Klimatic:
A Virtual Data Lake for Harvesting and Distribution of Geospatial Data

Tyler J. Skluzacek[1,2], Kyle Chard[2], and Ian Foster[1,2,3]

***Abstract*— Many interesting geospatial datasets are publicly accessible on web sites and other online repositories. However, the sheer number of datasets and locations, plus a lack of support for cross-repository search, makes it difficult for researchers to discover and integrate relevant data. We describe here early results from a system, Klimatic, that aims to overcome these barriers to discovery and use by automating the tasks of crawling, indexing, integrating, and distributing geospatial data. Klimatic implements a scalable crawling and processing architecture that uses an elastic container-based model to locate and retrieve relevant datasets and to extract metadata from headers and within files to build a global index of known geospatial data. In so doing, we create an expansive geospatial virtual data lake that records the location, formats, and other characteristics of large numbers of geospatial datasets while also caching popular data subsets for rapid access. A flexible query interface allows users to request data that satisfy supplied type, spatial, temporal, and provider specifications; in processing such queries, the system uses interpolation and aggregation to combine data of different types, data formats, resolutions, and bounds. Klimatic has so far incorporated more than 10,000 datasets from over 120 sources and has been demonstrated to scale well with data size and query complexity.**

I. Introduction

New sensors, simulation models, and observational programs are producing a veritable deluge of high quality geospatial data. However, these data are often hard for researchers to access, being stored in independent silos that are distributed across many locations (e.g., consortium registries, institutional repositories, and personal computers), accessible via different protocols, represented in different formats (e.g., NetCDF, CSV) and types (e.g., vector, raster), and are in general, difficult to discover, integrate, and use [1]. These challenges are none more evident than in environmental and climate science. Here, vast collections of data are stored in dark, heterogeneous repositories distributed worldwide.

We aspire to make these large quantities of geospatial data accessible by creating the virtual data lake, a cached subset of a data lake paired with additional metadata for non-cached datasets. A data lake is "a centralized repository containing virtually inexhaustible amounts of raw (or minimally curated) data that is readily made available anytime to anyone authorized to perform analytical activities" [2]. Such a system allows for the local caching of raw data in a standardized format, making integration and distribution more efficient at query-time. A geospatial data lake should allow for the straightforward alignment of spatial and time-based variables, and be able to manage and integrate heterogeneous data formats. Given the huge quantity of geospatial data, we extend the data lake model to encompass a metadata index of all processed data and the use of our virtual lake as a cache for popular raw data. This approach allows for the tracking of less popular datasets without giving up valuable performance and space availability for oft-accessed data.

To explore these ideas, we have prototyped Klimatic, a system for the automated collection, indexing, integration, and distribution of big geospatial data. Although there is prior research in both geospatial metadata extraction and data lakes, to the best of our knowledge this is the first example of a centralized, searchable index across disparate web-accessible resources, combined with a virtual lake cache for raw data. We adopt a scalable crawling and metadata extraction model, using a dynamic pool of Docker containers [3] to discover and process files. Thus, we pave the way for creation of a scalable system that has the capacity to scour an increasing number of available resources for geospatial data. To further reduce usage barriers, Klimatic supports the integration of heterogeneous datasets (in both file type and format) to match users' queries, while also ensuring data integrity [4], [5].

The rest of this paper is as follows. §II discusses challenges associated with the creation of a geospatial virtual lake. §III outlines Klimatic's architecture and implementation. §IV explores the data collected in Klimatic. §V discusses related work. Finally, §VI summarizes the impact of Klimatic while illuminating future research and applications.

II. Challenges

Geospatial data are stored in a variety of repositories, many accessible via HTTP or Globus GridFTP. Globus is a service-based research data management system that provides access to more than 10,000 storage systems (called "endpoints"), many of which are used for storing scientific data. Automating the collection and indexing of *all* geospatial data stored on Globus endpoints and the web would be of great benefit to researchers. However, this task is not without significant challenges, as we now discuss.

Discovery: Klimatic needs a way to discover and explore data stored across an extremely large number of storage systems. It must do so in such a way that file paths can be stored for purposes of data provenance and re-examination at a later time. Klimatic therefore requires a crawler that can scale to many sites and datasets. It needs to be able to identify potentially relevant datasets, for example by looking for relevant file extensions (e.g., .nc and .csv). For

[1]Department of Computer Science, The University of Chicago, Chicago, IL, USA **skluzacek@uchicago.edu**
[2] Computation Institute, The University of Chicago, Chicago, IL, USA
[3] Math & Computer Science Div., Argonne Nat. Lab., Argonne, IL, USA

each dataset identified, it needs to be able to introspect on its contents, which requires interfaces that support data in different formats accessible via different APIs. It must also be able to determine quickly whether the dataset already exists in the virtual data lake, and decide whether to cache or discard the dataset.

Indexing: Once Klimatic places a dataset in the Docker container it needs to acquire descriptive metadata that can identify datasets satisfying user-specific search criteria. Additionally, Klimatic must establish indices that allow users to quickly filter data by means of metadata. Metadata may be found in file names, in structured file headers, or within the file body. Thus, we require a flexible indexing model that can not only identify these metadata, but also allow for many geospatial queries while tracking provenance.

Integration: The purpose of our approach is to create a flexible virtual data lake from which users may retrieve not only individual datasets but also integrated datasets defined by a specification such as "all temperature measurements for the region (30W to 32W, 80N to 82N) for the period of January, 2016." It must process such requests efficiently, while also upholding the data's integrity. Geospatial data are particularly complicated to integrate as heterogeneous collection methods result in different representations (e.g., raster vs. vector) and different granularities (e.g., spatial and temporal). Furthermore, the units used to represent common data may be different (or even missing). Thus, Klimatic must effectively manage misalignments between datasets to curate a new dataset with near-equal integrity to its ancestors.

Ensuring integrity: Given the integrative nature of Klimatic, a number of geospatial integrity rules must be followed when integrating multiple geo-spatial datasets into one. These constraints include topological, semantic, and user-defined integrity constraints [4], [5], [6]. Topological constraints require that data be divided into mutually exclusive regions with all space covered. Semantic constraints require that geological relationships are maintained, meaning, for example, that a road cannot exist in the same space as a building. Finally, user-defined constraints require that data are minimally affected following post-processing. Additionally, integrated data should include information that tracks data lineage. If a dataset cannot fit these constraints, the user is asked whether to reject the integrated dataset.

III. Populating the Virtual Data Lake

The Klimatic architecture implements a three phase data ingestion pipeline to populate the virtual data lake: (1) crawling and scraping publicly accessible data, (2) extracting metadata and building a discovery index, and (3) loading data into virtual data lake storage.

A. Crawling and Scraping

The first step in the Klimatic pipeline works to identify and then download publicly accessible geospatial files. To provide scalability, we use an elastically scalable pool of *crawler instances*, implemented as Docker containers, each repeatedly retrieving a URL from a *crawling queue*, retrieving and processing any content at that address, and adding any new URLs identified during processing to the queue. Figure 1 illustrates this phase of the workflow.

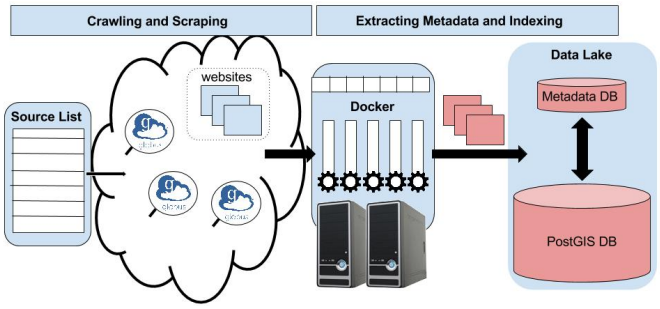

Fig. 1. Workflow for Klimatic's metadata extraction and storage.

Klimatic can initially retrieve data via either HTTP or GridFTP. In each case, our crawler looks for the commonly used NetCDF (.nc) [7] and CSV formats. The process by which the crawler discovers these datasets is dependent on the target repository.

For HTTP-accessible repositories, we seed the crawling queue with common repositories for geospatial data, such as the National Oceanic and Atmospheric Administration (NOAA) and the University Corporation for Atmospheric Research (UCAR). Using these links as an initial base, the crawler then explores those web sites and other linked web sites by scouring the links within pages. As a result of this crawling process, a list of datasets (with associated URLs) is appended to a second *extraction queue*. We have used this method to discover more than 10,000 climate files.

For GridFTP-accessible data, we use Globus APIs to seed the crawling queue with endpoints that analysis of access control lists show to be publicly accessible. The crawler then explores those endpoints recursively, filtering files by format and appending matching files to the extraction queue. Our crawler has so far identified 441 geospatial datasets, mainly in CSV format, residing on Globus endpoints.

The final challenge associated with the crawling phase is to determine whether files contain relevant spatial data, as well as dealing with false-positive datasets (i.e., datasets that seem to have spatial data during a scan, but do not). As NetCDF files contain structured headers (with time- and area-based keys) and raw data in-file, filtering NetCDF files for relevant metadata is straightforward. However, this task is more difficult when analyzing CSV files. To test whether a CSV file contains spatial data, the program checks for a number of pre-determined geo-spatial keys (e.g., 'lon,' 'lons,' 'long,' 'lng,' and 'longitude' for a longitude variable) in the first two rows of each column. If such keys are found, metadata are extracted. We have found that fewer than 10% of CSV files on the sites that we visited contain spatial data and CSV files rarely have informative headers and often require scanning the entire file to create metadata. Once metadata are stored, Klimatic briefly scans each new dataset's metadata to ensure that the geo-spatial data fit

within perceivable bounds (e.g., the latitudes and longitudes exist). If the metadata seem unlikely to be true, the data are flagged for human review.

B. Extracting Metadata and Indexing

We next process each dataset added to the extraction queue. This process is performed via an elastically scalable pool of Docker-based *extractor instances*. Each such instance repeatedly downloads datasets via HTTP or GridFTP and uses a metadata extraction library to complete the Klimatic metadata model. (We use the UK Gemini 2.2 [8] standard to represent geospatial metadata.) All metadata are loaded into a standard PostgreSQL database and indexed via a PostgreSQL text-search (TS) vector, an alternative to checksums that creates a unique string out of a dataset's metadata. This index allows the crawler to identify if a dataset is already known to the virtual warehouse, in which case, the duplicate is recorded in the index, so as to prevent redundant future accesses to the same file. The TS vector index also makes it easy for users to check for the availability of certain data parameters, such as lat, long, variables, start date, end date, and the dataset's publisher.

C. Data Storage

If a new dataset is not determined to be a duplicate, the Klimatic system next converts its contents to a relational format and loads them into a new PostgreSQL table, so as to accelerate subsequent retrieval and integration operations. The data are not otherwise modified, although future work could involve automatic transformation to reference grids, perhaps based on analysis of user query histories.

Given the virtually unlimited number of geospatial datasets, it is infeasible to retain the contents of every dataset. Thus, we operate a caching strategy. Metadata for every dataset located via crawling are stored in the index, but dataset contents are stored only if smaller than a predefined threshold and are subject to ejection via an LRU policy when the cache is full. Thus, larger and less popular datasets may need to be re-fetched when requested by a user. (In future work, we will also explore alternatives to discarding datasets, such as compression and transfer to slower, cheaper storage.)

D. Responding to User Queries

Having loaded some number of datasets into the virtual data lake, we are next concerned with responding to queries. We show our query model in Figure 3. Our initial query interface is a simple web GUI using Flask and Python. With the goal of making the query interface as simple as possible, we allow users to query using minimum and maximum latitudes and longitudes (i.e., a bounding box for their data); the variable(s) they would like included in their dataset; the begin and end dates; and (optionally) the data provider(s) from which data is wanted. Klimatic then estimates the amount of time required to conduct the join and deliver the dataset. Many queries require more than two minutes for the join, as many datasets have upward of 2 million cells.

The multiple possible encodings for climate data, most notably vector and raster, creates challenges when attempting

$$M_1 = \begin{bmatrix} . & . & . & 4 & . \\ . & 4 & . & . & 4 \\ 6 & . & 2 & . & 8 \end{bmatrix}$$

$$M_2 = \begin{bmatrix} . & . & . & 4 & 4 \\ 5 & 4 & 3.3 & 4.2 & 4 \\ 6 & 4.1 & 2 & 4.3 & 8 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 5 & 4.3 & 3.9 & 4 & 4 \\ 5 & 4 & 3.3 & 4.2 & 4 \\ 6 & 4.1 & 2 & 4.3 & 8 \end{bmatrix}$$

Fig. 2. F_1 on Matrix M to format a vector as a raster. Black values are original, red are created on first sweep, and orange created on second.

to integrate multiple datasets into one. A vector is a data structure that represents many observations from a single point, but at different times (e.g., precipitation levels measured at a fixed weather station). A raster can be represented by a two dimensional grid, in which each cell is a certain area identifiable on a map. Each cell contains the value of some variable: for example, the percentage of pollen in the air. Thus, to enable users to retrieve integrated datasets we require a method for integrating these two formats for cross-format data analysis: an integration that may involve a sparse set of vectors and a large raster database. (For example, ∼180,000 weather stations record precipitation in the U.S., each with a fixed latitude and longitude, while a complete radar mapping of the U.S. results in over 760,000 5 km^2 raster cells [9].)

We implement this integration via an interpolation from point values to a scalar field (a raster). We use a series of sweeping focal operations for some raster M, where each point in M represents a cell of a given region denoted by latitudinal and longitudinal boundaries. A focal operation is defined as the operation on a certain cell with regards to a small neighborhood around the cell [10]. Our implementation of this algorithm begins with a focal neighborhood of 1, or the eight diagonal or adjacent cells of a selected empty cell. If there are at least two neighbors, the new cell becomes the non-weighted average of all cells in region F_1. The center of F_1 is moved from cell-to-cell until either all cells are full or there exist F_1s such that there are not at least two value-bearing cells inside. The algorithm then adds one more series of neighbors (i.e., neighbors of neighbors), which we call F_2, F_3 through F_n, where F_n results in a complete matrix.

Figure 2 illustrates this process, where M_1 is the original sparse matrix and M_2 and M_3 are the second and third sweeps. As far as the data's user-defined, post-processing integrity is concerned, we record in Klimatic's output header the number of sweeps necessary to make the vector compatible with rasters. We may infer that a higher number of sweeps results in less 'pure' data. Our interface will also prompt users with information regarding the data's post-processing integrity as well as related data that could be selected to increase this integrity.

Klimatic currently supports the creation of integrated

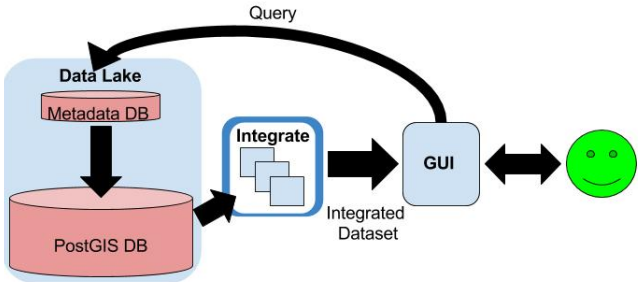

Fig. 3. Work flow for Klimatic's data integration and distribution.

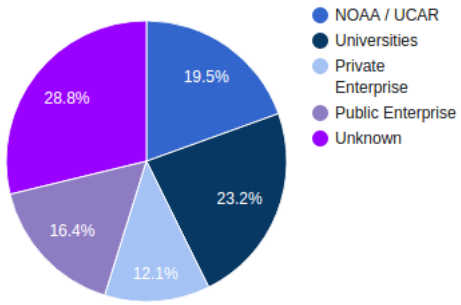

Fig. 4. Distribution of Klimatic's total datasets by provider-type

NetCDF and CSV files. NetCDF conventions simplify the creation of an integrated NetCDF dataset. NetCDF files can be conceptualized as having containers for multiple variables, while assuming that matching indices across the containers refers to a specific data point; index 0 in each container refers to the first data point, index 1 the second, and so on.

If a query response requires integration of both vector and raster data, Klimatic currently uses the grid dictated by the raster. Each vector always lies within a raster cell, so each cell containing one vector becomes the value of the vector at a given time. If multiple vectors fall within the same raster cell, we currently choose to average their values. (Here and elsewhere, we apply one data conversion strategy automatically in our prototype. Ultimately, we will want to allow the user to control such actions.) Once a standardized grid is achieved, the addition of a variable only requires the addition of another variable container, as long as the spatial and temporal bounds align. If the resolutions and time-bounds are different (e.g., if one dataset is measured in months and the other in years), we aggregate to the larger period (i.e., years). Future work could involve imputing values for missing areas and time periods, but this will require statistical distribution analysis.

IV. Evaluation

We aim in Klimatic to include geospatial data that span all areas and many variables and years, originating from both large repositories (e.g., UCAR and NOAA) and smaller private research, educational, and industrial sources. The importance of considering smaller sources is shown by the fact that only 19.5% of Klimatic's data are known to originate from large sources, as shown in Figure 4. (We classify providers based on information obtained from the

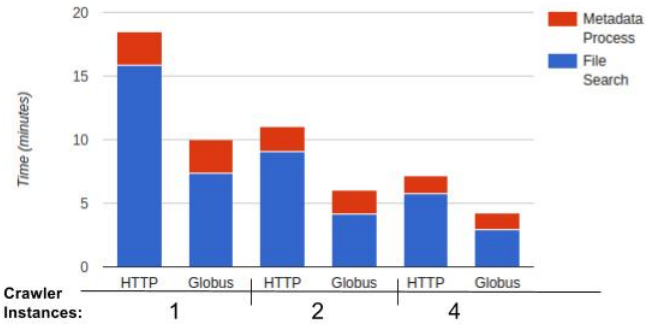

Fig. 5. Time (minutes) to find, extract and store metadata from, and add data to virtual warehouse for, 750 ∼100 MB files, via Globus and HTTP

crawled data locations. 28.8% of providers did not supply this information at the time of the tests.)

Klimatic has so far extracted metadata and constructed a searchable index for 10,002 datasets (∼11.5 TB). The area covered by Klimatic's collected data is expansive, with the least-covered regions of the world (e.g., South America, Australia, Antarctica) having ∼1,250–3,350 datasets each and the most-covered areas (e.g., North America, Europe, Australia, and Asia) having ∼8,900–9,500 datasets apiece. The datasets vary in resolution, from coarse 100 km x 100 km cells to fine 50 m x 100 m cells. To increase uniformity of coverage across regions of the globe, Klimatic could prioritize data in the less-covered areas.

From the perspective of computational efficiency, Klimatic performs well on dataset ingestion. To test data ingest performance, we evaluated the system on 750 randomly selected datasets averaging 100 MB each (for a total of 75 GB) stored in remote Globus and web sources, using 1, 2, and 4 crawler instances. As shown in Figure 5, the Globus scraper outperforms the web scraper due to the overhead inherent in the web scraper, as it concurrently traverses all links on a page to find—and explore—applicable paths to datasets before moving on to the next source.

V. Related Work

Related work encompasses areas such as data lakes and other integrated data approaches, geospatial data distribution, and metadata extraction. Klimatic expands upon prior work by addressing the challenges of collecting, indexing, and distributing geospatial data from diverse sources via a system based on data lake concepts [2]. We build upon others' efforts to encapsulate the steps necessary to bring raw geospatial data from its source to a user, including its acquisition, processing, and distribution [11].

The motivation for our work aligns with other efforts to scrape scientific data and extract metadata. For example, similar approaches have been applied to collect scientific information from papers, including our own work on extracting polymer properties from journal publications [12]. Others have used metadata extraction and indexing for business and industrial purposes, as there are noticeable increases in I/O performance and decreases in required human effort [13]. In biomedicine, data commons are proposed for integrating

genomics data [14]. In the geosciences there is growing emphasis on making data broadly accessible, as in the Earth Grid System Federation [15], which links climate simulation data archives worldwide; NCAR's Research Data Archive [16], which provides access to NCAR data; and DataOne [17], an online service that indexes a large number of geospatial datasets housed in various repositories. Our approach is differentiated as Klimatic aims to scrape arbitrary distributed data, rather than only those housed in repositories.

Other applications such as ESRI's ArcGIS [18] and Cadcorp's SIS [19] allow for metadata collection and dataset integration, but require significant human input. One is also limited to the data stored in those systems and one's own unindexed data. By removing the human element from metadata extraction, Klimatic ensures that a source's original metadata are cited correctly, leaving no room for human error [20], [21]. Klimatic follows the standard UK Gemini metadata storage convention [8], but given the broad scope of the data that Klimatic processes, some metadata are often justifiably missing, as when vector data, which correspond to a single point, lack bounding coordinates.

VI. Summary

Klimatic effectively provides an accessible architecture for the collection and dissemination of large, distributed geospatial data. It is able to automatically crawl huge amounts of data distributed across various storage systems and accessible via HTTP and Globus.

With continued work to add additional datasets to Klimatic and make the indexed data more broadly accessible to applications, we hope that Klimatic will become a great asset to the many communities that use geospatial data. In addition to seeking more data for Klimatic, a number of software improvements can occur in the short run. First we can create smarter metadata extraction. Additions to the current Klimatic process could include comparing sources that contain conflicting data, and using the geographic distributions to determine which data better fit a physical phenomenon. For example, we find that latitude and longitude are often encoded inconsistently, particularly in CSV files, as when 154.3° is used in some files to mean 154° and 0.3°, and in other files 154° and 3 minutes). Klimatic could look at additional dataset elements (e.g., city names, if available) to determine the intended interpretation of degrees versus minutes, and convert it to the standard convention (degrees, minutes, and seconds).

Additionally, the user experience can be improved through an enhanced interface and better caching strategies. UI enhancements could include allowing users to choose data from a map, providing better areas to include for a research study by analyzing the underlying statistics of a dataset (i.e., "These adjacent datasets share correlations in [selected variable]"), or allowing users to trace an outline of their desired data area on a map and getting a *very* specialized dataset in return, perhaps as a shapefile—an area bounded by a *connect-the-dots* convex hull commonly used in geographic analysis. Shapefiles are helpful in analysis of non-rectangular neighborhoods or odd-shaped natural features, including lakes and mountains. Furthermore, we plan to provide support for other, less popular file types to fully encompass the geospatial data domain. To allow the fast processing between datasets, the caching algorithm used in the data lake can better learn which files to hold on local disk in order to minimize the time required for the average user's queries. We also plan to implement a periodic checker to search each indexed dataset's origin for updates.

Other future work will focus on developing collaborative applications and expanding functionality. Klimatic is built to support external applications that may access data via APIs. We will collaborate with diverse disciplines to develop plug-ins to our system that notify a person or a decision-system when some threshold is reached, which allows that person or system to react to changes in data in a timely manner.

Acknowledgment

We thank Raffaele Montella for discussions on these topics, as well as Computation Institute at The University of Chicago for providing the resources necessary to undertake this work. This work was supported in part by DOE contract DE-AC02- 06CH11357 and by NSF Decision Making Under Uncertainty program award 0951576.

References

[1] P. B. Heidorn, "Shedding light on the dark data in the long tail of science," *Library Trends*, vol. 57, no. 2, pp. 280–299, 2008.

[2] I. Terrizzano, P. M. Schwarz, M. Roth, and J. E. Colino, "Data wrangling: The challenging journey from the wild to the lake." in *Conf. on Innovative Data Systems Research*, 2015.

[3] D. Merkel, "Docker: Lightweight Linux containers for consistent development and deployment," *Linux Journal*, no. 239, p. 2, 2014.

[4] R. Elmasri and S. Navathe, *Fundamentals of Database Systems, 2nd Edition*. Addison-Wesley, 1994.

[5] K. A. Borges, A. H. Laender, and C. A. Davis Jr, "Spatial data integrity constraints in object oriented geographic data modeling," in *7th ACM International Symposium on Advances in Geographic Information Systems*. ACM, 1999, pp. 1–6.

[6] S. Cockcroft, "A taxonomy of spatial data integrity constraints," *GeoInformatica*, vol. 1, no. 4, pp. 327–343, 1997.

[7] "netCDF: Network common data form," http://www.unidata.ucar.edu/software/netcdf/. Visited August 16, 2016.

[8] A. for Geographic Information, "UK GEMINI v.2.2 specification for discovery metadata for geospatial data resources," vol. 2.2, pp. 1–61, 2010.

[9] "Weather Underground," http://wunderground.com/about/data.asp. Visited August 25, 2016.

[10] S. Shashi and C. Sanjay, "Spatial databases: A tour," 2003.

[11] D. J. Maguire and P. A. Longley, "The emergence of geoportals and their role in spatial data infrastructures," *Computers, Environment and Urban Systems*, vol. 29, no. 1, pp. 3 – 14, 2005, geoportals. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0198971504000456

[12] R. Tchoua, K. Chard, D. Audus, J. Qin, J. de Pablo, and I. Foster, "A hybrid human-computer approach to the extraction of scientific facts from the literature," *Procedia Computer Science*, vol. 80, pp. 386–397, 2016.

[13] M. Chisholm, *How to build a business rules engine: Extending application functionality through metadata engineering*. Morgan Kaufmann, 2004.

[14] R. L. Grossman, A. Heath, M. Murphy, M. Patterson, and W. Wells, "A case for data commons: Toward data science as a service," *Computing in Science & Engineering*, vol. 18, no. 5, pp. 10–20, 2016.

[15] D. Bernholdt, S. Bharathi, D. Brown, K. Chanchio, M. Chen, A. Chervenak, L. Cinquini, B. Drach, I. Foster, P. Fox *et al.*, "The Earth System Grid: Supporting the next generation of climate modeling research," *Proceedings of the IEEE*, vol. 93, no. 3, pp. 485–495, 2005.

[16] “CISL Research Data Archive,” http://rda.ucar.edu. Accessed September 1, 2016.
[17] M. B. Strasser Cook, “DataOne,” *DataOne Best Practices Primer*, pp. 1–11, 2014.
[18] “ESRI ArcGIS,” http://www.esri.com/software/arcgis. Visited August 16, 2016.
[19] “Cadcorp SIS,” http://www.cadcorp.com/. Visited August 16, 2016.
[20] J. K. Batcheller, B. M. Gittings, and S. Dowers, “The performance of vector oriented data storage strategies in ESRI’s ArcGIS,” *Transactions in GIS*, vol. 11, no. 1, pp. 47–65, 2007.
[21] N. Kussul, A. Shelestov, M. Korbakov, O. Kravchenko, S. Skakun, M. Ilin, A. Rudakova, and V. Pasechnik, “XML and grid-based approach for metadata extraction and geospatial data processing,” in *5th International Conference on Information Research and Applications*, 2007, pp. 1–7.