Towards Benchmarking Large Arrays in Databases

H. Stamerjohanns  P. Baumann

Computer Science
Jacobs University Bremen

WBDB12
An Array DBMS: Rasdaman

Goal of rasdaman database:
- handle raster data
- massive n-dimensional Sensor-, Image-, Model & Statistics DB
- Tile-based architecture
  - n-D array → set of n-D tiles
- adapting storage to access pattern (preserve locality of reference)

1Baumann 1992, Baumann VLDBJ 1994
An Array DBMS: Rasdaman

- declarative, minimal, safe Array Algebra:
  - Intensive user studies: statistics, image, signal processing
- minimally invasive DBMS integration
- new attribute type: array<celltype, extent>

- maps d-dimensional Euclidean hypercube $X$ onto value set $V$

Array is function $a : X \rightarrow V$
An Array DBMS: Rasdaman

- implements SQL-embedded DML with array operators
  - select / insert / update / delete + *partial update*

```sql
select img.scene.green[x0:x1, y0:y1] > 130
from LandsatArchive as img
where some_cells(img.scene.nir > 127)
```

- Web mapping, image & signal processing statistics, linear algebra, pattern mining, scientific analytics
What is Big Data?

- somehow connected to volume
- but volume is moving target
- not only petabytes are Big Data
What is Big Data?

- unless you are reeeaally big, storage volume is not biggest problem
- to do proper analysis then is the difficulty
- suboptimal access patterns show up
- → inability of existing DB to scale
  - cardinality of data is typically small compared to volume
  - repeated observations of time or space
  - many datasets have inherent temporal or spatial dimensions
  - but not ordered accordingly to preserve locality
  - analysis then results in random-access patterns → sloow.
What is Big Data?

- ETL may not be the right solution...
- big volumes need to be transferred for further processing

Meta-definition:
"Any point in time when data volume forces us to look beyond the tried-and-true methods that are prevalent at that time"²

² A. Jacobs 2009
Array database domain

Diverse world

- different approaches to implement arrays on databases exist
  - MonetDB\(^3\)
  - SciDB\(^4\)
- no unified query language available
- different usage scenarios
  - (web-) service providing access to many users
  - but also personal research tool to analyse data

---

\(^3\)van Ballegoji et al., 2005, www.monetdb.org

\(^4\)P. Cudre-Mauroux et al., 2009, www.scidb.org
### Benchmarks should be...

**relevant**
- map real-world needs
- rather practice driven
- systematically cover features and data properties
- apply to different application domains

**simple**
- obviously some trade-off to previous point needed

**portable**
- as no unified query language available
  - high level description of tasks to fulfill

**scalable**
Benchmarking Array DBMS

Need to test

- array features
  - dimensionality, cell types
- data properties
  - volume, sparsity
- array query operations
- domain specific features
  - special operations, transformations
What needs to be tested... relevance

number of dimensions

- low-dimensional (1-D - 5-D)
  1-D environmental sensor time series
  2-D satellite images, seafloor maps
  3-D x/y/t image time series
  and x/y/z geophysics data
  4-D x/y/z/t climate and ocean data

- medium-dimensional (6-D - 12-D)
  OLAP

- high-dimensional (up to thousands)
  Data-Mining, collection of features
What needs to be tested... relevance

number of dimensions

- low-dimensional (1-D - 5-D)
  1-D environmental sensor time series
  2-D satellite images, seafloor maps
  3-D x/y/t image time series
  and x/y/z geophysics data
  4-D x/y/z/t climate and ocean data

- medium-dimensional (6-D - 12-D)
  OLAP

- high-dimensional (up to thousands)
  Data-Mining, collection of features
number of dimensions

- low-dimensional (1-D - 5-D)
  - 1-D environmental sensor time series
  - 2-D satellite images, seafloor maps
  - 3-D x/y/t image time series and x/y/z geophysics data
  - 4-D x/y/z/t climate and ocean data

- medium-dimensional (6-D - 12-D)
  - OLAP

- high-dimensional (up to thousands)
  - Data-Mining, collection of features
What needs to be tested... relevance

number of dimensions

- **low-dimensional (1-D - 5-D)**
  1-D environmental sensor time series
  2-D satellite images, seafloor maps
  3-D x/y/t image time series
  and x/y/z geophysics data
  4-D x/y/z/t climate and ocean data

- **medium-dimensional (6-D - 12-D)**
  OLAP

- **high-dimensional (up to thousands)**
  Data-Mining, collection of features
What needs to be tested... relevance

number of dimensions

- low-dimensional (1-D - 5-D)
  1-D environmental sensor time series
  2-D satellite images, seafloor maps
  3-D x/y/t image time series
  and x/y/z geophysics data
  4-D x/y/z/t climate and ocean data

- medium-dimensional (6-D - 12-D)
  OLAP

- high-dimensional (up to thousands)
  Data-Mining, collection of features
  precipitation
  x/y/z/t
What needs to be tested... relevance

Space time cube

- Satellite creates several scenes
- Satellite scene referenced by latitude/longitude + time
- at least twice per year each point should be mapped
- set of scenes that have temporal and spatial overlap

Example query:

- give me the Near-field infrared (NIR) values between 2007 and 2009 in Vienna
Dimensions and cell type constitute array model features

- cell types
  - single
  - records (e.g. colored pixel)
  - domain specific data structures
What needs to be tested... scaleability

Data properties
- Volume of data
  - range MB to PB
- Sparsity of data
  - sparse arrays like statistical data cubes
  - dense arrays like satellite imagery
Relevance in array database domain

Array is function $a : X \rightarrow V$

Query operations
- on $X$: trimming, slicing
- on $V$: pixel-wise addition of images
- on the function itself: histogram
Relevance in array database domain

Array is function $a : X \rightarrow V$

Query operations
- de-arraysing functions: aggregations
- querying irregular time axis (most rain in june in last years)
Relevance in array database domain

Array is function \( a : X \rightarrow V \)

Irregular time axis
- calendar is highly irregular, month lengths differ, leap years
- but need to analyse by month, season
- \( \rightarrow \) create additional dimensions
- has effect on tiling strategies
Ease of use in array database domain

Array is function \( a : X \rightarrow V \)

Query operation support
- natively supported?
- via User Defined Functions (UDF)?
  - expertise needed
  - additional costs involved

..how to implement in benchmark?
Suitability cube

Combination of assessments can be called a *suitability cube*

- addresses challenges from all relevant sides
- developers want to address all possibilities
- users want one single number...

Does modern technology help?

(modified image from qrarts.com)
Existing array DB benchmarks

Early attempts: Sequoia 2000\textsuperscript{5}, Paradise\textsuperscript{6}
Standard Science DBMS Benchmark (SS-DB)\textsuperscript{7}

- applies space-science use case
- \textit{relevant}, performs nine queries on astronomical data
  - load data
  - queries raw data
  - creates derived data (cooking)
  - queries derived data
- \textit{portable}, source-code available (but difficult to find...)
  → repeatable
- \textit{scalable}, covers small to big data volumes, data generator

\textsuperscript{5}Stonebraker 1993
\textsuperscript{6}Patel et al. 1997
\textsuperscript{7}Cudre-Mauroux et al. 2010
Existing array DB benchmarks, SS-DB

However...

- only single-user queries
- selection of queries seems rather limited
does not address higher-dimensions, such as 4-d, 5-d→ does not fully cover other application domains, such as
gephysics, climate and ocean data
- only regular time axis

Trade-off between simplicity and functional coverage
- *ease of use*, no analysis of array queries used
  - natively supported?
  - user defined functions
- result is not a single number...
Conclusion

- arrays inherent in Big Data
- benchmarks for big data should consider array operations as well
- suitability cube tries to address many metrics
- SS-DB good basis for discussion

benchmarks will make us work harder...