Efficient Metadata Generation to Enable Interactive Data Discovery over Large-scale Scientific Data Collections

Sangmi Lee Pallickara[1], Shrideep Pallickara[1],
Milija Zupanski[2]
Department of Computer Science [1],
Cooperative Institute for Research in the Atmosphere[2],
Colorado State University
{sangmi, shrideep}@cs.colostate.edu,
zupanskim@cira.colostate.edu

Stephen Sullivan[3]
University Cooperation for Atmospheric Research[3],
steves@ucar.edu

Abstract— **Discovering the correct dataset efficiently is critical for computations and effective simulations in scientific experiments. In contrast to searching web documents over the Internet, massive binary datasets are difficult to browse or search. Users must select a reliable data publisher from the large collection of data services available over the Internet. Once a publisher is selected, the user must then discover the dataset that matches the computation's needs, among tens of thousands of large data packages that are available. Some of the data hosting services provide advanced data search interfaces but their search scope is often limited to local datasets. Because scientific datasets are often encoded as binary data formats, querying or validating missing data over hundreds of Megabytes of a binary file involves a compute intensive decoding process. We have developed a system, GLEAN, that provides an efficient data discovery environment for users in scientific computing. Fine-grained metadata is automatically extracted to provide a micro view and profile of the large dataset to the users. We have used the Granules cloud runtime to orchestrate the MapReduce computations that extract metadata from the datasets. Here we focus on the overall architecture of the system and how it enables efficient data discovery. We applied our framework to a data discovery application in the atmospheric science domain. This paper includes a performance evaluation with observational datasets.**

Keywords- metadata, data discovery, cloud computing, atmospheric sciences, large-scale datasets

I. INTRODUCTION

Several government agencies and research groups publish large scientific datasets. In the earth sciences this includes NOAA [1], NCAR [2] and NASA [3]. NCBI provides biomedical and genomic datasets [4] for biologists, while NASA operates a data center that hosts astrophysics datasets. As the size of the generated data grows, scientific data hosts often package and organize datasets based on different metrics to manage it more efficiently. It is quite common to organize observational earth science datasets based on geospatial and temporal information. Bioinformatics researchers prefer to package their dataset based on structural information and useful annotations. The size of the published data collection varies from few kilobytes to several terabytes [5]. Such data collections are used for scientific computing, simulations, and processes that do data analysis.

Scientists trying to access public dataset collections available over the Internet do so in several phases. This process can be broadly broken down into five stages as classified in [6, 7]. These include: (1) recognition of the information problem, (2) issuing the search query over a selected search engine, (3) scanning and evaluating the search results prior to accessing the web document, (4) scanning and evaluating the selected web document with regard to its relevance to the search goal, and (5) comparing information from different pages. Although the current web-based data search technologies enable us to discover useful text documents using these stages, applying them directly to the discovery of scientific datasets poses several challenges. For instance, in stage-2, the binary encoding format that is used for improving performance by reducing data transfer sizes often precludes text-based searching and ranking algorithms. For stage-3, browsing datasets that are in the order of several megabytes is difficult. Similarly, accessing, browsing, and comparing (as required in stage-4 and stage-5) large scientific datasets is time consuming and often involves computing intensive operations to decode and visualize the dataset.

There have been several multi-disciplinary efforts encompassing atmospheric sciences and computer science in the area of the atmospheric data discovery. MyLEAD [8] provides features of data discovery for public data and personal data by means of cataloging and tracking the user's computational activities. GEONGRID [9] is a cyberinfrastructure for integrating 3D and 4D earth science data. The interface provided by GEON to its users is based on keywords, resource type, temporal filtering, and interactive maps. THREDDS [10] provides a metadata catalog to provide an advanced data search interface and a data subset service for the files based on the Netcdf format [11, 12]. For datasets in biology, MicroSeeds [13] provides a data discovery environment for microarray researchers. For more general scientific users, SDSC's iRODS [14], which is a successor of SRB, hosts more than 1,000 TB of data (more than 200 million files) for scientific users and projects. Metadata catalogs provide search query capabilities to the users; while large databases for scientific experiments such as SciDB [15] provide a relational database style data storage for supporting multi-dimensional arrays. However, as the number of public datasets available over the Internet increases, users require an integrated environment to discover the dataset over data collections not just from a single host but also from multiple data hosts. Each of the aforementioned web-based data search stages need suitable technologies tuned for scientific datasets, so that users can process the data discovery efficiently.

We have developed a system, GLEAN, to cope with issues related to the data discovery in large-scale data collections. This infrastructure provides the following features:

- Efficient and automated fine-grained metadata extraction schemes to support advanced queries
- Data summarization schemes for browsing large datasets
- Support for customizable datasets
- Automated collection of provenance metadata
- A programmable interface to the system's capabilities
- Extensibility to different scientific domains

The primary contribution of this paper is providing schemes for the efficient discovery of large scientific data. To enable advanced features, access to rich metadata is critical. We maintain three types of metadata: fine-grained metadata, provenance metadata, and summary of the datasets. First, we extract the fine-grained metadata automatically if it is needed. For large-scale binary datasets, extracting fine-grained metadata often involves a computing intensive operation. To provide efficient decoding, we have broken down the processing as a set of MapReduce [16] computations that are orchestrated using the Granules runtime. Fine-grained metadata allows users to issue a more detailed search query. Second, we track the user's data discovery activities to collect provenance metadata. This includes maintaining the history of data accesses and search requests that are statistically summarized and provided to the users. Finally, summary of dataset is generated for the dataset to enable browsing.

In scientific experiments, interesting datasets often lead the experiment to a meaningful result. Scientists try to discover data not only through a conventional database query over static metadata, but also based on statistical properties of the dataset and access patterns of other users. GLEAN provides various data summarization schemes and provenance data so that users can narrow down their selections without having to download or process large amounts of data to browse and compare them locally.

We have validated our ideas in the context of observational data in atmospheric sciences. Here, observational data arrives in a timely manner and the size of the data is considerably large. The datasets are organized based on geospatial coordinates.

The rest of the paper is organized as follows: Section 2 describes the related work in this area. In section 3, we describe the architecture of GLEAN. In section 4, we describe an application in the atmospheric sciences that uses GLEAN's features. We present results from our performance evaluation in section 5. Finally, we outline conclusions and future work in section 6.

II. RELATED WORK

Metadata provides critical clues to discovering a scientific dataset. The Metadata Catalog Service (MCS) [17] and MCAT Metadata Catalog associated with iRODS manage metadata for scientific datasets. MCAT is tightly coupled with iRODS and it provides metadata about logical and physical datasets [18]. In contrast, MCS manages the metadata of logical data objects and it provides more flexibility to support various data storage mechanisms. Both the metadata catalogs rely on the user's input to collect the metadata. In GLEAN, the metadata are collected automatically without user-intervention; these metadata can be fine-grained or coarse-grained.

There is a good body of research in the discovery of atmospheric or geospatial datasets. Thematic Real-time Environmental Distributed Data Services (THREDDS) [10] provides an integrated environment for data discovery, data analysis, display, and live access to real-time atmospheric data. Metadata of the datasets are stored and managed by metadata catalogs. The Earth System Grid (ESG) [19] supports discovery and access to important climate modeling datasets. ESG provides several ways to search datasets: Google-style text search, based on pre-generated key terms, and an interactive map interface. In GLEAN, users can browse summaries of metadata that are generated automatically in addition to the metadata encoded into the datasets.

MyLEAD [8] catalogs observational and modeling data. Additionally, myLEAD also tracks a user's computational activities such as running workflows while cataloging both the intermediate and output data of a computation. To catalog newly added datasets, the system relies on cooperation from the participating data sources. In contrast, GLEAN runs a crawler outside the purview of the data source to provide more flexible management of data sources. Community driven discovery of useful datasets is made possible in GLEAN by allowing users to register datasets.

In the geoscience domain, [20] provides advanced data search interfaces including keyword and data-type constrained searches with interfaces for applications based on web services. NCDC provides an interactive map [21] to specify a user's query and to view datasets. It provides regional, data type, and temporal filters for the data search. NCDC also provides an hourly summary of the published datasets. Similarly, the data library of NOAA's Climate Services provides hierarchical browsing based on the type of the data along with text and interactive map based search [22]. This interface provides a statistical overview of the data based on the search criteria. Finally, [23] provides gateways to the tens of data servers through the community data portal.

In biology area, MicroSEEDS [13] provides a web based data discovery environment for microarray researchers. MicroSEEDS maintains the metadata for a number of microarray experiment sites and sources added by other researchers and administrators of the system. Users narrows down their selection through interactive clustering process of sites and references.

The diversity of metadata formats in the targeted datasets is emerging as a critical problem for data discovery. There have been significant activities in standardizing the metadata format in different domains: a metadata scheme for visualizing data in Astronomy has been proposed in [3]. The MGED [12] initiative provides a standard for describing experiments in the life sciences. For geospatial information, metadata based on [24, 25] have been widely adopted in the earth sciences.

III. ARCHITECTURE

Figure 1 depicts the architecture of GLEAN. The system includes a set of local servers and a high-performance computing cluster. The following subsections describe the various components that comprise the infrastructure.

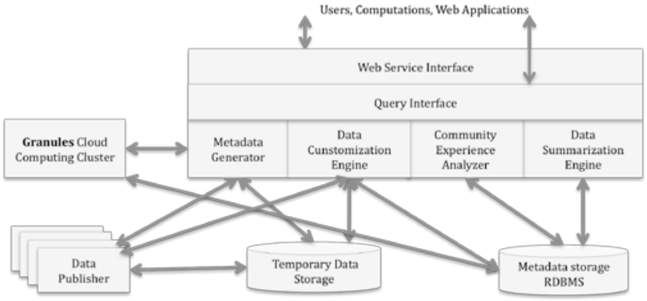

Users, Computations, Web Applications
Web Service Interface
Query Interface
Granules Cloud Computing Cluster
Metadata Generator
Data Cunstomization Engine
Community Experience Analyzer
Data Summarization Engine
Data Publisher
Temporary Data Storage
Metadata storage RDBMS

Figure 1. Architecture of the GLEAN system

A. Access Interfaces

At the topmost layer, there are two interfaces available to the users: computations and Web Applications. Users are allowed to access features through the command line and web service interfaces. Users can add new dataset entries, issue queries, and customize datasets through these interfaces. Remote computations and applications are able to access features through a standard web service interface. Here, a data search query can be integrated directly into the computation via the web service interface.

B. Granules for data-driven MapReduce computations

Our Granules [26, 27] runtime is targeted at data driven computations. Granules incorporates support for two models for developing cloud applications: Map-Reduce [16] and graph-based orchestration[28]. Computations in Granules can specify two types of inputs: files and streams. Individual computations specify a scheduling strategy that governs their lifetimes. This scheduling can be specified along three dimensions: data availability, periodicity, and a maximum limit for the number of times that they can be executed. One can also specify a custom scheduling strategy that is a combination along these three dimensions. Thus, one can specify a scheduling strategy that limits a computation to be executed a maximum of 500 times either when data is available or at regular intervals. Computations are held dormant till such time that their scheduling constraints are satisfied: data should be available on any one of their input streams or the specified interval between successive executions should have elapsed. A computation can change its scheduling strategy during execution, and Granules enforces the newly established scheduling strategy during the next round of execution. This scheduling change can be a significant one – for example, from data driven to periodic. The scheduling change could also be a minor one with changes to the number of times the computation needs to be executed or an update to the periodicity interval.

Computations can have multiple, successive rounds of execution and retain state across iterations. To maximize resource utilizations Granules interleaves the execution of multiple computations on a resource. Though the CPU burst times for individual computations during a given execution is short (seconds to a few minutes), these computations can be long running with computations toggling between activations and dormancy for several weeks to months. By sizing thread pools Granules can effectively utilize the availability of multiple execution pipelines on modern multicore machines. Some of the domains that Granules is currently being deployed in include atmospheric science, brain-computer interfaces, epidemiological simulations, and handwriting recognition.

Using Granules allows us to (1) develop the processing as MapReduce computations with the results being communicated between the MapReduce stages using streams rather than files (2) activate these computations when data is available without having to do busy waits or polling (3) Interleave multiple dataset computations on the same resource to maximize utilizations.

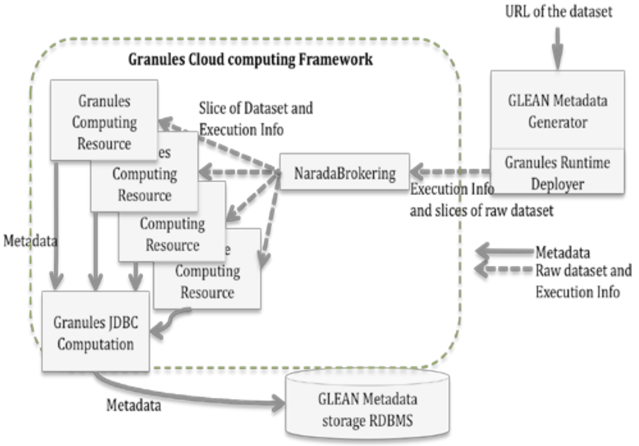

URL of the dataset
Granules Cloud computing Framework
Granules Computing Resource
es Computing Resource
Computing Resource
e Computing Resource
Slice of Dataset and Execution Info
NaradaBrokering
GLEAN Metadata Generator
Granules Runtime Deployer
Execution Info and slices of raw dataset
Metadata
Granules JDBC Computation
Metadata
GLEAN Metadata storage RDBMS
Metadata
Raw dataset and Execution Info

Figure 2. Process of Extracting and Summarizing Metadata

C. *Extracting Fine-Grained Metadata*

Rich metadata is a precursor to providing advanced features such as supporting complex queries, generating summaries of large datasets, and also for coping with dataset faults. Users can narrow down the result set more accurately with complex queries. Metadata encapsulating information about data quality such as the error rates or missing elements within a dataset can be useful in guiding a user's decision to access the data.

Many of data hosting services do not provide rich metadata externally; detailed metadata is often encoded within the dataset. To extract the metadata, we decode the binary encoded dataset. This decoding process is developed as a set of MapReduce computations that can be orchestrated on the available machines. There are several steps involved in extracting metadata using the Granules runtime. We first split the original file to distribute the dataset over a cluster. Computations are then pushed to nodes that hold portions of the file. Each computation then extracts fine-grained metadata by parsing the XML document, and ensuring that the data is stored in the metadata storage. Reducers are activated as soon as data is available from the mappers.

Figure 2 depicts the process of extracting metadata based on the specified URL of a binary dataset. The data is first downloaded to a temporary storage, and spliced into small chunks that contain all the information that need to be decoded. Figure 3 shows an example of the data splitting process for the BUFR format. Since the header contains the definition of the functional codes, the slices of the datasets refer to the header to be decoded. Hence, each of the data chunks must include the header part. Once the data chunks are ready, we use Granules to orchestrate the MapReduce computations that are responsible for decoding these data chunks. Metadata is extracted from the decoded dataset along with statistical metadata relating to averages, means, and variances of various metrics along with a synopsis of the faults such as error rates or missing elements in the dataset.

Extracted and generated metadata are stored in a relational database. The storage computation is managed by Granules, and is activated based on the availability of metadata for storage. There is only one active JDBC connection to the database; the system interleaves all storage requests on this connection. We only store the metadata in the database and not the dataset; only a pointer to the dataset is stored. None of the decoding computations need to know either the physical location of the database or authorization information needed to set up the JDBC connection.

We decode the binary BUFR file using our open-source wmoBUFR decoder. The WmoBufr system [29] is a Java software package to decode WMO BUFR files. The decoded data are available either as in-memory Java objects or as XML files. Since there are a wide variety of XML interpreters in nearly every programming language, generally the XML format is more accessible and easier to use than the original BUFR format. Currently the WmoBufr system supports WMO version 13 BUFR tables.

D. *Generating Advanced Metadata*

The rich metadata extracted from the dataset is processed further to provide summarization, which is useful for data browsing and comparing large datasets. There are several aspects that we consider in the summarization scheme. One such aspect is the degree of summarization; data is much more reliable and easier to interpret as the degree of summarization increases. In contrast, data becomes more opaque as the degree of summarization decreases. The summarization can be based on statistical summary, ranking, or categorizing methods that are useful for data discovery. Compact presentation of summaries is also important.

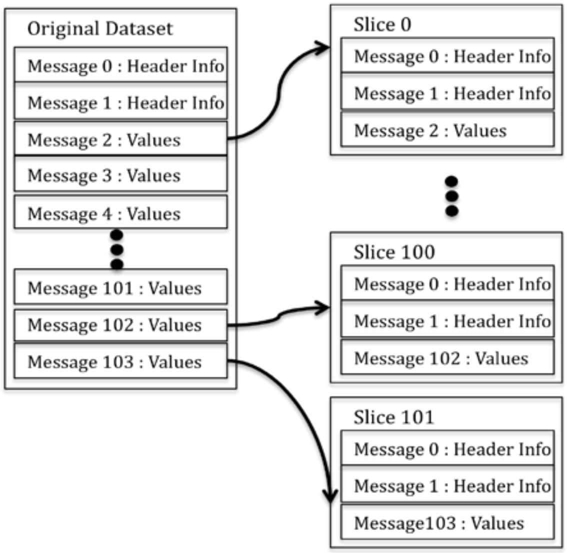


Figure 3. Splicing BUFR dataset and generating chunks of the dataset to distribute to the computing nodes of Granules runtime cluster

Currently, we provide average, maximum, and minimum value of key indexing items: geospatial coordinate, and time series. The summarization is done in a hierarchical fashion. Based on the data format, users can access summaries that cover only partial data from the full dataset. Users can also issue the search query over such partial summaries. This feature helps the users to pick interesting portions of the dataset to customize dataset later on. We also allow the users to track requests for data discovery. Users can browse information relating to (1) the statistics of data accesses for a specific dataset, (2) recently added datasets, and (3) their personal history of accesses or dataset registration requests.

E. Customizable Datasets

Users can also customize datasets. A customizable dataset is useful for improving the performance of a computation. Instead of downloading and decoding several hundreds megabytes of data, users can download specific portions of the dataset that will be injected into their computation. Furthermore, a customizable dataset is useful when the simulation requires data that falls between the ranges of different published datasets. The data customization engine supports the creation of a customized dataset from existing datasets.

Figure 4 shows the basic query styles for customized datasets. Some of the queries can involve multiple merging and splitting of files to create the final datasets. Type A requires only a small portion of the dataset from the dataset D_a. In surface observational datasets, the published dataset typically contains a large number of small records. The actual data that is requested is often a very small portion of the dataset. In this case, if the users can access the customized dataset that is much smaller than original dataset, the computation's performance will improve significantly through reduction in delays for downloading and decoding the data.

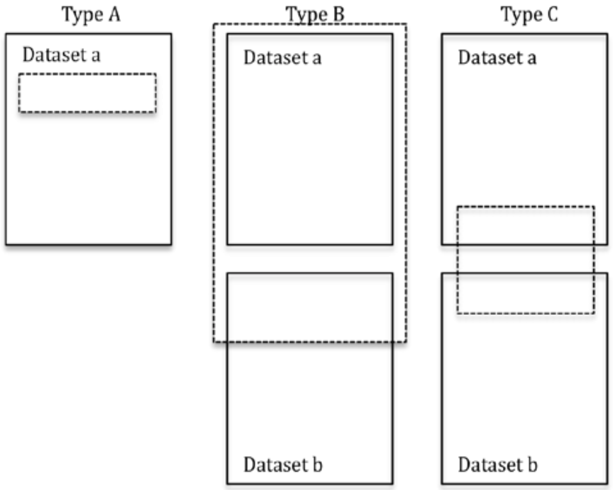


Figure 4. Basic Types of Customized Datasets: Type A requires only small portion of the dataset a. Type B requires complete dataset of Dataset a and small portion of Dataset b. Type C requires merging two small portions of the dataset a and b.

Type B and C happen when the range of the required dataset does not match the one in the published packages. In the type B, we need to store the offsets of the data chunks of the dataset D_b to retrieve a part of the dataset and merge it with dataset D_a. In type C queries, the offsets of the data chunks of dataset D_a and dataset D_b have to be maintained to retrieve the partial datasets from two different datasets. Type B and C also entail adjusting the header information if it is needed so that rest of the dataset refers to valid header information.

In our infrastructure, we store the offset information of the data chunk along with the metadata. The customized dataset is generated by collecting data chunks and generating the header information accordingly. Each of the data chunks are retrieved from the original dataset based on the offset information that was stored in the metadata storage.

IV. APPLICATION

We have applied GLEAN's features to the Atmospheric Data Discovery System (ADDS), which is a community-driven data discovery environment for atmospheric observational datasets. Here, we describe this system, and how it harnesses capabilities provided by GLEAN.

A. Collecting Metadata of Datasets

New datasets are added in two ways. First, a newly published dataset is periodically checked from a list of sites. If a new entity is detected, the dataset is downloaded and indexed. This process includes extracting finer-grain metadata if needed. Second, users are allowed to initiate the addition of a new dataset. The dataset downloaded from the requested URL is indexed if this was not done previously. Since any dataset with a well-formed URL and an encoding format that is supported by the framework is indexed, datasets published over the Internet can be made available using ADDS.

B. *Decoding and Collecting Fine-Grained Metadata*

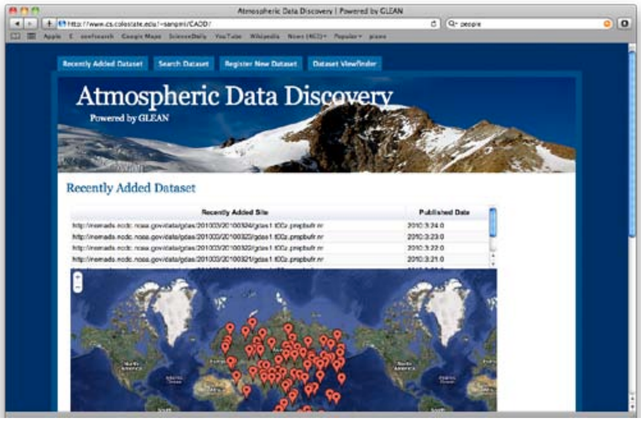
The collected URL of the dataset is then delivered to GLEAN's metadata generator. BUFR is one of the most popular data formats for observational data in the atmospheric science domain. We decode the BUFR datasets using the wmoBUFR decoder. Decoding a typical BUFR data file on a single machine with the wmoBUFR decoder takes several minutes to an hour. To improve the decoding performance, we relied on MapReduce computations executing on different machines.

C. *Query over the datasets*

Users are allowed to issue advanced queries over the fine-grained metadata to search the dataset. There are three types of queries:

- Name-value query (e.g. category = XYZ)
- Geospatial query (subset, superset, intersect, and exclusive)
- Temporal query (subset, superset, intersect, and exclusive)

These queries can be used either individually or combined to formulate a compound query. The result of a search query is a list of links to the datasets and their basic metadata. Users are allowed to browse the summary of the datasets before they download the dataset.


Atmospheric Data Discovery | Powered by GLEAN
http://www.cs.colostate.edu/~sangmi/CADD/
people
Recently Added Dataset
Search Dataset
Register New Dataset
Dataset Viewfinder
Atmospheric Data Discovery
Powered by GLEAN

Recently Added Dataset

Recently Added Site	Published Date
http://nomads.ncdc.noaa.gov/data/gdas/201003/20100324/gdas1.t00z.prepbufr.nr	2010.3.24.0
http://nomads.ncdc.noaa.gov/data/gdas/201003/20100323/gdas1.t00z.prepbufr.nr	2010.3.23.0
http://nomads.ncdc.noaa.gov/data/gdas/201003/20100322/gdas1.t00z.prepbufr.nr	2010.3.22.0
http://nomads.ncdc.noaa.gov/data/gdas/201003/20100321/gdas1.t00z.prepbufr.nr	2010.3.21.0

Figure 5. Screenshot of the Micro-view of the Dataset in Atmospheric Data Discovery System

D. *Browsing Large Datasets*

Browsing and comparing datasets are important steps in deciding whether a dataset should be downloaded. Selecting the right datasets to access can involve transferring and processing datasets that are tens or hundreds of MB in size. We use an interactive map interface for presenting geospatial information. We present summaries for large datasets that usually contain thousands of subsets inside. As depicted in Figure 5, users can select a geographical area and browse detailed metadata of the subsets.

E. *Access to the history of a user's activities*

The history of activity (index, access, and request) provides a first filter for users. As seen in Figure 4, a list of the newly added datasets is maintained; this is sorted based on when they were published or indexed. We also provide information about the total number of indexing requests and datasets that were selected to be downloaded. Although this portal requires users to register before they use this service, statistical information about collective actions do not reveal a user's identity.

F. *Creating and accessing the customized dataset*

The dataset packaged by data hosting services often contain various types of data. Likewise, very large geospatial area or long temporal range is covered within single dataset. We provide a feature that generates a customized dataset based on the user's queries to optimize the data access and processing for the user's computation.

The customized dataset is generated based on the stored offset information. If the dataset is cached, the customized dataset is created from the cached dataset, otherwise the original dataset is downloaded. The stored offset information contains the location of beginning and ending bytes of a minimum sized chunk of data. Here the minimum chunk is specified based on the data formats. For our prototype, we indexed observational datasets published by NCEP. These datasets are encoded using the BUFR data format and the encoding unit is defined accordingly. For the surface observational dataset spanning 6 hours, around 3000~4000 units are typically included.

The user's query requesting partial information will return a list of offset information. The bytes according to the returned offset information are collected and are stored at a Web-accessible temporary space. Finally, the user gets the URL of the customized dataset.

G. *Automated Search Interface for the Computation*

We provide a standard web service interface for programming accesses from a scientific application or computation. The scientific application is able to issue a search query within their code and retrieve datasets according to query results. Such a retrieval of datasets is useful for other data processing tools. For example, once a dataset is available, the Ensemble Data Assimilation (EnsDA) [30] filtering method quantifies the uncertainty associated with the dataset. When applying EnsDA to geosciences a limiting factor is the discovery of the observational dataset without the user having to specify the physical location of the dataset. Using Glean's web service interface it is possible for EnsDA to discover these datasets programmatically and subsequently process them for fidelity.

V. Performance Measurements

Our benchmarks report on several aspects of Glean, which indexes observational datasets published every 6 hours by NCEP. Published data is encoded using the BUFR format, and the original data size was 32 MB with 3363 chunks. All machines involved in the benchmarks were 3.0 GHz Intel Xeon processors with 8 cores and 16 GB of memory. The machines were part of a 100 Mbps LAN. Each data chunk is decoded and stored as separate XML files. We utilized our Granules runtime to orchestrate our MapReduce computations that decode datasets and extract metadata. Granules was configured to run with 4 worker threads per machine.

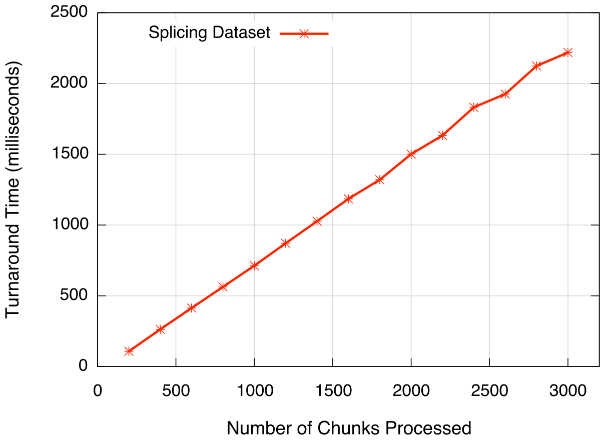


Figure 6. Turnaround time for splicing the original file to run on Granules cluster

Our first experiment measured the turnaround time for splicing the dataset and creating a set of decodable datasets with common header information on a single machine. We created splices with 100 data chunks each. As shown in Figure 6, as we increased the total number of data chunks, we observed a corresponding increase in the turnaround time; since each of the slices is stored in files, as the number of slices increases there is a corresponding increase in file I/O.

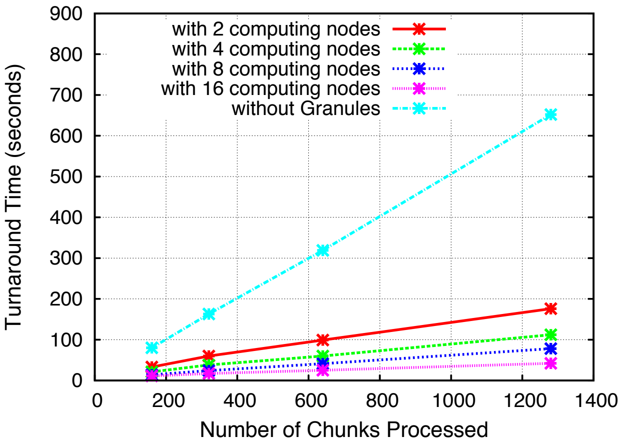


Figure 7. Turnaround time for decoding with different number of computing nodes

The next experiment measured the total turnaround time for decoding with different number of computing nodes. As shown Figure 7, as we increase the number of nodes, we observe a significant speed gain. The measurement involving 16 nodes resulted in a 16-fold speedup in the turnaround time when compared to the single node case.

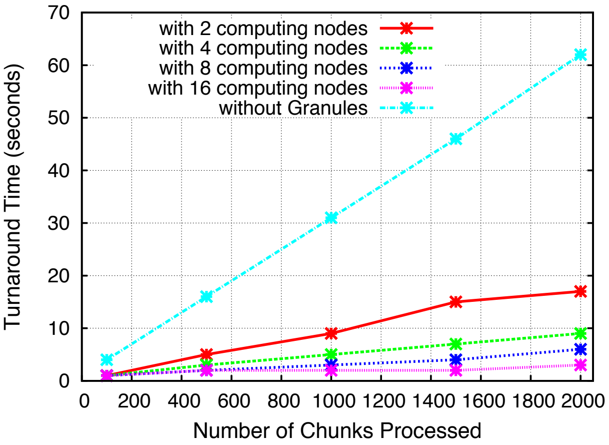


Figure 8. Turnaround time for extracting metadata and generating summary with various numbers of computing nodes

We performed another experiment to measure the turnaround time for extracting metadata, generating statistical information, and storing these metadata in a database. We have a separate computation designated for the initiating the JDBC connection to control the number of active connections that will be maintained to the database. As the number of the computing nodes increased, there was significant speed gain. As shown in Figure 8, with 16 nodes, we observed about a 20-fold speedup compared to the single node case.

VI. CONCLUSIONS AND FUTURE WORK

The amount of scientific data in the public domain has been growing rapidly. Since these datasets are encoded in a variety of binary formats, directly applying techniques that work well with text-based data search is difficult. We have developed a system, GLEAN that provides an interactive and efficient data discovery environment for scientific datasets.

Rich metadata and advanced tools allow users to perform their data discovery efficiently. We utilized our Granules runtime to orchestrate the execution of MapReduce computations that are responsible for extracting different types of metadata from the datasets. The extraction process is substantially faster in distributed settings, with almost linear performance gains as additional machines are made available.

Web service interfaces in GLEAN allow it to interoperate with the existing application components easily and across domains. These web service interfaces also allow web applications to selective of capabilities that they choose to harness from GLEAN, while also allowing computations programmatic access to these capabilities.

We have deployed our system in an application that provides a dataset search environment for atmospheric scientists: ADDS. This application catalogs observational atmospheric datasets with fine-grained metadata.

As part of our future work, we will focus on the quality of the datasets in observational settings. Unambiguously identifying missing data prior to the actual execution can improve the accuracy of the computations. We will also focus on expanding support to data formats in other domains.

REFERENCES

[1] G. K. Rutledge, *et al.*, "NOMADS: A Climate and Weather Model Archive at the National Oceanic and Atmospheric Administration," *Bull. Amer. Meteor. Soc.*, vol. 87, pp. 327-341, 2006.

[2] *The National Center for Atmospheric Resource*. Available: http://www.ncar.ucar.edu/

[3] R. L. Hurt, *et al.*, "Astronomy Visuallization Metadata (AVM) in action!," *American Astronomical Society, AAS Meeting* vol. 213, 2009.

[4] E. W. Sayers, *et al.*, "Database resources of the National Center for Biotechnology Information," *Nucleic Acids Research,* vol. 38, 2010.

[5] *Public Data Sets on AWS.* Available: http://aws.amazon.com/publicdatasets/

[6] S. Brand-Gruwel and P. Gerjets, "Instructional support for enhancing students' information problem solving ability," *Computers in Human Behavior,* vol. 24, pp. 615-622, 2008.

[7] G. Marchionini, *Information seeking in electronic environments*. Cambridge, UK: Cambridge University Press, 1995.

[8] B. Plale, *et al.*, "Cooperating Services for Managing Data Driven Computational Experimentation," *Computing in Science and Engineering(CiSE) magazine,* vol. 7, pp. 34-43, 2005.

[9] K. Lin and A. K. Sinha, "Discovery and Semantic Integration of Geologic Data," *Geoinformatics,* pp. 2006-5201, 2006.

[10] B. Domenico, *et al.*, "Thematic Real-time Environmental Distributed Data Services (THREDDS): Incorporating Interactive Analysis Tools into NSDL," *Journal of Interactivity in Digital Libraries,* vol. 2, 2002.

[11] R. K. Rew and G. P. Davis, "NetCDF: An Interface for Scientific Data Access," *IEEE Computer Graphics and Applications,* vol. 10, pp. 76-82, 1990.

[12] B. CA, *et al.*, "Standards for Microarray Data," *Science,* vol. 298, p. 539, 2002.

[13] C. A. Ball, *et al.*, "Submission of Microarray Data to Public Repositories," *PLos Biology,* vol. 2, pp. 1276-1277, 2004.

[14] R. W. Moore, "Building Preservation Environments with Data Grid Technology," *American Archivist,* vol. 69, 2006.

[15] P. C.-M. Kimura, *et al.*, "A Demonstration of SciDB: A Science-Oriented DBMS," *VLDB,* vol. 2, pp. 1534-1537, 2009.

[16] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," *Communications of the ACM,* vol. 51, pp. 107-113, 2008.

[17] E. Deelman, *et al.*, "Grid-Based Metadata Services," in *International Conference on Scientific and Statistical Database Management* Santorini Island Greece, 2004.

[18] R. W. Moore, "Managing Large Distributed Data Sets Using the Storage Resource Broker," *ITEA Journal of Test and Evaluation,* 2007.

[19] D. N. Williams, *et al.*, "The Earth System Grid: Enabling Access to Multimodel Climate Simulation Data," *Bull. Amer. Meteor. Soc.,* vol. 90, pp. 195-205, 2009.

[20] L.-A. Dupigny-Giroux, *et al.*, "NOAA's Climate Database Modernization Program: Rescuing, archiving, and digitizing history," *Bulletin of the American Meteorological Society,* vol. 88, pp. 1015-1017, 2007.

[21] N. C. D. Center. *Map Interface to Selected Online Data*. Available: http://www.ncdc.noaa.gov/oa/dataaccesstools.html#climate

[22] NOAA. *NOAA Climate Services: Data Library*. Available: http://www.climate.gov/#dataServices/dataLibrary

[23] NCAR. *Community Data Portal*. Available: http://cdp.ucar.edu/

[24] "Content Standard for Digital Geospatial Metadata," ed: Content Standard for Digital Geospatial Metadata, 1998.

[25] "ISO 19115 Geographic Information-Metadata," ed: International Organization for Standardization, 2003.

[26] S. Pallickara, *et al.*, "An Overview of the Granules Runtime for Cloud Computing," in *the IEEE eScience Conference*, Indianapolis, USA, 2008.

[27] S. Pallickara, *et al.*, "Granules: A Lightweight, Streaming Runtime for Cloud Computing With Support for Map-Reduce.," in *the IEEE International Conference on Cluster Computing*, 2009.

[28] M. Isard, et al., "Dryad: Distributed data-parallel programs from sequential building blocks," in *European Conference on Computer Systems*, Lisbon, Portugal, 2007.

[29] S. J. Sullivan. *WmoBufr*. Available: http://sourceforge.net/projects/wmobufr/

[30] M. Zupanski, "Maximum Likelihood Ensemble Filter: Theoretical Aspects," *Monthly Weather Review,* vol. 133, pp. 1710-1726, 2005.

[31] W. Thorpe, "A Guide to the WMO Code Form FM 94 BUFR.," ed. Monterey, CA.: Numerical and Oceanography Center, 1991.