CLIMATE ANALYTICS AS A SERVICE

John. L. Schnase [a,]*, Daniel Q. Duffy [b], Mark A. McInerney [c],
W. Phillip Webster [a], and Tsengdar J. Lee [d]

[a] Office of Computational and Information Sciences and Technology,
[b] NASA Center for Climate Simulation, [c] NASA Climate Model Data Services,
NASA Goddard Space Flight Center, Greenbelt, Maryland USA
(john.l.schnase, daniel.q.duffy, mark.a.mcinerney, phil.webster)@nasa.gov

[d] NASA High End Computing Program, NASA Headquarters,
Washington, DC USA – tsengdar.j.lee@nasa.gov

ABSTRACT

Climate science is a big data domain that is experiencing unprecedented growth. In our efforts to address the big data challenges of climate science, we are moving toward a notion of Climate Analytics-as-a-Service (CAaaS). CAaaS combines high-performance computing and data-proximal analytics with scalable data management, cloud computing virtualization, the notion of adaptive analytics, and a domain-harmonized API to improve the accessibility and usability of large collections of climate data. MERRA Analytic Services (MERRA/AS) provides an example of CAaaS. MERRA/AS enables MapReduce analytics over NASA’s Modern-Era Retrospective Analysis for Research and Applications (MERRA) data collection. The MERRA reanalysis integrates observational data with numerical models to produce a global temporally and spatially consistent synthesis of key climate variables. The effectiveness of MERRA/AS has been demonstrated in several applications. In our experience, CAaaS is providing the agility required to meet our customers’ increasing and changing data management and data analysis needs.

***Index Terms* —** Hadoop, MapReduce, MERRA, API

1. INTRODUCTION

The term ‘‘big data’’ is used to describe data sets that are too large or complex to be worked with using commonly-available tools [1]. Climate science represents a big data domain that is experiencing unprecedented growth [2]. Some of the major big data challenges facing climate science are easy to understand: large repositories mean that the data sets themselves cannot be moved: instead, analytical operations need to migrate to where the data reside; complex analyses over large repositories requires high-performance computing; large amounts of information increases the importance of metadata, provenance management, and discovery; migrating codes and analytic products within a growing network of storage and computational resources creates a need for fast networks, intermediation, and resource balancing; and, importantly, the ability to respond quickly to customer demands for new and often unanticipated uses for climate data requires greater agility in building and deploying applications [3]. It is useful to situate our big data challenges in this larger context, because doing so helps us understand where innovation can yield improvements.

2. BACKGROUND

Our understanding of the Earth’s processes is based on a combination of observational data records and mathematical models. The size of NASA’s space-based observational data sets is growing dramatically as new missions come online. However a potentially bigger data challenge is posed by the work of climate scientists, whose models are regularly producing data sets of hundreds of terabytes or more [2, 4].

The NASA Center for Climate Simulation (NCCS) provides state-of-the-art supercomputing and data services specifically designed for weather and climate research. The NCCS maintains advanced data capabilities and facilities that allow researchers within and beyond NASA to create and access the enormous volume of data generated by weather and climate models. Tackling the problems of data intensive science is an inherent part of the NCCS mission.

There are two major challenges posed by the data intensive nature of climate science. There is the need to provide complete lifecycle management of large-scale scientific repositories. This capability is the foundation upon which a variety of data services can be provided, from supporting active research to large-scale data federation, data publication and distribution, and archival storage.

* Corresponding author.

The other data intensive challenge has to do with how these large datasets are used: data analytics — the capacity to perform useful scientific analyses over large quantities of data in reasonable amounts of time. In many respects this is the biggest challenge: without effective means for transforming large scientific data collections into meaningful scientific knowledge, our mission fails. It is against this backdrop that the NCCS began looking at CAaaS as a potential element in our technological and organizational response to changing demands.

3. CLIMATE ANALYTICS AS A SERVICE

We believe there are five essential technology elements that contribute to building Climate Analytics-as-a-Service: high-performance, data-proximal analytics; scalable data management; cloud computing virtualization; adaptive analytics; and domain-harmonized APIs. In this section, we describe three of the more important of these elements; additional information can be found in [3].

3.1. High-performance, data-proximal analytics

Clearly, at its core, CAaaS must bring together data storage and high-performance computing in order to perform analyses over data where the data reside. MapReduce is of particular interest to us, because it provides an approach to high-performance analytics that is proving to be useful to many data intensive problems [5, 6]. MapReduce enables distributed computing on large data sets using high-end computers. It is an analysis paradigm that combines distributed storage and retrieval with distributed, parallel computation, allocating to the data repository analytical operations that yield reduced outputs to applications and interfaces that may reside elsewhere. Since MapReduce implements repositories as storage clusters, data set size and system scalability are limited only by the number of nodes in the clusters. While MapReduce has proven effective for large repositories of textual data, its use in data intensive science applications has been limited, because many scientific data sets are inherently complex, have high dimensionality, and use binary formats.

MapReduce distributes computations across large data sets using a large number of computers (nodes). In a “map” operation a head node takes the input, partitions it into smaller sub-problems, and distributes them to data nodes. A data node may do this again in turn, leading to a multi-level tree structure. The data node processes the smaller problem, and passes the answer back to a reducer node to perform the reduction operation. In a “reduce” step, the reducer node then collects the answers to all the sub-problems and combines them in some way to form the output — an answer to the problem it was originally trying to solve.

3.2. Adaptive analytics

Data intensive analysis workflows bridge between a largely unstructured mass of archived scientific data and the highly structured, tailored, reduced, and refined analytic products that are used by individual scientists and form the basis of intellectual work in the domain. In general, the initial steps of an analysis, those operations that first interact with a data repository, tend to be the most general, while data manipulations closer to the client tend to be the most specialized to the individual, to the domain, or to the science question under study. The amount of data being operated on also tends to be larger on the repository-side of the workflow, smaller toward the client-side end products.

This stratification can be exploited in order to optimize efficiencies along the workflow chain. MapReduce, for example, seeks to improve efficiencies of the near-archive operations that initiate workflows. In our work so far, we have focused on building a small set of canonical near-archive, early-stage analytical operations that represent a common starting point in many analysis workflows in many domains. For example, average, variance, max, min, sum, count, and difference operations of the general form:

$$\text{result} \leftarrow \text{avg}(\text{var}, (t_0,t_1), ((x_0,y_0,z_0),(x_1,y_1,z_1))),$$

that return, in this example, the average value of a variable when given its name, a temporal extent, and a spatial extent. Because of their widespread use, we refer to these simple operations as "canonical ops" with which more complex analytic expressions can be built. They provide a template for users as they begin their exploration of MapReduce analytics and are useful in their own right as steps in larger analyses. We tend to think of them as a type of assembly language instruction for climate data analysis.

The goal is to deploy the canonical ops within a framework that is able to capture their patterns of use and enable more complex analyses to be assembled and incorporated back into the system. The notion of engaging the broader community to deal with big data challenges has been used successfully in other settings, perhaps most notably with GalazyZoo, where a large user community is helping search the Sloan Digital Sky Survey for patterns and observations of potential scientific value [7]. We believe that this type of social networking can play an important role in the future of climate analytics. The approach we are taking sets the stage for the community construction of new capabilities that are adapted to the socially expressed requirements of those who use the system.

3.3. Domain-harmonized APIs

In order to knit these capabilities together and deliver them into practical use, we are building the Climate Data Services (CDS) application programming interface (API). APIs specify how software components interact with each other; they can take many forms, but the goal for all APIs is to make it easier to implement the abstract capabilities of a system. In building the CDS API, we are trying to provide for climate science a uniform semantic treatment of the combined functionalities of large-scale data management and data-proximal analytics. In doing so, we are combining concepts from the Open Archive Information Systems (OAIS) reference model, highly dynamic object-oriented programming APIs, and Web 2.0 resource-oriented APIs.

The OAIS reference model, defined by the Consultative Committee on Space Data Systems, addresses a full range of archival information preservation functions including ingest, archival storage, data management, access, and dissemination — full information lifecycle management. OAIS provides examples and some "best practice" recommendations and defines a minimal set of responsibilities for an archive to be called an OAIS [8]. These high-level services provide a vocabulary that we have adopted for the CDS Reference Model and associated Library and API.

The CDS Reference Model is a logical specification that presents a single abstract data and analytic services model to calling applications. The Reference Model can be implemented using various technologies; in all cases, however, actions are based on the following six primitives:

Ingest – Submit data to a service.
Query – Retrieve data from a service (synchronous).
Order – Request data from a service (asynchronous).
Download – Retrieve data from a service.
Status – Track progress of service activity.
Execute – Initiate a service-definable extension.

Within this OAIS-inspired framework, we are creating a Python-based CDS Library that contains methods that support the basic primitives (ingest, query, order, etc.) as well as extended utilities that combine the primitives into automated multi-step canonical ops (avg, max, min, etc.).

The Library sits atop a RESTful web services client that encapsulates inbound and outbound interactions with various climate data services. These provide the foundation upon which we have built a CDS command line interpreter (CLI) that supports interactive sessions. In addition, Python scripts and full Python applications also can use methods imported from the API. The resulting client stack can be distributed as a software package or used to build a cloud-based service (SaaS) or distributable cloud image (PaaS).

Unlike other APIs, our approach focuses on the specific analytic requirements of climate science and unites the language and abstractions of collections management with those of high-performance analytics. Doing so reflects at the application level the confluence of storage and computation that is driving big data architectures of the future.

4. MERRA ANALYTIC SERVICES

MERRA Analytic Services (MERRA/AS) pull these elements together in an end-to-end demonstration of CAaaS capabilities. MERRA/AS enables MapReduce analytics over NASA's Modern-Era Retrospective Analysis for Research and Applications (MERRA) data. The MERRA reanalysis integrates observational data with numerical models to produce a global temporally and spatially consistent synthesis of 26 key climate variables [9]. Spatial resolution is 1/2° latitude × 2/3° longitude × 72 vertical levels extending through the stratosphere. Temporal resolution is 6-hours for three-dimensional, full spatial resolution, extending from 1979-present, nearly the entire satellite era. MERRA data are typically made available to the general public through NASA Earth Observing System Distributed Information System (EOS DIS). A subset of the data is made available to the climate research community through the Earth System Grid Federation (ESGF), the research community's data publication infrastructure.

We are focusing on the MERRA collection because there is an increasing demand for reanalysis data products by an expanding community of consumers, including local governments, federal agencies, and private-sector customers. Reanalysis data are used in models and decision support systems relating to disasters, ecological forecasting, health and air quality, water resources, agriculture, climate energy, oceans, and weather.

In simple terms, our vision for MERRA/AS is that it allows MERRA data to be stored in a Hadoop Distributed File System (HDFS) on a MERRA/AS cluster. Functionality is exposed through the CDS API. The API exposures enable a basic set of operations that can be used to build arbitrarily complex workflows and assembled into more complex operations (which can be folded back into the API and MERRA/AS service as further extensions). The complexities of the underlying (Java) mapper and reducer codes for the basic operations are encapsulated and abstracted away from the user, making these common operations easier to use.

4.1. The MERRA/AS analytics platform

The Apache Hadoop software library is the classic framework for MapReduce distributed analytics. We are using Cloudera, the 100% open source, enterprise-ready distribution of Apache Hadoop. Cloudera is integrated with configuration and administration tools and related open source packages. The total size of the MERRA/AS HDFS repository is approximately 480 TB. MERRA/AS is running on a 36-node Dell cluster that has 576 Intel 2.6 GHz SandyBridge cores, 1300 TB of raw storage, 1250 GB of RAM, and a 11.7 TF theoretical peak compute capacity. Nodes communicate through a Fourteen Data Rate (FDR)

Infiniband network having peak TCP/IP speeds in excess of 20 Gbps.

The canonical operations that implement MERRA/AS's average, variance, max, min, sum, count, and difference calculations are Java MapReduce programs that are ultimately exposed as simple references to CDS Library methods or as web services endpoints. There is a substantial code ecosystem behind these apparently simple operations, nearly 6000 lines of Java code being offloaded from the user to the MERRA/AS service.

4.2. MERRA/AS in use

Our initial exposure for client applications that wish to consume MERRA/AS results is the MERRA/AS Web Service. We are using a Representational State Transfer (REST)-style architecture, which is the predominant web API design model. REST provides scalability of component interactions, accommodates intermediaries like firewalls and proxies without the need to change interfaces, and allows independent deployment of components where implementations can change without the need to change interfaces.

In one application, MERRA/AS's web service is providing data to the RECOVER wildfire decision support system, which is being used for post-fire rehabilitation planning by Burned Area Emergency Response (BAER) teams within the US Department of Interior and the US Forest Service. This capability has lead to the development of new data products based on climate reanalysis data that until now were not available to the wildfire management community.

In our largest deployment exercise to date, the CDS Client Distribution Package and the CDS API have been used by the iPlant Collaborative to integrate MERRA data and MERRA/AS functionality into the iPlant Discovery Environment. iPlant is a virtual organization created by a cooperative agreement funded by the US National Science Foundation (NSF) to create cyberinfrastructure for the plant sciences. The project develops computing systems and software that combine computing resources, like those of TeraGrid, and bioinformatics and computational biology software. Its goal is easier collaboration among researchers with improved data access and processing efficiency. Primarily centered in the US, it collaborates internationally and includes a wide range of governmental and private-sector partners.

MERRA/AS is currently in beta testing with about two dozen partners across a wide range of organizations and topic areas. Initial results have shown that analytic engine optimizations can yield near real-time performance of MERRA/AS's canonical operations and that the total time required to assemble relevant data for many applications can be significantly reduced.

CONCLUSIONS

Climate data are generally moved to client applications for analysis and use. As climate model outputs increase in size and complexity, and as customer demands for this important class of information increase, existing data practices in this domain must change. Rather than deliver data as a service, it will become necessary to deliver data analytics as a service. Our work suggests that such an approach offers great promise in efforts to address the big data challenges of the climate sciences.

REFERENCES

[1] Snijders, C., Matzat, U., & Reips, U.D. 2012. Big data: big gaps of knowledge in the field of Internet science, *International Journal of Internet Science*, Vol. 7, pp. 1-5.

[2] Edwards, P.N. 2010. A Vast Machine: Computer Models, Climate Data, and the Politics of Global Warming, Cambridge, Mass: MIT Press, 518 pp.

[3] Schnase, J.L., Duffy, D.Q., Tamkin, G.S., Nadeau, D. Thompson, J.H., Grieg, C.M., McInerney, M.A., and webster, W.P. 2014. MERRA Analytic Services: Meeting the Big Data challenges of Climate Science through Cloud-Enabled Climate Analytics-as-a-Service. *Computers, Environment and Urban Systems*, http://dx.doi.org/10.1016/j.compenvurbsys.2013.12.003 (In press).

[4] Webster, Phil. 2013. Supercomputing the Climate: NASA Big Data Mission. *CSC World*, Computer Sciences Corporation, http://www.csc.com/cscworld/publications/81769/81773-supercomputing_the_climate_nasa_s_big_data_mission.

[5] Duffy, D.Q., Schnase, J.L., Clune, T.L., Kim, E.J., Freeman, S.M., Thompson, J.H., Hunter, K.A., Theriot, M.E. 2011. Preliminary Evaluation of MapReduce for High-Performance Climate Data Analysis, AGU Fall Meeting Abstracts, 2011 AGU Fall Meeting, (San Francisco, December), http://adsabs.harvard.edu/abs/2011AGUFMIN44A..08D.

[6] Tamkin, G.S. 2013. Hadoop for High-Performance Climate Analytics, Hadoop Summit (Palo Alto, July), http://www.youtube.com/watch?v=1UMBiVlsPp0.

[7] Szalay, A. S., Kunszt, P. Z., Ani Thakar, A., Gray, J., Slutz, D., and Brunner, R. J. 2000. Designing and mining multi-terabyte astronomy archives: the Sloan Digital Sky Survey. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data* (SIGMOD '00). ACM, New York, NY, USA, pp. 451-462.

[8] Open Archive Information System (OAIS) Reference Model, http://public.ccsds.org/sites/cwe/rids/Lists/CCSDS%206500P11/Attachments/650x0p11.pdf.

[9] Rienecker, M.M., & Coauthors. 2011. MERRA: NASA's Modern-Era Retrospective Analysis for Research and Applications. *Journal of Climate*, Vol. 24, No. 14, pp. 3624–3648. Available online at http://dx.doi.org/10.1175/JCLI-D-11-00015.