SciDB: An Array Database

PIYUSH RAI[1]

[1]School of Informatics and Computing, Bloomington, IN 47408, U.S.A.
*Corresponding authors: piyurai@iu.edu
+HID - S17-IO-3014

SciDB is an open-source array database developed and maintained by Paradigm4. This paper explores it's archtectural overview and it's features designed to optimize the array data processing.

Keywords: Array Database, SciDB

https://github.com/piyurai/sp17-i524/tree/master/paper2/S17-IO-3014/report.pdf

---

1. INTRODUCTION

SciDB is an open source DBMS based on multi-dimensional array data model and runs on Linux platform. The data store is optimized for mathematical operations such as linear algebra and statistical analysis. The data is organized into arrays which can be stored in different forms including vectors and matrices with general form being n-dimensional arrays. It can be used to used to analyze scientific data from various data acquisition devices like sensors and wearables [1]. Array databases are optimized for homogeneous data models which are found in the field such as IOT and earth and space observation.

2. ARCHITECTURAL OVERVIEW

The array and their dimensions are named. The data tuple stored in each cell contain values for one or more attributes which are named and typed. The operations such as to filter and join arrays and aggregation over the cell values are supported. The arrays can be partitioned and per-partition values such as sum can be computed. Operations on large sized arrays could be performed without requiring them to fit in-memory. It also has 'missing data' facilities to reduce noise and impute missing values in the data [1].

The arrays are divided into chunks and partitioned across the nodes in the cluster, with provision of caching some of them in the main memory. SciDB uses shared-nothing architecture. Each node in the cluster has it's own resources such as memory, storage and CPU. The nodes can be added to or removed from the cluster without stopping the system. SciDB is ACID compliant. It uses Multiversion Concurrency Control (MVCC) to retain old values of data and maintains log of the changes in database state over time. In case the data history isn't required, the facility of purging the redundant data is also available.

Scalability and parallelism are the core requirements behind the design philosophy. Multiple machines connected over a network cluster are running server software instances. Disjoint subsets of data are stored locally on each machine. The input query is divided into smaller components which is applied by each instance to it's locally stored data. The information regarding which data is stored at what node is maintained by using Multidimensional Array Clustering (MAC) [2]. Data values close to each other are stored in the same chunk of storage media for faster access of range selects. It helps to minimize data movements among nodes. An optional user-settable chunk overlap can also be employed which enhances the performance of window queries that straddle two chunks. SciDB maintains a liveness-list of functioning instances which keeps getting updated as and when the nodes are added or removed from the cluster.

3. EXTENSIBILITY

To be able to cover different applications for scientific computations, the existing computing functionality can be extended by embedding user-defined functions and algorithms into the system. It uses registries to maintain information about data such as types and functions available in the system. This information is used to break down the query into sequence of operations. A number of client APIs are provided so that the system can be interfaced with tools like R and Python. With SciDB-R and SciDB-Py libraries, data in SciDB can be referenced as data frame objects.Bulk and parallel loading is supported for various formats such as binary and text.

4. ALTERNATIVES

SciDB tries to overcome the problem of scalability with traditional RDBMS and and lack of ACID support in NOSQL databases [3]. There are a few other array databases such as Rasdaman database. Rasdaman is one of the oldest array databases to be rolled out [4]. It's open source and it's source code is easily available. It's also available in form of VM image. However,

it's source code is divided into two communities, one open-source and other enterprise. SciDB provides Amazon Machine Images to try out the database. SciDB project also started as an open source project. However, instructions and source code for installing SciDB are available at [5] which can be accessed only after registration. SciDB has an in-depth documentation available explaining it's core features and design.

5. CONCLUSION

SciDB is designed for scalability and performance while at the same time complying with ACID properties. It also employs multiple methodologies to enhance the performance of the queries by dividing the data across multiple nodes and to reduce the data flow among nodes. It also provides the facility to embed user defined functions and algorithms into the system so that large variety of big data applications requiring scientific computation on homogeneous collection of data can benefit from it. SciDB is developed and supported by Paradigm4.

REFERENCES

[1] Paradigm4, "Architectural summary and motivation for paradigm4's scidb," Paradigm4, Whitepaper. [Online]. Available: http://discover.paradigm4.com/scidb-database-for-21st-century.html

[2] "Multidimensional array clustering," Web Page, 2017.

[3] J. Vaughan, "High programming overhead for nosql," blog, dec 2011. [Online]. Available: http://searchmicroservices.techtarget.com/blog/Microservices-Matters/Stonebraker-sees-high-programming-overhead-for-NoSQL

[4] "Rasdaman - impact and uses," Web Page.

[5] A. Poliakov, "Index of scidb releases," Web Page, mar 2015. [Online]. Available: http://forum.paradigm4.com/t/index-of-scidb-releases/773