Towards a Systematic Benchmark for Array Database Systems

Peter Baumann and Heinrich Stamerjohanns

Center for Advanced Systems Engineering (CASE), Jacobs University, Bremen, Germany
{p.baumann,h.stamerjohanns}@jacobs-university.de

Abstract. Big Data are a central challenge today in science and industry. Typically, Big Data are characterized from application perspectives. From a data structure perspective, among the core structures appearing are sets, graphs, and arrays. In particular in science and engineering we find arrays being a main contributor to data volumes. In fact, large, multi-dimensional arrays represent an important information category in earth, life, and space sciences, but also in engineering, business, and e-government.

Having long been neglected by database research, arrays today increasingly receive attention leading to a whole new field of investigation, Array Databases. As more and more Arry Database Systems emerge, similarities and differences can be observed. This calls for complementary research on benchmarks for Array DBMSs.

We present work in progress on such a comprehensive Array DBMS benchmark, which is based on our 15 years of pioneering Array DBMSs and also designing a geo raster query language standard and its corresponding functionality benchmark.

1 Motivation

Large, multi-dimensional arrays represent a major Big Data contributor in science, industry, and e-government. For example, spatio-temporal Earth science data include 1-D time series, 2-D satellite imagery, 3-D x/y/t image timeseries and x/y/z geophysical data, and 4-D x/y/z/t atmosphere and ocean data, among others. Likewise, in Life Sciences bio/medical modalities like computerized tomography (CT) scans and confocal microscopy produce increasing amounts of spatio-temporal data. In Astrophysics, optical and radar sensors deliver high-resolution raster data as continuous streams and with large numbers of spectral bands. Statistical data sets transcend spatio-temporal dimensions by using user-defined measures as dimension axes, but still yielding n-D data cubes. Multimedia databases use vectors of hundreds to thousands of features for content-based image retrieval. Figure 1 symbolizes some relevant applications.

Arrays appear as low-dimensional spatio-temporal data, medium-dimension statistics data (such as 3 to 12 dimensional OLAP [17]), and high-dimensional feature vectors (with thousands of dimensions) [15]. A further distinguishing

T. Rabl et al. (Eds.): WBDB 2012, LNCS 8163, pp. 94–102, 2014.

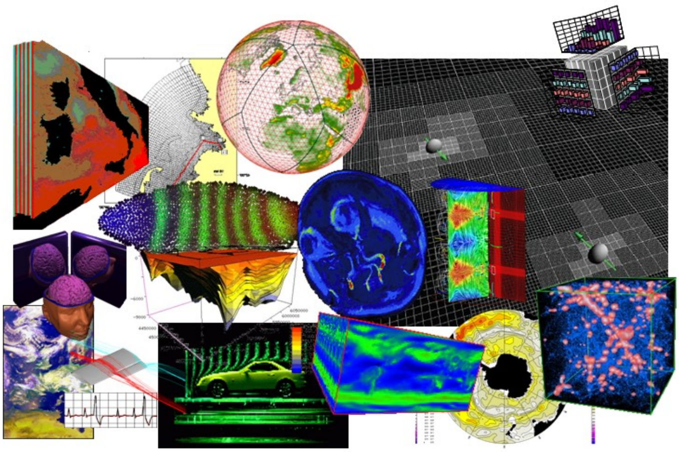

Fig. 1. Array data: A collage of applications

criterion is the number of cells carrying meaningful information: sparse data, with typically 3 to 5 % of cell positions being occupied by data, appear in OLAP and statistics data cubes; dense data, with 100% or not much less of cells carrying data, such as satellite imagery.

Operations applied on such arrays can be studied by investigating image and signal processing, statistics, and linear algebra, to name a few. Finally, arrays regularly appear as "Big Data" with terabyte-sized single objects and petabyte archives, such as the holdings of Earth Observation (EO) data centers like the European Space Agency (ESA) and NASA archives.

Although arrays form an essential data structure in science and engineering and although this structure is well defined and known, database research has long neglected arrays, categorizing them as "unstructured data" to be stored as BLOBs. Consequently, no semantics and no operations can be offered by the database system, and hence users like large-scale data centers did not get any value from using databases for their array data. Still today, therefore, databases in science are mainly used for metadata while array and similar data are maintained by specially crafted data management tools with specialized service interfaces, but without fexible general-purpose query languages.

Only recently Array DBMSs have become a mainstream area of research. The pioneering system is rasdaman ("raster data manager") with its 24 years since its first publication and with a fully-fledged implementation used in operational installations since many years. Rasdaman is based on a minimal algebra on which query language, optimization, query evaluation, and storage layout is based. Among recent research approaches are SciQL, an array extension to the column-store MonetDB system, and SciDB, a standalone Array DBMS utilizing User-Defined Functions for providing array functionality.

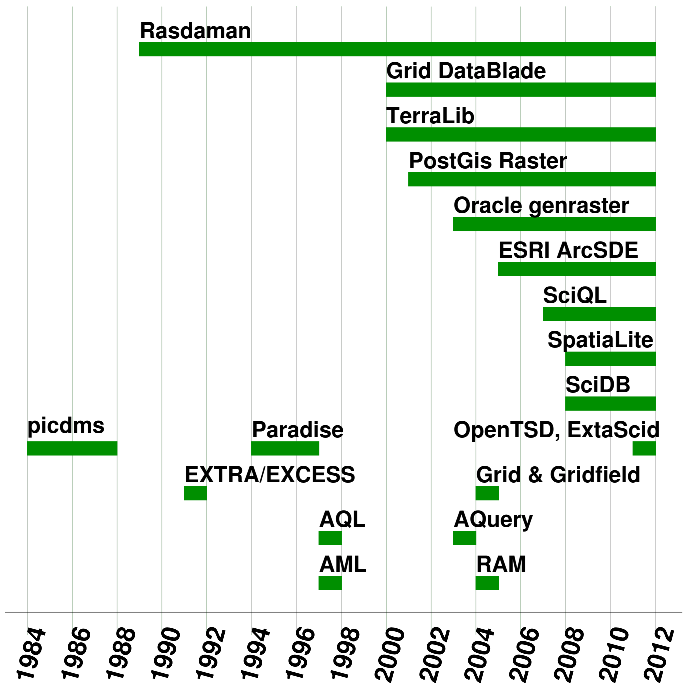

Fig. 2. A Brief History of Array DBMSs

In terms of data service standards, we can find arrays with ISO SQL:1998 and its successors and with the OGC Web Coverage Processing Service (WCPS) [7], a geo raster query language.

Given this relevance of array support in databases and with various, slightly varying systems emerging the quest for benchmarks arises. A comprehensive, well documented, and maintained benchmark can be of significant value to both deployers - like data centers - and database vendors, but also individual scientists and engineers. Further, it should allow to not only assess Array DBMSs as such, but also the large number of array supporting tools which are not using database technology, such as MatLab [19], R [23], and OPeNDAP [3].

In this paper we describe first concepts of a benchmark for large-scale, multi-dimensional array services. Currently, we are structuring the various facets of the possible and useful benchmarking tests. As a first result, we suggest a "suitability cube" framework in which all assessment aspects can be embedded. Under work is the refinement and breakdown of this concept.

In the next section we describe related work. Section 3 introduces the Suitability Cube. Practical examples are presented in Section 4; Section 5 concludes the contribution.

2 Related Work

Benchmarking of databases has been thoroughly addressed in the eighties and nineties, including periods of hot "benchmark wars" between vendors. Today, a set of generally accepted benchmarks is available for relational databases. Among the most popular are the SPEC [26] and TPC [10] database benchmarks.

There is no equivalent, though, for array databases. Given the only recently broadened interest of the database community there are no established benchmarks yet - actually, not even a commonly agreed conceptual data and query model.

Figure 2 gives a brief visual overview of the historical development. A notable precursor was PICDMS [9] which offered a conceptual model of a stack of same-resolution 2-D arrays with operators on them, although a generic array query language was not yet present, and no suitable architecture was indicated. Several publications emerged from relatively short-lived investigations. Most of today's systems, like PostGIS Raster [22] and rasdaman [18], add arrays as an additional attribute type, in sync with ISO SQL [14] which establishes arrays as a collection (i.e., column) type. A deviating approach is pursued by SciQL [16] and SciDB [24] where arrays are modeled similarly to tables, reusing much of standard SQL syntax albeit with a different semantics.

On commercial side, Oracle GeoRaster has to be mentioned, although - similar to PICDMS - it supports only 2-D arrays and lacks query support. ESRI ArcSDE has attempted to utilize databases for its 2-D rasters, but seems to not pursue development any further.

In terms of standards, we can find array support in two places. ISO SQL:1998 and its successors offer array support through an array collection type, although no array operators; a currently proposed new work item is aiming at closing this gap. The Open Geospatial Consortium (OGC) establishes and maintains Web service standards for geospatial intelligence. Arrays form a subcategory of so-called coverages [6], aka space-time varying phenomena. In 2008, OGC has adopted the Web Coverage Processing Service (WCPS) standard [7], a spatio-temporal geo raster query language, conceptually influenced by the rasdaman Array Algebra.

There have been several early attempts to benchmark geospatial databases [25,20], but these included e.g. a limited number of temporal queries or focused on domains like remote sensing exclusively so that evaluations outside their specific application domain were not feasible.

SS-DB has been proposed as a benchmark for science oriented databases[11]. By applying a space science use-case, it performs nine queries on astronomical array data. This case study is an important contribution towards understanding astrophysical workloads. However, the benchmark remains on application level and does not provide a thorough evaluation on model or algebra level. This benchmark has been run against SciDB [12] and MySQL [1]. It is available as open source, although similar results have not been reported yet by other groups.

For the geospatial domain, an analysis of relevant functionality has been pursued in [13]. Based on a broad survey of operations used in geo imaging, functionality has been classified and described by a uniform algebraic framework to allow for a systematic inspection. Representative array query examples further have been published for web mapping [4], genetic research [21], among others.

All these efforts are characterized by selecting particular use cases, without proof of covering the respective domain adequately. Further, there is no rigorous conceptual analysis of array queries which might characterize a concrete system's performance in its entirety. Therefore, the field currently is dominated by ad-hoc attempts. Our work aims at consolidating them into an Array Database benchmarking framework.

3 Benchmarking Arrays

3.1 Conceptual Array Modeling

Based on the common definition of an array as a function $a : X \rightarrow V$ from some d-dimensional Euclidean hypercube X into some value set V we naturally find some first query operation candidates:

- Changing the domain set X, often called subsetting; this can be differentiated into trimming (cropping the domain while retaining the number of dimensions) and slicing (extracting hyperslabs, thereby reducing dimensionality).
- Manipulating the value set V; this leads to a common set of unary and binary array operations, such as pixel-wise addition of images.
- Changing the array function itself, like establishing new mappings (examples include histograms and matrix multiplication).
- De-arraying functions, like aggregation.

When it comes to storing arrays, all systems uniformly perform partitioning - as practiced in image processing since long under the name *out-of-core processing* - into sub-arrays called chunks or tiles. Systems differ in the degree of variability. On one end there are static partitionings into square blocks where only the block size can be modified; on the high end are freely definable tiling schemes with and without overlapping, which forms an important tuning parameter. Also, these partitions naturally induce a tile streaming architecture which allows to keep only few parts of an array in server main memory during query evaluation, thereby achieving scalability in data volumes.

3.2 Benchmarking Dimensions

Based on the above outline of the concepts under test, we group features of an Array DBMS into several categories: Overall, we currently consider the following data categories as relevant for a benchmark:

- **Array Model Features:** Assess the expressive power of the data model:

  - **Number of Dimensions:** This can be low-dimensional spatio-temporal data (1-D, 2-D, 3-D, 4-D, 5-D), medium-dimensional (6-D through 12-D), or high-dimensional (such as thousands of dimensions). Note that there is not a rigorous limit between boundaries, but we feel that the orders of magnitude separate good enough for focused testing.
  - **Cell Type:** Array cells can contain single values, records of values (such as hyperspectral satellite imagery), as well as theoretically any other data structure. In practice, variable-length cell types like strings are avoided by all models inspected, due to the added complexity in storage management.
- **Array Data Properties:**
  - **Volume of Data:** Object sizes may range from a few kB for 1-D time-series over a few hundred MB for a satellite image up to PB size climate model output. Sizes of object sets can be massive as well – e.g., the European Space Agency (ESA) plans to have 10^{12} satellite images under their custody with their ngEO project.
  - **Sparsity of Data:** How does access and processing performance depend on sparsity, i.e., the percentage of non-null values within a data array. OLAP data, for example, have a density of typically 3% to 5% while satellite images often have a density of 100%.
  - **Storage Features:** What partitioning schemes does the Array DBMS support? Can partitions be compressed? Distributed?
- **Array Operations:** This encompasses questions like: what primitives are offered? Are operations executed natively or as UDFs? An open question is how to systematically scale query complexity for benchmarking.
  - **Isolated Position Relevance:** How does access to a large array depend on the size, shape, and position of the subsetting box?
  - **Coupled Position Relevance:** How does access to a large array vary when two subsettings are done in sequence, for different bounding boxes? What about non-trivial access patterns like in convolutions, statistical operations, Fourier Transforms, or simply mirroring an array?
  - **Processing Capabilities:** What array operations are offered? Formalizations like Array Algebra help to find comprehensive operations and operation combinations.
  - **Processing Implementation:** To what extent are array operations natively supported by the query engine, and where does it resort to UDFs? How efficient is the architecture, utilizing optimization, parallelization, etc.?
  - **Data Ingest and Update:** How fast can arrays be loaded? How fine-grain can updates to parts of arrays be applied?
- **Updates:** In view of the large size of single objects, it is not sufficient to only test creation, replacement, and deletion of whole objects. Updating an object typically will address selected areas within an array, which poses specific performance challenges.
- **Application Specific Features:** Geo imagery, for example, requires specific operations like orthorectification, coordinate transformation; statistical data require algebra operations like matrix multiplication and inversion.

4 Application Scenarios

Our research in array databases is based on both theoretical investigation, like finding a declarative, minimal Array Algebra [8], and extensive practical evaluations with users and in standardization bodies [5]. A number of domains in engineering and science have been investigated in close collaboration with large-scale data centers, including remote sensing, oceanography, geology, climate modeling, astrophysics, planetary science, computational fluid dynamics, genetics, and human brain imaging.

There is a diverse audience of users for these use cases. For the public at large, the database serves a large number of clients with typically a limited set of well defined queries wrapped in visual clients. Power users and researchers may use the database query language - possibly again with visual support - to conveniently wade through their raw data and run individual analyses.

Typically, the hardware to be used - like cloud, cluster and tape silos - is already present so the main question is not to determine the best hardware but to find the right data management and service tool for the given scenario. Here we hope to provide guidance with a reproducible benchmark.

5 Conclusion and Outlook

Arrays comprise an information category whose importance is just now being acknowledged by the database community at large. In science, engineering, business, social media, and statistics large arrays are of prime importance. With further systems emerging in addition to the pioneer Array DBMS, rasdaman, a standardized benchmark is useful for both system designers and data providers using such technology.

As part of ongoing activities towards a systematic benchmark we propose a first structuring of benchmark facets for a quantitative assessment of Array DBMSs. Aspects considered include conceptual data and query model capabilities, scalability in data volume, dimensionality, and query complexity, native query support vs UDFs, and application domain requirements. Therefore, we consider our work as a generalization of the application specific SS-DB performance comparison. In particular, experience from writing functional conformance test suites for geo raster services within OGC [2] has provided useful insights into test design and structuring.

Currently we are implementing the first slate of tests, focusing on storage access and array operations. Once a sufficient slate of tests is available, it is planned to run them against the available Array DBMSs and publish both benchmark code and results.

Acknowledgement. This research work is being supported by the European Community's Seventh Framework Programme (EU FP7) under grant agreement no. 283610 "European Scalable Earth Science Service Environment (EarthServer)".

References

1. MySQL, http://www.mysql.com/
2. OGC Compliance Testing, http://www.opengeospatial.org/compliance
3. OpenNdap, http://www.openndap.org
4. Baumann, P., Jucovschi, C., Stancu-Mara, S.: Efficient map portrayal using a general-purpose query language (a case study). In: Bhowmick, S.S., Küng, J., Wagner, R. (eds.) DEXA 2009. LNCS, vol. 5690, pp. 153–163. Springer, Heidelberg (2009)
5. Baumann, P. (ed.): Web Coverage Processing Service (WCPS) Implementation Specification. No. 08-068r2, OGC, 1.0.0 edn. (2008)
6. Baumann, P.: Beyond rasters: introducing the new OGC web coverage service 2.0. In: Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS 2010, pp. 320–329. ACM, New York (2010), http://doi.acm.org/10.1145/1869790.1869835
7. Baumann, P.: The OGC web coverage processing service (WCPS) standard. Geoinformatica 14(4), 447–479 (2010), http://dx.doi.org/10.1007/s10707-009-0087-2
8. Baumann, P.: A database array algebra for spatio-temporal data and beyond. In: Pinter, R., Tsur, S. (eds.) NGITS 1999. LNCS, vol. 1649, pp. 76–93. Springer, Heidelberg (1999)
9. Chock, M., Cardenas, A.F., Klinger, A.: Database structure and manipulation capabilities of a picture database management system (picdms). IEEE Transactions on Pattern Analysis and Machine Intelligence 6(4), 484–492 (1984)
10. Council Transaction Processing Performance, TPC C Benchmark (2010), Standard Specification, http://www.tpc.org/tpcc/spec/tpcc_current.pdf
11. Cudre-Mauroux, P., Kimura, H., Lim, K.T., Rogers, J., Madden, S., Stonebraker, M., Zdonik, S., Brown, P.: SS-DB: A standard science DBMS benchmark (2010)
12. Cudre-Mauroux, P., Kimura, H., Lim, K.T., Rogers, J., Simakov, R., Soroush, E., Velikhov, P., Wang, D.L., Balazinska, M., Becla, J., DeWitt, D., Heath, B., Maier, D., Madden, S., Patel, J., Stonebraker, M., Zdonik, S.: A demonstration of SciDB: a science-oriented DBMS. Proc. VLDB Endow. 2(2), 1534–1537 (2009), http://dl.acm.org/citation.cfm?id=1687553.1687584
13. Garcia, A., Baumann, P.: Modeling fundamental geo-raster operations with array algebra. In: Proc. IEEE SSTDM, October 28-31, pp. 607–612 (2007)
14. ISO9075:1999: Information Technology-Database Language SQL. Standard No. ISO/IEC 9075:1999, International Organization for Standardization (ISO) (1999), (Available from American National Standards Institute, New York, NY 10036, (212) 642-4900)
15. Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. In: Nédellec, C., Rouveirol, C. (eds.) ECML 1998. LNCS, vol. 1398, pp. 137–142. Springer, Heidelberg (1998), http://dx.doi.org/10.1007/BFb0026683
16. Kersten, M.L., Zhang, Y., Ivanova, M., Nes, N.: SciQL, a query language for science applications. In: Baumann, P., Howe, B., Orsborn, K., Stefanova, S. (eds.) EDBT/ICDT Array Databases Workshop, pp. 1–12. ACM (2011), http://dblp.uni-trier.de/db/conf/edbt/array2011.html#KerstenZIN11
17. Kimball, R., Caserta, J.: The data warehouse ETL toolkit. John Wiley & Sons (2004)

18. LSIS Research Group Jacobs University. The array database rasdaman, Rasdaman is available at http://www.rasdaman.org
19. MATLAB: version (R2013a). Natick, Massachusetts (2013)
20. Patel, J.M., Yu, J.B., Kabra, N., Tufte, K., Nag, B., Burger, J., Hall, N.E., Ramasamy, K., Lueder, R., Ellmann, C.J., Kupsch, J., Guo, S., DeWitt, D.J., Naughton, J.F.: Building a Scaleable Geo-Spatial DBMS: Technology, Implementation, and Evaluation. In: Peckham, J. (ed.) SIGMOD Conference, pp. 336–347. ACM Press (1997), http://doi.acm.org/10.1145/253260.253342
21. Pisarev, A., Poustelnikova, E., Samsonova, M., Baumann, P.: Mooshka: a system for the management of multidimensional gene expression data in situ. Inf. Syst. 28(4), 269–285 (2003), http://dx.doi.org/10.1016/S0306-4379(02)00074-1
22. PostGIS, http://www.postgis.org
23. R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2013), http://www.R-project.org
24. Stonebraker, M., Brown, P., Poliakov, A., Raman, S.: The architecture of SciDB. In: Bayard Cushing, J., French, J., Bowers, S. (eds.) SSDBM 2011. LNCS, vol. 6809, pp. 1–16. Springer, Heidelberg (2011), http://dl.acm.org/citation.cfm?id=2032397.2032399
25. Stonebraker, M., Frew, J., Gardels, K., Meredith, J.: The Sequoia 2000 Benchmark. In: Buneman, P., Jajodia, S. (eds.) SIGMOD Conference, pp. 2–11. ACM Press (1993), http://doi.acm.org/10.1145/170035.170038
26. The Standard Performance Evaluation Corporation. SPEC Benchmark (2012), http://www.spec.org/benchmarks.html