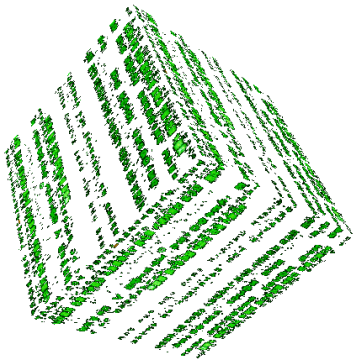




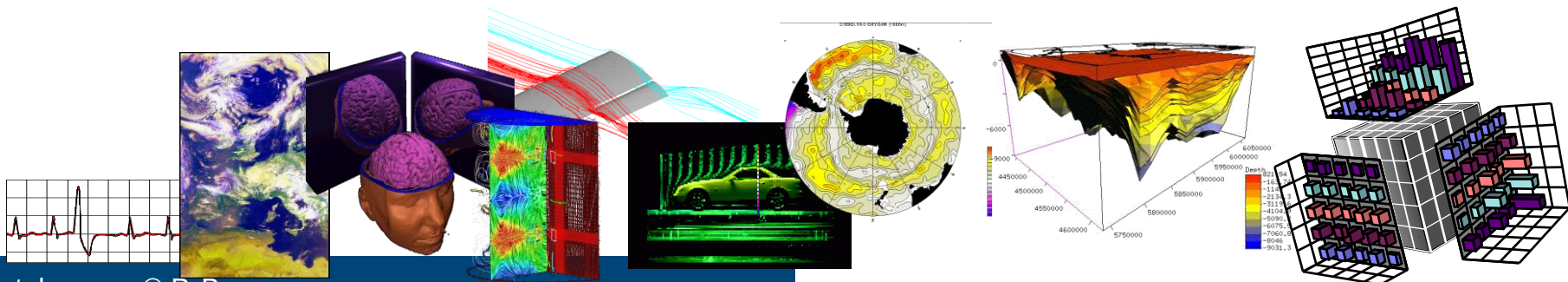
Array Databases

<http://l-sis.org> → publications

http://en.wikipedia.org/wiki/Array_DBMS

Who Needs Arrays?

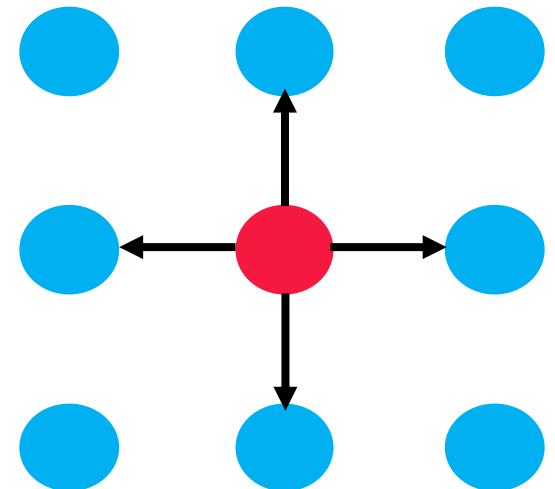
- **Sensor, image, simulation, statistics data**
 - **Earth:** Geodesy, geology, hydrology, oceanography, climate, earth system, ...
 - **Space:** optical / radio astronomy, cosmological simulation, planetary science, ...
 - **Life:** Pharma/chem, healthcare / bio research, bio statistics, genetics, ...
 - **Engineering & research:** Simulation & experimental data in automotive/shipbuilding/aerospace industry, turbines, process industry, ...
 - **Management/Controlling:** Decision Support, OLAP, Data Warehousing, census, statistics in industry and public administration, ...
 - **Multimedia:** distance learning, prepress, ...
- „80% of all data have some spatial connotation“ [C&P Hane, 1992]



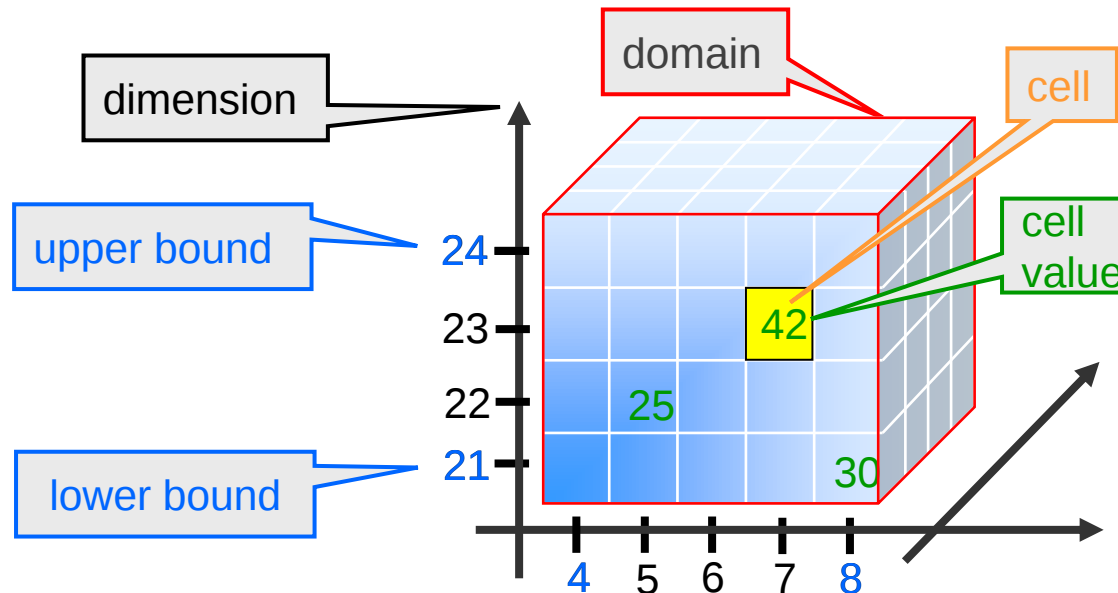
CONCEPTUAL MODELLING

Array Analytics

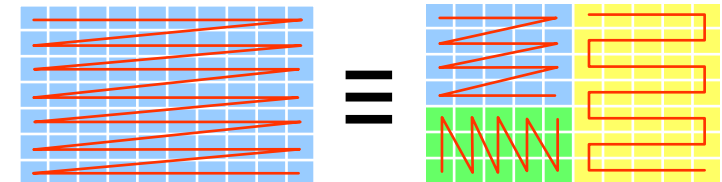
- Array Analytics :=
Efficient analysis on multi-dimensional arrays of a size several orders of magnitude above evaluation engine's main memory
- Essential **data** property: n-dimensional Cartesian neighborhood
 - Secondary: #dimensions, density, ...
- **Operations**: Tensor Algebra++



The Array Data Model



- Minimal **algebra** for model, QL, storage mapping, optimization
- Array iteration implicit
→ **no explicit loops** → declarative, safe
- Expressiveness up to, eg, Discrete Fourier Transform



Arrays in SQL

[SSDBM 2014]

rasdaman as blueprint



```
create table LandsatScenes(
  id: integer not null, acquired: date,
  scene: row( band1: integer, ..., band7: integer ) mdarray [ 0:4999,0:4999] )
```

```
select id, encode(scene.band1-scene.band2)/(scene.band1+scene.band2)), „image/tiff“ )
from LandsatScenes
where acquired between „1990-06-01“ and „1990-06-30“ and
      avg( scene.band3-scene.band4)/(scene.band3+scene.band4)) > 0
```

Sample Linear Algebra Ops

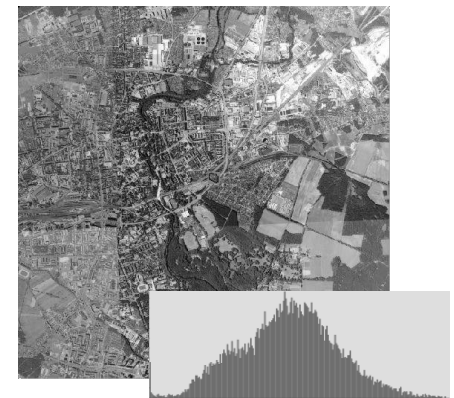
- $n \times p$ Matrix multiplication

$$(\mathbf{AB})_{ij} = \sum_{k=1}^m A_{ik} B_{kj}$$

```
select marray i in [1:n], j in [1:p]
      values condense +
            over      k in [1:m]
            using      a [ i, k ] * b [ k, j ]
from    matrix as a, matrix as b
```

- Histogram

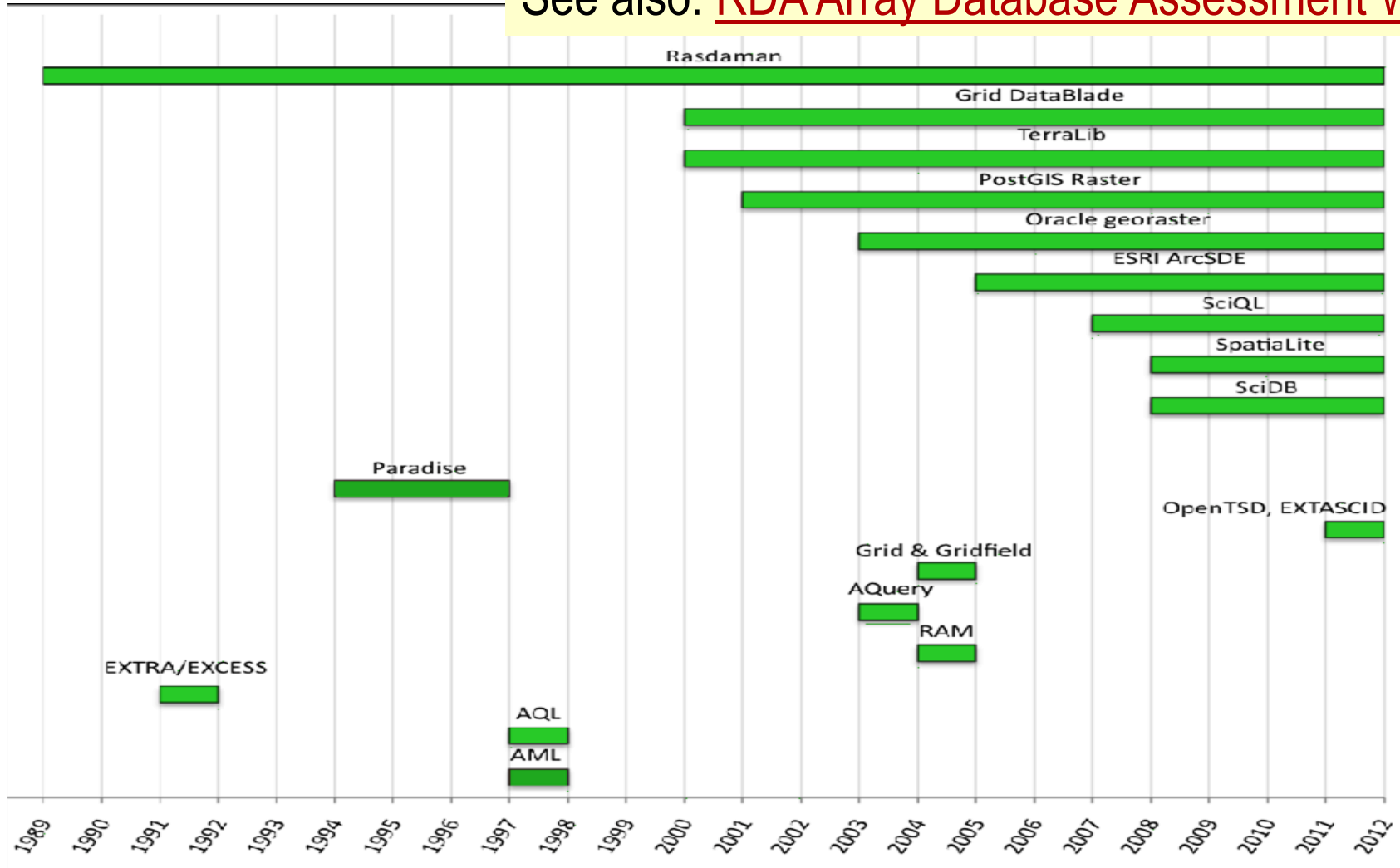
```
select marray bucket in [0:255]
      values count_cells( img = bucket )
from    img
```



RELATED WORK

Early History of Array Databases

See also: [RDA Array Database Assessment WG](#)



Array DBMSs Landscape Today

rapidly evolving ecosystem
→ necessarily incomplete

■ Array Databases = Tensor Databases

- query language, multi-user operation, storage management, access control
- Ex: rasdaman, SciDB, EXTASCIID, PostGIS Raster, Oracle GeoRaster

■ Array tools: command-line tools & libraries, but no service

- no query concept, but procedural API
- Ex: OpenDataCube, OPeNDAP, Wendelin.core, TensorFlow, boost::geometry, xtensor, TileDB, ArrayStore, Ophidia

■ Map/Reduce: Hadoop & Spark as cloud parallelization paradigm

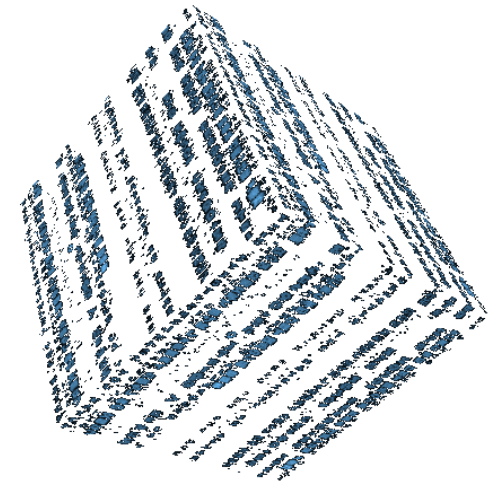
- Array layers on top of Hadoop, Spark
- Ex: SciHadoop, Spatial Hadoop, GeoTrellis, MrGeo, SciSpark, ClimateSpark

[survey of 19 tools]

RASDAMAN

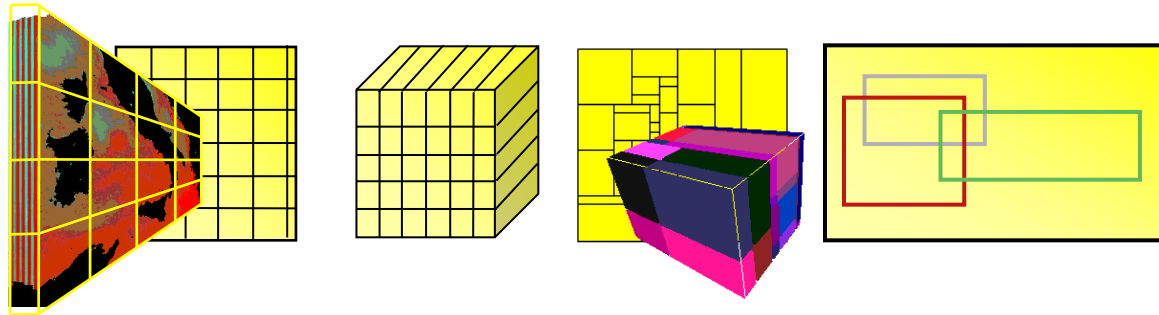
rasdaman

- „raster data manager“: **SQL + n-D arrays**
 - Scalable parallel “tile streaming” architecture
 - 200+ publications
- In operational use: agencies, industry, research
- Blueprint for standards
- Innovation awards



Adaptive Partitioning („Tiling“)

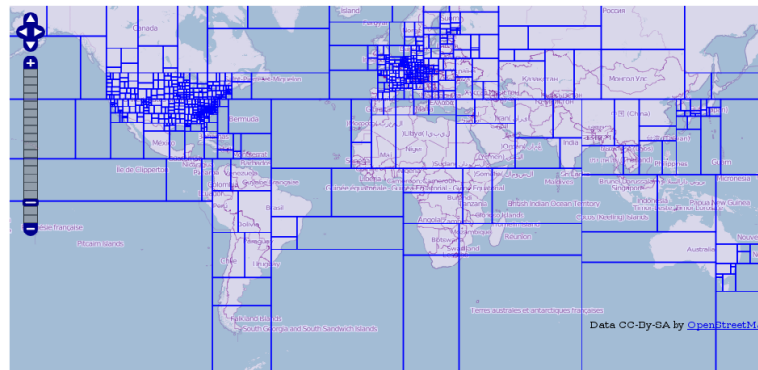
- Any tiling [Furtado 1999]
 - Cast into strategies



- rasdaman storage layout language

```
insert into MyCollection
values ...
tiling
    area of interest [0:20,0:40], [45:80,80:85]
    tile size 1000000
    index d_index storage array compression zlib
```

- Why irregular tiling?

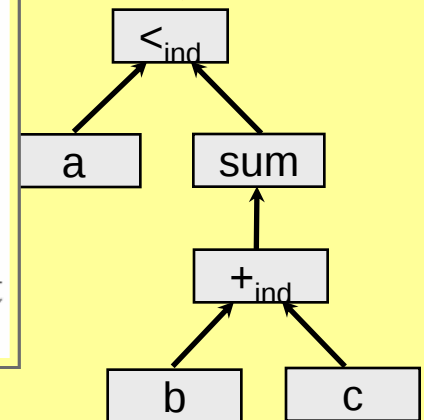
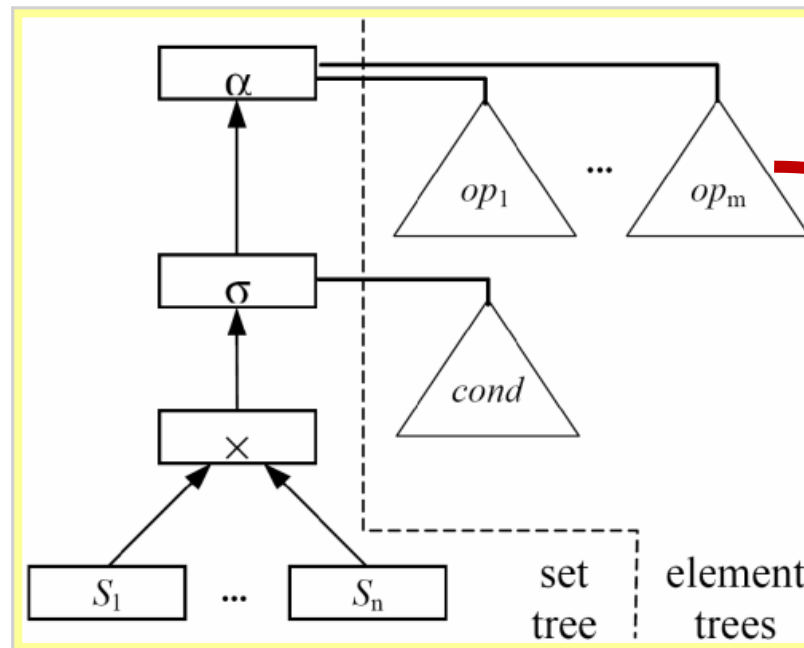


[OpenStreetMap]

Query Processing

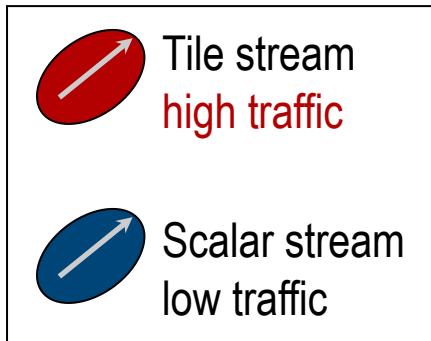
- Clear separation:
set vs array trees
 - Arrays as 2nd order attributes
- Optimization
- Tile-based evaluation

```
select a.array <
      sum_cells( b.array + c.array )
from   a, b, c
```

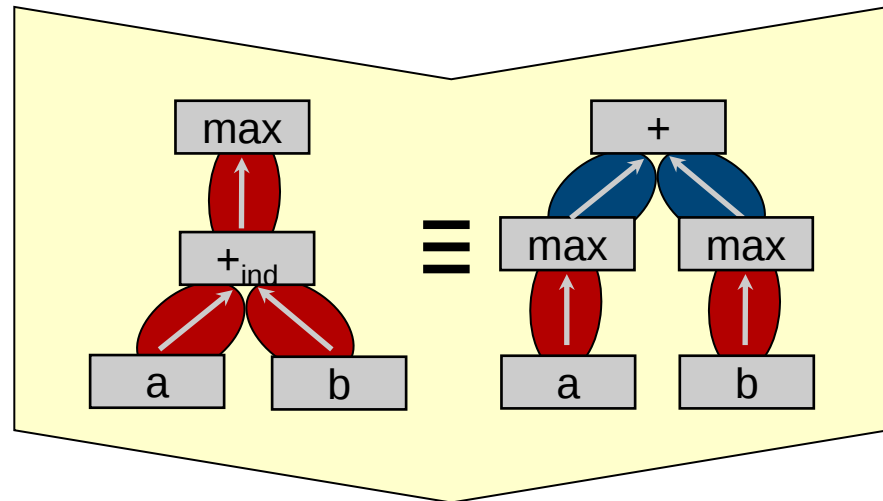


Query Optimization

- Algebraic
+ cost-based
- 150+ rules
currently
- Example:



```
select some_cells( a or b )
from   a, b
```



```
select some_cells( a )
      or some_cells( b )
from   a, b
```

Parallel / Distributed Query Processing

```
select
```

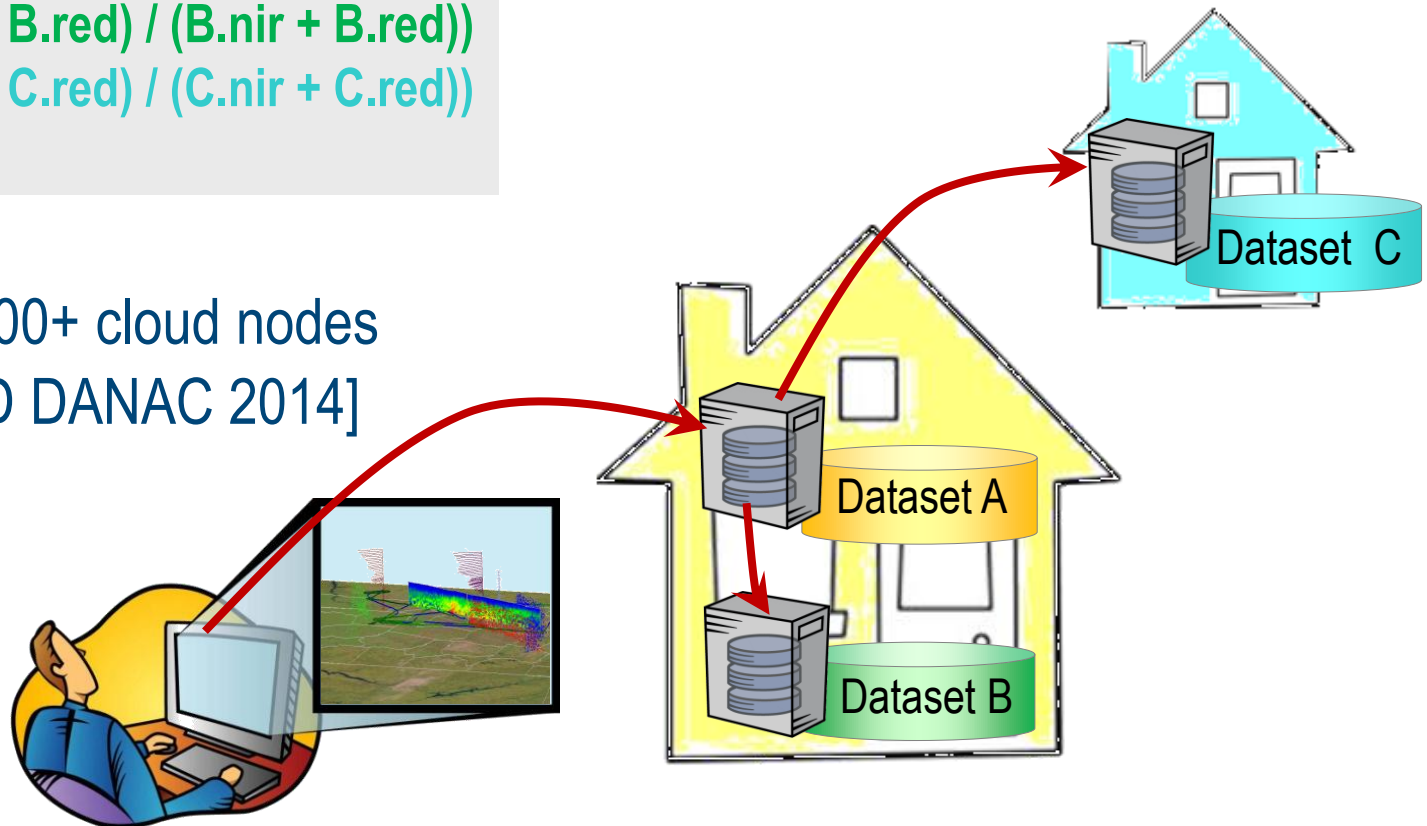
```
  max((A.nir - A.red) / (A.nir + A.red))
```

```
- max((B.nir - B.red) / (B.nir + B.red))
```

```
- max((C.nir - C.red) / (C.nir + C.red))
```

```
from A, B, C
```

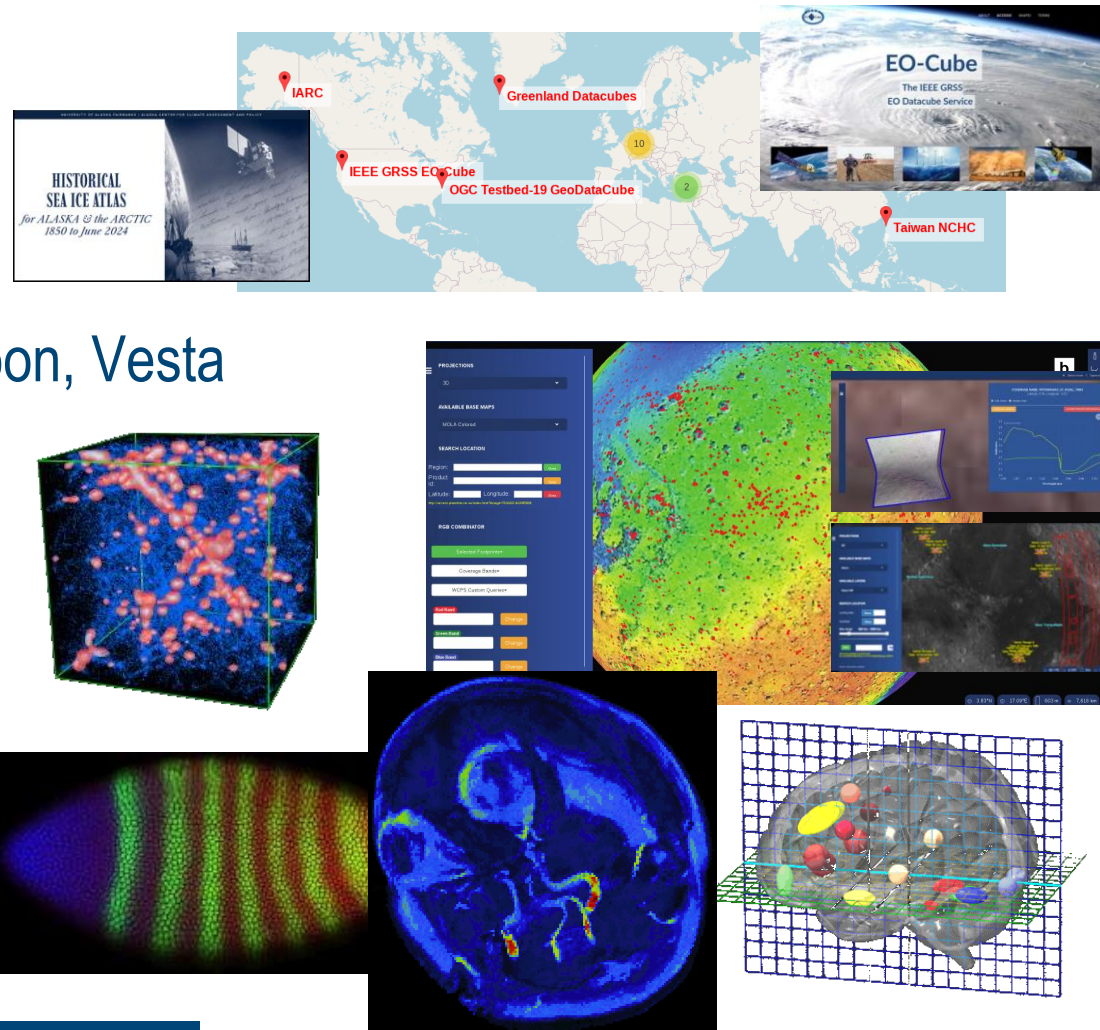
1 query → 1,000+ cloud nodes
[ACM SIGMOD DANAC 2014]



APPLICATIONS

Applications using rasdaman

- Earth datacube federation: satellite, weather/climate, ...
- Radio networks planning
- Planetary science: Mars, Moon, Vesta
- Cosmological simulations
- Human brain imaging
- Gene Expression Analysis



AI-Cubes™

- AI + datacubes query integration

```
for $c in (RGB_coverage)
return rsllava.chat( $c, "Describe this scene." )
```

“A river flows through farmland, surrounded by green fields and trees on both sides.”

- Model fencing:

restrict model evaluation to its comfort zone

- **Benefit:** enhanced model accuracy
- EU FAIRgeo:
Fencing AI for Enhanced Reliability in Geo Services



- ChatGPT query writing assistant

**Say Hello
to Datacubes**

Let AI write & explain
datacube analytics

To the Edge: Services in Motion

■ Rasdaman-based services done on:

- Airborne drone
- Ship
- Cubesat, demonstrated in orbit



■ Benefits:

- **Faster:** avoid full download
- **Quality of Service:** deliver insight, not pixels



■ Next: federation of data centers + drones + sats

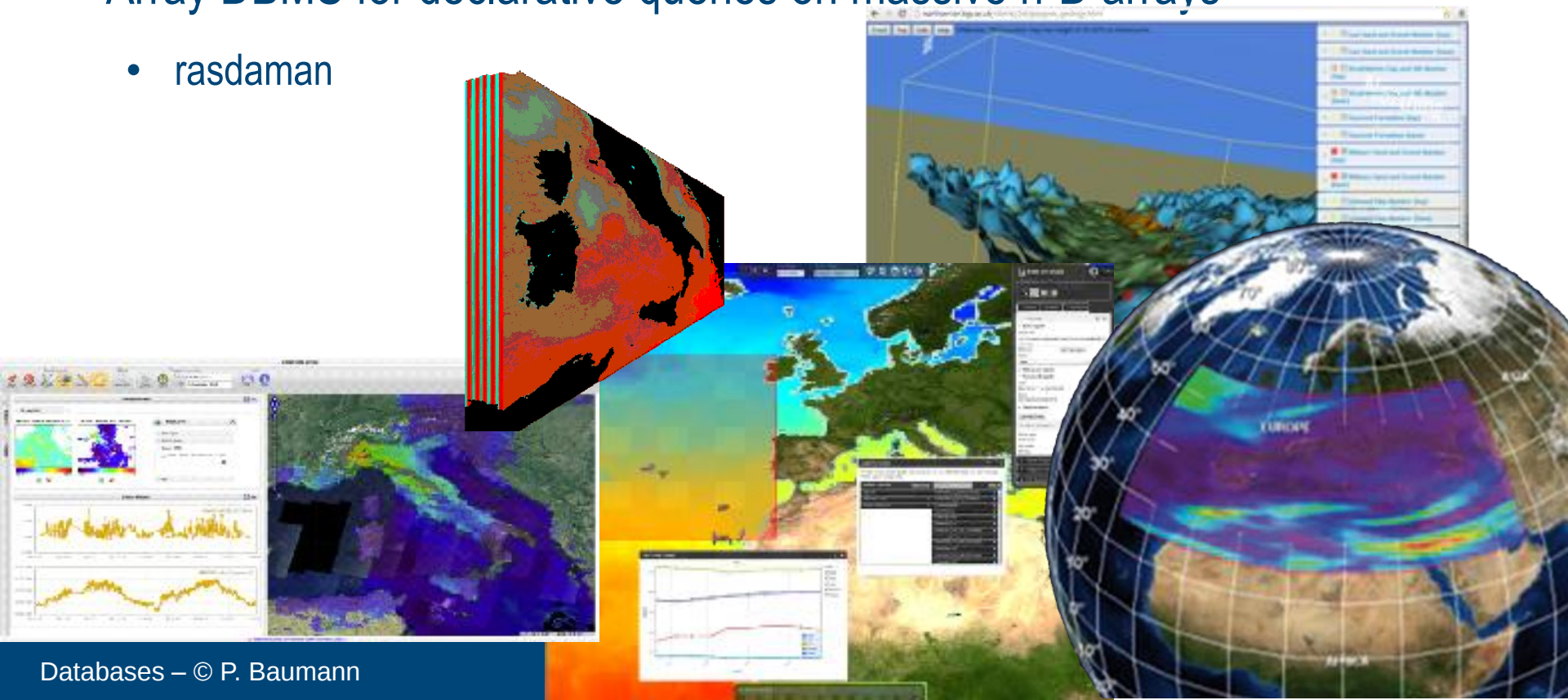
- Precursor: EU SkyFed project



WRAP-UP

Summary

- Arrays are fundamental data structure next to sets, graphs, hierarchies
 - sensor, image, simulation, statistics datacubes
- Array DBMS for declarative queries on massive n-D arrays
 - rasdaman



DEMO

Advertisement

- Seeking datacube coders
 - Thesis - [current thesis topics](#)
 - Research projects
- Common requirement:
strong coding skills
 - JavaScript. frameworks
 - Java
 - C++

