EarthCycle

A Flexible, Extensible System for the Collection, Integration, Analysis, and Curation of Geoscience Data

Prepared By:

The University of Washington
- eScience Institute
- Joint Institute for the Study of the Atmosphere and Ocean (JISAO)
- Applied Physics Laboratory, Polar Science Center

Incorporated Research Institutions for Seismology (IRIS)

IBM Corporation

Overview

Geoscientists today have at their disposal an unprecedented and growing wealth of data and data processing resources. These offer unprecedented opportunities for scientific discovery. Yet, this potential is unrealized due to challenges associated with the distribution of data and resources, the maturity and power of software tools, and the design of existing high performance computer systems.

Achieving the EarthCube goal of an "integrated system to access, analyze and share information that is used by the entire geosciences community"[1] will require transformative advances in approaches to data ***collection, integration, analysis***, and ***curation***. Success depends on identifying productive solutions from inside and outside the community while building on decades of effort in developing the existing community cyberinfrastructure.

This white paper will outline a powerful, accessible, flexible cyberinfrastructure plan built around best practices from industry and academia - a plan which exploits the strengths of the existing community software and systems, while anticipating the evolution of technology and user requirements.

Data Integration and Analysis

Geoscience data come from myriad sources across many domains. Sources include a wealth of simulation codes, developed over decades and accepted within their communities. Sensors located on satellites to the ocean floor provide ever more data in an ever expanding variety of formats. Expecting the geoscience community to abandon existing formats and conform to some standard is unrealistic, and probably undesirable. Yet the problem remains: For scientists to be productive, they must have a convenient means by which to integrate data from many sources.

A Common Data Language

Astronomy provides valuable guidance in terms of empowering scientists through a common, high level language for accessing complex data. The Sloan Digital Sky Survey (SDSS)[2] , a catalogue of more than 15TB, comprising more than half a billion objects, began supplying data to users more than a decade ago. Users ranging from grade school students to professional astronomers all interact with the system by way of standard Structured Query Language (SQL) commands. A library of query templates is available to users, flattening the learning curve and simplifying the process of extracting relevant data subsets for subsequent analysis. Similar approaches have been adopted in the life sciences where SQL is used not only for data integration and extraction, but also to drive complex analysis and even to manage the data production pipeline[3]. In the decade since Sloan first came online, SQL has been embraced in even the most demanding data management environments, such as the Large Hadron Collider Atlas experiment[4] and has become the data processing standard in industry and government.

Federated, not Unified

Unlike SDSS where data come from a single source and comprise a manageable data volume, an EarthCube system cannot, and probably should not, dictate data formats. Moreover, as we will discuss in more detail later, no single site can hope to maintain a complete copy of all

EarthCube data. Fortunately, the power, flexibility, and ease of use of SQL remain relevant through the use of database federation approaches.

Federation allows a database system to address arbitrary external datasets as if they were native database tables through the use of *wrappers*. This approach is available through the *engines* feature of the open source MySQL database, and has been adopted for the integration of data for *omics analysis pipelines[5]. Similarly, IBM's flagship DB2 relational database has a robust wrapper implementation. IBM has invested in the development of wrappers for various life science data sources[6] as part of a growing library of wrappers.

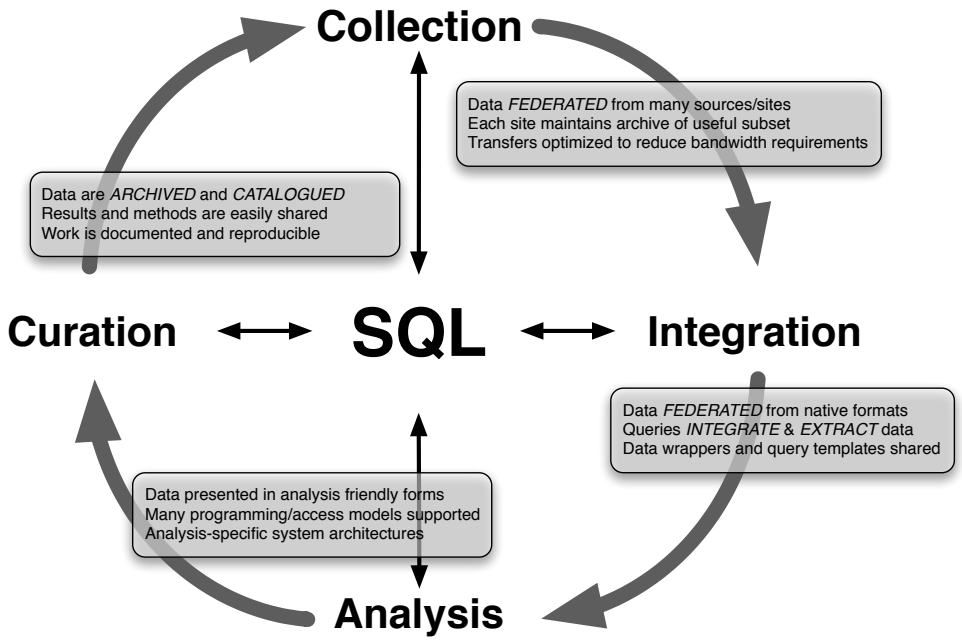

figure 1

The number of geoscience data types is large and growing, but not so large or growing so fast that the development of a library of wrappers is an intractable problem. And, unlike current ad hoc approaches to data integration in the geosciences, wrappers are reusable software. Adding a new data source in the same format as a previously used source requires no additional programming.

For example, a climate scientist interested in analyzing data from atmospheric, ocean, and land surface models, each conducted at multiple scales, along with data from a variety of terrestrial and space based instruments is confronted with an immense data management challenge before even beginning to do anything she would recognize as science.

Equipped with wrappers for the standard model and instrument data types, and a library of SQL query templates, this hurdle seems much more manageable. The user simply selects the desired data sources from a catalogue (the wrappers take care of making the data accessible to the SQL query engine), indicates the attributes of interest to her from each dataset and specifies

any scaling or other transformations necessary to align data from one source with the others (accomplished by either writing succinct SQL commands, selecting example commands from a library of templates, or some combination). The result is an **integrated subset**, **extracted** from the disparate input data by harnessing the database engine's ability to both integrate data and productively apply SQL operators to the integrated view.

Data Analysis System Architecture

Just as it is impractical and unwise to dictate data formats, EarthCube should embrace and encourage a variety of approaches to data analysis, ranging from HPC analytics, to SQL-based data warehousing, to distributed data intensive frameworks such as MapReduce. Regardless of the specific approach, the data to be analyzed must be accessible to the CPU resources in a useful fashion. Moreover, because the volumes of data are large and growing, working data sets must be ephemeral, with the input and results from each new analysis effort replacing those from previous efforts.

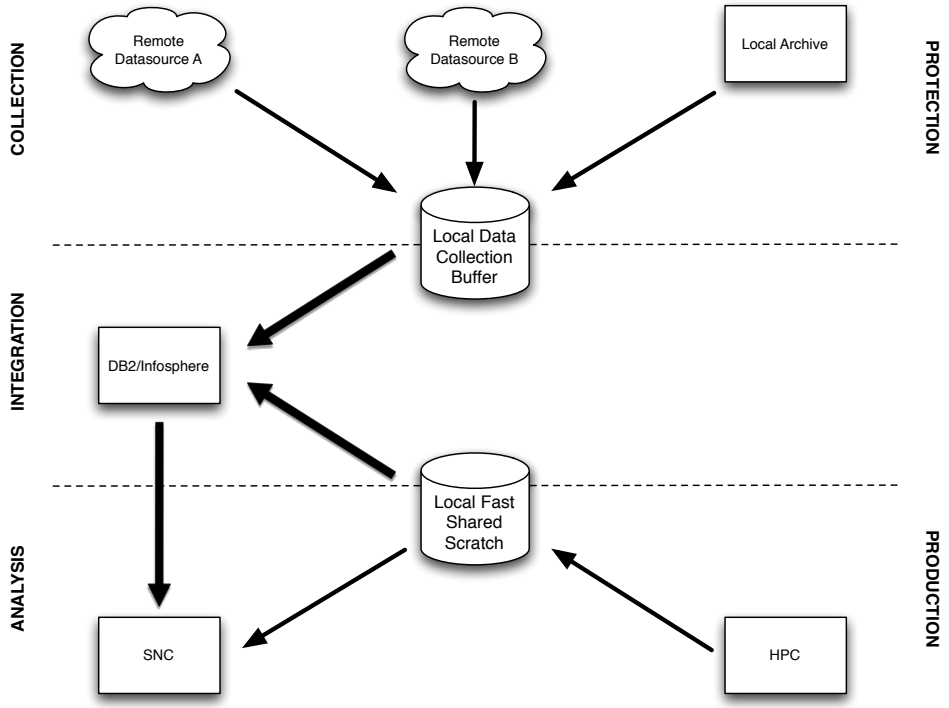

figure 2

Figure 2 illustrates an approach to system design which builds on existing cyberinfrastructure investments, while addressing EarthCube challenges. In this example, input data **collected** from remote sites, the local site's data archive, and produced by models run on the local HPC facility are **integrated** using the federation capabilities of a DB2/Infospehere server, with the results provisioned among the nodes of a DB2/GPFS shared nothing cluster (SNC) for **analysis**. GPFS (General Parallel File System) is the IBM file system capable of handling high

performance data I/O in parallel and its SNC version will extend the function over a collection of distributed nodes.

Data may be provisioned as a distributed SQL database, or as a set of XML or plain text files, and analysis may be performed using SQL data warehousing approaches, Hadoop, or any other approach which benefits from the attributes of a shared nothing cluster. Users also have the option of leaving the working dataset on the local fast shared scratch storage and performing their analysis entirely within the master DB2 server or leveraging the power of the site's existing HPC facility for HPC analytics.

The approach described here is flexible and portable. Sites may deploy some or all of the elements, depending on their requirements. The specific features of each element may also vary from site to site. For example, some sites may have neither an HPC facility nor an analysis-specific SNC, some may operate petascale, tape-based archives, while others have no archives at all, and some may choose to run MySQL on a single PC with lots of direct attached storage, while others may opt for DB2 configured to manipulate multi-TB datasets in-core.

Data Collection and Curation

Multiple supercomputer centers operated by multiple entities, each running multiple generations of equipment are now and will remain the reality of the high performance and scalable computing landscape. This ensures data producers (simulation output, intermediate data analysis products), and consumers (simulations, analysis) will forever diverge. Add to this the growing abundance of sensors and sensor networks, and two facts become obvious:

1) No single site can hope to maintain a complete and current copy of the community's data
2) Each site must collect (sometimes large) datasets from remote sites as needed

Too Much Data, Too Little Bandwidth

While the speed and power of data collection and processing equipment is advancing at the rate of Moore's Law, the networks connecting remote data sources and consumers have been more or less static for a decade with no sign of improving substantially any time soon. Some context: the current generation of Northwest regional climate models comprise ~40TB of data, the next generation of IPCC climate models are expected to produce tens of petabytes of data[7], yet the typical effective bandwidth between supercomputer centers in the NSF XSEDE system (formerly the Teragrid) has been stuck at < 100MBs throughout the same period[8].

Transferring 1PB at 100MBs requires 116 days. Even at 1GBs 12 days are required. Clearly, for geoscientists to be effective, the amount of data transferred must decrease while transfer speeds increase. Fortunately, the solution to the former may provide the means to achieve the latter, particularly if we recognize that different classes of data storage, with varying cost/performance attributes are appropriate for different tasks.

Distributed Archives, Local Caches

Today, data analysis often proceeds by the scientist first identifying data sources and downloading them, if necessary, to the site where the computation will occur, often writing

directly into high performance/high cost, relatively volatile, working disk. Imagine instead users at each computer center downloading the desired data into the site's data archive on cost effective storage. All subsequent uses of a data set downloaded by one user are then available to all subsequent users from the local archive copy rather than needing to be downloaded over the net. Transfer speeds between the local archive and working disk often exceed one Gigabyte/sec (1GBs) while storage costs are measured in pennies per GB/year. Over time, each site's local archive will grow, becoming a local cache of all data downloaded. The need for new transfers will never be eliminated, but they will be reduced.

Further, imagine a **global data catalogue** to which participating sites publish the contents of their archives. This enables a scheme where transfers may proceed in parallel from multiple secondary sources and are optimized to use the fastest available links from partners in the closest proximity. As archives at each site grow, the aggregate performance and reliability of the overall system improve, too. Of course, precedents exist for distributed, parallel archives with local caches. In fact, the BitTorrent network is built on these principles, and by many estimates it accounts for more than half of all Internet traffic[9]. The Tranche project[10] also embraces some of these concepts.

A Catalogue for Data, Results, Methods, and Tools

The process of identifying data sources in the first step of the cycle illustrated in *Figure 1* clearly benefits from a global catalogue of data sources. Not only does it simplify the collection of known datasets, but it can also assist in the discovery of useful data from unexpected sources. Less often considered, but potentially of equal importance is the contribution a catalogue can make in closing the data curation cycle.

A well designed catalogue may be extended to support not only input data sets, but data analysis results and the steps used to generate them. The use of a formal system for data integration and extraction, as described earlier, simplifies the process of cataloging methods (SQL queries) and tools (wrappers, etc.). All elements in the data analysis cycle may be catalogued, archived, queried, and recalled using a common interface and command set.

Proposed Proof of Concept

The University of Washington is well situated to develop a proof of concept EarthCycle system. This white paper, coordinated by the UW eScience Institute, was developed with input from the university's climate science and seismology communities and affiliates, which are leaders in their fields, along with valuable guidance from the IBM Corporation. UW operates a centrally supported research compute and storage system designed to accommodate the elements described here[11]. IBM, a world leader in the development of data-centric computing technologies, including DB2/Infosphere and the GPFS filesystem, as well as various advanced hardware solutions, is an active partner in the development of our campus research computing cyberinfrastructure. The eScience institute employs research staff with extensive experience in developing many of the underlying EarthCycle technologies[12]. The entire ~200TB IRIS seismology data collection, and the entire ~40TB set of NW regional climate model data are already stored on-site.

[1] http://earthcube.ning.com/profiles/blogs/welcome

[2] http://www.sdss3.org/index.php

[3] http://www.dynameomics.org/

[4] http://public.web.cern.ch/public/en/lhc/ATLAS-en.html

[5] http://escience.washington.edu/get-help-now/david-beck-bioinformatics-pipelines-using-databases

[6] http://www.ibm.com/developerworks/data/library/techarticle/0203haas/0203haas.html

[7] http://en.wikipedia.org/wiki/IPCC_Fifth_Assessment_Report

[8] http://speedpage.psc.teragrid.org/speedpage/www/speedpage.php

[9] http://en.wikipedia.org/wiki/BitTorrent_(protocol)

[10] https://trancheproject.org/

[11] http://escience.washington.edu/what-we-do/campus-compute-storage

[12] http://escience.washington.edu/who-we-are/our-team