Chapter 12
GeoBase: Indexing NetCDF Files for Large-Scale Data Analysis

Tanu Malik
University of Chicago, USA

ABSTRACT

Data-rich scientific disciplines increasingly need end-to-end systems that ingest large volumes of data, make it quickly available, and enable processing and exploratory data analysis in a scalable manner. Key-value stores have attracted attention, since they offer highly available data storage, but must be engineered further for end-to-end support. In particular, key-value stores have minimal support for scientific data that resides in self-describing, array-based binary file formats and do not natively support scientific queries on multi-dimensional data. In this chapter, the authors describe GeoBase, which enables querying over scientific data by improving end-to-end support through two integrated, native components: a linearization-based index to enable rich scientific querying on multi-dimensional data and a plugin that interfaces key-value stores with array-based binary file formats. Experiments show that this end-to-end key-value store retains the features of availability and scalability of key-value stores and substantially improves the performance of scientific queries.

INTRODUCTION

Advances in remote sensing technology have significantly lowered the cost of data acquisition via satellites and aircrafts. Well-known satellites, such as GOES (NOAA), Landsat (NASA), and Aqua/Terra (NASA) continuously stream 20-60 GB of remotely sensed image datasets to receiving stations per day. To use these datasets for analysis, scientists typically determine relevant datasets through a metadata search. Given the large data volumes, a metadata search, however, is increasingly becoming insufficient in retrieving datasets of interest. Content-based searches, such

DOI: 10.4018/978-1-4666-4699-5.ch012

as determining ranges of pressure or temperature within the data files, can improve retrieval. To conduct these searches, however, currently scientists have to download datasets, and perform data management tasks, such as format conversions, pre-process the datasets, index, and visualize, and then subset the content. It is not uncommon for scientists to work with many different systems to perform the entire data management task, Alternatively, it is desirable to have an *end-to-end* system that ingests datasets in their native format, indexes them as part of the ingestion process, and provides a simple API for content-based searches.

We are interested in designing such an end-to-end system for geoscience datasets. In several sub-domains of geosciences, end-to-end issues are addressed through data access libraries. For instance most geoscience datasets are stored in self-describing formats, such as NetCDF (Rew 1990), and HDF5 (Folk 1999), and are made available for exploration and analysis through the NetCDF-Java API or HDF5 API. These libraries address some end-to-end issues in that they hide the nitty-gritty details specific to their format. They, however, do not address other end-to-end issues. In particular, the libraries do not provide a mechanism for indexing the datasets or the ability to conduct content-based analysis on multiple datasets in parallel.

Another alternative is to explore data through parallel file systems tailored for NetCDF files (Li 2003), but these file systems introduce other end-to-end issues, namely, communication between high-level data analysis operators that must be controlled by the programmer (Buck 2011). Datasets can also be explored by using geospatial databases, such as PostGIS Raster and MySQL. Most database systems, however, require datasets resident in a file system to be imported into their native format. For instance, raster images can be ingested into the databases through the Geospatial Data Abstraction Library (GDAL), but bulk loading of large volumes of data is known to be slow. For analysis, the storage model in these traditional databases is row-oriented, i.e., they store multi-dimensional array data as a single record. They do not leverage the performance benefit derived from a column-oriented design, which has shown to provide orders of magnitude of performance improvement (Abadi 2008).

For geoscience applications, a useful alternative is distributed key-value stores, e.g., BigTable (Chang 2008) and their open-source counterparts, e.g., HBase (Apache 2010) that are built on a parallel storage framework and have shown to scale to millions of updates while being fault-tolerant and highly available. Key-value stores, however, do not address all end-to-end issues in data exploration of geoscience datasets. In particular, key-value stores do not natively support files in self-describing formats, which are commonly used in geoscience disciplines. They also do not provide for efficient exploration of multi-dimensional datasets. Lack of support for self-describing files leads to format transformation, and lack of support for multi-dimensional data leads to inefficient indexing, and additional data management that soon become too cumber- some given the size of common scientific data sets.

In this chapter, we present GeoBase an end-to-end system that enables efficient exploratory data analysis on geoscience datasets without incurring any additional data transformation costs. To achieve this vital goal, GeoBase uses space-filling curves to store multi-dimensional spatio-temporal data in a one-dimensional key-value store. In adopting space-filling curves as an indexing technique, GeoBase does not introduce any data transformation costs that are often implicit in the linearization process. GeoBase intercepts requests for the sub-queries on the space-filling curve and corresponds them to contiguous, low-level byte extents at the physical level. Through this modification, data can continue to reside in