Towards Efficient Data Search and Subsetting of Large-scale Atmospheric Datasets

Sangmi Lee Pallickara, Shrideep Pallickara
Department of Computer Science
Colorado State University
{sangmi, shrideep}@cs.colostate.edu

Milija Zupanski
Cooperative Institute for Research in the Atmosphere, Colorado State University
zupanskim@cira.colostate.edu

Abstract— **Discovering the correct dataset in an efficient way is critical for effective simulations in atmospheric sciences. Compared to text-based web documents, many of the large scientific datasets contain binary or numerically encoded data that is hard to discover through the popular search engines. In the atmospheric sciences, there has been a significant growth in public data hosting. However, the ability to index and search has been limited by the metadata provided by the data host. We have developed an infrastructure – Atmospheric Data Discovery System (ADDS) – that provides an efficient data discovery environment for the observational datasets in the atmospheric sciences. To support complex querying capabilities, we automatically extract and index fine-grained metadata. Datasets are indexed based on the periodic crawling of popular sites and files requested by the users. Users are allowed to access subsets of a large dataset through our data customization feature. Our prototype has been implemented based on the MapReduce framework. Our focus is the overall architecture, data subsetting scheme, and a performance evaluation of our system.**

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

I. INTRODUCTION

Discovering and providing the right input is critical to getting accurate results in atmospheric simulations. Over the last decade there has been a sustained increase in Internet-enabled access to massive pools of public observational datasets. This includes data collected and published by government-supported agencies [1-4], research institutes [5], and organizations targeting specific projects [6, 7]. Datasets are processed and packaged based on their targeted use. Since observational datasets are naturally geospatial and arrive in a timely manner a common key for organizing them is the geospatial range, temporal charachteristics, and type of the data.

Although Web based data search engines, such as Google or Bing, provide an index of more than 1 trillion documents [8], massive datasets published for the scientific researches have not really benefited from the powerful web search engines. Many of the observational datasets are encoded in binary formats [9, 10] to improve performance by reducing data transfer sizes. This often precludes use of text-based search and ranking algorithms for these datasets.

Most data services provide some methods to discover the data. Most conventionally [11], key information is encoded within the full path to the dataset. Metadata based on pathnames can limiting in its ability to describe a large dataset. Some data services [12] provide advance query interfaces using text filtering or interactive maps. THREDDS [13] provides a metadata catalog to provide an advanced data search interface and a data subset service for files based on the Netcdf [14] format. These services allow the users query over the datasets within their domain.

There have been multi-disciplinary efforts in the area of the atmospheric data discovery. MyLEAD [15] provides data discovery for both public and personal data by means of cataloging and tracking the user's computational activities. GEONGRID [16] 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.

In this paper we describe our system, Atmospheric Data Discovery Network (ADDS), which enables users to discover observational datasets in the Atmospheric science domain. Our data location service sits between the data hosting services and the individual users; this allows users to search for data from multiple data hosting services. This infrastructure provides the following features:

- Third-party observational data discovery spanning multiple data hosting services.
- Support for complex queries over the automatically generated fine-grained metadata of the datasets.
- Indexing of datasets based on automated crawling and a user's request.
- Efficient extraction of fine-grained metadata and indexing.

- The ability to browse large datasets efficiently prior to downloading them.
- Support for subsetting datasets.
- Programmable interfaces.

A. Motivation and Scientific Challenges

As the volume of data generated by observational instruments has increased, simulations in the Atmospheric sciences need to cope with the large number of the massive datasets that are available over the Internet. Here we describe the computational challenges that we faced that in turn motivated us to develop the ADDS framework.

Challenge 1: Automated data discovery and assimilation
Ensemble Data Assimilation (EnsDA) [17] is an advanced filtering method that simultaneously estimates the optimal state of a system and quantifies its uncertainty. In its relatively short and successful existence, EnsDA has been applied to many problems of geosciences. An intrinsic part of EnsDA is access to observations, which in weather applications involves the need for reading files with formatting defined by the World Meteorological Organization (WMO) and by the regional and national data centers. However, the capability to discover and access input data sets is not automatically satisfied on a specific computational platform. This seriously limits the portability of the EnsDA code and ultimately the benefit to a user.

Challenge 2: Does this result also include the datasets most recently published?
A key characteristic of atmospheric observational datasets is that they arrive in a timely manner. Data hosting services publish the datasets daily, hourly, or even every several minutes. Data processing and indexing of observational datasets should be efficient enough so that users can access the most recently listed dataset as early as possible.

Challenge 3: Optimize processing for computations that need to access only small portion of the large dataset?
Decoding binary data for the large size dataset is an IO intensive process. Some data hosting services package datasets that span wide geospatial and chronological ranges resulting in computations having to download and process the whole dataset unnecessarily. Similarly, published data often include datasets from global weather stations. For computations requiring only regional data, accessing only a small portion of the dataset can improve the performance of the computation.

Challenge 4: Is the metadata provided with the dataset rich enough to process the user's queries?
Text based web-based search engines are powerful but are unsuitable for binary datasets in atmospheric science. Therefore, users can search or browse dataset only as much as the metadata provided allows. Some of the data hosting services [12, 13] provide search interfaces but it limits the search capability within the datasets hosted by the service. Some of them provide very minimum metadata, which is not enough to process any of the rich queries that user might need to issue.

B. Paper Contributions

This paper's contributions are in the areas of data discovery, subsetting datasets, and metadata extraction. Our mechanisms for discovery of observational data rely on multiple techniques to not just improve the accuracy of the discovery, but also the performance of the computation that uses this dataset. We have developed a framework that automatically collects newly published observational datasets and extracts fine-grained metadata using the Map-Reduce paradigm to improve performance. Instead of relying on information encoded in the path of the file or URL, rich SQL-queries can be used to filter fine-grained metadata to ensure accuracy. The list of the datasets returned by data search query can be browsed graphically and statistically. To browse data graphically we summarize datasets by grouping geospatial locations. Statistical browsing involves retrieving information such as average, maximum, minimum, or standard deviation of various metrics within the dataset.

We support subsetting datasets. This feature enables users access to portions of a large dataset without having to download and process the complete dataset. This feature is useful for computations that only need access to a small portion of a dataset or combined portions of datasets from multiple, large data files.

Metadata underpins efficient access to these datasets. When the metadata provided by the host is not sufficiently expressive, fine-grained metadata is extracted from the dataset. Metadata extraction is automated and efficient so that newly listed data can show up in the search-results in a timely fashion. We implemented this using the MapReduce framework.

The rest of the paper is organized as follows: Section II describes the flow of observational data from the observational instruments to the users. Section III describes the system architecture. Performance evaluation of this system is presented in section IV. Related work is presented in section V. Our conclusions and future work are outlined in section VI.

II. Flow of Observational Data in Atmospheric Science

Data collected from the observational instruments is processed through multiple steps before it is made available to the users. Figure 1 depicts an example of the flow of the datasets from the observational instruments to the user's

desktop/computation using the ADDS framework. Data processing levels for data products generated as part of a research investigation are categorized [18] as:

- Level 0: Reconstructed unprocessed instrument/payload data at full resolution; any and all communications artifacts
- Level 1A: time-referenced, and annotated with ancillary information, including radiometric and geometric calibration coefficients and georeferencing parameters
- Level 1B--Level 1A data that have been processed to sensor units
- Level 2--Derived geophysical variables at the same resolution and location as the Level 1 source data
- Level 3--Variables mapped on uniform spate-time grid scales, usually with some completeness and consistency
- Level 4--Model output or results from analyses of lower level data

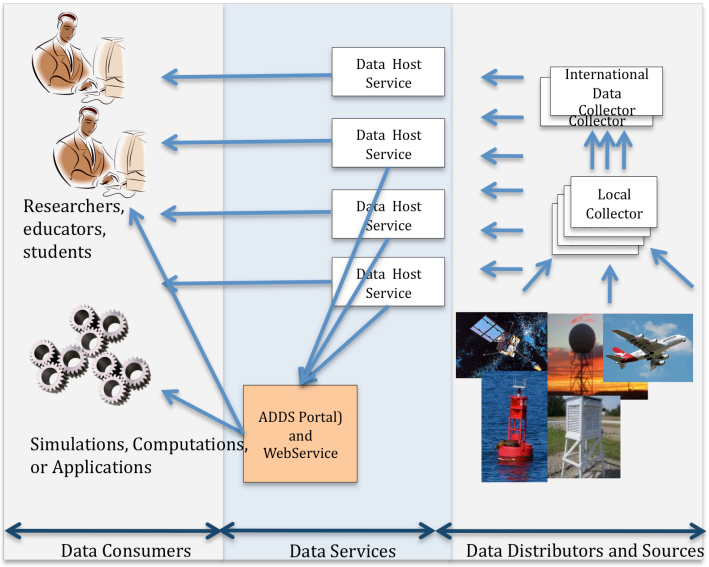

Figure 1. Example flow of observational data and ADDS

Most users access the datasets in Level 1 ~ 4. Since level 4 is model output or results, users of our system will access datasets in Level 1 ~ 3. For example, datasets available through our system are generated by observational instruments and collected by either local or national data collectors. These datasets are collected by higher level data systems such as WMO's Global Telecommunication Systems (GTS) [19]. GTS collects observational data from the globally located participants and distributes them at the national level. Datasets are collected from the World Meteorological Centers (Melbourne, Moscow and Washington), 15 Regional Telecommunication Hubs (including Beijing, Brasilia, Cairo, and Tokyo), and satellite data centers.

After the datasets are received nationally, they are processed and published for valid users by the authorized organization such as NCEP. For example, NCEP hosts datasets delivered from GTS and the National Environmental Satellite and Information Service (NESDIS) [11]. NCEP also receives datasets from various instruments such as the NEXRAD radar and the aviation circuit. These datasets can be in different levels of data processing. For example, wind data from NEXRAD radar includes level 2 and 3 data. There are also several data formats. NCEP packages datasets based on data similarity (but it still maintains original structure of reports) and observational cycles. Finally, integrated and encoded datasets are published periodically. Most of these datasets are available publically through the Internet.

These datasets are accessed by forecasting systems such as the Global Forecast System [20], which runs regional model and data assimilation system. The output data from Global Forecast System is also periodically published and used by research projects.

Our infrastructure is located between the data hosting services and the users (or computations) and enables users to discover the published observational datasets more accurately and efficiently. Based on our observations these datasets have the following characteristics:

- Users need to discover the newly available datasets in a timely manner.
- Datasets can contain multiple data types.
- A packaged dataset can contain data from a large geospatial area.
- Datasets can be packaged in different styles and coverage over the various data hosting services.
- Metadata provided by the data hosting services can have various qualities and levels of detail.

III. ARCHITECTURE

The ADDS framework comprises a set of local servers, web services, and a computing cluster. This depicted in Figure 2. ADDS utilizes a data cataloging engine, GLEAN [21], for scientific datasets. GLEAN provides general-purpose mechanisms for metadata management and summarizing. In this section we describe the various components that comprise the infrastructure.

A. Data Format

Currently, we support the Binary Universal Form (BUFR) [9] for the representation of meteorological data. BUFR is a binary data format maintained by the World Meteorological Organization. BUFR was originally designed to describe position-driven meteorological data such as surface observations, upper air soundings, and monthly climatological data. The BUFR format is a table-driven code

format: the meaning of data elements is determined by referring to a set of tables that are kept and maintained separately from the message itself. A BUFR message is composed of six sections including those for static metadata, data descriptors and binary data stream. We extract static metadata to describe the dataset. In addition, the binary data stream is decoded to extract fine-grained metadata.

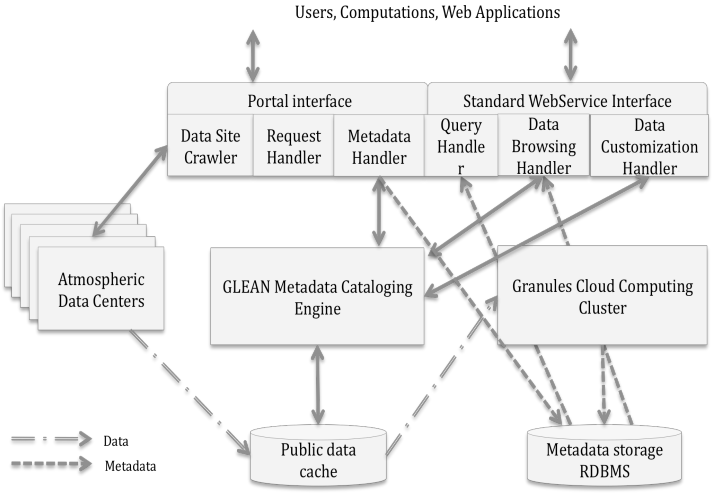

Figure 2. System Architecture

B. Interfaces

Clients can access ADDS capabilities either through a portal interface or a web service interface. The portal interface provides convenient access to the ADDS service using a web browser. The portal manages the user's account. Clients can use the portal interface to issue data discovery queries, browse, and creating a subset of the dataset.

A standard web service interface is provided for command line users and applications such as simulations. Data discovery can be integrated directly into the computation through the Web service interface.

C. Granules for data-driven MapReduce computations

Granules [22, 23] is a lightweight runtime for cloud computing that is targeted at data driven computations. Granules incorporates support for two models for developing cloud applications: Map-Reduce [24] and graph-based orchestration [25]. 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, earthquake science, 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.

D. Data Site Crawler

A new dataset can be added in two ways. First, a list of sites is visited periodically to check for the availability of a newly added dataset. If a new entity is detected, the dataset is downloaded and indexed. This process includes extracting finer-grain metadata if needed.

Second, a user initiates addition of a new dataset by specifying the corresponding URL. The dataset from the requested URL is indexed if it is not already indexed. Since any URL may be submitted, so long as the URL is well-formed and the data format of the dataset that it points to is supported by the framework, datasets from the special interest groups can be made available using ADDS.

E. Metadata Handler

For the advanced data discovery features, maintaining fine-grained metadata is essential. Generating metadata in this system comprises three phases: collecting available metadata, decoding the file, and tracking a user's activity.

First, we collect metadata already available with the dataset. For the datasets from the sites encoding the metadata into their URL paths, the key items will be extract from the string of the URL. For example, consider the URL: http://nomads.ncdc.noaa.gov/data/gdas/201005/20100502/gdas1.t06z.prepbufr.nr. This is a dataset published by the gdas server hosted by NCEP. This dataset contains observational data generated on May 2nd, 2010, between 03:00AM and 09:00 AM. Metadata will be constructed based on the parsed information.

Second, if the metadata provided by the host service is not rich enough to execute advanced query, finer-grained metadata is extracted automatically. For the binary datasets, decoding and extracting useful information is the first step of constructing fine-grained metadata. Decoding and extracting large size dataset is the most expensive process of the metadata generation. We discuss and analyze the cost of metadata generation in the section IV. We use our runtime, Granules, to orchestrate the distributed decoding computations written using the MapReduce framework.

Finally, the metadata about the user's usage pattern is collected. This includes the statistics of users' requests for the new dataset, and access to the dataset.

F. Query Handler

Indexing fine-grained metadata generated from datasets allows users to issue advanced queries over the datasets. Users can construct 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 this metadata before they download the dataset.

G. Data Browsing Handler

In the data discovery process, browsing the dataset is an important part of the decision-making process: the dataset may or may not be downloaded eventually. Unlike web documents that are less than few MB, deciding the right datasets to access can involve datasets that are tens or hundreds of MB in size.

Using interactive map interface is natural for the geospatial information. We provide summarized look for the large datasets that usually contain thousands of subsets inside. As depicted in Figure 3, users can select a geographical area and browse the metadata of the subsets.

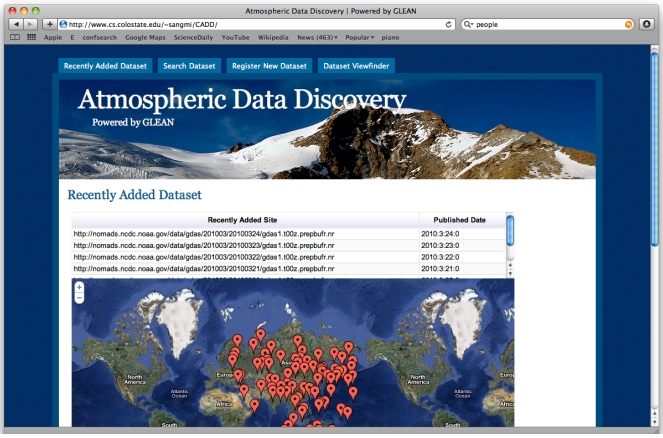

Figure 3. Snapshot of the Data Browsing Interface

The portal interface provides text-based browsing. From the set of results returned by their search query, users can select the dataset whose metadata they will browse. The metadata shown to the users is sometimes a summary data from the decoded values. Since datasets contain a large number of values even for key information such as geospatial and temporal information, the corresponding values are using statistical methods such as average, or a range of the values.

H. Request Handler

The requester of a new dataset determines whether the metadata will be shared with other users. Each of the newly added metadata contains information about one or more requester(s). Once the requesters publish the metadata, other users are allowed to search and browse this metadata. However, datasets submitted for cataloging by the community are not cached or redistributed.

I. Data Subsetter

Data hosting services often package various types of data into a dataset. Often, the published datasets cover a much larger geospatial area or longer temporal range than what the user actually needs. We provide a feature that creates a subset of the dataset based on the user's queries to optimize the data access and processing for the user's computation.

The subset of the dataset is created based on the stored offset information and cached dataset. If the dataset is not cached, the original dataset is downloaded. Offset information contains the location of beginning byte and

ending byte of minimum chunk of the data. Here the minimum chunk is defined based on the data formats. For our prototype, we indexed observational datasets published by NCEP, and used the data encoding unit provided by BUFR data format. For the dataset spanning 6 hours, around 3000~4000 minimum units are typically included.

The result of a query requesting partial information contains a list of offset information and the subset of the dataset is created by means of collecting the bytes according to the offset information. This dataset is then moved to a Web-accessible temporary space and the user then gets the URL of the dataset.

IV. PERFORMANCE EVALUATION

We have developed a prototype of this system, which indexes the observational datasets published every 6 hours by NCEP. Data encoding follows BUFR format, and the 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.

To extract detailed metadata from the binary file, we decode the data file using the open-source wmoBUFR decoder developed by UCAR. The decoder extracts metadata and generates XML files. We used Granules to orchestrate our MapReduce computations. Granules was configured to run on each of the 16 machines with 4 worker threads per machine.

1) System Benchmarks

Our first benchmark measured the total turnaround time for decoding different number of chunks included in the dataset. The turnaround time includes delays from multiple stages of data processing: preparation, decoding, and storing the metadata in an RDMBS storage. The preparation includes overheads for launching the Map-reduce job on Granules. The original file is downloaded and split into smaller chunks to be processed by each of Granules nodes. Our results in Figure 4 show that decoding is the largest contributor to the overall turnaround time.

Figure 5 contrasts the costs for processing chunks on 1 machine with that of processing these chunks on 16 machines. The overheads introduced by Granules in orchestrating these MapReduce computations is acceptable; we get an almost linear speedup as we increase the size of the dataset. We observed that there is a twelve-fold speed gain for a dataset that included 1200 chunks.

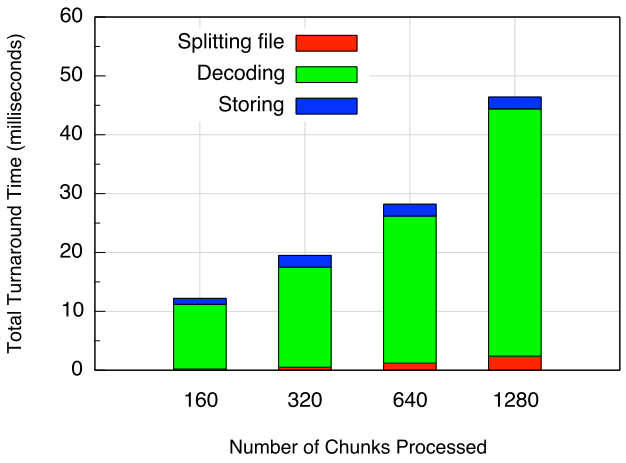

Figure 4. Turnaround time for generating fine-grained metadata

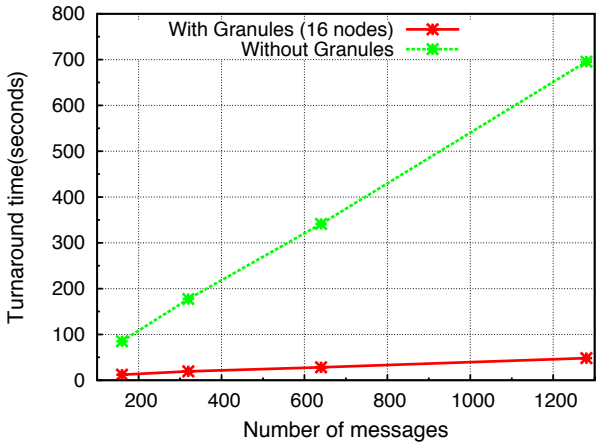

Figure 5. Comparing turnaround time for generating fine-grained metadata with Granules and without it.

We performed another benchmark to measure the total turnaround time for creating a dataset subset based on a user's query. This turnaround time included query processing, and generating a new file accordingly. In our benchmark we filtered datasets based on the number of chunks that they contain. As shown in Figure 6, when we query datasets containing larger number of chunks, we observed that the turnaround time increased as well. Since different datasets contain chunks of different sizes, the turnaround time is not exactly proportional to increases in the number of chunks. The largest dataset subset had more than 1038 messages, and it took 728 milliseconds. This overhead is reasonable for users performing dataset subsetting on demand through the ADDS portal.

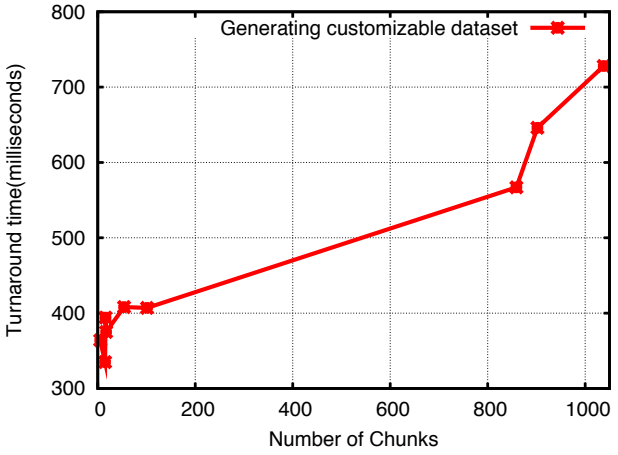

Figure 6. Total turnaround time for subsetting datasets

2) Subsetting in the context of a Scientific Data Assimilation Software

We have also tested our subsetting feature with Ensemble Data Assimilation (EnsDA) which is a filtering method for estimating the optimal state of a system and providing a measurement of its uncertainty. We measured the turnaround time for EnsDA using ADDS. The test scenario involved (a) the original dataset with 3332 message chunks and (b) a dataset subset with 259 message chunks. For the original dataset, it took 23 seconds. With the dataset created using our data subsetting feature, it took 3 seconds: an approximately 8-fold improvement.

V. Related Work

Thematic Real-time Environmental Distributed Data Services (THREDDS) [13] 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 using metadata catalogs.

The Earth System Grid (ESG) [27, 28] is an infrastructure for discovery and access to the important climate modeling datasets. ESG provides several ways to search datasets: Google-style text search, based on pre-generated key terms, interactive map interface.

MyLEAD [15] catalogs observational and modeling data. This also tracks the user's computational activities such as running workflows, and catalogs the intermediate and output data of the computation. To catalog newly added datasets, the aforementioned systems rely on the cooperation of the participating data sources. In contrast, ADDS runs a crawler outside of the data source to provide more flexible management of the data sources. In addition, the community effort to discover a useful data is harnessed by allowing each of the individual users to register a dataset.

In geospatial science domain, [16] provides advanced data search interface including keyword and data type search. It also provides interface for the application based on the Web service technology.

The National Climate Data Center (NCDC) under NOAA distributes observational, climate, and product data [12]. This data service provides an interactive map to specify the user's query and 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, data library of NOAA's Climate Services provides hierarchical browsing based on the type of the data along with text search and interactive map based search. This interface provides a statistical overview of the data based on the search criteria. There has also been an effort to provide gateways [28] to the tens of data servers through the community data portal.

VI. Conclusions and Future Work

Scientific data volumes in the public domain have been growing rapidly. Since datasets are encoded with various formats, applying text based data search technologies is not straightforward. In this paper, we presented our data discovery framework for observational datasets in Atmospheric science. ADDS provides an environment for users to discover large-scale observational datasets including datasets encoded with binary formats. We also incorporate support for discovering data generated by different hosting services, specification of rich queries, and support for subsetting datasets while obviating unnecessary dataset downloads by incorporating support for metadata browsing.

To provide these advanced features, maintaining rich metadata is critical. Fine-grained metadata is automatically generated by the system when needed. Metadata extraction was developed using the MapReduce framework, and orchestrated by the Granules cloud runtime. Our benchmarks also profiled subsetting datasets.

As part of our future work, we will focus on extending this service to support more data formats. We have observed that the quality of the dataset must be considered a part of the metadata. For the observational datasets, data can often be missing. We plan to enhance our service to cope with missing data items and outliers. Browsing and identifying the missing data prior to the actual execution can improve the accuracy of the computations.

REFERENCES

[1] Atmospheric Science Data Center, http://eosweb.larc.nasa.gov/. last accessed July 2010.

[2] National Centers for Environmental Prediction, http://www.ncep.noaa.gov/. last accessed July 2010.

[3] National Climatic Data Center, http://www.ncdc.noaa.gov/oa/ncdc.html. last accessed July 2010.

[4] The National Center for Atmospheric Resource, http://www.ncar.ucar.edu/. Last accessed July 2010.

[5] Global Change Master Directory, http://gcmd.nasa.gov/Resources/pointers/meteo_university.html. last accessed July 2010.

[6] NOAA's El Nino Page, http://www.elnino.noaa.gov/. last accessed July 2010.

[7] British Antarctic Survey, http://www.antarctica.ac.uk/. last accessed July 2010.

[8] Google official blog, "We know Web was big..", 2008, http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html. last accessed July 2010.

[9] Thorpe, W. (1991), "A Guide to the WMO Code Form FM 94 BUFR." http://dss.ucar.edu/docs/formats/bufr/bufr.pdf, Fleet Numerical and Oceanography Center, Monterey, CA

[10] World Meteorological Organization (WMO) (2009). WMO Operational Codes TAC-BUFR-CREX-GRIB. http://www.wmo.ch/pages/prog/www/WMOCodes/OperationalCodes.html. last accessed July 2010.

[11] Rutledge, G. K., J. Alpert, and W. Ebuisaki, 2006: NOMADS: A Climate and Weather Model Archive at the National Oceanic and Atmospheric Administration. Bull. Amer. Meteor. Soc., 87, 327-341.

[12] Dupigny-Giroux, L.-A., T. F. Ross, J. D. Elms, R. Truesdell, and S. R. Doty, 2007: NOAA's Climate Database Modernization Program: Rescuing, archiving, and digitizing history. Bulletin of the American Meteorological Society, 88(7), 1015-1017.

[13] Domenico, B., J. Caron, E. Davis, R. Kambic, S. Nativi, "Thematic Real-time Environmental Distributed Data Services (THREDDS): Incorporating Interactive Analysis Tools into NSDL," Journal of Interactivity in Digital Libraries, 2002 Vol. 2, No. 4

[14] Rew, R. K. and G. P. Davis, "NetCDF: An Interface for Scientific Data Access," IEEE Computer Graphics and Applications, Vol. 10, No. 4, pp. 76-82, July 1990.

[15] Plale, B., D. Gannon, Y. Huang, G. Kandaswamy, S. L. Pallickara, and A. Slominski, "Cooperating Services for Managing Data Driven Computational Experimentation," Computing in Science and Engineering(CiSE) magazine, (Vol. 7, No. 5) pp. 34-43. September/ October 2005.

[16] Lin, K., A.K. Sinha, "Discovery and Semantic Integration of Geologic Data," Geoinformatics, 2006, Reston, VA

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

[18] MTPE EOS Reference Handbook 1995, available from the EOS Project Science Office, code 900, NASA Goddard Space Flight Center, Greenbelt, MD 20771.

[19] The Global Telecommunication System, http://www.wmo.int/pages/prog/www/TEM/GTS/index_en.html

[20] Wu, W., R. J. Purser, D. F. Parrish, "Three-Dimensional Variational Analysis with Spatially Inhomogeneous Covariances" Mon. Wea. Rev., 2002, 130, 2905-2916

[21] Pallickara, S. L., S. Pallickara, M. Zupanski, and S. Sullivan. "Efficient Metadata Generation to Enable Interactive Data Discovery over Large-scale Scientific Data Collections," In Proceedings of the IEEE International Conference on Cloud Computing Technology and Science. Indianapolis. November 2010.

[22] Pallickara, S., J. Ekanayake, and G. Fox, "Granules: A Lightweight, Streaming Runtime for Cloud Computing With Support for Map-Reduce." In Proceedings of the IEEE International Conference on Cluster Computing. 2009.

[23] Pallickara, S., J. Ekanayake, and G. Fox, "An Overview of the Granules Runtime for Cloud Computing." In Proceedings of the IEEE eScience Conference. 2008.Indianapolis, USA.

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

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

[26] Middleton, D.E., Bernholdt, D. Brown, M. Chen, A. L. Chervenak, et al., "Enabling worldwide access to climate simulation data: the earth system grid (ESG)," Scientific Discovery Through Advance Computing (SciDAC), 2006

[27] ESG project, The Earth System Grid, www.earthsystemgrid.org, last visited July 2010.

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

[28] Community Data Portal, http://cdp.ucar.edu/ last visited July 2010.