the MDA as it targets a larger phenomenon (storms). The two algorithms can be used concurrently do identify storms and tornadoes to aid in severe weather prediction and forecast.

Storm clustering algorithm (SCA)

In many cases, the SDA and the MDA can produce numerous event detections for a given spatial region of interest. Brute force approach of spawning a finer resolution model run for each detection can be computationally prohibitive and can produce redundant forecast especially if the detections have close spatial proximity. A better approach is to spatially group the detections, determine which groups are most interesting and spawn limited finer resolution models runs for groups that ranked high. We have developed a Storm Clustering algorithm to address this problem. Our algorithm is based on the DBSCAN algorithm (Ester et al. 1996). The DBSCAN algorithm is a density-based clustering algorithm, which regards a spatial cluster as a region in data space that contains data samples with certain density. Data density around a data sample is determined as the number of samples N within a distance ε to the sample. Two data samples are connected if their distance is less than ε. DBSCAN algorithm connects neighboring samples into spatial clusters and can identify a spatial cluster of any shape. The N and ε parameters determines the number of clusters in a data set and clustering performance and indirectly determines the number of clusters in a data set.

Since the purpose of the storm clustering algorithm is to automatically group data samples into optimal number of clusters, we have employed the use of the Hartigan index (1975) criterion. The Hartigan index (1975) is a statistical index that examines the relative change of fitness as number of clusters changes. For a given data set containing N samples, $\boldsymbol{X}_i$, $i=1,N$ where $\boldsymbol{X}_i$ is a $\boldsymbol{M}$-component vector representing $\boldsymbol{M}$ features for sample i, the overall fitness for the clustering can be expressed as the square of error for all samples:

$$\mathrm{err}(k) = \sum_{i=1}^{k} \sum_{j=1, j\in Ci}^{N} d^2\left(X_j, X_{Ci}\right),$$

Where d is the distance between data sample $\boldsymbol{X}_j$ and the center $\boldsymbol{X}_{Ci}$ that it belongs to. err(k) is the total square of error for k cluster partitioning.

The Hartigan index $H(k)$ for k partitioning, as a ratio, is expressed as follow:

$$H(k) = (n-k-1)\frac{\mathrm{err}(k) - \mathrm{err}(k+1)}{\mathrm{err}(k+1)}$$

Since err(k) is monotonically non-increasing with increasing k, the ratio is a relative measure of the reduction of square error when number of clusters increases from k to $k+1$. The multiplier correction term of $(N-k-1)$ is a penalty factor for large number of cluster partitioning. The optimal k number is the one that maximizes the $H(k)$.

Once the spatial clusters have been identified, they need to be ranked based on observed measures. This ranking allows the CEP system to only spawn detailed forecast runs on limited regions. We have developed a heuristic measure-Storm Severity Factor (SSF) to rank the spatial clusters. SSF is a weighted average of these geophysical attributes and is estimated as follow:

$$\mathrm{SSF} = \omega 1 \cdot c_s + \omega_2 \cdot c_{md} + \omega_3 \cdot c_{mi},$$

where c_s, c_{md} and c_{mi} are the normalized attributes of C_s, C_{md} and C_{mi}, respectively. C_s is the total size of all storms in a cluster; C_{md} is the mean storm depth in a cluster; C_{mi} is the mean storm intensity in a cluster. ω_1, ω_2 and ω_3 are the weights for the three geophysical attributes c_s, c_{md} and c_{mi}, respectively. These weights indicate the relative importance of the three geophysical attributes in evaluating the storm severity and $\omega_1+\omega_2+\omega_3=1$. Statistical z-score is used to normalize each geophysical attribute. Therefore, c_s is calculated as follow:

$$c_s = \frac{C_s - \mu_{c_s}}{\sigma_{c_s}},$$

where μ_{c_s} and σ_{c_s} are the mean and standard deviation of C_s, respectively. Same is true for c_{md} and c_{mi}. A large SSF value indicates severe weather for a spatial cluster and is ranked at top of the list.

Figure 4 shows the storm clusters identified over the borders of Oklahoma, Kansas and Missouri on 21:00–21:30 UT, May 6, 1994. The SDA was applied to the reflectivity field of WSR88D-II radar product. The threshold was set at 30.0 dBz, implying all radar volume with reflectivity intensity over 30 dBz were considered as part of a storm. Total of 19 storms from six radar volume scans were detected during this time period. The SCA algorithm was then applied to the identified storms. Three storm clusters were automatically identified and were labeled as C1, C2 and C3. Table 1 shows the statistics of geophysical properties of the three storm clusters, as discussed above, with the ranking using the SSF measure. The total storm size, mean storm depth and mean storm intensity were considered equally important to characterize the severity of storm clusters. Consequently, each of the weights ω_1, ω_2 and ω_3 in SSF equation was set as 1/3, respectively.

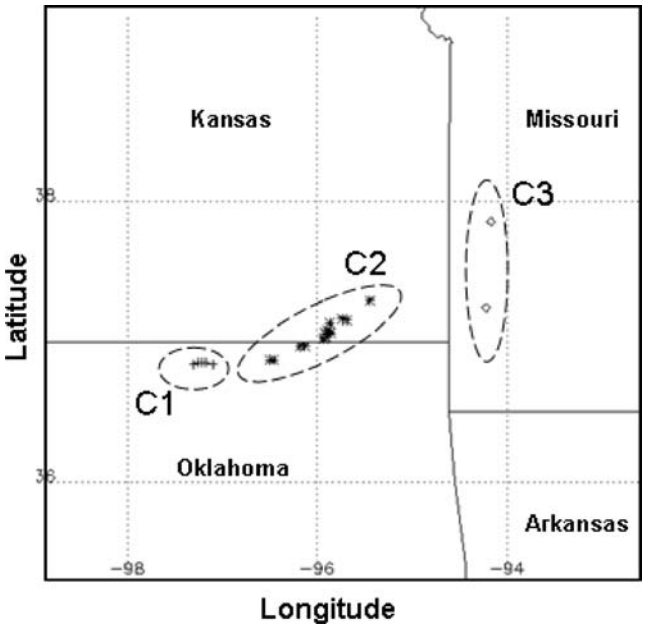

Fig. 4 Storm clusters identified over the borders of Oklahoma, Kansas and Missouri on 21:00–21:30 UT, May 6, 1994. Three clusters were identified and labeled as C1, C2 and C3

Table 1 Statistics of geophysical properties (storm size C_s, mean storm depth C_{md}, and mean storm intensity, C_{mi}), SSF measure and severity rank of the storm clusters identified using the SCA on 21:00–21:30 UT, May 6, 1994

Storm cluster	Storm number	C_s	C_{md}	C_{mi}	SSF	Rank
1	5	398.75	11.59	39.63	1.976	2
2	14	1,096.62	5.0	38.39	4.027	1
3	2	73.0	3.74	38.11	−0.742	3

Storm cluster 2 was ranked first, followed by storm cluster 1 (see Table 1). Storm cluster 3 was the least significant of the three. Detailed examination revealed that storm cluster 1 was more severe than storm cluster 2 with respect to mean storm depth and mean storm intensity. However, the total size of storm cluster 2, as contributed by the total number of storms in the cluster, was significantly larger than that of storm cluster 1. As a result, its SSF measure is larger than that of storm cluster 1. Different weight setting may cause different ranking result. The true (non-normalized) values of total storm size C_s, mean storm depth C_{md}, and mean storm intensity, C_{mi}, are presented in Table 1.

Dynamic triggering of forecast models

Detecting severe storm patterns in real time by looking though observational data is like sorting through hay in a haystack to find a needle; it requires working through large volumes of data in a short period of time. This events processing requires a combination of domain-independent operators (e.g., to fuse streams, filter unwanted data, count instances, and create new streams) plus domain-dependent operators, such as the data mining algorithms we describe here. There are two main philosophies of event processing (Liu et al. 2006): the first are the SQL-based approaches, which use a declarative, query-like language to specify the

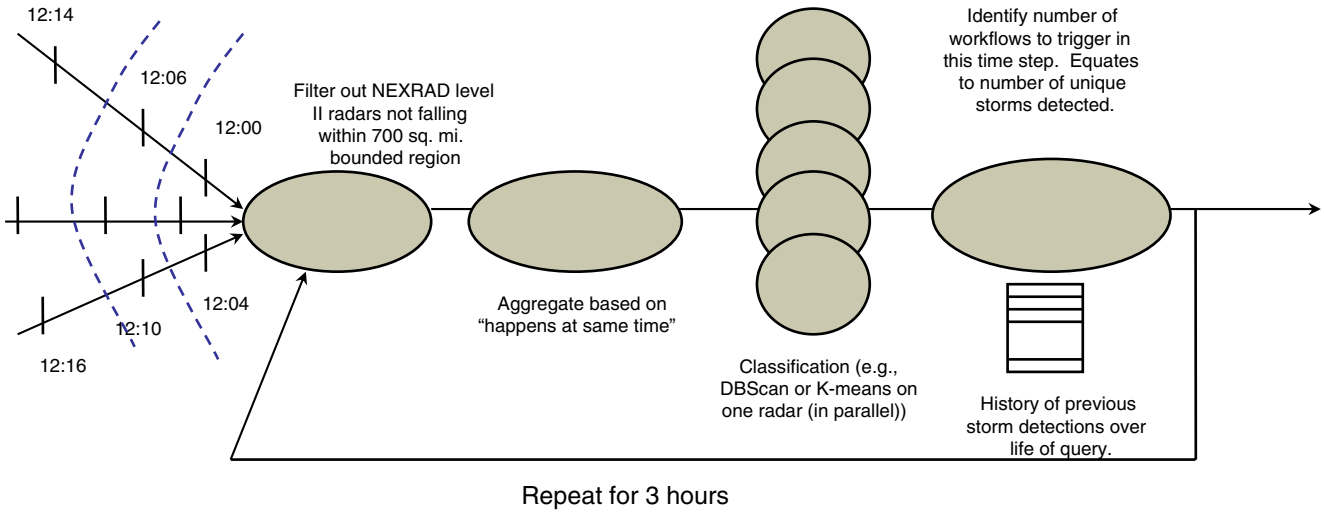

Fig. 5 An example event-processing query. Events travel from left to right. The NEXRAD Level II Doppler data arrives from multiple radars and is filtered according to the spatial area in which the scientist is interested. The detection algorithm executes in parallel on the radar data. The previous storm detections are retained from one execution of the query to the next

desired events processing behavior. The second is the rule-based approaches that view anomaly detection as behavior that can be detected by means of a series of rules that are executed in a forward or backward manner. Interaction between the events processing system and a workflow engine can best be illustrated by an example. A scientist through a graphical interface in the LEAD portal bounds a 700 square mile region using a graphical map and specifies a request that there be a "watch on a 700 square mile region of the country for the next 4 h, with a 6-h weather forecast triggered on each occurrence of a severe storm." She would also through the LEAD portal configure the 6-h weather forecast over a similarly spatially-configured region. Weather triggered workflows, which our research has enabled, are a scientific research outcome of LEAD, not having been demonstrated in a repeatable way before. In LEAD, we use the Calder events processing system and ADAM data mining toolkit to mine the real-time weather data and trigger forecast workflows.

The events processing is carried out as a sequence of operators where events are pushed from downstream operators to upstream operators. This is best illustrated through an example as given in Fig. 5. NEXRAD Doppler WSR 88-D data is ingested at the events processing service, shown in the left of the diagram, marked with the timestamp at which the data were generated at the instrument. The sensed data for the radars falling within the region selected by the meteorologist are passed on to an "aggregate" node where they are fused based on their timestamp. These event snapshots are passed to a parallel classification operation that executes one instance of the detection algorithm such as SDA per radar stream. The final node merges the results, and applies the SCA to determine the relevant storm clusters and their spatial bounds (as seen in Fig. 4). For each storm cluster, a notification event is generated and sent to the LEAD-CI workflow engine, which triggers a latent weather forecast workflow. The newly generated information can then be used in subsequent executions of the query. The latter is shown as a stream of derived data feeding back into the first operator.

The model we propose here for bridging the observational data from sensor networks and instruments into a SOA such as the LEAD-CI is through an events processing system which combines filtering, aggregating, and temporal joins with sophisticated data mining algorithm libraries. These systems are built for the continuous processing required by constantly arriving data. The interoperability between the events processing system and the remaining SOA is achieved by means of "triggers" that the workflow engine can understand and react to. Triggers are delivered on a common enterprise service bus.

Summary and future work

Information technology research has given rise to cyberinfrastructure frameworks that serve the needs of next generation computational science knowledge discovery. Challenges exist in making these frameworks responsive to the physical environment around them. While workflow engines are quite adept at executing known sequences of tasks, to date they lack responsiveness to the dynamic behavior required to respond to weather events as they occur. We propose in this paper a model and algorithms for bridging the gap between the physical environment and the cyberinfrastructure framework by means of an events processing approach to responding to anomalous behavior and sophisticated data mining algorithms that apply

classification techniques to the detection of severe storm patterns. The ideas presented in this paper have been implemented in the LEAD-CI prototype.

As with any research, several interesting research questions both at the system and the mining algorithm level were discovered while designing and developing this prototype. At the systems level, questions for future work focus on finding the optimal frequency of executing the clustering algorithm, on differentiating the triggers for storms for which the forecast runs has already been launched, and on sharing the results across users in order to limit redundant forecasts.

Our basic assumption to ignore the temporal aspect of weather events in order to simplify the mining algorithms has to be modified. Our future research will look at adding mining algorithms that can track the growth and the movement of the storms in both space and time. The other algorithm level research questions that require further research focus on exploring the performance of generic detection algorithms as compared to developing specialized mining algorithms per phenomenon. The cluster ranking metric is a simple statistical *Z*-score measure and has not been whetted by the science community and will require further investigation.

Acknowledgements This research was funded in part through Department of Energy DE-FG02-04ER25600, and through the National Science Foundation under Cooperative Agreements: ATM-0331594, ATM-0331591, ATM-0331574, ATM-0331480, ATM-0331579, ATM03-31586, ATM-0331587, and ATM-0331578. The authors acknowledge the numerous contributions and insights from members of the LEAD team.

References

Arasu A, Cherniack M, Galvez E, Maier D, Maskey A (2004) Linear road: a stream data management benchmark. VLDB Conference

Burgess DW (1976) Single Doppler radar vortex recognition. Part I: mesocyclone signatures. 17th Conf. on Radar Meteorology, Seattle, WA, Amer. Meteor. Soc., pp 97–103

Cranor C, Johnson T, Spataschek O, Shkapenyuk V (2003) Gigascope: a stream database for network applications. Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, pp 647–651

Crum TD, Alberty RL, Burgess DW (1993) Recording, archiving, and using WSR-88D data. Bull Amer Meteor Soc 74:645–653

Desrochers PR, Donaldson RJ Jr (1992) Automatic tornado prediction with an improved mesocyclone-detection algorithm. Weather Forecast 7:373–388

Donaldson RJ (1970) Vortex signature recognition by a Doppler radar. J Appl Meteorol 9:661–670

Droegemeier KK, Gannon D, Reed D, Plale B, Alameda J, Baltzer T, Brewster K, Clark R, Domenico B, Graves S, Joseph E, Morris V, Murray D, Ramachandran R, Ramamurthy M, Ramakrishnan L, Rushing J, Weber D, Wilhelmson R, Wilson A, Xue M, Yalda S (2005) Service-oriented environments in research and education for dynamically interacting with mesoscale weather. IEEE Comput Sci Eng 7:24–32

Ester M, Kriegel H-P, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of 2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD-96), Portland, OR, pp 226–231

Gannon D, Plale B, Marru S, Kandaswamy G, Simmhan Y, Shirasuna S (2007) Dynamic, adaptive workflows for mesoscale meteorology. In: Taylor IJ, Deelman E, Gannon DB, Shields M (eds) Workflows for e-science: scientific workflows for grids. Springer, New York , pp 129–145, Jan

Gartner (2007) Real time agility through event processing and business activity monitoring. In: Gartner event processing summit

Gonzalez RC, Woods RE (1992) Digital image processing. Addison-Wesley, USA

Hartigan JA (1975) Clustering algorithms. Wiley series in probability and mathematical statistics. Wiley, New York

Huang Y, Slominski A, Herath C, Gannon D (2006) WS-messenger: a web services based messaging system for service-oriented grid computing. 6th IEEE International Symposium on Cluster Computing and the Grid (CCGrid06)

LEAD: Linked Environments for Atmospheric Discovery Project (2007). [Available online from http://lead.ou.edu.]

Li X, Ramachandran R, Rushing J, Graves S, Kelleher K, Lakshmivarahan S, Douglas K, Jason L (2004) Mining NEXRAD Radar Data: an investigative study. Interactive Information and Processing Systems (IIPS), Seattle, WA, American Meteorological Society

Liu Y, Vijayakumar NN, Plale B (2006) Stream Processing in Data-driven Computational Science. 7th IEEE/ACM International Conference on Grid Computing (Grid'06), Barcelona, September

Mainwaring A, Culler D, Polastre J, Szewczyk R, Anderson J (2002) Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications. Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications, Atlanta, Georgia, USA, 88–97

Marzban C, Stumpf GJ (1996) A neural network for tornado prediction based on Doppler radar-derived attributes. J Appl Meteorol 35:617–626

Pielke R, Carbone R (2002) Weather impacts, forecasts, and policy: an integrated perspective. Bull Am Meteorol Soc 83(3):393–403

Rushing J, Ramachandran R, Nair U, Graves S, Welch R, Lin A (2005) ADaM: a data mining toolkit for scientists and engineers. Comput Geosci 31:607–618

Russell N, ter Hofstede AHM, van der Aalst WMP, Mulyar N (2006) Workflow control-flow patterns: a revised ViewBPM Center Report BPM-06-22

Stumpf GJ, Witt A, Mitchell ED, Spencer PL, Johnson JT, Eilts MD, Thomas KW, Burgess DW (1998) The national severe storms laboratory mesocyclone detection algorithm for the WSR-88D. Weather Forecast 13:304–326

Vijayakumar NN, Plale B, Ramachandran R, Li X (2006) Dynamic filtering and mining triggers in mesoscale meteorology forecasting. IEEE International Geoscience and Remote Sensing Symposium (IGARSS'06), Denver, CO, August

Zrnic DS, Burgess DW, Hennington LD (1985) Automatic detection of mesocyclonic shear with Doppler radar. J Atmos Ocean Technol 2:425–438