FASTDB: An Array Database System for Efficient Storing and Analyzing Massive Scientific Data

Hui Li[1,2(✉)], Nengjun Qiu[1,2], Mei Chen[1,2], Hongyuan Li[1,2],
Zhenyu Dai[1,2], Ming Zhu[3], and Menglin Huang[3]

[1] Department of Computer Science,
Guizhou University, Guiyang 550025, China
{cse.HuiLi, gychm, cse.zydai}@gzu.edu.cn
[2] Guizhou Engineering Laboratory of ACMIS, Guiyang 550025, China
[3] National Astronomical Observatories,
Chinese Academy of Sciences, Beijing 100016, China
{mz, huangmenglin}@nao.cas.cn

Abstract. With the development of science and technology, the data size and complexity of scientific data are increased rapidly, which made efficient data storage and parallel analysis of scientific data become a big challenge. The previous techniques that combine the traditional relational database with analysis software tends cannot efficiently meet the performance requirement of large scale scientific data based analysis. In this paper, we present FASTDB, a distributed array database system that optimized for massive scientific data management and provide a share-nothing, parallel array processing analysis. In order to demonstrate the intrinsic performance characteristics of FASTDB, we applied it into the interactive analysis of data from astronomical surveys, and designed a series of experiments with scientific analysis tasks. According to the experimental results, we found FASTDB can be significantly fast than traditional database based SkyServer in many typical analytical scenarios.

Keywords: Scientific database system · Array database · Scientific analysis · Performance measurement · Massive data

1 Introduction

With the development of science and technology, the volume of scientific data approximately doubling every year. The relational database system have been widely used and proved its values in various areas. However, as the scientific data has the characteristic of high noise, massive computation and often prefer in array model in nature, RDBMS tends cannot to be the satisfactory solution to achieve scalable analysis and storage for data-intensive scientific areas [1, 2], e.g., astronomy survey like the Sloan Digital Sky Survey (SDSS) [3, 23], which aiming at creating a digital map of a big part of the universe by spectroscopic data. There are urgent needs to develop efficient and scalable array model based scientific database systems for scientific areas.

Given a concrete example, consider data from the five hundred meters Aperture Spherical radio Telescope (FAST) [4], the world's largest astronomical radio telescope.

© Springer International Publishing Switzerland 2015
G. Wang et al. (Eds.): ICA3PP 2015 Workshops, LNCS 9532, pp. 606–616, 2015.
DOI: 10.1007/978-3-319-27161-3_55

It can make the neutral hydrogen observation extends to the edge of the universe, make dark matter and dark energy observations, enable the arrive time precision of pulsar from the current 120 ns to 30 s and search for interstellar communication signal and extraterrestrial civilizations, etc. The data such as the strength of the radio celestial objects, spectrum and polarization data deriving from FAST has multidimensional characteristics that cannot matched to traditional relational database. For example, the process of cross-identification of multi-band radio data would merge multiple data sets, then it will become more complicated when it comes to the matrix inverse operation.

To meet the requirements of large-scale scientific data storage and analysis, the array model based database system have been proposed. It often provides with inherent support for multi-dimensional arrays and opt for the array as the first-class citizen and it is designed to be a share-nothing architecture [5–7].

In this paper, we present the analytical database system FASTDB [11], which builds on array model based SciDB engine [8–10] to provide a share-nothing, parallel processing and data management engine for needs of FAST [4]. In the internal of database engine layer, we hacked its implementation to enhanced SciDB's adaptive storage capability. Furthermore, we optimized its parallel loading capability and multiple join implementation, which can enable efficient complex astronomical data analysis such as cross-identification.

2 Related Work

In this decade [1, 2], there has been a trend to employ array model as the data schema to exploit large-scale scientific data management and analysis. SciDB [8–10] is an representative open source array database system proposed for management and analysis of scientific data. It is mainly developed Paradigm4 Inc [12]. It uses the array data model that like the array structure in mathematics, and it provide good support for multidimensional scientific data processing.

There several major technical featrues make SciDB engine is optimized to scientific areas by design. At first, there is no overwrite during the update. SciDB store different version of data and uses timestamp as the symbol of different historical data. At second, data compression algorithms are employed to save storage space. At third is Named Versions, in SciDB, changing the array will produce a new version of data. At fourth, it support the characteristic of data provenance, which can meet the need of data repeatability. The last is its capability for manage data uncertainty. Scientific data, especially the scenarios that data obtained by certain observational approaches, which often contain some inaccurate information. To solve this problem, SciDB's built-in mechanism allows data has error or approximation.

A typical SciDB array structure is illustrated as Fig. 1. Its basic unit is cell, each cell has the same value types. The value of cell can be one or more scalar value, also can be one or more array. Each dimension has contiguous integer values. SciDB uses the message mechanism for communication between nodes and uses array as the first-class citizen. Thus SciDB often to be efficient for processing large array data.

Generally, the operations scientist conducted over scientific data can be categorized into two types: data management and data analysis. The common data management task

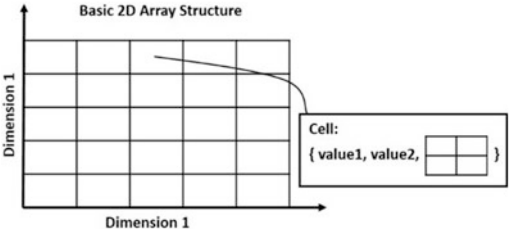

Fig. 1. Example of 2D array

of scientific data including the process of ingest, cook, group of raw data, etc. In scientific domain, scientist can obtain the value of the raw data through a series of processing, such as remove noise in the raw imagery, multiband fusion, cross-identification. The analysis of scientific data involves computing and statistics from the processed data, such as computing the distances to celestial objects.

In order to support efficient parallel query processing of large-scale astronomical celestial bodies, the university of Washington developed AscotDB system [13, 14] for large-scale astronomical data management and analysis. It take the SciDB cluster as the backend database engine and employ SciDB-Py package as system interface, which allow user to connect SciDB database using Python language.

Rasdaman (raster data manager) [15, 16] is an open source array database system originally developed by the Jacobs University Bremen and Rasdaman Inc [17]. It is fast and flexible, and have been successfully applied in geoscience and other scientific areas. MonetDB [18, 19] is another representative array-oriented database system. It is designed to provide high performance of the query support for large-scale scientific data, it can also be applied in analytical areas such as data mining, on-line analytical processing, text retrieval and multimedia retrieval.

Since MapReduce has been a trend for distributed processing, there also some analytical solution proposed for scientific data analysis by leverage the Hadoop system [20–22]. Howerver, since the complexity of analytics in scientific areas are go well beyond the MapReduce frameworks, and matching all the functionality delivered by array database like SciDB would require many Hadoop ecosystem components, including HDFS, PIG, HBase, Hive and analytical package like R, which will reslut the technical solution tends to be more complexity. Therefore, scientific data management and analysis community often prefere the array database system as their solution.

3 Overview of FASTDB

The FASTDB project is a collaboration of an inter-disciplinary team comprising astronomy and database researchers. The architecture of FASTDB is shown in Fig. 2. FASTDB is a share-noting distributed system for massive scientific data management and analytics. FASTDB integrates several pieces of technology: the system interface, the optimized SciDB engine for data storage and processing, a cluster monitor system and a front-end UI for interactive and exploratory data analysis. Since the FASTDB

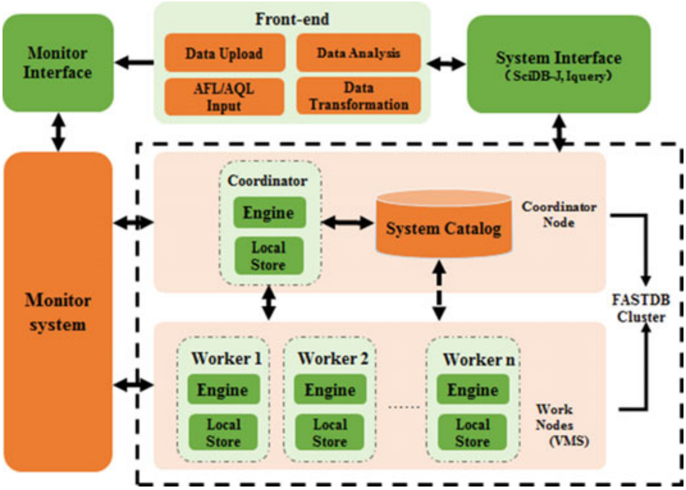

Fig. 2. Architecture of FASTDB

engine is implemented based on SciDB engine, we just present our enhancement and the difference of them, and ignore the rest of them which are identical.

System Interface: The system interface is the bridge between the interactive front-end and underlying database cluster, it is implement based on the command line interface Iquery and JDBC driver SciDB-J. In Iquery, both the functional-style array query language AFL and the SQL-like array query language AQL are can be used to interact with underlying database cluster. As to SciDB-J, we optimized its implementation to support error handling and enhanced its support for AFL.

Interactive Front-end UI: It provide the capability to visually interact and explore data. Since FASTDB is initiated for meet the needs of FAST [4], currently, all the functionality of the front-end are developed for astronomical data exploration like SkyServer [3, 23]. In the front-end of FASTDB, user can upload their own data into FASTDB cluster and then analyze it by either AFL/AQL statements or the built-in interactive visual explorative tools.

Monitor: The monitor subsystem consists of two components: agent and server. Each FASTDB node will be installed and configured the monitor agent. Via SNMP, the agent ends can collect various fine-grained system information of cluster server. The monitor server will analyzing the incoming monitoring data by preconfigured rules, and send out event alarms if necessary. All the monitored information, including the reports, static data and the configuration will be shown in a web based dashboard, which is convenient for users to know about the system information of the server.

Parallel Data Loading: FAST [4] will produce tens of data every day, which require FASTDB can load most of them into system for archive and further analysis. In order to achieve this goal, we optimization the implementation of SciDB's data load component as a new subsystem named FASTLoad. Its architecture is as Fig. 3.

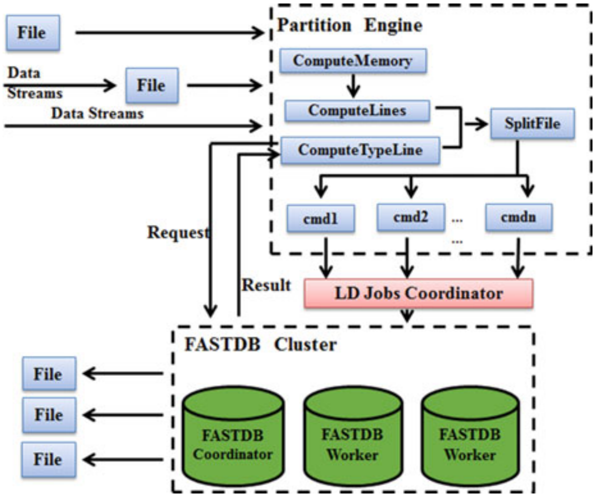

Fig. 3. Architecture of FASTLoad subsystem

When the FASTLoad system load data into system, the Partition Engine split the file into n parts with specific sizes. n is determined by the monitor (it know which FASTDB worker is busy) and FASTDB Coordinator, it means there will n FASTDB worker. Then the task will scheduled by LD Jobs Coordinator to load the corresponding file later. Currently, we only use the simple FIFO strategy for job schedule in LD Jobs Coordinator. That means the loading order is same as the order of data file loading request received by the FASTDB Coordinator.

Adaptive Chunk Determination: Chunk is the basic storage unit of both SciDB and FASTDB. In SciDB and other array database systems, they primarily employ the fixed length strategy for chunk segmentation. If the length of the chunk is too big, it will increase the overhead of cell location during query analysis. On the contrary, it will make too many small chunks and increase the memory overhead. In the implementation of FASTDB, we developed an adaptive chunk length determination (CLD) algorithm to improve the performance of array database by reducing the number of track and prefetching technique when the data is read. Due to the space limitation, we don't present it in this paper, the detail of our CLD approach and its performance evaluation can be found in literature [11].

Query Optimizer: Database engine's optimizer aims produce better execution plan to achieve performance goal. As traditional database system, optimize the execution of join operators will also lead a performance improvement in SciDB or FASTDB. In the state-of-the-art SciDB 14.12 and the coming new edition, the join of SciDB is implement as a “cross-join” operator, and it only have several heuristic based rule for join optimization, which is far beyond the needs of produce good enough execution play for complex analytical task. In FASTDB, we developed an array statistics based cost-based optimization to solve this problem. Our evaluation in literature [11] proved that, towards the analytical task has multiple array joins, the execution plan produced by our optimization can reduce 40%–60% overheads than SciDB.

4 Design of Experimental Study

In order to characterize the performance between FASTDB the representative astronomical database system SkyServer [3], we designed a micro benchmark for performance evaluation. This benchmark are based on the real dataset of SDSS (Sloan Digital Sky Survey [24]), it consist of three types of astronomy analytical tasks from [25], which including 8 representative analytical queries:

Q1: Find the Objects with movement speed equal to 1336, field equal to 11 from PhotoObj table of SDSS.
Q2: Find the galaxies that its luminosity less than 22 and local extinction is greater than 0.175.
Q3: Find galaxies in a given area of the sky, using a coordinate cut in the unit vector cx, cy and cz.
Q4: Search for Cataclysmic variables and pre-CVs with white dwarfs and very late secondary from PhotoPrimary table.
Q5: Find quasars from the Star table.
Q6: It is a query with single Join and aims to find galaxies that are blended with a star and output the deblended galaxy magnitudes.
Q7: Find all objects within 30 s one another that have very similar colors.
Q8: Search for merging galaxy pairs, as the prescription in [25].

In above analytical tasks, Q1–Q5 is belong to the first type of task, which consist of several statements in the form of “SELECT * FROM * WHERE *”. Q1 is to find objects in a particular field while Q2 is to find all galaxies with a special value of r and extinction_r. Q3, Q4, Q5 are to find the Objects in a given area of the sky. Q6 is belong to the second type task, which are composed by several statements in the form of “SELECT * FROM * JOIN * ON * WHERE *”. The third type of task is consist of several statements in the form of “SELECT * FROM * AS * JOIN * ON * AS * JOIN * ON * (WHERE * AND *)”, e.g., Q7 and Q8. Besides, we generated five scales of data sets from SDSS DR9 to use in our evaluation, their size are number of records are as Table 1.

Table 1. Number of records in different datasets

Data size	1 GB	10 GB	20 GB	50 GB	100 GB
#Records	80000	800000	1600000	4800000	8000000

5 Performance Evaluation and Discussion

5.1 Experimental Setup

In this section, we will compare FASTDB with traditional database based astronomic data analytical solution SkyServer. All of our evaluation are based on the real dataset of SDSS Data Release 9, which is an astronomy survey aiming at creating a digital map of a big part of the Universe, and it is about 12 terabytes of zipped data.

The experiments conduct on a cluster and each nodes have two Intel(R) Xeon(R) CPU E5-2620 @ 2.00 GHz processor, the operating system is CentOS 6.4 and Windows 2008 (SkyServer only can running on a Windows System). The Coordinator node has 40 GB main memory and 1 TB hard disk space, while each of 15 worker node has 8 GB main memory and 1 TB disk respectively. The backend Database engine of FASTDB and SkyServer is SciDB 14.3 and Microsoft SQL Server 2008 R2. During the evaluation, both system are use the default configuration.

5.2 Experimental Results and Analysis

In our evaluation, we measure the response time of FASTDB and SkyServer in different workload. Figure 4(a)–(h) illustrated their performance when they run aforementioned 8 queries over five different data set sizes. The x-axis denotes the data size, while y-axis measures the time to complete the analysis task. When FASTDB run the tasks of Q6, Q7 and Q8, since it consume too long time to complete the tasks than SkyServer, we just ignore the specific response time in corresponding figures.

As to Q1–Q5, FASTDB often to be better than SkyServer for its parallel processing capability. As to Q5–Q8, although Q5 and Q6, Q7, Q8 belong to different type of analytical tasks, they still have a common characteristic that they have at least one Join operation. Currently, the distributed scientific database like SciDB didn't have indexing mechanism and cannot support the Join operation very well. While the underlying database engine of SkyServer is SQL Server 2008, it has mature indexing capability which could enable high performance index join, it may become one of the major factor which significantly improved SkyServer's performance.

In our evaluation, since the data are uniformly distributed in all FASTDB Worker nodes, the Join operation need to manipulate all of the data in all worker nodes of FASTDB, thus the execution tends lead large network traffic. Additionally, FASTDB will compress data when store into the database engine, and the decompression overhead of large data size needed for the join query often to be very expensive, thus it may degrade the performance of join queries. Furthermore, the join operation needs a lot of computation in the coordinator node, thus FASTDB is relative easy to incur main memory bottleneck when the involved data size is large.

Let's take Q5 to characterize FASTDB's performance when it execute the first type of analytical tasks. As show in Fig. 4(e), when data sets become larger, we can know that the response time has rising in both two systems. Especially, on the 20 GB and 50 GB data sets, there is an obvious rising trend on SkyServer. Meanwhile, only a relative gradual growth trend is occurs on FASTDB due to it distributes the data to all

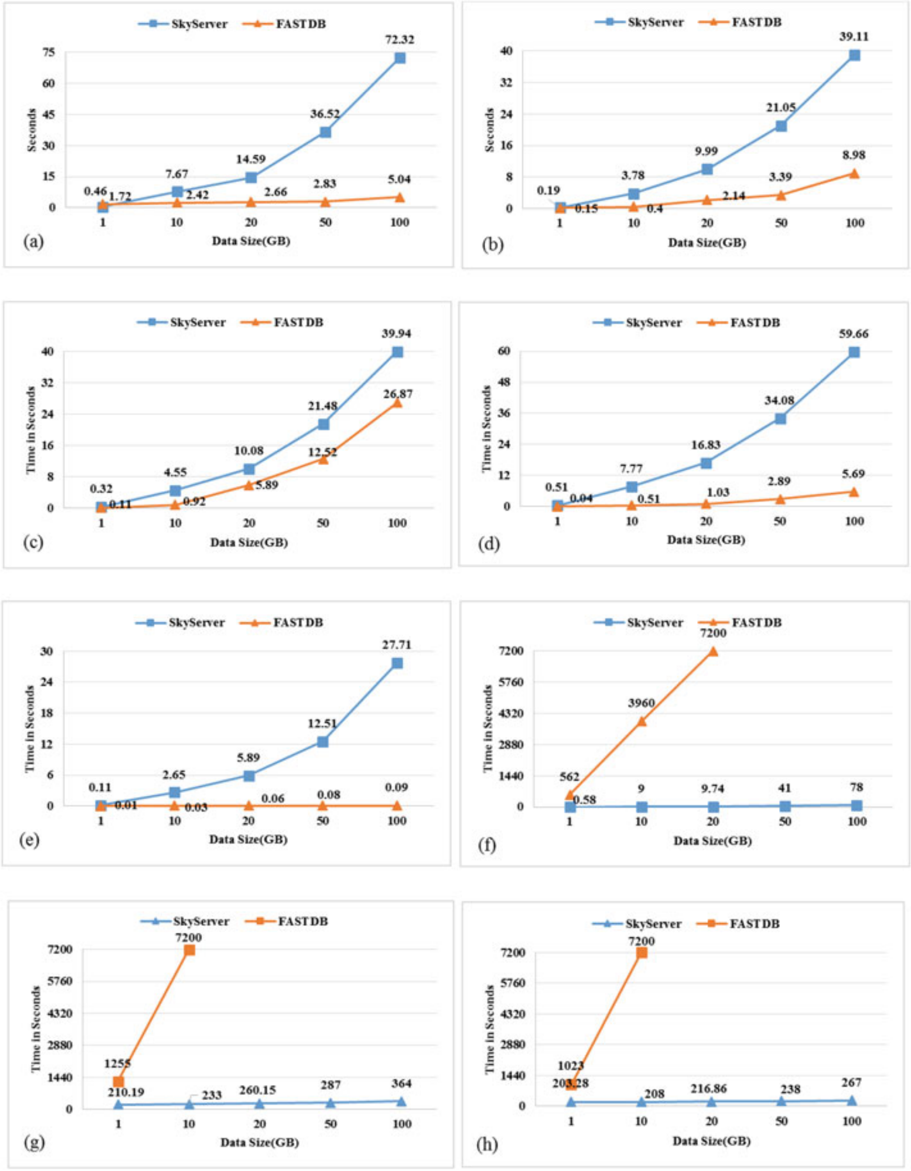

Fig. 4. Overall performance of FASTDB and SkyServer on different data sizes. (a)–(h) represent the Response Time of Q1–Q8.

of worker nodes by its chunking mechanism, which make each worker node can execute analysis tasks in parallel and results in the FASTDB based solution is about 10 to 30 times faster than SkyServer.

Besides evaluate the parallel processing capability of FASTDB, we also conduct a series of experiment to evaluate single node FASTDB with SkyServer. The results are tabulated in Table 2. As to Q1–Q5, FASTDB tends 2 times faster than SkyServer when data size equal or larger than 10 GB (the performance number are mark as bold), this is most benefit from FASTDB's array model for it can convenient to obtain data by dense packing, which make it tends to obtain better performance than relation model based system like SkyServer. As in cluster scenarios, when FASTDB running tasks which have Join operators, due to the lack of indexing capability, and other reasons mentioned in the analysis of Fig. 4, most of task Q6–Q8 are timeout in FASTDB.

Table 2. Overall performance of Single FASTDB node and SkyServer. " —" means the response time is more than 7200 s.

System	Dataset	Query overheads (second)							
		Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
SkyServer	1 GB	0.46	0.19	0.32	0.51	0.11	0.58	210	203.3
	10 GB	7.67	3.78	4.55	7.77	2.65	9.01	233	208
	20 GB	14.59	9.99	10.08	16.83	5.89	9.74	260	216.86
	50 GB	36.5	21.05	21.48	34.08	12.5	41	287	238
	100 GB	72.32	39.11	39.94	59.66	27.71	78	364	267
FASTDB	1 GB	0.76	0.72	0.65	0.99	0.89	1900	—	—
	10 GB	**1.02**	**1.94**	**2.89**	**3.96**	**2.26**	—	—	—
	20 GB	**1.16**	**4.23**	**7.34**	**5.69**	**2.71**	—	—	—
	50 GB	**1.36**	**8.02**	**15.56**	**10.21**	**3.85**	—	—	—
	100 GB	**3.11**	**24.45**	**31.11**	**19.59**	**5.12**	—	—	—

6 Conclusions

To storing and analyzing massive scientific data efficiently, we developed a scalable share-nothing parallel array database system named FASTDB. Benefit from its array model and optimization of backend SciDB database engine, it has been proved significantly better than traditional relation model based database system SkyServer in non-join analytical tasks in both single node and cluster scenarios. In the future, we will continually working on optimize FASTDB to make it achieve better scientific analysis with multiple joins capability.

Acknowledgments. This work was supported by the China Ministry of Science and Technology under the State Key Development Program for Basic Research (2012CB821800), Fund of National Natural Science Foundation of China (No. 61462012, 61562010, U1531246), Scientific Research Fund for talents recruiting of Guizhou University (No. 700246003301), Science and Technology Fund of Guizhou Province (No. J [2013]2099), High Tech. Project Fund of Guizhou Development and Reform Commission (No. [2013]2069), Industrial Research Projects of the Science and Technology Plan of Guizhou Province (No. GY[2014]3018) and The Major Applied Basic Research Program of Guizhou Province (No. JZ20142001, No. JZ20142001-05).

References

1. Gray, J., Liu, D.T., DeWitt, D., Heber, G., Nieto-Santisteban, M., Szalay, A.S.: Scientific data management in the coming decade. MSR-TR-2005-10 (2005)
2. Hey, T., Tansley, S., Tolle, K.: The fourth paradigm: data-intensive scientific discoveries. Microsoft research, p. 10 (2009)
3. Gray, J., Szalay, A.S., Thakar, A.R., Kunszt, P.Z., Stoughton, C., Slutz, D., et al.: Data mining the SDSS SkyServer database. MSR-TR-2002-01 (2002)
4. Five-hundred-meter aperture spherical telescope. http://fast.bao.ac.cn/en/
5. Abadi, D.J., Madden, S.R., Ferreira, M.: Integrating compression and execution in column-oriented database systems. In: Proceedings of SIGMOD, pp. 671–682 (2006)
6. Soroush, E., Balazinska, M., Wang, D.: ArrayStore: a storage manager for complex parallel array processing. In: Proceedings of SIGMOD, pp. 253–264 (2011)
7. Seering, A., Cudre-Mauroux, P., Madden, S., Stonebraker, M.: Eficient versioning for scientific array databases. In: Proceedings of ICDE, pp. 1013–1024 (2012)
8. Cudre-Mauroux, P., Kimura, H., Kimura, H., Lim, K.-T., Rogers, J., Simakov, R., Soroush, E., et al.: A demonstration of SciDB: a science-oriented DBMS. In: Proceedings of VLDB, pp. 1534–1537 (2009)
9. Stonebraker, M., Becla, J., Dewitt, D., Lim, K.-T., Maier, D., Ratzesberger, O., Zdonik, S.: Requirements for science data bases and SciDB. In: Proceedings of CIDR (2009)
10. Brown, G.: Overview of SciDB: large scale array storage, processing and analysis. In: Proceedings of ICDE, pp. 963–968 (2010)
11. Hui, L., Nengjun, Q., Hongyuan, L., Mei, C., Min, Z., Menglin, H.: FASTDB: a array database system for efficient storing and analyzing massive scientific data. Technical report, GZU-ACMIS-TR-2014-07, pp. 1–104. (in Chinese)
12. Paradigm 4 Inc. http://www.paradigm4.com/
13. Marcos, D., Connolly, A.J., et al.: ASCOT: a collaborative platform for the virtual observatory. In: Proceedings of ADASS XXI, vol. 461, pp. 901–904 (2012)
14. Vanderplas, J., Soroush, E., Krughoff, S., Balazinska, M., Connolly, A.: Squeezing a big orange into little boxes: the AscotDB system for parallel processing of data on a sphere. IEEE Data Eng. Bull. **36**, 11–20 (2013)
15. Baumann, P., Dehmel, A., Furtado, P., Ritsch, R., Widmann, N.: The multidimensional database system RasDaMan. In: Proceedings of SIGMOD, pp. 575–576 (1998)
16. Baumann, P., Dumitru, A.M., Merticariu, V.: The array database that is not a database: file based array query answering in rasdaman. In: Nascimento, M.A., Sellis, T., Cheng, R., Sander, J., Zheng, Yu., Kriegel, H.-P., Renz, M., Sengstock, C. (eds.) SSTD 2013. LNCS, vol. 8098, pp. 478–483. Springer, Heidelberg (2013)
17. Rasdaman Inc. http://www.rasdaman.com/
18. Ivanova, M., Nes, N., Goncalves, R., Kersten, M.L.: MonetDB/SQL meets SkyServer: the challenges of a scientific database. In: Proceedings of SSDBM, pp. 7–13 (2007)
19. Kersten, L., Zhang, Y., Ivanova, M., Nes, N.: SciQL, a query language for science applications. In: Proceedings of Array Databases Workshop, pp. 1–12 (2011)
20. Xiangsheng, K.: Scientific data processing using MapReduce in cloud environments. J. Chem. Pharm. Res. **6**, 1270–1276 (2014)
21. Lai, W.K., Chen, Y.-U., Wu, T.-Y., Obaidat, M.: Towards a framework for large-scale multimedia data storage and processing on Hadoop platform. J. Supercomputing **68**, 488–507 (2014)

22. Buck, B., Watkins, N., LeFevre, J., et al.: SciHadoop: array-based query processing in Hadoop. In: Proceedings of International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 66:1–66:11 (2011)
23. SkyServer. http://skyserver.sdss.org/
24. Data Release 9 of Sloan Digital Sky Survey. http://skyserver.sdss.org/dr9/en/
25. Typical astronomic queries. http://cas.sdss.org/dr9/en/help/docs/realquery.asp