COS 787– Array databases

V Rautenbach

Introduction

- The significant increase in scientific data that occurred in the past decade
  - such as NASA's archive growth from some hundred Terabytes in 2000 to 32 Petabytes of climate observation data
  - marked a change in the workflow of researchers and programmers.
- Early approaches consisted mainly of:
  - retrieving several files from an FTP server, followed by
  - manual filtering and extracting, and then
  - either running a batch of computation processes on the user's local workstation, or
  - tediously writing and optimizing sophisticated single-use-case software designed to run on expensive supercomputing infrastructures.

Introduction

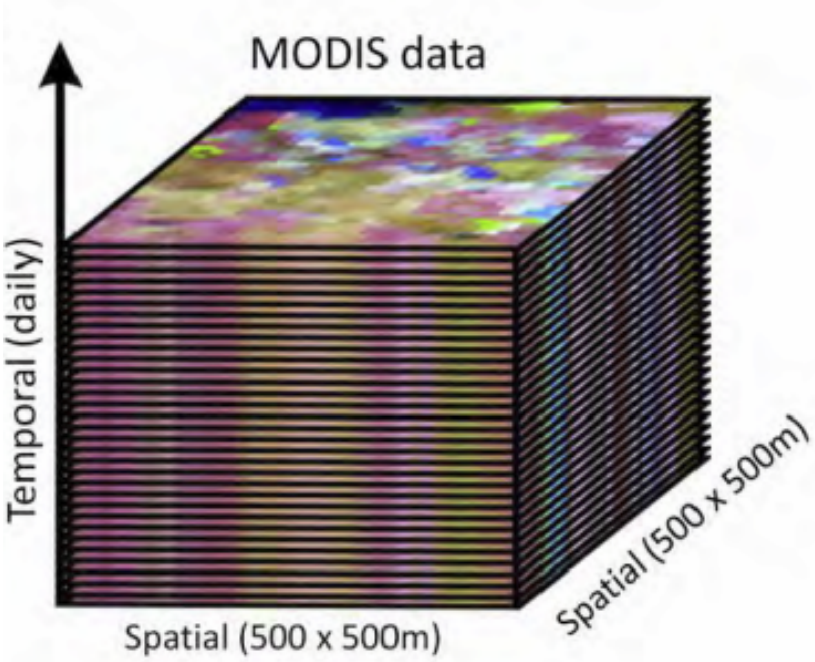
- This is not feasible any more when dealing with Petabytes of data which need to be stored, filtered and processed beforehand.
- For decades now, SQL has proven its value in any-size data services in companies as well as public administration.
- Part of this success is the versatility of the query language approach, as well as the degree of freedom for vendors to enhance performance through server-side scalability methods.
- Unfortunately, scientific and engineering environments could benefit only to a limited extent.

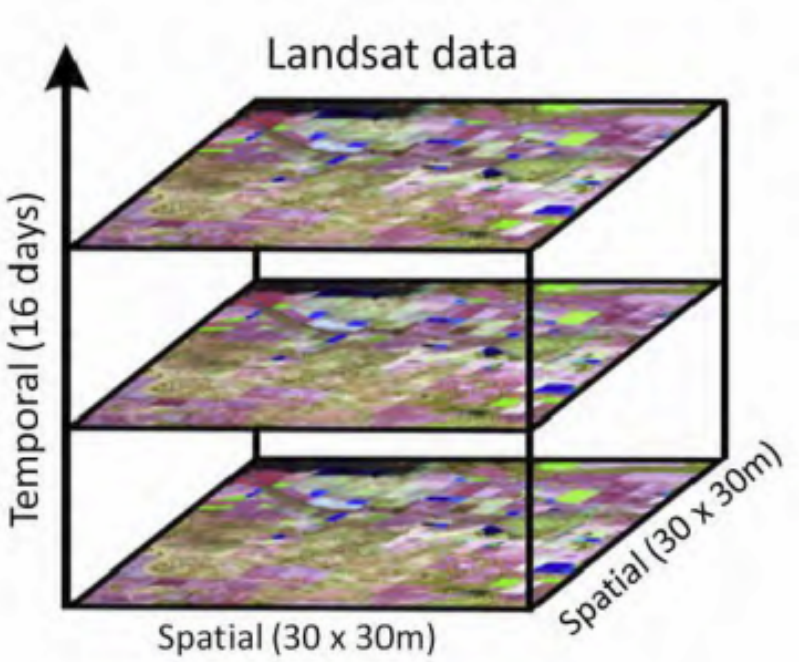
Introduction

- The main reason is a fundamental lack in data structure support:
  - While flat tables are suitable for accounting and product catalogues, science needs additional information categories, such as hierarchies, graphs, and arrays.
  - The consequence of this missing support has been a historical divide been “data” (large, constrained to download, no search) and “metadata” (small, agile, searchable).

Array databases

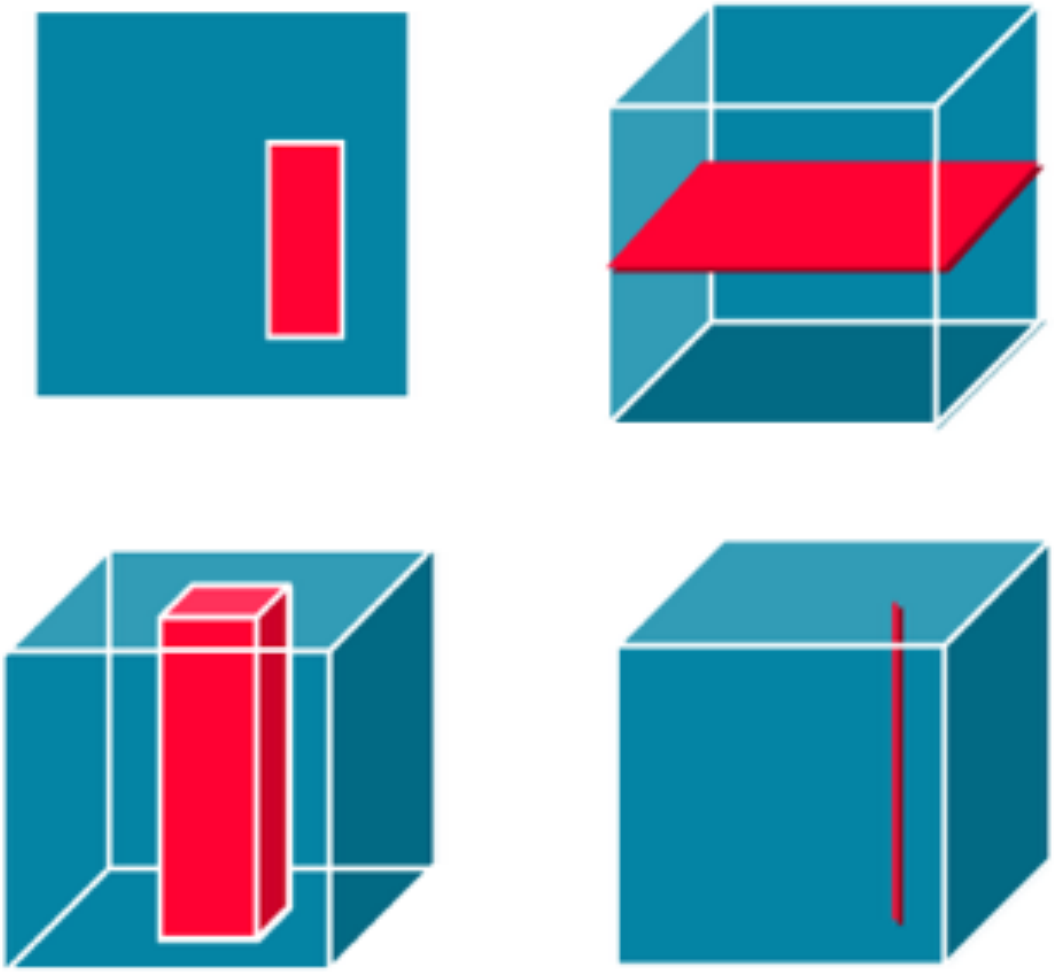
- Array databases provide flexible, scalable services on massive multi-dimensional arrays.
- Consisting of storage management and processing functionality for multi-dimensional arrays which form a core data structure in science and engineering.
- Specifically designed to fill the gaps of the relational world when dealing with large binary datasets of structured information and have gained traction in the last years.
  - In scientific communities as well as in industrial sectors like agriculture, mineral resource exploitation etc.
  - 1-D sensor data, 2-D satellite and medical imagery, 3-D image timeseries, 4-D climate models are all at the core of virtually all science and engineering domains.
- The currently most influential array database implementations are, rasdaman and SciDB.


MODIS data
Temporal (daily)
Spatial (500 x 500m)
Spatial (500 x 500m)


Landsat data
Temporal (16 days)
Spatial (30 x 30m)
Spatial (30 x 30m)

Querying Arrays

- Although array query languages heavily overlap there is not yet a common consensus on operations and their representation.
- For the brief introduction we rely on Array Algebra because it is a minimal formal framework of well understood expressiveness and forms the theoretical underpinning of the forthcoming ISO Array SQL standard, SQL/MDA



Some examples of trimming and slicing on a 3-D array,
https://www.rd-alliance.org/group/array-database-assessment-wg/wiki/introduction-array-databases

Deriving arrays

- The *marray* operator creates an array of a given extent and assigns values to each cell through some expression which may contain occurrences of the cell's coordinate.
- Let us start simple:
  - assume we want to obtain a subset of an array A.
  - This subset is indicated through array coordinates, i.e., we extract a sub-array.
  - For a d-dimensional array this subset can be defined through a d-dimensional interval given by the lower corner coordinate (lo_1, ..., lo_d) and upper corner coordinate (hi_1,...,hi_d), respectively.
  - To create the subset array we write

```
marray x in [ lo1:hi1, ..., lod:hid ]
values a[x]
```

Deriving arrays

  - This extraction, which retains the dimensionality of the cube, is called trimming.
  - Commonly this is abbreviated as

$a[\ lo_1{:}hi_1, \ \ldots, \ lo_d{:}hi_d\ ]$

- We can also reduce the dimension of the result by applying slicing in one or more coordinates.
  - This means, instead of the "$lo_i{:}hi_i$" interval we provide only one coordinate, the slice position "s_i".
  - Notably, if we slice d times we obtain a single value (or, if you prefer, a 0-d array), written as:

```
marray x in [ s_1, ..., s_d ]
values a[x]
```

victoria.rautenbach@up.ac.za