Cyber Infrastructure for Community Remote Sensing

Arcot Rajasekar[1], Reagan W. Moore[1], Mike Wan[2], Wayne Schroeder[2]
Data Intensive Cyber Environments Center
[1]University of North Carolina at Chapel Hill
Chapel Hill, NC USA 27599-3360
[2]University of California at San Diego
La Jolla, CA USA 92093-0505
{sekar,moore,mwan,schroeder}@diceresearch.org

Introduction: Community Remote Sensing (CRS) is an emerging field where information is collected about the environment by the general public and then integrated into collections to provide a holistic view of the environment with local details. Citizen scientists may use sophisticated sensors and tools to collect increasingly precise information about our environment. Such holistic views can serve as ground truthing for information collected by traditional sources such as satellites and deployed sensor systems. With social networking tools and crowd sourcing technologies, the data collected by the CRS systems can grow exponentially. One of the challenges of the CRS community is the problem of how to manage such data in a coherent manner such that it can enable new science and aid decision making. In this paper, we propose a cyber infrastructure that can be deployed for CRS systems that is scalable and that can organically grow to meet the needs of an expanding CRS community. We demonstrate the implementation challenges for a scalable data management system for CRS and the solutions to meet the challenges. We also propose an architecture based on the integrated Rule-Oriented Data Systems as an exemplar for the Community Resource Sensing cyber infrastructure (CRS-CI).

CRS-CI Challenges: Community-driven data collection can produce large amounts of environmental data (such as rainfall, temperature, humidity, water shed level, crop yields, etc.) including sensor-based point measurements, textual data capturing information in free form, photographic images and video. We expect these data to be distributed geo-spatially and contain metadata about the data collector, time information, and other contextual information that provide additional attributes about the collection process. As technology for data collection and data ingestion/acquisition becomes simpler and easier to use, more and more citizen scientists will be involved in data gathering which in turn will increase the collection size. We expect that in the near future, such collections will total 100s of Gigabytes to Terabytes in size with millions of files. Managing such distributed data collections requires facilities for 1) ingesting, 2) organizing, 3) storing, 4) discovery and access, 5) analysis, and 6) long term preservation. The CRS-CI needs to provide an overall framework for these capabilities. Specific requirements can be identified for a viable data management framework.

Since the data being collected are nation-wide (and possibly global-wide) one needs a distributed data collection environment. This is necessary not just for fault-tolerance and disaster-recovery but also to give the users (both data gatherers and data users) fast access to the system resources. Since the CRS community will be interested in collection of data for different disciplines (not just climate-related but also data for hydrology, plant biology, ornithology, etc), it may be necessary to have different instances of the cyber-infrastructure running for each of these sub-communities. One of the goals may be to provide easy federation of these diverse data collections to support inter-disciplinary research.

Since these data are being gathered by citizen scientists, it would be helpful and a motivating driver if they can see their results used in analysis and forecasting. The CRS-CI should provide tools that the citizen scientists can use to analyze their own data (along with others) and also tools for visualizing the data. Another important capability in the CRS-CI tool kit would be scientific workflow tools (such as Kepler) that can be used by data gatherers and users to analyze their data by chaining multiple analytical components and feeding them to appropriate visualization engines. Such workflows can be used for analysis of gathered data in real-time, can be run continually, and can be used for comparing current results with archived data. The analysis tools can also be used for fusing data of multiple types (say rainfall measurement along a river, and river depth measurements) to show a composite view of an emerging

disaster in a flood plain. Integrating data gathered by CRS with data from deployed sensor networks (by governmental agencies and research units) will also provide a more complete picture of the environment. Enabling such analysis and synthesis should be possible under the CRS-CI.

An important criterium for CRS-CI is validation and error-correction of CRS collected data. Metrics and processes for data quality assessments, correlation of data for validation purposes, and cross-checking with inter-disciplinary data will be of importance to document the usefulness of the results that can be inferred from the data gathered under CRS. Tools for performing such validation should be automatically applied on data ingestion. Moreover, the concept of validation-before-publication should be a policy that is advocated in CRS-CI, along with automated discipline-centric triage.

Since the data collected by CRS are of multiple types, they may need to be stored in heterogeneous systems from relational databases to file systems. Organizing this data is extremely important. Indeed as new data come in from different user bases, organizing the data into coherent groups will be harder but very much needed. Self-organizing data collections are needed, such that the characteristics of the data being ingested will "place" the data into the appropriate CRS-CI data organization. Policies for how data gets self-organized in such a diverse data space is a research topic that needs to be addressed. Similarly, the identification of relationships between data sets is also important and needs to be captured to help in analyzing the data in a meaningful manner. These relationships (similar to functional relationships between tables in relational data bases) need to be codified for usage.

Standardization also plays an important role in CRS-CI. Since data are being gathered by multiple persons, the vocabulary used for describing the data and the data units can lead to incompatibilities. Hence, one needs to ensure that self-consistent ontologies and reserved key words are used when ingesting data into the CRS-CI. This may require automatic form generation mechanisms. Also, the data values need to be converted to a standard set of units. Hence if the CRS-CI is standardized to ingest temperature data in Celsius units, then all temperatures need to be converted to this unit before storage and publication. Similarly date and time formats need to be standardized to minimize confusion and error.

CRS-CI Solution: The CRS system needs to be based on a cyber infrastructure that is robust and extensible and can meet the multiple challenges posed by the diverse data gathering and usage models. We propose a system that has the following principles as an ideal system that meets these challenges.

Scalable Federated Data Grid Architecture: Data Grids provide access to large collections of data whose sizes are measured in petabytes and hundreds of millions of files. A data grid provides access to geographically distributed heterogeneous data resources assembled from file systems, tape archives, relational databases, semi-structured data systems, video streaming systems, and sensor data streams. They support operations for sharing data across inter and intra-disciplinary groups without having to aggregate the data into a centralized data warehouse. Data grids provide a virtualized interface to applications, hiding the idiosyncrasies of the underlying infrastructure from users and applications. The data grids also provide a means for long-term preservation by implementing technology transparency. Federation of data grids allows for independent data grids to interact with each other based on trusted relationships and allow users from one data grid to access data from another data grid.

Semantics-enabled Discovery and Access: Data without metadata that defines the context are worthless for automatic sharing (for example try imagining the web without any search engines). Associating metadata to describe the content and context of the data provides a means of discovering relevant data for users and applications. Free form descriptions are one type of metadata that can be textually indexed and used for discovery (as done in the web). With scientific data a more robust semantic indexing is needed. This can be in the form of keyword-value-unit triplets that can define the properties of data (e.g. Measurement = temperature in Celsius) and describe the content and context of the data. Search on the metadata supports discovery of desired files in a collection. One can also associate structured metadata (relational) and semi-structured metadata (XML and RDF) to capture a more complex context. Searching for metadata will require its own engine relative to the type of metadata being stored. Querying across multiple types of metadata and joining the results in a meaningful way is still a research problem.

Policy-based Data Organization and Management: The CRS-CI system will cater to a wide-variety of data (both in discipline, context, format, gathering and usage) and the management and organization of this heterogeneity will bring its own challenges. One way to lessen the burden is to automate as much of the organization of the data collection and management of the data system as possible. Organization of data may need to be done on demand and through self-organizing management policies. Similarly administrative management of the system can be based on defined policies for retention, disposition, distribution, replication, and synchronization. The self-organizing policies and management policies can be captured as computer actionable rules and executed on demand. These rules will be akin to ECA-rules of active databases so that when a event occurs, the condition gets checked and appropriate rules are fired to perform the necessary actions. Capturing data organization policies and data system management policies in a CRS system will be a research problem that needs to be addressed. Our experience with iRODS system (described below) shows that this is feasible.

User-friendly workflow systems for analysis and synthesis: The data collected in a CRS system will be used in multiple ways – in near real time for analysis and forecasting, in delayed analysis for modeling, and in data minining for specific events. The CRS-CI should provide facilities for including services and tools for performing modeling, analysis, synthesis and data mining. The CRS-CI by itself may not have them in its core part, but should support plug and play mechanisms.

Virtualization and Standardization of all aspects of the CI: Standards for the CRS-CI system data collections are necessary for making it easier for discovery, organization and management. These can be captured as policies coded as ECA-type rules. Virtualization allows for hiding the intricacies of the CRS-CI from the users. Also by virtualizing the tools, users are not tied to physical characteristics of the data collection. Again, the metadata plays a vital role in virtualization.

The above five principles provide a guide the development of a solution for the challenges posed for implementing a cyber infrastructure for Community Remote Sensing. By providing workflow mechanisms and hooks for including tools and procedures, ingestion pipelines, quality control and integrity maintenance can be achieved. We next describe an architecture that is built on these principles.

CRS-CI Architecture: The iRODS (Figure 1) middleware [1] is an exemplar architecture for implementing the cyber infrastructure for the Community Remote Sensing data system. It encodes the five principles discussed above. It is a peer-to-peer data grid architecture that federates distributed and heterogeneous data resources (including files, web pages, tape archives and data bases) into a single logical file system (called the collection hierarchy) (Figure 2) and provides a modular interface to integrate new client-side applications as well as server-side data and compute resources. Developed by the Data Intensive Cyber Environments Center at the University of North Carolina at Chapel Hill and the University of California at San Diego, iRODS (Fig 1) is open source software that is used by multiple projects and organizations for their data management and data sharing functionality. iRODS also supports optimized protocols for transfer of files of different sizes as well as provides normal file system functionalities for user login, access control and creation, operation and management of hierarchical collections. iRODS also has a built-in metadata catalog called iCAT for managing state information. Because of this, the users get a logical view of the full data collection independent of the location, system type and access protocol. The metadata catalog also supports multiple user-defined and discipline-centric metadata formats including keyword-value-unit triplets, extensible relational schemas and XML metadata. Users can associate metadata to their data either by ingestion or through automated extraction with appropriate routines. iRODS also supports a rule engine that acts as a distributed workflow execution system. Rules are of the ECA-type and are fired based on events, on user command or at a periodic rate or in delayed fashion. The rules are used by data system administrators, data providers, data generators, and the system designer to implement management policies. The policies for data management as well as for error checking and data organization can be encoded as iRODS rules and executed at appropriate times. iRODS also provides services and management policy hooks for data ingestion, organization into collections, association of metadata and annotations, and publication and disposition of data. Also, one can associate different policy sets for different data collections leading to an extensible and flexible management system implementation.

Figures 3 and 4 shows some use cases for iRODS in large projects. For support of Community Remote Sensing, the architecture can be similar but the various policies and rules need to be encoded to deal with the requirements of the diversity of data providers and the usage models. Figure 3 shows an astronomy data grid using iRODS where telescopic data from Chile is ingested into the data grid and stored and accessed from the US. Figure 4 shows a data grid assembled for the Temporal Dynamics Learning Center (one of the six Science of Learning Centers funded by NSF) whose scientists are distributed all over the US and share multi-format data among themselves Because of data taken from human subjects the data grid needs additional requirements for access control for each collection. The policy-based management mechanism provides the necessary infrastructure to deal with file-specific access controls, while sharing data across a distributed group of people. Other projects using iRODS include the National Archives and Records Administration's Transcontinental Persistent Archive Testbed, the Australian Research Coordination Service and the Carolina Digital Repository.

Conclusion: We have proposed the need for a common architecture for the cyber-infrastructure that will be needed to cater to the needs of Community Remote Sensing systems. We have identified the challenges that such a cyber infrastructure (CRS-CI) has to meet and also proposed five principles as solutions to meet these challenges. Finally, we have described the integrated Rule Oriented Data System, a data grid middleware that is built upon these principles and provides an ideal and exemplar implementation for CRS-CI. In the future, we propose to implement a CRS-CI using iRODS and study the effectiveness of the system to meets the needs of the Community Remote Sensing paradigm.

Acknowledgement: The research in this paper is funded by the NARA supplement to NSF SCI 0438741, (2005-2008) and the NSF OCI-0848296 grant (2008-2012). Funding was also provided by NSF ITR 0427196 (2004-2007) and NSF SDCI 0721400 (2007-2010).

References:

[1] IRODS: integrated Rule Oriented Data System, https://www.irods.org/index.php. (This reference has links to several papers and tutorials on iRODS).

Figures 1, 2, 3, 4, denoting the iRODS architecture in the upper left, virtualization of collections in the upper right, an astronomy application in the lower left, and a cognitive science application in the lower right.