Proceedings of the International Conference , "Computational Systems for Health & Sustainability"
17-18, April, 2015 - by R.V.College of Engineering,
Bangalore,Karnataka,PIN-560059,INDIA

A Study for Handelling of High-Performance Climate Data using Hadoop

Ajay K P
Department of Computer Science,
Mangalore Institute of Technology and Engineering,
Badaga Mijar,
Moodbidri-574225

Dr K C Gouda
CSIR- Centre for Mathematical modeling and Computer Simulation,
National Aerospace Laboratories,
Wind Tunnel road, Bangalore-37

Dr Nagesh H R
Department of Computer Science,Mangalore Institute of Technology and Engineering,
Badaga Mijar,
Moodbidri-574225

Abstract—Introduction of Hadoop has become the de factor for large-scale data analysis in commercial applications, and nowadays finding its prominence in scientific applications also. In climate research, where there is a need for high-performance analytics, the Hadoop MapReduce may be useful in providing solution to data intensive problems. It makes use of parallel computation for analysis paradigm that uses clusters of computers and combines distributed storage of large data sets. This paper presents the potential of MapReduce for scientific data sets which is in the NetCDF format, and performs basic operations common to a wide range of analyses. This provides a prototype for series of canonical MapReduce operations for number of observational and climate simulation datasets. Our work provides solutions on how to tackle with arbitrary spatial and temporal global climate data which is in NetCDF form. This approach can improve efficiencies within data intensive analytic workflows.

Keywords—Hadoop,MapReduce, high-performance analytics,scientific data,NetCDF.

I. INTRODUCTION

In recent years the volume of data generated by the scientific community has been increasing. The size of global observational climate data sets is growing dramatically as new missions come online. However today, science is all about big data, and making those data sensible, requires analysis tools that scale to meet the demands of scientists, and utilize resources efficiently.

In climate research, it is very important to understand the Earth's process. To achieve that, we need to combine observational data records and mathematical models which contain huge volume of data. Potentially, bigger data challenge is posed by the work of climate scientists, whose models are regularly producing data sets ofhundreds of terabytes or more.

The NASA Center for Climate Simulation (NCCS) provides data services and state-of-the-art supercomputing specificallydesigned for weather and climate research [1].The NCCSallowresearchers within and beyond NASA to createand access theenormous volume of data generated by weather and climatemodels. Tackling the problems of data intensive science is an inherent part of the any research organization mission.There are two important challenges set by the data intensivenature of climate science.

- There is a necessity to provide complete lifecycle management of large-scale scientific repositories. This laid a foundation upon which a variety ofdata services can be provided, from supporting active researchto data publication and distribution,and archival storage, large-scale data federation. In this work, we think of this aspect asour mission of climate data services.
- The next challenge is to do deal with how theselarge datasets are used: data analytics —the capacity of analyzing scientific data over enormous quantities of data,in reasonable amounts of time. In many aspects, thisis the mainstreamchallenge; without transforming largescientific data collections into meaningful scientific knowledge,our mission fails.

In climate research, MapReduce is an approach to high-performance analyticsthat may be useful to data intensive problems [2]. For huge data sets, MapReduce enables distributed computing using large number of clusters of computers. It is an analysis paradigmthat combines distributed storage and retrieval with distributed, parallel computation, allocating to the data repository analyticaloperations that yield reduced outputs to applications and interfacesthat may reside elsewhere. The implication of networking and communication plays a pivotal role in the architecture of MapReduce.For large repositories of textual data, MapReduce has proven effective, but its usage in data intensive science applicationshas been limited [3][4], because many scientific data setshave high dimensionality ,inherently complex, , and use binaryformats.

There are many methods of scientific data analysis that already exist. One popular set of tools is, NetCDF Operators (NCO), which is designed to execute common queries over data stored in theNetCDF [5] file format. Unfortunately, the scalability of NCO is limited by its design which takes a centralized approach to processing data. In NCO, all computation is performedon single node, and data is read serially from the file system. Through this, we propose a way to optimize the dataset written in NetCDF by data reduction as a sub-setting method and to process the dataset using Hadoop,a MapReduce framework. These methods can be applied to pre-processing and post-processing in scientific data experiments.

II. BACKGROUND

Data intensive analytic workflows, bridge between highly structured, reduced, tailored and refined products usedby scientists and thelargely unstructured mass of stored scientific

data and the in their research. In general, the initial steps of an analysis is those operations that first interact with a data repository, tends to be the most general, while data manipulationscloser to the client tends to be most specialized to the individual, to domain, or to the science question under study. The amount of data being operated also tends to be larger onthe repository-side of the workflow, smaller toward the client sideend products.

For example, European Soil Moisture and Ocean Salinity(SMOS)[6] satellite is been providing global maps of soil moisture forevery three days at a spatial resolution of 50km and the globalmaps of sea-surface salinity over an area of 200x200km.The NCCS manages a small subset of these data for SMOS scientists. These scientists will subsetthe data provided by the SMOS satellite and find spatial zones around the Earth that do notchange significantly over time. These zones can be used tovalidate and calibrate SMOS instruments.

The data sets from the SMOS collection aredownloaded to local work stations where they are organized, andmoved to the NCCS computers, and then analyzed using MatLab. Thistraditional type of workflow for large-scale data analysis limits the scientists and hampers their ability to look atphenomena over broader spatial extents. More importantly, this exampleshows how ,a very simple canonical operation such asfinding the average value over a portion of the Earth and measuringhow the value changes over time can be quite cumbersometo the scientists.

III. SCIENTFIC DATA EXPERIMENT FRAMEWORK

We describe an overall framework that will help us to build a MapReduce application, which supports to transform data in the field of scientific data experiments and its mainstream considerations. The framework makes composure of seamlessly integrating pre-processing and post-processing phases with data analysis application. The designed framework for the scientific data experiments do consists of three entities:

- Scientific experimental workspace,
- Science data farm, and
- Domain data repository.

In scientific experimental workspace, a researcher who has a plan of a conducting scientific data experiment defines a workflow of this experiment consisting of processes with a set of activities. Each activity corresponds to asingle process block, can be linked to another process block if control or data dependencies exist between them. An activity might be something that is straightforward as a data transformation that extracts specific values in a raw dataset. It can ease a researcher to use large-scale back-end computational resources such as cloud computing and High Performance Computing Service. In a scientific data experiment, workflow helps to organize tasks of the data experiment. When a researcher conducts a scientific data experimentlike simulation, other researcher working in some other domain may clearly define variables that can be ambiguous. In defining different phases of a scientific data experiment, ontology is referred to resolve conflict of variables unit and usage as well as ambiguity of them.

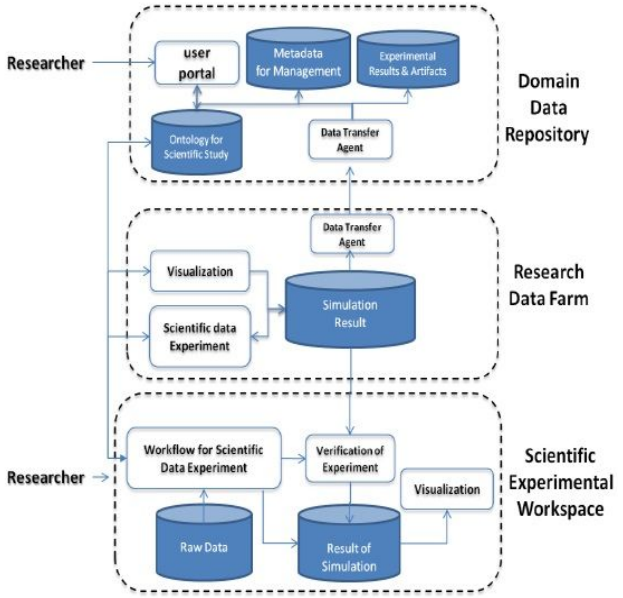

Fig 1:The overall framework for scientific data experiments.

According to workflow defined in scientific experimental workspace, a scientific data experiment is being progressed on the scientific research data farm. Then, the intermediate and final data is being moved to the domain data repository. In the domain data repository, the researchers could share data, expertise and knowledge by the user portal. The portal is designed to aggregate multiple information sources and applications to provide its users with uniform, personalized and access. Fig.1 shows the overall framework of scientific data experiments.

In the post-processing phase, the dataset generated during simulation is typically validated, and then subjected to some initial processing and formatting and annotated with appropriated auxiliary information (i.e., metadata) to form a large amount of data collection. The type of post-processing varies with the application such asgenerating multi-resolution representations of the data for visualization, or applying scientific codes to interpret the data in fusion applications. In some cases the amount of data generated in post-processing phase is equal to or greater than the original data, especially if indexes are being built to be usedin the analysis phase. Post-processing of data can be done at the multiple computational sites. Thus, the large subsets of the original data can be moved to those computational sites and can be replicated at themultiple locations. Different combinations of the data from one or more data collections have to be accessed and processed to produce the desired result. Unfortunately in many areas, bandwidths of data transfer are growing at a much slower pace, making it extremely hard for the scientists to download these rapidly growing datasets. Hence, data reduction is pretty muchnecessary to efficiently move a subset in the dataset. A typical workflow for scientific data experiments in the application structure is shown in Fig.2.

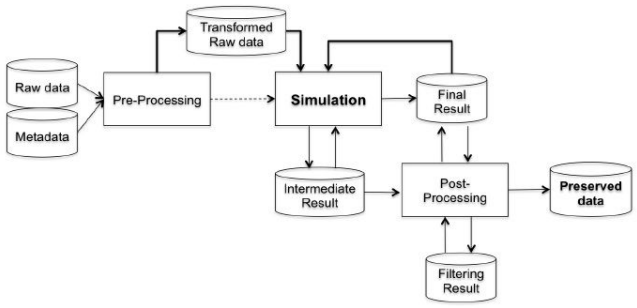

Fig 2: Basic workflow for scientific data experiments.

IV. HADOOP

Apache Hadoop [7] is an open source implementationof Google’s MapReduce frameworkwhich was described in the previous section.Hadoop supports the execution of applicationson large clusters consisting of commodity machines,where an application is divided intomany fragments of work that are executed parallelly.In addition, it provides its own distributedfile system (HDFS) [8], which was derivedfrom Google File System (GFS).HDFS can achieve very high aggregatethroughput and computational power acrossthe cluster by replicating data on DataNodes,coordinated by the Namenode. Hadoop is writtenin Java programming language and is alarge scale project maintained by Yahoo! andApache.

HDFS:

HDFS is a common example of master/slavearchitecture. The Namenode is consideredto be the master and DataNodes the slaves.DataNodes consist of data blocks, while theDataNodes themselves are grouped into Racks. The architecture of HDFS is shown in Fig.3.It should be pointed out that the defaultreplication factor of HDFS is 3. This meansthat a data block that is initially saved in asingle Datanode on a random rack is thenreplicated on two other DataNodes. One ofthem, by default, is on the same Rack with theinitial Datanode, while the second one is on adifferent Rack.The default HDFS replication policy definesthat no Datanode is allowed to contain morethan one replica of any block and no rack containsmore than two replicas of the same block,provided there are sufficient racks on the cluster

The greatest benefits of HDFS are [8][9]:

- Low cost per byte.
- Data redundancy and replication.
- Large storage capacity.
- Balanced storage utilization.
- High fault-tolerance.
- High throughput rate.
- Scalability.

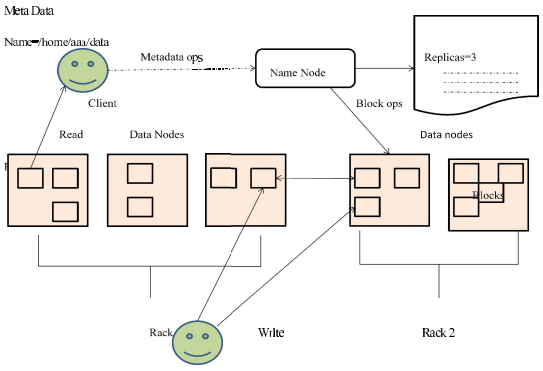

Fig 3: Architecture of HDFS

Namenode:

The role of the Namenode is the coordinationof the whole cluster. There exists onlyone Namenode per cluster, which is responsiblefor the storage and update operations ofthe namespace as well as job tracking. Thenamespace is a tree-like hierarchic structure ofthe file system, files and directories – whichmaintains a physical address mapping of datablock to the equivalent Datanode.Since every action on the cluster uses thistree, it is stored in the Name Node’s main memoryfor faster queries. Due to the fact thatthere is a single Namenode per cluster, techniquessuch as metadata images, checkpointsand backup nodes exist to guarantee fault tolerance and recovery. To sum up, the Namenode is responsible for load balancing, job tracking andnamespace handling. The latter consists of operations suchas physical address and block mapping, write,read, open, close and rename.

Datanode:

DataNodes are the cluster’s storage spaceand processing units. Every Datanode in HDFShas a unique and permanent storage identifier that is assigned to it by the Namenode, themoment it joins the cluster. The Namenodeuses this ID to recognize the DataNodes andbe able to communicate and exchange informationwith them.Once a new Datanode is connected to thecluster, the Namenode registers a unique storageID to it and the newly added node is readyfor use, as soon as the handshake between thetwo nodes is complete. The handshake is nothingmore than an ID verification process.

ADatanode regularly sends heartbeats tothe Namenode in order to confirm that it is operational. The interval between heartbeats is3 seconds by default. If no heartbeat has beenreceived by the Namenode for 10 minutes, the Data Node is deemed out of service. In theabove scenario, data stored in the unavailableDatanode are replicated to other DataNodes inorder to sustain data redundancy and fault toleranceproperties.Heartbeats, apart from notifying the Namenodeabout the availability of a Datanode,are used for carrying additional informationconcerning work load, storage capacity and utilizationas well as current data traffic. Such information combined with all the other heartbeatsreceived – allow the Namenode to makeload balancing decisions and efficient scheduling.

V. MAPREDUCE ANALYTIC WORKFLOWS

We have used the SMOS example as a foil to design a MapReduce [10]to this analysis scenario. MapReduceis a framework designed for processing highly distributable problemsacross huge data sets using a large number of computers(nodes). In a "map" operation the head node takes the input,partitions it into smaller sub-problems, and distributes them across thedata nodes. A data node may in turn do it again, leading to amulti-level tree structure. The data node will then processes the smallerproblem, and passes the answer back to reducer node to performthe reduction operation. In a "reduce" step, the reducernode will then collects the answers to all the sub-problems andcombines them to form the output — the answerto problem it was originally trying to solve. The map andreduce functions of MapReduce are both defined with respectto data structured in <key, value>pairs.

For this work, we used Modern Era Retrospective-Analysis for Research and Applications (MERRA)[11] data. MERRA uses the newest version of the Goddard Earth Observing System (GEOS)[12][13]. Data Assimilation to create a reanalysis ofthe last 30 years of observation data. The project provides aglobal view of the hydrological cycle across a broad range ofweather and climate time scales. Retrospective-analyses (orreanalysis) have been a critical tool in studying weather andclimate variability for the last 15 years. Reanalysis blend thecontinuity and breadth of the output data of a numerical modelwith the constraints of vast quantities of observational data.The result is a long-term continuous data record.

Fig.4 shows a snapshot of MERRA data displaying theseasonal averages of temperature for the winter of 2010. TheMERRA product includes 2D and 3D data for a large numberof relevant climate parameters and has the following resolutions:

- Native: 1/2 degree by 2/3 degree using the model conventions.
- Reduced: 1 1/4 degree by 1 1/4 degree, dateline-edge, andpole-edge.
- Reduced FV: 1 degree by 1 1/4 degree, using the model conventions.

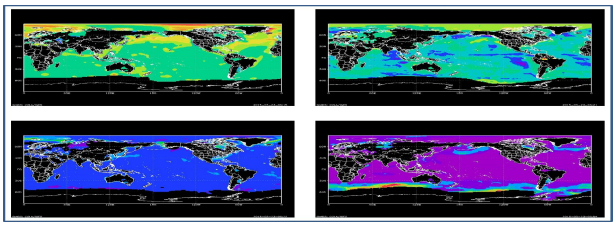

Fig 4:Example of seasonal averages of temperature for thewinter of 2010 taken from MERRA.

The use of MERRA data to compare with observations likethose provided by SMOS is increasing in the climate community day by day. Since MERRA data holdings are about 200 terabytes, the traditionalanalysis approaches are simply inadequate.

Fig.5 shows the flow of data in MapReduce processing. In MapReduce, application is supported by a MapReduce library; all the map operations can be executed independently. Each of the reduce operation may depend on the outputs generated by any number of map operations. All of the reduce operations, however, can also be executed independently.

Programming Model:

The computation takes a set of input key/value pairs, and produces a set of output key/value pairs. The user of the MapReduce library expresses the computation as two functions: Map and Reduce. Map, written by the user, takes an input pair and produces a set of intermediate key/value pairs. The MapReduce library groups together all intermediate values associated with the same intermediate key I and passes them to the Reduce function. The Reduce function, also written by the user, accepts an intermediate key I and a set of values for that key. It merges together these values to form a possibly smaller set of values. Typically just zero or one output value is produced per Reduce invocation. The intermediate values are supplied to the user's reduce function via an iterator. This allows us to handle lists of values that are too large to fit in memory.

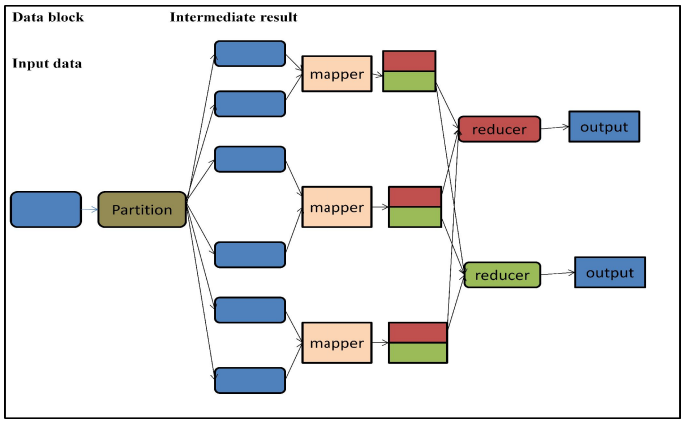

Fig 5: The flow of data in MapReduce processing.

VI. DATA REDUCTION OF DATASET

Network common file (NetCDF) is used to provide a common data access method for the atmospheric science applications to manipulate a variety number of data types that encompass single-point observations,spaced grids, time series, regularly, and radar or satellite images [14]. Atmospheric science communication can have access to datasets and transfer datasets encoded in NetCDF that is an interface for array-oriented data access and a library that provides an implementation of the interface. The NetCDF library defines a machine-independent format to representscientific data. There are many methods developed for scientific data analysis. As one of such tools, NCO is been designed to execute common statistical operations over data in the NetCDF file format. However, the scalability of NCO is been limited by the size of processing data.

The data generation phase consists of running large simulations that require the full use of computing resources for many hours. During the phase, the main concern is to store the resulting data as fast to avoid slowing the computer simulation. Datasets written in the NetCDF format may also needs to be moved to the archival storage at the same rate to make room for data from the next simulation run. For example, climate data models will generate terabytes of data per simulation run, and increase in simulation resolution will greatly increase data the storage requirements[15][16]. Even a conservative increase in granularity of the model meshes will generate a factor of 16 and more data. In the post-processing phase, the datasets generated during a simulation run is typically validated, subjected to some of the initial processing and formatting and annotated with appropriated auxiliary information. Moving such datasets isoften prohibitive; rather a subset selection and filtering of the data sets needs to be performed at the simulation site, and only results are sent to the scientist for further analysis. The NetCDF in Fig.6 represents the output of the ensemble based climate simulation and the reduced NetCDF represents a filtered output with specific variables in NetCDF. Fig.5 shows the process of data reduction in a dataset written in NetCDF as a result of a simulation. This process is used to generate the input of statistical data analysis such as MapReduce.

Fig 6:Basic Workflow of Data Reduction Process for Scientific Data Experiments.

The graph in Fig.7 plots the access time of the sub-setting in the data reduction process. After sub-setting of five variables in the original NetCDF file with 79 variables, the access time of one variable reduces up to 276 % in the part of the graph.

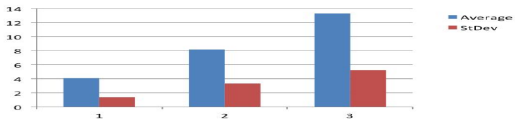

Fig 7: Speedup over sub-setting of NetCDF.

VII. BUILDING NETCDF INTO HADOOP

In order to execute MapReduce operations on the climatedata, the data first needs to be ingested into the Hadoop FileSystem (HDFS). The climate data is stored within a self describingNetCDF format that contains both data andmetadata. We considered several ways to ingest the binaryNetCDF data into the HDFS:

• Ingest NetCDF Data Unchanged: Hadoop allowsthe flexibility for any data sets to be stored within the HDFSwithout changes. However, for the binary data, Hadoopwould have no clear knowledge of how to split a binary fileinto <key, value>pairs. In order to map the data sets, a custommapper would have to be written.

• Custom Input Format Reader: If the data is storedin HDFS natively, then a custom input format reader has to be created to sequence the data as it wasread. This would impose a significant overhead whenreading the data and would have to be executed eachtime a file is opened.

• File Name as the Key:Another simple method foringesting the data into Hadoop is to translate thename of the file to be the key and the contents of the fileto be the value. Similar to above, Hadoop would nothave any insight into data itself and would not beable to intelligently place mappers in relation to the databeing processed. This would likely have a large performanceimpact due to a significant amount of the databeing transferred over the network to the mapping functions.

• Single NetCDF File to Hadoop Sequence File: Byapplying our knowledge of binary file formats, acustom sequence file can be created, so that the data islogically stored as <key, value> pairs within the resultingsequence file. For this case, a custom sequencerwould have to be written and applied to the each data fileseparately prior to ingesting the data into the HDFS.There would be a one-to-one mapping of sequencefile to the original data file. One benefit of this approachis that, the NetCDF metadata could be preserved withina file even when the file was stored in HDFS.

• Single NetCDF Variable to the Hadoop Sequence File:Similar way of generating new sequence files, the individualparameters within the NetCDF file could be separatedinto many individual files. For each original NetCDF file, this might result in many sequence files (approximately 20)that would contain one and only one variable with allvalues for that variable. However, this poses theproblem on how to keep the linkage between the datawithin the individual sequence files and the metadata ofthe original NetCDF file.

• Single NetCDF Value to the Hadoop Sequence File: Insomeanalyses, it might be useful tobreak the data up even further. The climate data couldbe broken down into a single value for each parameterat each time step and geographic location. This wouldresult in a large number of small files within the HDFSand bring up the issue of linkage between the data andmetadata. In addition, there could be serious performanceissues with this particular approach, as Hadoop is not basically designedfor high performance on a large number of verysmall files.

VIII. CONCLUSION

New scientific methods to analyze and organize data are required to handle large-scale datasets. Hence, these methods need to support effective infrastructure composed of computing resources that are used for pre-processing and post-processing data. We described the empirical studies to handle dataset associated with a scientific data experiment which supports data reduction and data transformation, which are an essential phase to handling large scale data in scientific data experiments. This approach introduces a new way to process raw data by Hadoop, a MapReduce framework in a parallel manner and for high-performance climate analysis in efficient

way and for future perspective; it can further be extended to integrating HDFS with a virtualized climate data service [17]

REFERENCES

[1]. NASA Center for Climate Simulation (NCCS), http://www.nccs.nasa.

[2]. J. Dean and S. Ghemawat, MapReduce: Simplified Data Processing on LargeClusters,Google,Inc., http://research.google.com/archive/mapreduce.html

[3]. J. Buck, et al., SciHadoop: Array-based Query Processing in Hadoop, UC Santa Cruz, https://systems.soe.ucsc.edu/node/439.

[4]. C. Ranger, et al., Evaluating MapReduce for Multi-core and Multiprocessor Systems, Computer Systems Laboratory, Stanford University.

[5]. Network Common Data Form(NetCDF). http://www.unidata.ucar.edu/software/netcdf.

[6]. Soil Moisture and Ocean Salinity (SMOS) Satellite, http://www.esa.int/SPECIALS/smos/index.html.

[7]. Apache, http://hadoop.apache.org/.

[8]. Konstantin Shvachko,Hairong Kuang,Sanjay Radia,Robert Chansler,2010,The Hadoop Distributed File System,IEEE 2010.

[9]. Abhishek Sagar,Umesh Bellur,2011,Distributed Computation on Spatial Data on Hadoop Cluster,Dept. of Computer Science and Engineering Indian Institute of Technology,Bombay.

[10]. MapReduce, http://en.wikipedia.org/wiki/MapReduce.

[11]. Modern Era Retrospective-Analysis for Research and Applications(MERRA), http://gmao.gfsc.nasa.gov/merra.

[12]. Goddard Earth Observing System (GEOS), http://gmao.gsfc.nasa.gov/systems/geso5.

[13]. Global Modeling and Assimilation Office (GMAO), http://gmao.gsfc.nasa.gov/.

[14]. Ncar, http://www.unidata.ucar.edu/software/netcdf/.

[15]. http://gcmd.nasa.gov/records/UCAR_WACCM.html.

[16]. http://www.cesm.ucar.edu/models/atm-cam/.

[17]. J. Schnase, et al., The Virtual Climate Data Server (vCDS): An iRODS-Based Data Management Software Appliance Supporting Climate Data Services and Virtualization-as-a Service in the NASA Center for Climate Simulation,2012 iRODS users Group Meeting,in review.

