Geophysical Research Abstracts
Vol. 17, EGU2015-7570, 2015
EGU General Assembly 2015

Array Processing in the Cloud: the rasdaman Approach

Vlad Merticariu and Alex Dumitru
Jacobs University Bremen, Germany (v.merticariu@jacobs-university.de)

The multi-dimensional array data model is gaining more and more attention when dealing with Big Data challenges in a variety of domains such as climate simulations, geographic information systems, medical imaging or astronomical observations. Solutions provided by classical Big Data tools such as Key-Value Stores and MapReduce, as well as traditional relational databases, proved to be limited in domains associated with multi-dimensional data. This problem has been addressed by the field of array databases, in which systems provide database services for raster data, without imposing limitations on the number of dimensions that a dataset can have. Examples of datasets commonly handled by array databases include 1-dimensional sensor data, 2-D satellite imagery, 3-D x/y/t image time series as well as x/y/z geophysical voxel data, and 4-D x/y/z/t weather data. And this can grow as large as simulations of the whole universe when it comes to astrophysics.

rasdaman is a well established array database, which implements many optimizations for dealing with large data volumes and operation complexity. Among those, the latest one is intra-query parallelization support: a network of machines collaborate for answering a single array database query, by dividing it into independent sub-queries sent to different servers. This enables massive processing speed-ups, which promise solutions to research challenges on multi-Petabyte data cubes. There are several correlated factors which influence the speedup that intra-query parallelisation brings: the number of servers, the capabilities of each server, the quality of the network, the availability of the data to the server that needs it in order to compute the result and many more. In the effort of adapting the engine to cloud processing patterns, two main components have been identified: one that handles communication and gathers information about the arrays sitting on every server, and a processing unit responsible with dividing work among available nodes and executing operations on local data.

The federation daemon collects and stores statistics from the other network nodes and provides real time updates about local changes. Information exchanged includes available datasets, CPU load and memory usage per host.

The processing component is represented by the rasdaman server. Using information from the federation daemon it breaks queries into subqueries to be executed on peer nodes, ships them, and assembles the intermediate results.

Thus, we define a rasdaman network node as a pair of a federation daemon and a rasdaman server. Any node can receive a query and will subsequently act as this query's dispatcher, so all peers are at the same level and there is no single point of failure. Should a node become inaccessible then the peers will recognize this and will not any longer consider this peer for distribution. Conversely, a peer at any time can join the network.

To assess the feasibility of our approach, we deployed a rasdaman network in the Amazon Elastic Cloud environment on 1001 nodes, and observed that this feature can greatly increase the performance and scalability of the system, offering a large throughput of processed data.