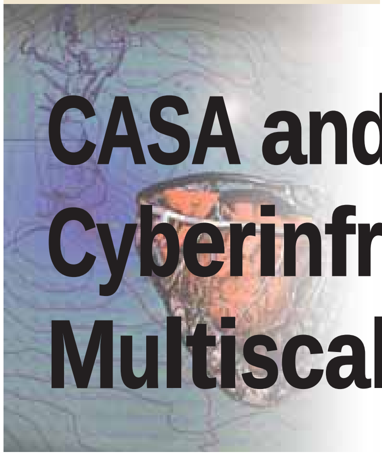
COVER FEATURE



CASA and LEAD: Adaptive Cyberinfrastructure for Real-Time Multiscale Weather Forecasting

Beth Plale and Dennis Gannon, Indiana University Bloomington; *Jerry Brotzge and Kelvin Droegemeier,* University of Oklahoma, Norman; *Jim Kurose and David McLaughlin,* University of Massachusetts Amherst; *Robert Wilhelmson,* University of Illinois at Urbana-Champaign; *Sara Graves,* University of Alabama in Huntsville; *Mohan Ramamurthy,* University Corporation for Atmospheric Research; *Richard D. Clark and Sepi Yalda,* Millersville University; *Daniel A. Reed,* University of North Carolina at Chapel Hill; *Everette Joseph,* Howard University; *and V. Chandrasekar,* Colorado State University

Two closely linked projects aim to dramatically improve storm forecasting speed and accuracy. CASA is creating a distributed, collaborative, adaptive sensor network of low-power, high-resolution radars that respond to user needs. LEAD offers dynamic workflow orchestration and data management in a Web services framework designed to support on-demand, real-time, dynamically adaptive systems.

Recent hardware and software advances are ushering in a revolution in meteorological research and forecasting. Whereas today's forecasts are generated on a fixed time schedule, new radar technologies as well as improved model physics are enabling on-demand forecasts in response to current weather events. These forecasts ingest regional atmospheric data in real time and can consume large computational resources in real time as well.

Two highly complementary projects, Collaborative Adaptive Sensing of the Atmosphere (www.casa.umass.edu) and Linked Environments for Atmospheric Discovery (http://lead.ou.edu), are developing a hardware and software framework to enable real-time multiscale forecasting. CASA and LEAD are stand-alone systems that offer distinct benefits to their respective user communities, but when used together, they represent a paradigm shift in atmospheric science in two respects:

- For the first time, meteorologists can directly interact with data from instruments as well as control the instruments themselves.
- Unlike traditional forecasting approaches, which generate static, fixed-cycle predictions, CASA and LEAD establish an interactive closed loop between the forecast analysis and the instruments: The data drives the instruments, which, to make more accurate predictions, refocus in a repeated cycle.

The "Hypothetical CASA-LEAD Scenario" sidebar provides an example of the unprecedented capabilities these changes afford.

Mesoscale meteorology is the study of smaller-scale weather phenomena such as severe storms, tornadoes, and hurricanes. System-level science in this context involves the responsiveness of the forecast models to the weather at hand as well as conditions on the network at large and the large-scale computational resources on which forecasts rely. This responsiveness can be broken down into four narrowly defined goals:

- *Dynamic workflow adaptivity*. Forecasts execute in the context of a workflow, or task graph. Workflows should be able to dynamically reconfigure in response to new events.
- *Dynamic resource allocation*. The system should be able to dynamically allocate resources, including radars and remote observing technologies, to opti-

Published by the IEEE Computer Society

0018-9162/06/$20.00 © 2006 IEEE

Hypothetical CASA-LEAD Scenario

Daytime surface heating on a late spring afternoon in southwestern Oklahoma creates a wind convergence zone at the boundary between ripening winter wheat and adjacent plowed fields. CASA feeds real-time data collected by small-scale radars scanning in general surveillance mode to weather-detection algorithms that classify the convergence zone with respect to location, orientation, depth, and intensity. It combines this information with other data—including user priorities for controlling the radars at that particular moment, the radar network's geometry, local terrain parameters, and input from National Weather Service (NWS) next-generation radar satellites—to yield an optimal remote-sensing configuration for the next 30 seconds that lets the radars zero in on the wind convergence zone.

As real clouds and precipitation develop within the surface convergence zone, the radars adjust their mode of operation and, via the optimization system, produce extremely fine-scale, calibrated precipitation rate estimates. CASA makes this information available to a private regional company that specializes in hydrologic models and stream-flow decision support systems, as well as to a university researcher in California. The researcher configures a LEAD workflow to run a forecast model in a triply nested mode that can predict weather at scales ranging from the continental US down to a few kilometers surrounding the storms. She searches the community data collection registry and her own personal space for data products that will strengthen the forecast accuracy.

The researcher sends output from a 100-member parallelized forecast run to her personal workspace for further refinement through data mining. She also processes the output within another workflow to generate probabilistic forecasts that, when combined with observations and analyses, yields statistically reliable conditional probabilities in 1,000 categories—for example, a probability of precipitation greater than 0.5 inches in 1 hour. The researcher feeds this information into a power company's off-site, proprietary risk assessment model to which she and her students have access as part of a summer project. If the probability of the forecast exceeds a predetermined risk-assessment threshold, the power company automatically reduces energy production at a hydroelectric plant downstream.

The researcher can leverage the combined systems to respond to this emerging severe weather condition in many ways. As the storms move, she can change the grid spacing in the forecast model to maintain fine resolution and keep the features of interest as far away as possible from the nested grid boundaries. As computing resources become saturated for this on-demand application, LEAD moves one of the domains previously computed on supercomputers in Illinois to Pittsburgh and automatically allocates the bandwidth needed to transfer the associated data. The researcher then decides to add three more domains and to launch an ensemble of 10 runs, requiring that the three-hour forecasts finish in 10 minutes. When one of the processors begins to overheat, LEAD detects this and automatically shifts the affected model components to other available resources.

While all of this is occurring, the detection of a small circulation within one of the storms triggers the two radars nearest the circulation to hand off tracking responsibility to neighboring radars as the evolving tornado progresses eastward. CASA automatically reports the tornado's location, intensity, movement, and projected path via wireless links to the NWS, local media outlets, and emergency managers and publishes them to the community metadata catalogues. When the tornado destroys two CASA radars within the network and disrupts local communication links, other nearby radars assume responsibility, via automated fault-tolerant software both at the data transport and application levels, and the network reroutes local traffic to ensure quality of service.

mize data collection as well as request vast amounts of computational resources on the fly.

- *Continuous feature detection and data mining.* The system should be able to continuously detect significant weather features and mine data from observational instruments to refocus detection efforts.
- *Model adaptivity.* Computational models often run for one to two hours or longer. Ideally, the models themselves should respond to changes in atmospheric conditions during execution.

A fully adaptive framework could not be realized within LEAD or CASA alone; it is the combination of the two projects that makes achieving these goals possible.

CASA-LEAD CYBERINFRASTRUCTURE

Figure 1 on the next page shows how CASA and LEAD cooperate to meet the multiple and often conflicting needs of users. CASA comprises an observational loop with data streams linking the radars to the *meteorological command and control* (MC&C) module,[1] which in turn generates control messages back to the radars. LEAD comprises a modeling loop, which executes forecast models in response to weather conditions. It requires data storage tools to automate data staging and collection and monitoring tools to enhance reliability and fault tolerance. LEAD has a path back to the radars for steering the radar location, while CASA mediates potential conflicting interests in determining the next radar position.

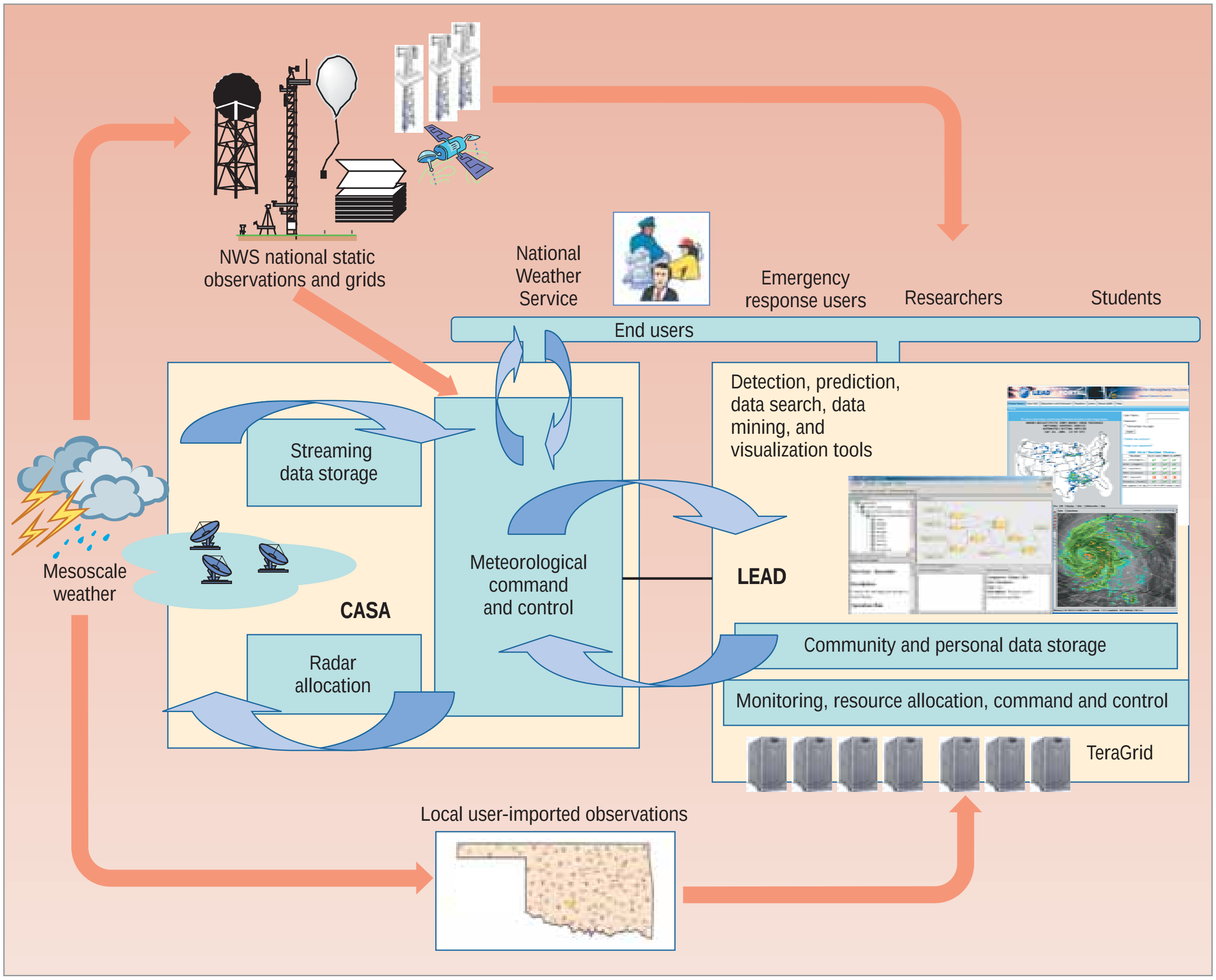

NWS national static observations and grids
National Weather Service
Emergency response users
Researchers
Students
End users
Detection, prediction, data search, data mining, and visualization tools
Streaming data storage
Meteorological command and control
LEAD
Mesoscale weather
CASA
Radar allocation
Community and personal data storage
Monitoring, resource allocation, command and control
TeraGrid
Local user-imported observations

Figure 1. CASA-LEAD interaction. The two systems cooperate to meet the multiple and often conflicting needs of users.

CASA overview

CASA is designed to detect and predict hazardous weather in the lowest few kilometers of the earth's atmosphere using distributed, collaborative, adaptive sensing (DCAS).[2]

- *Distributed* refers to the use of numerous small and inexpensive radars, spaced near enough to "see" close to the ground despite the earth's curvature and thereby avoid resolution degradation due to radar-beam spreading.
- *Collaborative* refers to the coordination of beams from multiple radars to view the same region in space, thus achieving greater sensitivity, precision, and resolution than is possible with a single radar.
- *Adaptive* refers to the ability of these radars and their computing/communications infrastructure to dynamically reconfigure in response to changing weather conditions. Rather than "pushing" data from radars, DCAS "pulls" data, allowing the system to allocate resources to best meet user needs.

CASA's first network of radars, NetRad, is a prototype DCAS system currently deployed in southwestern Oklahoma. As Figure 2 shows, the initial testbed consists of four mechanically scanning X-band radars with overlapping footprints. Other principal DCAS components include meteorological algorithms that detect, track, and predict hazards; user preference and policy modules that determine the relative utility of performing alternative sensing actions; the MC&C, an underlying substrate of distributed computation, communication, and storage that dynamically processes sensed data and manages system resources;[1] and control interfaces that let users access the system.

The MC&C currently executes on a cluster of three Intel Xeon processors with an attached 3.6 Tbytes of IDE RAID storage. As winds, precipitation, and other detected features unfold, the MC&C posts these on the *blackboard*;[3] this feature repository decouples the data ingested from the radar response, allowing radar commands to be generated asynchronously. The MC&C's task-generation component in turn takes features from

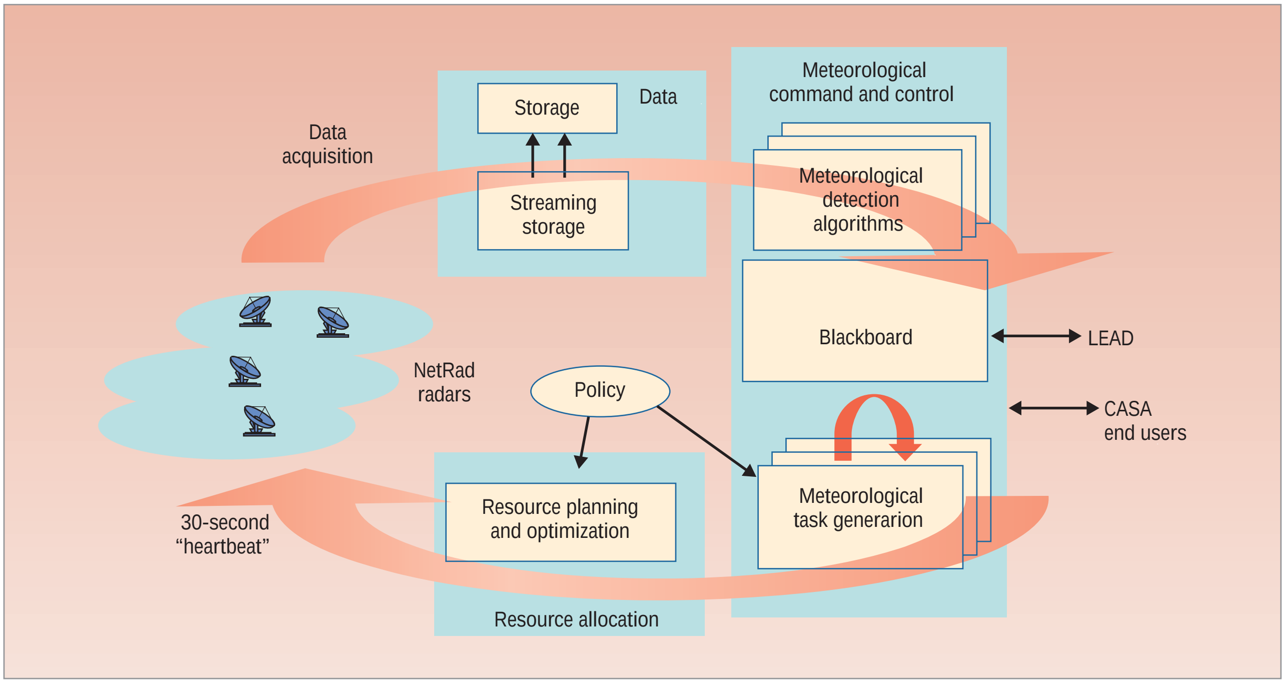

Data acquisition
Storage
Data
Streaming storage
Meteorological command and control
Meteorological detection algorithms
Blackboard
LEAD
CASA end users
NetRad radars
Policy
Resource planning and optimization
Meteorological task generarion
30-second "heartbeat"
Resource allocation

Figure 2. CASA NetRad architecture. (Upper loop) Reflectivity and wind velocity data flow from the distributed radars, through the network, to storage and meteorological feature detection algorithms, which post detected features on the blackboard. (Lower loop) NetRad responds to posted meteorological features by redirecting the radars to focus on users' greatest needs.

the blackboard and produces a list of tasks, with an associated utility.

NetRad's optimization routines scan the list of tasks and redirect the radars to focus on regions of interest during the next system cycle in a configuration that maximizes overall utility. To avoid stalling due to unanticipated network and processing delays, NetRad only acts upon data already posted on the blackboard; it waits until the next cycle to generate tasks based on late-arriving data.

A scan strategy incorporates preferences from multiple users—including emergency response managers, next-stage consumers of NetRad data such as LEAD, and meteorological researchers—as well as radar-specific considerations such as how well a certain radar can observe a particular object. NetRad's policy rules determine the relative weights of these competing needs.[1]

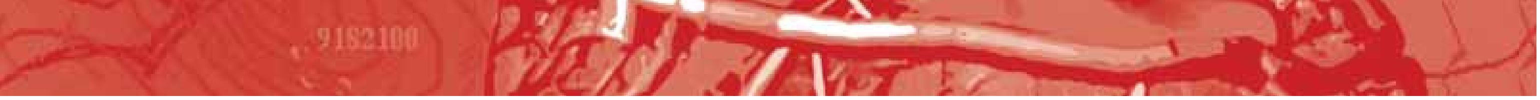
LEAD overview

LEAD is developing the middleware that facilitates adaptive utilization of distributed resources, sensors, and workflows. Constructed as a service-oriented architecture,[4] the system decomposes into services that communicate with one another via well-defined interfaces and protocols. LEAD uses SOA concepts at both the application and middleware level, and its grid has about 20 distinct services that interoperate in various ways.

For example, the Analysis Regional Prediction System (ARPS) Data Assimilation System has been decomposed into individual services in which a Web services tool wraps the original Fortran code.[5] The Algorithm Development and Mining system[6] has similarly been "service-ized" into component services. ADaM uses decision trees, neural networks, pattern-recognition algorithms, and knowledge discovery techniques to, among other things, identify rotating storms in data streaming from radars.

Data subsystem. The meteorology community requires access to numerous observational and model-generated data collections—including from Geostationary Operation Environmental Satellite systems, upper-air balloons, Meteorological Aerodrome Report observations, and next-generation radar (NEXRAD) Level II and Level III Doppler systems. The LEAD data subsystem consists of about a dozen services that provide online data mining and filtering of streaming data in support of on-demand forecast initiation, indexing and accessing of heterogeneous community and personal collections, personal workspace management, querying of rich domain-specific metadata utilizing ontologies, and automated metadata generation.

Tools. Community tools simplifying human access to these collections are the foundation of LEAD. These tools include Unidata's Internet Data Distribution (www.unidata.ucar.edu/software/idd/iddams.html), Thematic Real-Time Environmental Distributed Data Services (www.unidata.ucar.edu/projects/THREDDS), and Local Data Manager (www.unidata.ucar.edu/software/ldm), as well as the Open Source Project for a



Network Data Access Protocol (OPeNDAP; www.opendap.org).

Search support. LEAD provides search support over heterogeneous collections as well as concept-based searching. These features require advances in metadata schema definition, ontologies, automated metadata generation, vocabularies, and efficient XML support. For example, the LEAD Metadata Schema[7] is a semantic, domain-specific XML schema that extends the Federal Geographic Data Committee (www.fgdc.gov/standards) scheme and borrows from the Earth Science Markup Language[8] and THREDDS schemas.

Search attributes also can include more generic terms such as "precipitation." The Noesis ontology service, based on the Semantic Web for Earth and Environmental Terminology ontology (http://sweet.jpl.nasa.gov), resolves concepts into primitive terms that the system can then use to search the archives. Community data product holdings currently reside primarily in OPeNDAP servers that THREDDS catalogues index, while users implement their personal workspace using the myLEAD metadata catalogue[9] and supporting back-end storage.

Resource allocation in CASA involves the dynamic, collaborative tasking and retasking of radars to meet user needs.

User interface. Users access the rich suite of LEAD tools and services through the LEAD Portal (https://portal.leadproject.org/gridsphere/gridsphere). In the hypothetical weather scenario described in the sidebar, the researcher logs into the LEAD Portal and searches for streaming CASA data. Using this and a predefined workflow from her workspace,[9] she launches a request for on-demand resources on the TeraGrid[10] to generate a 1-km fine-scale gridded analysis to use as initial conditions for a high-resolution, 100-member Weather Research and Forecasting[11] model ensemble.

A monitoring toolkit based on SvPablo,[12] workflow tools based on the Business Process Execution Language for Web Services v1.1 (www-128.ibm.com/developerworks/library/specification/ws-bpel), and a network communication protocol, Web Services Eventing (www-128.ibm.com/developerworks/library/specification/ws-eventing), underlie the activity. Our WS-Eventing implementation is a content-based publish/subscribe protocol with persistent memory ferrying events from one part of the system to another in a broadcast fashion.

DYNAMIC WORKFLOW ADAPTIVITY

CASA uses a blackboard-based framework[3] in which cooperating agents post messages in response to a current situation or problem. Agents watch the blackboard to see if there is something within their domain of expertise that they can address. An agent can select one or more "facts" from the blackboard and propose a theory about the facts that leads to some conclusion. The agent can record this result on the blackboard to replace or augment the original facts. A new fact can trigger other fact-gathering agents to add more results.

The goal of a LEAD workflow is to carry a prediction scenario from the gathering of observation data to user visualization. A typical workflow would start with a signal from a data mining service or a signal from the CASA system indicating possible storm formation in a particular region. The workflow that this event triggers will usually begin with a set of data assimilation tasks that incorporate current weather data in the region of interest into the form the simulation service needs.

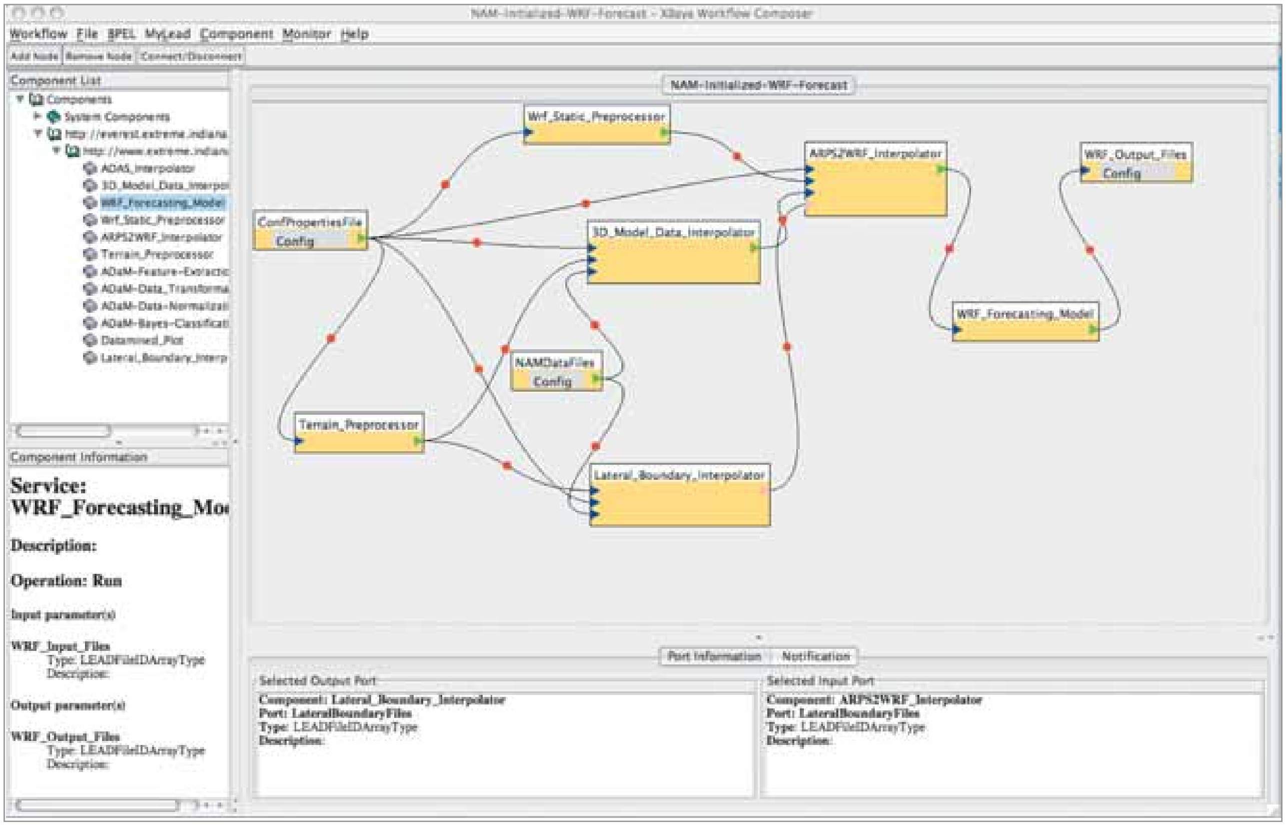
Scientists use a graphical tool to create a LEAD workflow control program. As Figure 3 shows, the workflow's individual component nodes are Web services that execute specific data analysis and simulation programs deployed on various hosts. The graph depicts data and control dependencies of the execution of the services representing the nodes. The graphical tool is actually a compiler that generates a standard BPEL4WS script that is the executable form of the workflow.

The unique way in which services notify the workflow control program instance that they have completed execution closely resembles the blackboard-based CASA workflow. Each service publishes a stream of events, based on WS-Eventing, to an event "bus" or channel. Various LEAD components, including each running workflow instance, subscribe to these events. When a workflow invokes a service, the invocation includes a *globally unique identifier* for the associated computation experiment. As the service executes the stream of events, it publishes all details about data products created by that execution and tags those events with the UID. The workflow instance learns about newly created data products through the event stream because it subscribes to all events having that UID.

Another benefit of this event-driven model of execution is the recording of a workflow's event stream in the LEAD data services. This makes it easier to debug workflows and provides a complete history of each data product's execution that is part of the user's computational experiments.

DYNAMIC RESOURCE ALLOCATION

Resource allocation in CASA involves the dynamic, collaborative tasking and retasking of radars to meet user needs. The initial system accomplishes retasking on a 30-second "heartbeat" interval based on the mechanically scanning radars' physical properties and the timescale over which atmospheric conditions change. As CASA's



Service:
WRF_Forecasting_Mo

Description:

Operation: Run

Input parameter(s)

WRF_Input_Files
Type: LEADFileIDArrayType
Description:

Output parameter(s)

WRF_Output_Files
Type: LEADFileIDArrayType
Description:

Component: Lateral_Boundary_Interpolator
Port: LateralBoundaryFiles
Type: LEADFileIDArrayType
Description:

Component: ARPS2WRF_Interpolator
Port: LateralBoundaryFiles
Type: LEADFileIDArrayType
Description:

Figure 3. Sample LEAD workflow. The control program feeds input data into terrain and surface data preprocessors. The terrain data preprocessor sends output to a 3D model data interpolator, while the surface data preprocessor generates lateral boundary conditions. All of these results are derived data products used by the ARPS forecasting program, which sends its output to two other services that generate the visualization.

radars evolve to rapidly reconfigurable, solid-state radars, we expect to relax the notion of a system heartbeat.

Dynamic resource allocation in LEAD takes the form of a very distributed Web services infrastructure to manage data repositories and launch jobs on large computer resources. Because of the better-than-real-time requirements of storm forecasting, the needed resources must be available on demand. If a workflow requires a large ensemble of simulations to launch, it is imperative to locate enough computational power to run the simulations as quickly as possible. TeraGrid researchers are working on procedures that will allow on-demand scheduling of large computation under emergency situations such as an impending tornado or hurricane.

However, LEAD also requires a strategy that lets workflows change how they use resources. For example, as a storm progresses, it might be necessary to dynamically change the workflow and launch additional simulations that were not initially anticipated. The LEAD workflow model's event-driven architecture is the key to creating workflows that adaptively respond to the weather and resource availability.

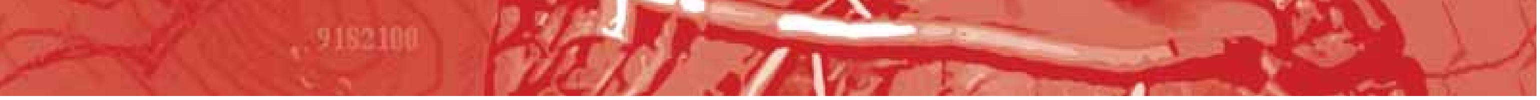
Computational resource allocation in a dynamic environment occurs following submission of a job and requires monitoring for adjustments as the task runs. LEAD uses a monitoring architecture from the University of North Carolina's Pablo[12] infrastructure that checks network bandwidth and processor instrumentation and provides a detailed view of the state of the computing hardware, including processor temperature sensors, to detect load and possible failure conditions. LEAD can summarize monitoring data and publish it to an event stream that provides status information to resource brokers. LEAD will use resource brokers from the Virtual Grid Application Development Software (VGrADS) project (http://vgrads.rice.edu).

CONTINUOUS FEATURE DETECTION AND DATA MINING

CASA continuously extracts features from data that weather-observing instruments gather, while researchers can use LEAD to dynamically mine such data to refocus detection efforts.

Feature detection

CASA converts arriving NetRad data, including reflectivity and wind velocity values, to a common format and stores it. Using the Warning Decision Support System—



Integrated Information (WDSS-II) software's linear buffer publish/subscribe mechanism, the MC&C distributes this data among various feature-detection modules.[13] Upon activation, these modules detect spatially coherent meteorological "objects" in the data such as precipitation areas, cloud systems, fronts, and pressure centers.

The MC&C writes the radar data as well as higher-level features into the blackboard, which makes meteorological objects available for delivery and display to users.[3] The blackboard can also hand off NetRad data to next-stage components such as LEAD; such components can likewise provide inputs to the blackboard. Model-driven prediction components external to NetRad can thus be easily integrated with NetRad's short-term detection components. The system also can store and merge assimilated exogenous data—for example, from NEXRAD radars and satellites—with NetRad-generated data.

Ultimately, adaptivity in CASA and LEAD must extend beyond the cyberinfrastructure and instruments to include computational models.

Data mining

In LEAD, users submit custom data-mining requests that run on their behalf for a specified period of time looking for weather activity over a particular geospatial region. Detection techniques range from using a simple threshold measure of echo intensity returned from a radar to complex clustering algorithms that identify couplets of incoming and outgoing radial velocities in NEXRAD radar data as mesocyclone signatures.[6]

The client view is of a Web service that provides SQL-like query access to a virtual collection of data streams. A system administrator sets up multiple *virtual stream stores* for an application and registers instrument streams to one or more stream stores. Clients then discover stream stores using standard UDDI-like Web service discovery techniques and submit time-bounded queries to the stream store such as, "Watch the Chicago region for the next four hours looking for mesocyclone signatures, and trigger workflow 'myforecast' if the signature threshold exceeds *x*." The queries execute continuously on a user's behalf for a bounded time period. Queries generate either a stream of results or a single trigger upon detecting behavior of interest.

The Calder data stream processing system[14] implements the virtual stream store abstraction, while the domain data-mining algorithms are part of the ADaM[6] toolkit. Calder manages multiple data stream stores simultaneously, tracking the active queries, streams, and computational nodes on which the queries are executed. It also tracks the provenance of streams, enabling users to track new streams generated as a result of query execution back to the original streams and queries that caused the creation of the new streams.

Calder has a 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. Data streams from instrument sources enter Calder through a point-of-presence gateway that maps the flows from their native dissemination protocol to an internal binary-based publish/subscribe system.

Calder subscribes to the LEAD notification system; when a query generates a trigger, that trigger goes to a gateway node that maps from the Calder publish/subscribe protocol to the LEAD notification protocol. The LEAD workflow engine is listening on this channel, so it can react when a trigger is received.

MODEL ADAPTIVITY

The science of adaptivity in large-scale simulations of complex nonlinear systems is rapidly developing, and emerging technologies will be an essential component of future mesoscale weather forecasting. Ultimately, adaptivity in CASA and LEAD must extend beyond the cyberinfrastructure and instruments to include computational models that make it possible to refine forecast simulation grids over specified regions.

This can be accomplished by launching an entirely new forecast simulation at finer spacing (workflow adaptation) or creating a nested grid within the simulation itself (application adaptation). This seemingly simple approach poses major optimization challenges with respect to potentially competing strategies for assessing the trigger condition and effectuating adaptation, and continuing the "parent" forecast run to provide boundary conditions for the next domain versus running a one-way nest. More sophisticated analysis might involve using multiple nests, adding ensembles based upon a specified condition, and using severe weather precursors such as tornado watches as a trigger to launch a coarse-grid background forecast in preparation for finer-grid nests.

Software systems that monitor and control weather-observing instruments and ingest and analyze their data are a key component of the national cyberinfrastructure needed to guide the response to disasters that result in loss of life and property. To do their job, these systems must be responsive to real-world data and agile in carrying out the computations needed to respond intelligently within a short period.

CASA and LEAD each offer highly adaptive capabilities, but the systems are far from complete. The first end-to-end CASA testbed radar system in Oklahoma is scheduled to begin operation later this year. CASA collaborators will operate other testbeds in Colorado and Puerto Rico. LEAD is at least one year from realizing the vision of fully dynamic workflows, but much of the infrastructure is already in place. A major outstanding component is resource brokers that can dynamically allocate time on supercomputers on demand. LEAD is working with the VGrADS and other TeraGrid partners who will supply these components.

Looking forward, the next step is to evaluate the combined CASA-LEAD system for its ability to meet system-level end-to-end policies drawn from the extensive use cases we have gathered. Integration of the two systems will take place over the next two years, but researchers have already used the architecture built thus far in numerous experiments. For example, complete simulations involving Hurricane Katrina data demonstrate that the LEAD distributed data, event, and application services framework is robust and operates correctly over resources involving TeraGrid computers and the LEAD testbed of machines and data archives distributed over five states.

Together CASA and LEAD represent an instance of systems science that is expected to change the paradigm for severe storm prediction and basic atmospheric science from a model based on static, fixed-schedule forecasts to one based on an adaptive response to weather. Closing the loop between the instruments and the analysis and simulations should enable more timely forecasts and provide more accurate information to policy makers, thereby saving the lives of those in the path of storms. ■

Acknowledgments

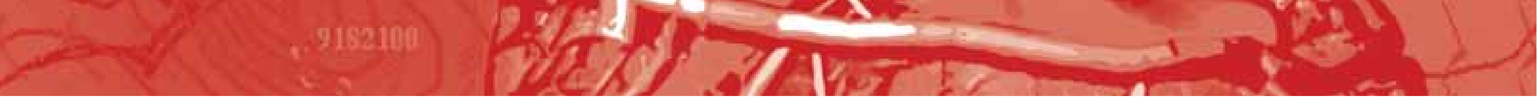
CASA is supported in part by the Engineering Research Centers Program of the National Science Foundation under NSF Cooperative Agreement no. EEC-0313747, and LEAD is supported by Cooperative Agreements ATM-0331594, ATM-0331591, ATM-0331574, ATM-0331480, ATM-0331579, ATM-0331586, ATM-0331587, and ATM-0331578. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the National Science Foundation.

References

1. M. Zink et al., "Meteorological Command and Control: An End-to-End Architecture for a Hazardous Weather Detection Sensor Network," *Proc. Workshop on End-to-End, Sense-and-Response Systems, Applications, and Services,* Usenix Assoc., 2005, pp. 37-42.
2. D.J. McLaughlin et al., "Distributed Collaborative Adaptive Sensing (DCAS) for Improved Detection, Understanding, and Predicting of Atmospheric Hazards," *Proc. 9th Symp. Integrated Observational and Assimilation Systems for the Atmosphere, Oceans, and Land Surfaces,* American Meteorological Soc., 2005; http://gaia.cs.umass.edu/networks/papers/casa_san_diego_2005.pdf.
3. V. Jagannathan, R. Dodhiawala, and L.S. Baum, *Blackboard Architectures and Applications,* Academic Press, 1989.
4. D. Gannon et al., "Service-Oriented Architectures for Science Gateways on Grid Systems," B. Benatallah, F. Casati, and P. Traverso, eds., *Proc. 3rd Int'l Conf. Service-Oriented Computing,* LNCS 3826, Springer, 2005, pp. 21-32.
5. G. Kandaswamy et al., "Building Web Services for Scientific Applications," *IBM J. Research and Development,* vol. 50, no. 2/3, 2006, pp. 249-260.
6. J. Rushing et al., "ADaM: A Data Mining Toolkit for Scientists and Engineers," *Computers & Geosciences,* vol. 31, no. 5, 2005, pp. 607-618.
7. R. Ramachandran et al., "LEAD Metadata Schema for Geospatial Data Sets Based on FGDC Standard," 27 Jan. 2005; www.unidata.ucar.edu/projects/LEAD/FinalSchema.doc.
8. R. Ramachandran et al., "Earth Science Markup Language (ESML): A Solution for Scientific Data-Application Interoperability Problem," *Computers & Geosciences,* vol. 30, no. 1, 2004, pp. 117-124.
9. B. Plale et al., "Active Management of Scientific Data," *IEEE Internet Computing,* vol. 9, no. 1, 2005, pp. 27-34.
10. D.A. Reed, "Grids, the TeraGrid, and Beyond," *Computer,* Jan. 2003, pp. 62-68.
11. J. Michalakes et al., "Design of a Next-Generation Regional Weather Research and Forecast Model," W. Zwieflhofer and N. Kreitz, eds., *Towards Teracomputing,* World Scientific, 1999, pp. 117-124.
12. D.A. Reed et al., "Scalable Performance Analysis: The Pablo Performance Analysis Environment," *Proc. Scalable Parallel Libraries Conf.,* IEEE CS Press, 1993, pp. 104-113.
13. K.D. Hondl, "Capabilities and Components of the Warning Decision and Support System—Integrated Information (WDSS-II)," *Proc. 19th Conf. Interactive Information and Processing Systems,* American Meteorological Soc., 2003; http://ams.confex.com/ams/pdfpapers/57283.pdf.
14. Y. Liu, N.N. Vijayakumar, and B. Plale, "Stream Processing in Data-Driven Computational Science," to appear in *Proc. 7th IEEE/ACM Int'l Conf. Grid Computing,* IEEE CS Press, 2006.

Beth Plale is an associate professor at Indiana University Bloomington, where she also directs the Data and Search Institute. She received a PhD in computer science from Binghamton University, State University of New York. Contact her at plale@cs.indiana.edu.

Dennis Gannon is a professor of computer science at Indiana University Bloomington. He received a PhD in computer science from the University of Illinois at Urbana-Champaign and a PhD in mathematics from the University of California, Davis. Contact him at gannon@cs.indiana.edu.



Jerry Brotzge is a senior research scientist at the Center for the Analysis and Prediction of Storms (CAPS) at the University of Oklahoma, Norman, and CASA's director of NetRad operations. He received a PhD in atmospheric science from The University of Oklahoma, Norman. Contact him at jbrotzge@ou.edu.

Kelvin Droegemeier is Regents' Professor of Meteorology, the Roger and Sherry Teigen Presidential Professor, associate vice president for research, and director of CAPS at The University of Oklahoma, Norman. He received a PhD in atmospheric science from the University of Illinois at Urbana-Champaign. Contact him at kkd@ou.edu.

Jim Kurose is a Distinguished University Professor in the Department of Computer Science at University of Massachusetts Amherst. He received a PhD in computer science from Columbia University. Contact him at kurose@cs.umass.edu.

David McLaughlin is a professor of electrical engineering in the College of Engineering at University of Massachusetts Amherst. He received a PhD in electrical engineering from University of Massachusetts Amherst. Contact him at dmclaugh@ecs.umass.edu.

Robert Wilhelmson is a professor in the Department of Atmospheric Sciences at the University of Illinois at Urbana-Champaign, where he also serves as chief scientist of the National Center for Supercomputing Applications. He received a PhD in computer science from the University of Illinois at Urbana-Champaign. Contact him at bw@ncsa.uiuc.edu.

Sara Graves is a professor in the Department of Computer Science at The University of Alabama in Huntsville, where she is also director of the Information Technology and Systems Center. She received a PhD in computer science from The University of Alabama in Huntsville. Contact her at sgraves@itsc.uah.edu.

Mohan Ramamurthy is director of the Unidata Program Center at the University Corporation for Atmospheric Research. He received a PhD in meteorology from The University of Oklahoma, Norman. Contact him at mohan@ucar.edu.

Richard D. Clark is chair of the Department of Earth Sciences and a professor of meteorology at Millersville University. He received a PhD in atmospheric science from the University of Wyoming. Contact him at richard.clark@millersville.edu.

Sepi Yalda is an associate professor of meteorology in the Department of Earth Sciences at Millersville University. She received a PhD in meteorology from Saint Louis University. Contact her at sepi.yalda@millersville.edu.

Daniel A. Reed is Chancellor's Eminent Professor as well as vice chancellor for information technology and CIO at The University of North Carolina at Chapel Hill, and director of the Renaissance Computing Institute. Reed received a PhD in computer science from Purdue University. Contact him at ran_reed@unc.edu.

Everette Joseph is an associate professor in the Department of Physics and Astronomy at Howard University, where he is also a principal investigator in the Center for the Study of Terrestrial and Extraterrestrial Atmospheres. Joseph received a PhD in physics from the University at Albany, State University of New York. Contact him at ejoseph@howard.edu.

V. Chandrasekar is a professor in the Department of Electrical and Computer Engineering at Colorado State University. He received a PhD in electrical engineering from Colorado State University. Contact him at chandra@engr.colostate.edu.