# Realtime Planetary-Scale Datacube Fusion

P. Baumann [1]
[1] Jacobs University, Bremen, Germany
p.baumann@jacobs-university.de

**Commission VI, WG VI/4**

**KEY WORDS:** datacube, fusion, distributed processing, query planning, rasdaman

## 1. INTRODUCTION

The datacube model has rapidly gained acceptance as a cornerstone for analysis-ready data, and also the corresponding service model which is more powerful, but easier to use than existing API-based interfaces. Mature and widely adopted datacube standards show the way how datacube functionality can be presented to clients, be they human or m2m connections. Specifically, the OGC Coverage Implementation Schema (CIS) data model and the Web Coverage Service (WCS) service model suite define a framework adopted by the main open-source as well as proprietary tools, including MapServer, GeoServer, GDAL, QGIS, ArcGIS, and python/OWSlib.

While the basic, most widely used functionality includes access, possibly extraction, and reformatting of a data sub(set) for download, processing and in particular data fusion represent complex, resource-intensive challenges. In the WCS suite this is addressed through a concept of modularity where WCS Core (which is mandatory for any WCS to implement) offers the GetCoverage request to accomplish "give me part of this spatio-temporal coverage, in my favourite format" and (optionally implementable) extensions add further functionality facets, up to the spatio-temporal datacube analytics language, Web Coverage Processing Service (WCPS) (Baumann, 2010).

For example, computing the NVDI over Europe on the 1st of July 2018 using Sentinel data stored in a datacube with the same name, and returning the result as a NetCDF file, can be achieved with the following WCPS query:

```
for $c in (Sentinel)
return
  encode(
   (((float) $c.nir – $c.red) / ((float) $c.nir + $c.red))
   [ Lat(35:70), Long(10:40), time("2018-07-01") ],
  "image/tiff"
  )
```

As WCPS allows combination of datasets ("joins" in database terminology) the question arises: what if datacubes to be combined reside on different computers (cloud scenario), or even in different data centers (federation scenario)? A suboptimal implementation obviously could cause massively degraded performance.

In this contribution we present federation methods implemented in the rasdaman datacube engine which is accepted technology leader, standards driver, and reference implementation for datacubes.

## 2. RASDAMAN ARCHITECTURE

The rasdaman engine resembles a fully-fledged Array Database System which utilizes a declarative query language enhancing standard SQL with high-level array operators; in fact, ISO in 2018 has adopted this language as an extension to the SQL language standard (ISO, 2019). This datacube language is domain agnostic and can serve as well medical imagery, cosmologic simulation results, and the like. A semantic layer on top of it offers geo semantics, i.e., it knows about spatial and temporal coordinates and, hence, also about regular and irregular grids. This semantics is offered via OGC WCS, WCPS, and WMS interfaces so that a plethora of clients can utilize massive datacubes managed by rasdaman. All such requests are translated to array SQL queries internally.

In the server, such incoming queries are optimized, parallelized, and ultimately mapped to datacubes that are tiled on disk (or tape). As opposed to standard regular tiling (i.e., equi-sized tiles) rasdaman allows for arbitrary tiling guided by several strategies (Baumann et al., 2010). Tiling remains invisible to the applications (and, hence, in the queries), it is a tuning parameter for the administrator. Tiles of the same datacube can even sit on different machines.

The rasdaman server is multi-parallel per se, allowing any number of simultaneous parallel requests (inter-query parallelization) as well as parallel execution of incoming queries (intra-query parallelization), see (Dumitru, 2014).

Datacube operations are usually triggered via standardized web services, such as WCS, WMS and WCPS. However, these services are only the interface to the client performing the operation, while the processing itself is handled at a lower level which enables optimized, efficient execution. In rasdaman, the geo layer intercepting the web requests representing datacube operations translates them into array database queries, which are then handled by rasdaman's query processing engine.

The engine analyses each incoming query and compiles it into machine code. During that process, a series of optimization steps is applied, one of which is distribution.

Such federations are being done routinely meantime (Baumann, 2017). In the EarthServer project, Petabyte-scale datacubes have been federated across ECMWF/UK and NCI/Australia. Another real-life use-case of the optimization is given in the federation which was set up between the German Sentinel hub, CODE-DE, established in the BigDataCube project (BigDataCube project team, 2019), and the Alfred Wegener Institute for Polar and Marine Research. Among others, temperature datacubes are available at CODE-DE, and SeaIce datacubes are available at AWI. In one of the use-cases, scientists need to compute the sea ice distribution at different temperature inter-

vals. The WCPS query that achieves that is presented below. The datacubes involved contain data at different resolutions and in different coordinate systems, which gets adjusted in the course of query processing. Most recently, FCU/Taiwain has been federated with CODE-DE.

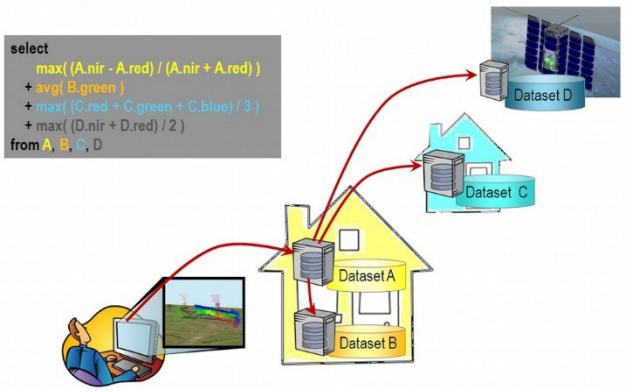


Figure 1. Sample query splitting and distribution in rasdaman

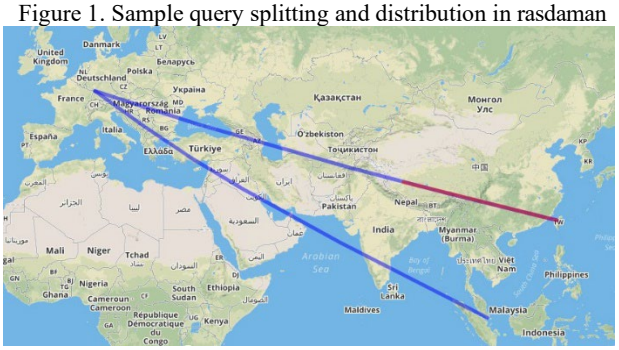

Figure 2. FCU/CODE-DE federation query path: query sent to CODE-DE from Singapore, subquery generated for FCU

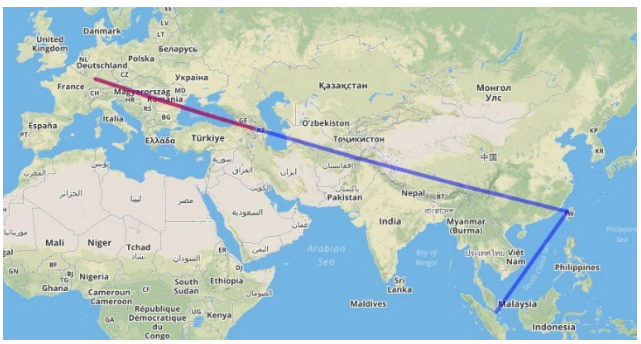

Figure 3. FCU/CODE-DE federation query path: query sent to FCU from Singapore, subquery generated for CODE-DE

## 3. CONCLUSION

In summary, rasdaman allows federated processing of declarative queries whereby complete location transparency is given: any federation member can receive the query, and the federation will dynamically orchestrate each query individually for optimized processing.

## ACKNOWLEDGEMENTS

This work is being supported by H2020 LandSupport, H2020 EOSC-hub, and German BMWi BigDataCube.

## REFERENCES

Baumann, P., 2010: The OGC Web Coverage Processing Service (WCPS) Standard. *Geoinformatica*, 14(4)2010, pp 447-479

Baumann, P., et al. 2010: Putting Pixels in Place: A Storage Layout Language for Scientific Data. *Proc. IEEE ICDM Workshop on Spatial and Spatiotemporal Data Mining*, Sydney, Australia, pp. 194 - 201

Dumitru, A. et al., 2014: Exploring cloud opportunities from an array database perspective. *Proc. ACM SIGMOD DanaC*, Snowbird, USA, pp. 1 - 4

ISO, 2019: Information technology — Database languages — SQL — Part 15: Multi-Dimensional Arrays. ISO IS 9075-15:2019

N.n., 2019: BigDataCube. http://www.bigdatacube.org/

P. Baumann, A.P. Rossi,et al., 2017: Fostering Cross-Disciplinary Earth Science Through Datacube Analytics. In. P.P. Mathieu, C. Aubrecht (eds.): Earth Observation Open Science and Innovation - Changing the World One Pixel at a Time, International Space Science Institute (ISSI), pp. 91 – 119