Geophysical Research Abstracts
Vol. 12, EGU2010-12147-2, 2010
EGU General Assembly 2010

Database Objects vs Files: Evaluation of alternative strategies for managing large remote sensing data

Chaitan Baru, Viswanath Nandigam, and Sriram Krishnan

University of California, San Diego, San Diego Supercomputer Center, La Jolla, CA, United States (baru@sdsc.edu)

Increasingly, the geoscience user community expects modern IT capabilities to be available in service of their research and education activities, including the ability to easily access and process large remote sensing datasets via online portals such as GEON (www.geongrid.org) and OpenTopography (opentopography.org). However, serving such datasets via online data portals presents a number of challenges. In this talk, we will evaluate the pros and cons of alternative storage strategies for management and processing of such datasets using binary large object implementations (BLOBs) in database systems versus implementation in Hadoop files using the Hadoop Distributed File System (HDFS).

The storage and I/O requirements for providing online access to large datasets dictate the need for declustering data across multiple disks, for capacity as well as bandwidth and response time performance. This requires partitioning larger files into a set of smaller files, and is accompanied by the concomitant requirement for managing large numbers of file. Storing these sub-files as blobs in a shared-nothing database implemented across a cluster provides the advantage that all the distributed storage management is done by the DBMS. Furthermore, subsetting and processing routines can be implemented as user-defined functions (UDFs) on these blobs and would run in parallel across the set of nodes in the cluster. On the other hand, there are both storage overheads and constraints, and software licensing dependencies created by such an implementation.

Another approach is to store the files in an external filesystem with pointers to them from within database tables. The filesystem may be a regular UNIX filesystem, a parallel filesystem, or HDFS. In the HDFS case, HDFS would provide the file management capability, while the subsetting and processing routines would be implemented as Hadoop programs using the MapReduce model. Hadoop and its related software libraries are freely available.

Another consideration is the strategy used for partitioning large data collections, and large datasets within collections, using round-robin vs hash partitioning vs range partitioning methods. Each has different characteristics in terms of spatial locality of data and resultant degree of declustering of the computations on the data. Furthermore, we have observed that, in practice, there can be large variations in the frequency of access to different parts of a large data collection and/or dataset, thereby creating "hotspots" in the data. We will evaluate the ability of different approaches for dealing effectively with such hotspots and alternative strategies for dealing with hotspots.