Evaluation and demonstration of Array DBMSs using national geospatial data

Draft 24.10.2017

Janne Kovanen
Ville Mäkinen
Tapani Sarjakoski
Finnish Geospatial Research Institute FGI in the National Land Survey of Finland
Emails: janne.kovanen@nls.fi, ville.p.makinen@nls.fi, tapani.sarjakoski@nls.fi

Executive Summary

The project evaluates and demonstrates the usage of multidimensional array database management systems (DBMSs) for storing and retrieving geospatial information and carrying out simple geospatial analysis tasks. The evaluations are based on the criteria set by the RDA Array Database Assessment WG and domain-specific benchmarks created by us.

The evaluation was based on the non-functional, functional, and performance tests. In short, SciDB is more mature as a generic DBMS and especially suitable for running analysis. SciDB may be used to store and access geographic data by implementing an access layer, which, for instance, supports coordinate reference systems. rasdaman, for its parts, is best suited for storing geospatial raster data through its web interface (Petascope); however, the data should be pretty dense and without non-zero null values. Memory availability affects both systems; in SciDB data ingestions need to be divided into memory-fitting parts whereas in rasdaman analysis requires that the associated data fits into memory.

The applicability of the criteria defined by RDA Array Database Assessment WG for in the evaluation of array DBMSs is still limited until the criteria have reached a more mature and stable status.

Objectives

The project evaluates and demonstrates the usage of multidimensional array database management systems (DBMSs) for storing and retrieving geospatial information and carrying out simple geospatial analysis tasks. The evaluations will be based on the criteria set by the RDA Array Database Assessment WG and domain-specific benchmarks created by us.

Initial State

A wide variety of papers regarding the particular array DMBSs have been published. In addition, the DMBSs are being actively developed, and releases are made on a regular basis. Two of the particular DBMSs, the array DBMSs, have been earlier evaluated on a single node up to 10 GB input data size concerning the ingestion and querying the data. This has been reported in *G. Merticariu, D. Misev, and P. Baumann. Towards a general array database benchmark: Measuring storage access. In T. Rabl, K. Sachs, M. Poess, C. Baru, and H.-A. Jacobson, editors, Big Data Benchmarking, Lecture Notes in Computer Science, 5th International Workshop, WBDB 2014, Potsdam, Germany, August 5-6-2014, volume 8991, pages 40–67. 2016*. The evaluation was, however, not performed in a cluster environment and did not include in-database analysis, test ingestion of pre-split data or multi-step ingestion, or use real geospatial data.

RDA Array Database Assessment WG has defined a set criteria to be used in the evaluation of array DBMSs. These criteria are under continuous development and were only preliminary form when this subproject was started.

Project Outcomes

The project reports two deliverables: 1) A report of the evaluation will be written at the end of the project, 2) A demonstration set up: Public access to the clusters of evaluated array database technologies will be made available. Access is available until the end of this subproject. The source data and setup files required for the demonstration set up will be made available online for download until end of February 2022.

Non-functional comparison

We created a criteria list based on general non-functional requirements. As a lot of non-functional requirements cannot be compared to each other, we used here a fuzzy rating based on use of the DBMSs.

Functional comparison

We created a criteria list based on the template[1] made available by the Array Database Assessment WG. The criteria list was written to be composed of the following main categories:

- general DBMS capabilities,
- data model & schemas,
- processing model, and
- geospatial capabilities

The last category is domain-specific but as the domain is closely related to array databases it was thought suitable to be added. Next a system assessment was performed system-wise against the criteria list. The assessment was performed for the rasdaman community and SciDB Community Edition (CE).

The assessment was initially performed against the documentation. Next, some of the functionalities were validated to be really available using test cases. At the same time, the test cases created a framework that demonstrates how the databases work. The test cases can later on be extended for further functionalities and run against other database instances.

The test cases were written with Python 2. The test cases do not include negative test cases. Neither do they contain test cases that corrupt the database or concern functionality that is found in the documentation but not in the implementation. Such cases were handled by writing down small error reports in the end of the assessment.

The test cases were implemented using the interfaces of the DBMSs. For SciDB this meant using its shim interface and for rasdaman its rasql web interface, which is a shim. For SciDB, this restricted the test cases to be run only with AFL and not AQL. The test cases do not need external data; instead, the data is loaded to the servers from the

[1] https://www.rd-alliance.org/group/array-database-assessment-wg/wiki/array-systems-assessment

notebook. Hence, for rasdaman, we also needed to implement a web service to put the input data to the server side.

The test cases can be run using Jupyter Notebook and basically against any SciDB and rasdaman implementation. The servlet source code is as attachment; the servlet just needs to be packaged with maven, the web archive renamed to correspond to the name defined in the Jupyter Notebook, and, finally, put to a servlet container.

Performance comparison

A set of queries was written for both DBMSs. The queries were written to use two data sets: the digital elevation model of Finland in ten meter resolution (2D data) and CORINE land cover data of 2000-2012 concerning Finland resampled to five meter resolution (3D data). The queries were run using bash scripts and timed on the server side. The queries are listed in DBMS-specific Jupyter Notebooks, which are as attachment. The performance comparison was made with basic configurations; a possibility is later to run the test cases against different configurations.

Summary & Conclusions

This is an evaluation based on the non-functional, functional, and performance tests. In short, SciDB is more mature as a generic DBMS and especially suitable for running analysis. SciDB may be used to store and access geographic data by implementing an access layer, which, for instance, supports coordinate reference systems. rasdaman, for its parts, is best suited for storing geospatial raster data through its web interface (Petascope); however, the data should be pretty dense and without non-zero null values. Memory availability affects both systems; in SciDB data ingestions need to be divided into memory-fitting parts whereas in rasdaman analysis requires that the associated data fits into memory.

The applicability of the criteria defined by RDA Array Database Assessment WG for in the evaluation of array DBMSs is still limited until the criteria have reached a more mature and stable status.

Attachments

- Functional comparison and evaluation of the SciDB and rasdaman (rda-systems-evaluation.pdf)
- Jupyter notebook containing queries that have been used to validate some of the statements concerning functionality and that acts as a demonstration setup (rda_master.ipynb). The file can be viewed without running an Jupyter Notebook server from https://nbviewer.jupyter.org/url/86.50.170.177/rda/rda_master.ipynb?flush_cache=true
- Web servlet for the validation of the rasdaman servers using the Jupyter notebook (upload-servlet.tar.gz)
- Jupyter notebook containing scripts that were used to ingest data and query data from the SciDB database (rda_scidb_scripts.ipynb). The file can be viewed without running an Jupyter Notebook server from https://nbviewer.jupyter.org/url/86.50.170.177/rda/rda_scidb_scripts.ipynb?flush_cache=true
- Jupyter notebook containing scripts that were used to ingest data and query data from the rasdaman database (rda_rasdaman_scripts.ipynb). The file can be viewed without running an Jupyter Notebook server from

https://nbviewer.jupyter.org/url/86.50.170.177/rda/rda_rasdaman_scripts.ipynb?flush_cache=true