IAC-18.B1.4.4x44924

BigDataCube: Making Big Data a Commodity

Dimitar Mišev[a*], Peter Baumann[a], Vlad Merticariu[a], Dimitris Bellos[b], Stefan Wiehle[c]

[a] *Department of Computer Science & Electrical Engineering, Jacobs University Bremen, Campus Ring 1, 28759 Bremen*, Germany, {first name initial}.{last name}@jacobs-university.de
[b] *cloudeo AG, Ludwigstrasse 8, 80539 Munich, Germany*, {first name initial}{last name}@cloudeo.group
[c] *Remote Sensing Technology Institute / SAR Signal Processing, German Aerospace Center (DLR), Henrich-Focke-Straße 4, 28199 Bremen, Germany*, {first name}.{last name}@dlr.de
* Corresponding Author

Abstract

The BigDataCube project aims at advancing the innovative datacube paradigm – i.e., analysis-ready spatio-temporal raster data – from the level of a scientific prototype to pre-commercial Earth Observation (EO) services. To this end, the European Datacube Engine (in database lingo: "Array Database System"), rasdaman, will be installed on the public German Copernicus hub, CODE-DE, as well as in a commercial cloud environment to exemplarily offer analytics services and to federate both, thereby demonstrating an integrated public/private service.

Started in January 2018 with a runtime of 18 months, BigDataCube will complement the batch-oriented Hadoop service already available on CODE-DE with rasdaman thereby offering important additional functionality, in particular a paradigm of "any query, any time, on any size", strictly based on open geo standards and federated with other data centers. On this platform novel, specialized services can be established by third parties in a fast, flexible, and scalable manner.

To this end, several features crucial for operational services need to be tested and/or implemented, such as securing access (in particular in a distributed processing context), tuning to the specific cloud configuration of CODE-DE, and further items to be determined in the initial requirements analysis phase. The result will be the prototype of a federation of rasdaman installations on CODE-DE, cloudeo, as well as further (external) data centers; further, best practices on the use of array databases in operational environments will be established. This will pave the way for individual value-adding services by third parties.

Keywords: datacube, array database, rasdaman, security, federations

1. Introduction

In face of the Sentinel data deluge, many users — especially value-adding service providers with no deep remote sensing expertise — are overwhelmed with the task of handling the myriad of satellite images. Catalogues are aiming to provide a remedy, although with only partial success at best:

- Users still have to manually select from a plethora of available scenes.
- The selected data is almost always in shape that grossly mismatches what the user actually needs; this leads to a less than satisfactory service quality, requiring unnecessarily large data transfers and additional post-processing and computer resources on the user side.

The novel *Datacube* approach is set to change this dynamic: all data from one instrument are *spatio-temporally aggregated* and offered to users as a *single* large object [9][25][24][3]. This entails multiple advantages:

- The quantity of objects considered is far better manageable (single/double digit numbers of cubes versus millions of scenes).
- Cumbersome, user-unfriendly measures such as semantic-carrying file names are unnecessary: extraction is intuitive through precise direct subsetting in space and time.
- As multidimensional objects, datacubes unify a multitude of previously separately managed data in one model: 1-D sensor time-series, 2-D satellite scenes, 3-D x/y/t Sentinel time-series and x/y/z geophysical voxel data, as well as 4-D x/y/z/t weather forecasts.
- It is possible to offer powerful, yet simple-to-use processing functionality on datacubes, such as "vegetation index over Portugal", "areas at risk of forest fires in Greece", "combine weather time-series and satellite image analysis".
- The datacube model demonstrably allows for far more efficient evaluation of such questions. The collocation of spatio-temporally close data is amenable to scaling on a single machine with CPU / GPU parallelization, while distant data often benefits from distributed evaluation in a cloud or a grid infrastructure.

With this, users get *exactly* the subset of data needed, confectioned for answering their very question, and in the data format expected — "what you get is what you need". For example, when determining average monthly rainfall over ten years, earlier users had to download these ten years of full data whereas in a queryable datacube setup only the resulting (minimal) data come back. Not only does this allow for more comfortable and performant services as already stated, but also opens the door for rapid development of novel and complex value-adding functionality — "one cube tells more than a million images".

Flexibility and scalability of such datacubes on multi-Petabyte holdings has been established convincingly in the EarthServer initiative [1][6], at the same time showing suitability of the OGC datacube standards suite for offering analysis-ready data in a user friendly way.

The BigDataCube project [20] builds on these insights and aims to make data from current Sentinels as well as German EO missions fully accessible through a Datacube-based interface. The main data and computing resources provider is the *Copernicus Data and Exploitation Platform – Deutschland* (CODE-DE), but the offerings are expanded through seamless and secure federation with other platforms, in particular cloud-based providers of satellite imagery.

Technically, in BigDataCube the support for massive Terabyte- and Petabyte-sized Datacubes is made possible with *rasdaman*, a pioneering technology that has coined the principle of queryable datacubes [14] and has been actively developed for more than two decades. The data and service model are based on the open OGC standards thereby opening up access to a wide variety of clients, and results can be processed further by subsequent tools in the pipeline, such as GISs and AI. The standardized datacube (*coverage* in OGC-speak) model lends itself well to rapid development of innovative cloud-based methods and applications which hide the technical details and instead present the functionality through visually-interactive web clients like the NASA WorldWind virtual globe.

In more detail, the BigDataCube platform provides the following functionality in an efficient and scalable manner:

- *Registration, processing, analysis and visualization of large time-series*: analytics along time is equally simple as it is along the spatial axes with the Datacube structure, allowing for complex analytics fully integrated across time and space.
- *Fusion and integration of EO data with other related data from various sources*: it's possible to combine Sentinel data with climate simulation data for example, which is particularly straightforward through securely federating Datacube servers at different data centers. Vector data is also supported allowing for precise subsetting (*clipping*) of n-D raster Datacubes.
- *Automated validation in various phases of the processing pipeline*: during data registration a multitude of plausibility checks are performed and missing data pieces are handled as configured; the registered data can then be verified through the same interface used for analysing it, and this can be extended to validation against internal or external reference data as well.

In the next section we describe the platform technology used. Two application services are established on top of the platform, demonstrating practical feasibility: land use, developed by cloudeo (Section 3), and marine economy covered by DLR (Section 4). Furthermore, a generic demonstrator is deployed directly at CODE-DE (Section 5). The paper concludes with Section 6.

2. Platform

Figure 1 shows the architecture of the BigDataCube platform on a single machine. At its core the rasdaman Array DBMS (Section 2.1) enables scalable storage, access, and analytics on large Datacubes. An administrator is responsible for server administration and setting up the data registration process. Various applications — maritime and terrain profile in this project (Sections 3 and 4) — are established via WCS and WCPS queries, similar to how typical web applications are built on top of SQL or NoSQL DBMSs.

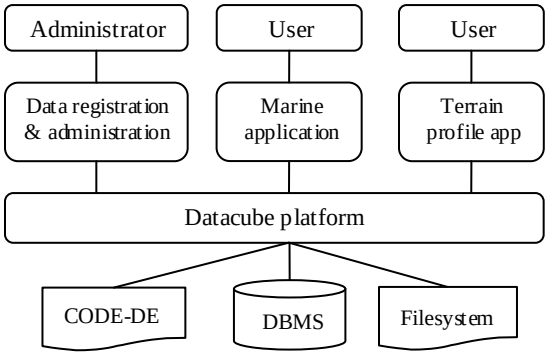

Figure 1: BigDataCube architecture diagram

Datacube servers running on separate machine are straightforward to connect into a federation as we will show below, allowing for transparent data sharing and distributed query evaluation across machines in the same data center, across data centers, or even on-board moving machines such as drones and satellites for on-the-fly access to recently acquired data.

Data access can be managed via flexible security rules, powerful enough to allow control down to single pixels (Section 2.1.3). This is combined with a billing

protocol and API allowing to fully automate services in a cloud environment.

2.1 rasdaman Datacube Server

The backbone of BigDataCube is the rasdaman Array DBMS. It has a multidimensional array data model with a declarative query language that is domain-neutral [7][8] and can likewise be used for datacubes in Earth observation, genetics, brain imaging, and cosmological simulations, to name a few applications done. Internally query processing is supported by heavily optimized storage management [5][17], query processing engine, and client-server network protocol. As such, it is an excellent base for development of more domain-specific extensions, such as the geo domain where a large percentage of the data is multidimensional arrays, ornamented with additional geo semantics. Rasdaman supports such geo data out of the box through the OGC *Web Coverage Service* (WCS) suite of standards [10] for download and on-demand processing of coverages with the *Web Coverage Processing Service* (WCPS) datacube language [13], as well as the Web Map Service (WMS) for map portrayal [16]. These standards-conformant interfaces offer the additional advantage of readily tapping into the large set of existing clients (Figure 2).

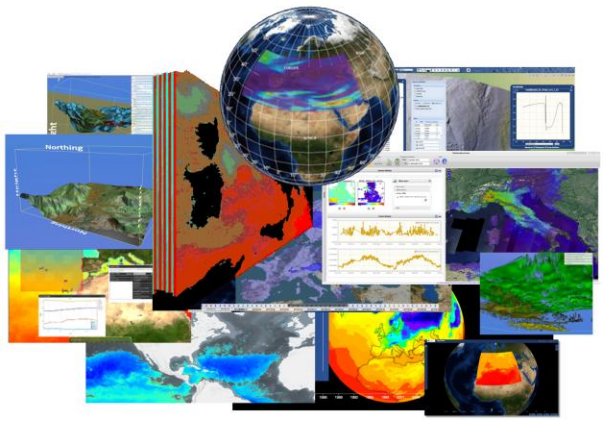

Figure 2: Kaleidoscope of clients and portals using rasdaman

The following sections cover developments in rasdaman relevant in BigDataCube: datacube creation and maintenance, support for (possibly federated) data access, processing, and portrayal, and security and billing aspects in the context of data center federations.

2.1.1 Continuous Data Registration

Large datacubes can be created in rasdaman through an OGC *Web Coverage Service - Transaction Extension* (WCS-T) standard interface [12], which enables users and machines to perform data insertion, updating, and deletion through simple Web requests secured by the database ACID properties. In particular, a choice can be made as to whether registering data (without copying them) or fully importing into the rasdaman database (which involves copying). Building large time-series datacubes, mosaics, etc. and keeping them up-to-date as new data becomes available is supported even for complex data formats and file/directory organizations.

Registering means that original data is used as-is in-situ [4], with no data duplication or modification. It requires data decoding on-the-fly during access and processing, however, which by experience usually is a minor and acceptable compromise allowing to reduce storage requirements. When processing speed is of high importance, e.g. in real-time interactive applications, data can be *imported* internally and optimally tiled / indexed in the process for the expected access patterns.

For convenience rasdaman provides a Python tool *wcst_import* which hides the complexity of building WCS-T requests for data import [19]. It allows for flexible, reproducible, and safe ingestion and maintenance of datacubes from heterogeneous datasets in all kinds of formats (GeoTIFF, NetCDF, HDF, JPEG2000, etc.) and of various degrees of data completeness.

All information necessary to build a datacube is specified in an *ingredients* file. This is the complete description of the whole ingestion process in a simple, small JSON configuration file. An ingredient is based on some supported *recipe*, such as `time_series_irregular`, `map_mosaic`, etc. The recipe, which usually administrators do not need to edit, defines the method of translating and modelling the files indicated in the ingredients file into the datacube in rasdaman. For example, with `map_mosaic` all files would be mosaiced on a single 2-D map, whereas with `time_series_irregular` the files would be placed at the correct time position on a 3-D cube (as derived from metadata or file name pattern), and further mosaicked spatially as with `map_mosaic`. Re-running wcst_import at any later point with a different set of files to be ingested will correctly update the datacube at the right positions.

To give an example of the flavour we present the ingredients `Sentinel_1A_VH.json` for Sentinel 1 data which looks like this:

```
{
"config": {
  "service_url":
    "http://localhost:8080/rasdaman/ows",
  "insitu": true
},
"input": {
  "coverage_id": "Sentinel_1A_VH",
  "paths": [
    "Sentinel_1A/S1A_*/*/*-vh-*.tiff"
  ]
},
"recipe": {
  "name": "time_series_irregular",
  "options": {
```

```
    "time_parameter": {
      "filename": {
        "regex":
          ".*-vh-(.*)-.*-.*-.*\\.tiff",
        "group": "1"
      },
      "datetime_format": "YYYYMMDDTHHmmss"
    },
    "time_crs":
      "http://localhost:8080/def/crs/
       OGC/0/AnsiDate?axis-label=\"unix\"",
    "tiling":
      "ALIGNED [0:1499, 0:1499, 0:0]"
  }
}
}
```

The *config* section holds some general settings which usually do not need to be edited, except for the *insitu* parameter which decides about registration versus import. The *input* section provides information about the source files to be considered. The recipe part contains structural information: what is the datacube type? In this case it is an image timeseries with irregular timesteps. Further, where does the timestamp of an image come from? Here it is given through a regular expression which extracts the timestamp from the file name. How should the cube be tiled internally? This is set in the *tiling* parameter, finally.

Figure illustrates the wcst_import workflow given an ingredients file.

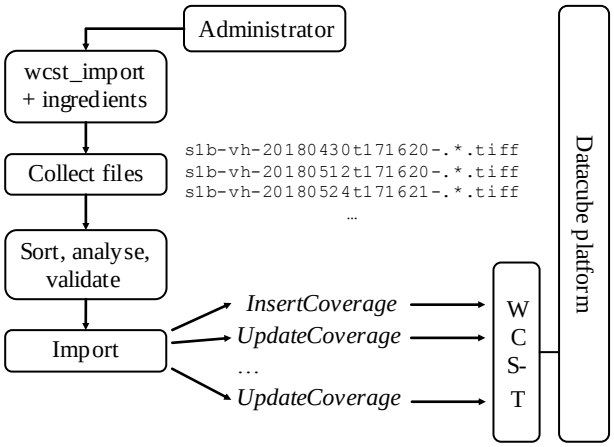

Figure 3: Ingestion process with wcst_import

In many cases a satellite mission is still running at the time of ingestion, and new data is constantly being added to the data center. This might even be organized as a rolling data archive, where most recent data pushes the oldest data to be moved to slower storage (tapes for example).

Very often datacubes are not set up in a single step, but rather get fed continuously with new imagery arriving. Therefore, rasdaman needs to continuously monitor for new data to be processed. For this purpose the wcst_import tool can continue running as a daemon process once it has finished importing the data that is currently available. In this mode the tool keeps watching the path wildcards specified in the ingredients file for any new files (the check interval is configurable). When new data appear that have not been previously seen these will be automatically imported according to the same ingredients file.

Oftentimes before importing a file some preparation, even some pre-processing with a third party tool, may be needed, usually followed by some clean up step for intermediate files etc. In wcst_import this can be handled with *pre-* and *post-import hooks*, i.e., calls to external programs or scripts before or after a file is processed. In the hooks several variables can be used, like `%filePath` for the file currently being processed.

Ultimately, this allows setting up an entirely automated datacube import and maintenance process, even for dynamic data repositories of active satellite missions. The one time step consists of crafting an ingredients file and executing wcst_import; the rest runs on autopilot.

2.1.2 *Data Access Interfaces & Federation*

As has been mentioned, rasdaman is a domain-agnostic Array DBMS. Hence, its array query language, rasql, is not used directly as it does not offer any spatio-temporal semantics. Rather, an application layer on top of it, named petascope, is used which offers geo-enriched datacubes with interfaces based on the open OGC standards (Figure 4). For pure visualization, Web Map Service (WMS) access is provided. Datacube handling is done via the Web Coverage Service (WCS) suite [10][11] which offers versatile functionality for accessing, extracting, formatting, and processing of data while retaining the original values (e.g., returning height values instead of color-coded pixels like WMS). One notable component of the WCS suite is the Web Coverage Processing Service (WCPS), a high-level geo datacube analytics language [13]. Lateron we will see some concrete examples of WCPS queries.

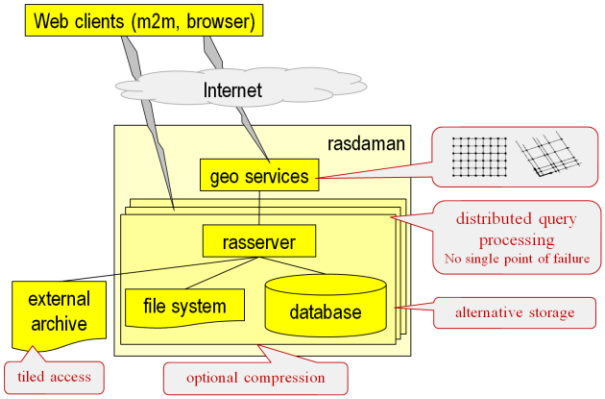

Figure 4: rasdaman schematic architecture

All WMS, WCS, and WCPS requests internally are translated into rasql queries which undergo all of rasdaman's optimization, multicore CPU/GPU parallelization, and distributed processing techniques for achieving optimal response times.

Distribution of queries for parallel processing on different nodes is, first of all, rasdaman's mechanism for cloud-based processing. As opposed to rigid, inflexible approach of MapReduce-type systems where first map functions and then reduce functions are executed in lock-step synchronization, rasdaman performs an individual splitting of incoming queries into sub-queries. To this end, every rasdaman node gets continuously updated with the data holdings of all partner nodes so that global optimization becomes possible. Splitting is optimized according to (i) ship as much work as possible to where the data sit and (ii) minimize inter-node communication; in future, we plan to enhance this with further strategies.

This very same principle of informed query splitting is also used for federated processing between data centers. Obviously, here minimization of internal data transport during query processing is of particular importance. Such federations have been established, for example, between the European Centre for Medium-Range Weather Forecast (ECMWF) and National Computational Infrastructure (NCI) Australia; cloud parallelization has been performed across more than 1,000 Amazon nodes [6]. In BigDataCube, the federation principle is being applied in a new context, between public and private federation partners CODE-DE and cloudeo.

2.1.3 Security and Billing

The transition from all-open data in EarthServer to partially protected assets in BigDataCube brought along an extra challenge – access control became mandatory. As every node in a federation retains full autonomy there is a strong requirement to enable local administrators with complete control over what their contributing nodes are allowed to do, and what not.

The first mechanism rasdaman offers is local definition of allowed communication partners. In the rasdaman configuration, so-called *inpeers* can be listed from which queries are accepted; all other senders of queries will not get honoured. Conversely, an *outpeers* list restricts the nodes to which the concrete instance wants to send sub-queries. Altogether, this gives each administrator control over the information flow in the network.

Complementarily, role-based access control operates on data level. Named users get associated with roles, and each role gets configured in turn with rules defining access rights. First and foremost, read and write access can be distinguished. Further, sets of objects or individual objects can be protected.

However, this is not enough in face of "Big Datacubes", rather protection of parts of a datacube is required. For example, ECMWF offers the long tail of its climate data for free; the most recent two weeks, however, are charged. In rasdaman, query expressions are used to determine particular areas, and this gives complete freedom down to the level of single pixels. Examples include protection of areas given by a bounding box; given by a mask; or given by a bounding polygon.

Information about the cost of a query actually is already available in rasdaman as this is used in its cost-based optimization to find the most efficient way of executing a query, where there typically are many different options. The concept includes relevant detail information like amount of data required to be read from disk; processing cost; internal data transfer cost; result size. This information now is made available externally for access control purposes in addition.

Altogether this fine-grain access control allows service providers to retain full control over their offerings. Further, it allows defining quota where excessive costs are avoided upfront. Users may even ask beforehand about the costs their query would incur.

Notably, such safeguarding is particularly important in situations where datacube processing is not done interactively by an end user, but as part of a larger processing chain where a chain of tools orchestrates itself without human interference.

3. Terrain Profile Application

The cloudeo BigDataCube web app provides user-friendly access to accurate and reliable terrain profiles anywhere on the globe, without the need to manage any data [5], see Figure 5. This service is particularly helpful for users in the telecommunication industry looking for detail information on terrain/surface structure and for line of sight and microwave link design when planing for new towers and upgrading existing networks coverage.

Locally, NEXTMap 5 and PlanetDEM 30 Plus elevation data are registered and accessed through rasdaman. The Intermap NEXTMap data has 5m resolution and absolute vertical accuracy 1m RMSE (1.65m LE90); a digital surface model (DSM), as well as a digital terrain model (DTM) with features such as buildings and trees removed are available. The Planet Observer PlanetDEM data is SRTM 30m resolution, and is bundled with other source data (ASTER, cartographic, etc.). Altogether this data is around 1.5TB in size at the moment, and is hosted on two VMs (each with 2 CPU cores and 8 GB of RAM) — an internal private one for the commercial data, and a public one for the open data. Behind the scenes, the terrain profile is calculated with a WCPS query like this:

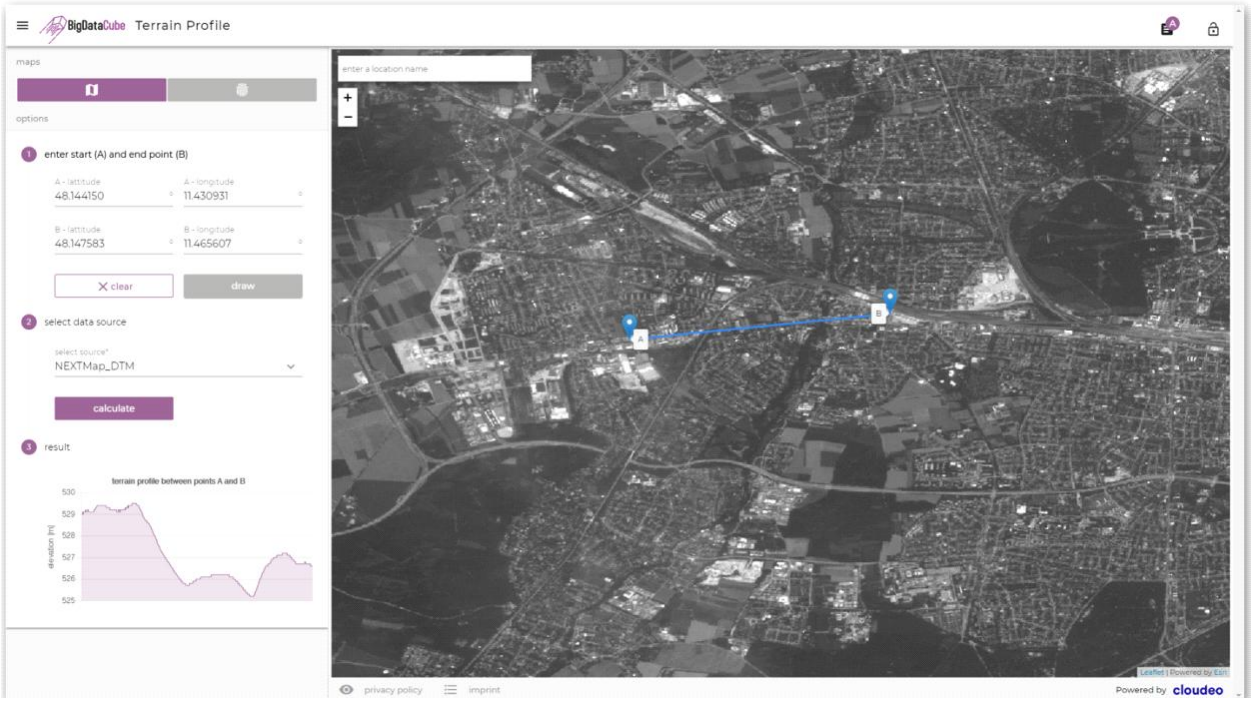

Figure 5: Terrain profile web application at cloudeo based on the BigDataCube platform: users select their path of interest (right), and the elevation profile get extracted on the fly (left).

```
for $c in (PlanetDEM_30_Plus__Western_Europe)
return encode(
  clip(c, LineString(48.210068 11.006927,
                     48.183975 11.199188))
, "json")
```

The local rasdaman instance is federated with the rasdaman running at CODE-DE, so that all the Sentinel-2 data is accessible at cloudeo as if it was available locally. For visualization, this data is used as a base map which can be further presented in different band combinations as chosen by the customer, depending on the current application.

4. Marine Application

Remote Sensing of the oceans is a crucial task for maritime safety and security, important for vessels at sea, coastal protection or offshore constructions. Due to its independence from sunlight and cloudiness, global coverage and high resolution, space-borne Synthetic Aperture Radar (SAR) is an indispensable source of information of the ocean surface and for coastal applications.

Within the BigDataCube platform, ocean wind and sea state are retrieved from Sentinel-1 SAR imagery. Wind retrieval is based on a Geophysical Modelling Function (GMF), relating radar reflection caused by sea surface roughness to wind speeds [1]. For sea state, an empirical algorithm was developed for TerraSAR-X X-band data [2] and adopted for Sentinel-1 C-band Interferometric Wide Swath (IW) data [3]. Coastal waters are generally acquired by Sentinel-1 in the IW mode with a swath width of 250 km, so most coastal Sentinel-1 scenes worldwide can be used for the wind and sea state retrieval.

Data are processed for subscenes of 2.5 by 2.5 km with a raster step of 6 km in both flight and range directions, resulting in about 1,200 values for one IW scene. A series of procedures are included for land masking, artefact filtering (outliers like ships, buoys, oil silks) and control of results. The algorithm's function is based on the spectral analysis of subscenes in wavenumber space using Fast Fourier Transform (FFT). The empirical function allows a direct estimation of the significant wave height *Hs* from image spectra without first converting them into wave spectra and uses integrated image spectra parameters as well as estimated local wind information. A texture analysis based on Grey Level Co-occurrence Matrices (GLCM) is also applied.

All algorithms operate fully automatically. No operator supervision is necessary and new products can be generated immediately once new acquisitions are available. Data from the Weather Research and Forecasting model (WRF) is automatically accessed to retrieve wind direction with which the wind speed is calculated. Using these results, the sea state is calculated for the given scene.

Below we list several examples for the application of this technology on the BigDataCube platform.

4.1 Sea state safe for construction

Offshore wind park construction can only be performed on days with low sea state conditions – concretely, wave height may not exceed 1.3 m, or it becomes too unsafe for workers. A statistical analysis at a wind park construction site can show the historical ratio of safe working days to support construction planning.

For example, let us take the area around the existing Amrumbank West wind park near Helgoland in the North Sea. The following WCPS query divides the number of time-slices where waves around the wind park area are on average less or equal than 13 decimetres (1.3 metres) with the total number of time-slices available in the `Sentinel_1A_hs_gc` Datacube:

```
for $c in (Sentinel_1A_hs_gc) return
count(
  coverage safeWaveHeight
  over $i unix(imageCrsDomain($c, unix))
  values avg($c[Lat(54.3: 54.6),
                Long(7.55:7.85),
                unix:"CRS:1"($i)]) <= 13))
/
 (condense +
  over $j unix(imageCrsDomain($c, unix))
  using 1.0)
* 100.0
```

For the Sentinel data available in the current CODE-DE precursor service we get that `60.0%` of the days were historically safe for construction; the query can be easily tweaked further to accommodate any specific needs, e.g. restrict only to the summer months or take the maximum rather than the average height.

4.2 Finding sustainable locations for wind farms

Wind speed is a main parameter for economic sustainability of a wind park. However, wind wakes from existing parks may reduce wind speed for several dozen kilometres. Using data derived from Sentinel-1 acquisitions, a temporal history of wind speed on a planned construction site can be retrieved, including wake effects which are often not included in wind models.

On the same area as in the previous example, we now go into the `Sentinel_1A_windspeed_gc` datacube to calculate average wind speed for each time slice with the following WCPS query:

```
for $c in (Sentinel_1A_windspeed_gc)
return encode(
  coverage safeWaveHeight
  over     $i unix(imageCrsDomain($c, unix))
  values   avg( $c[ Lat(54.3: 54.6),
                    Long(7.55:7.85),
                    unix:"CRS:1"($i) ] )
, "csv")
```

The result (Figure 6) is a 1-D array, here delivered in comma-separated values format (CSV), which can be directly plotted as a diagram in the Web client:

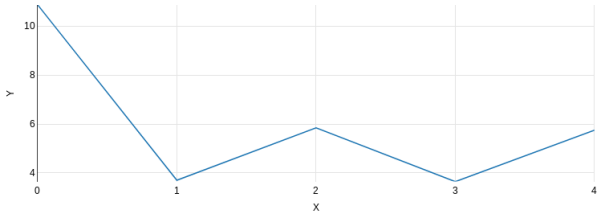

Figure 6: Average wind speed (Y) over time (X)

4.3 Model adjustments

Wind speed derived from a full Sentinel-1 scene can be compared to model data for a given time, allowing for a full 2-dimensional comparison of the wind situation compared to only having point measurements from buoys, and supports the adjustment of model parameters.

The query below overlays wind speed on 2018-03-31 (5pm) over the 4th band of Sentinel 2 data on 2018-02-07 (10:42am) on the same area:

```
for $c  in (Sentinel_1A_windspeed_gc),
    $b4 in (Sentinel_2A_B04_10m)
return encode(
  image.stretch(
  scale($c[Long(7.6550:8.16343),
           Lat(53.7: 54.021),
           unix("2018-03-31T17:00:00.000Z")],
        { imageCrsDomain(
       $b4[Long(7.6550:8.16343),
           Lat(53.7: 54.021),
           unix("2018-02-07T10:42:11.000Z")])
        }))
  overlay
  ((unsigned char)(
  $b4[Long(7.6550:8.16343),
      Lat(53.7: 54.021),
      unix("2018-02-07T10:42:11.000Z")]/5.0))
, "jpeg")
```

The wind speed product is lower resolution than the Sentinel-2 data so it needs to be scaled up to the same resolution; in addition the *`image.stretch`* user-defined function (UDF) is applied to enhance the contrast. Such UDFs represent administrator--provided bespoke code that gets linked dynamically into the server to extend the query language's capabilities. The result (Figure 7) is encoded in JPEG format; brighter sea areas indicate higher wind speed, while darker areas indicate lower wind speed.

In future, more maritime products generated from SAR acquisitions could be implemented to answer questions such as: Has the safety zone around this wind park been entered by any ships in the last month? How often was oil spill detected around that oil platform? Have any ice bergs been detected in that area in the last year? Which parts along this shipping route are currently covered by sea ice?

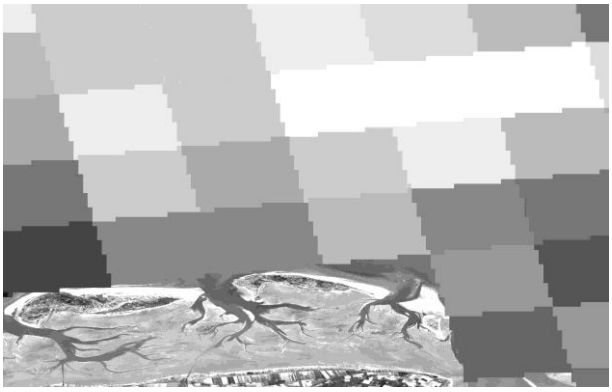

Figure 7: Wind speed overlayed on Sentinel-2 data

5. CODE-DE Datacube Services

While awaiting the larger-scale resources to be allocated by the CODE-DE operator, DLR, a precursor CODE-DE VM with 50 GB of Sentinel-1 and -2 data has been established for testing and demonstration. Once this infrastructure is available the full archive of CODE-DE will be accessible as datacubes.

The Datacube Services demonstrator [21] provides an access point for users, and showcases the capabilities according to datacube dimension, application domain, geo standards, and clients (Figure 8).

An example dialog using the WCPS console, an expert tool for writing queries directly, is shown in Figure 9.

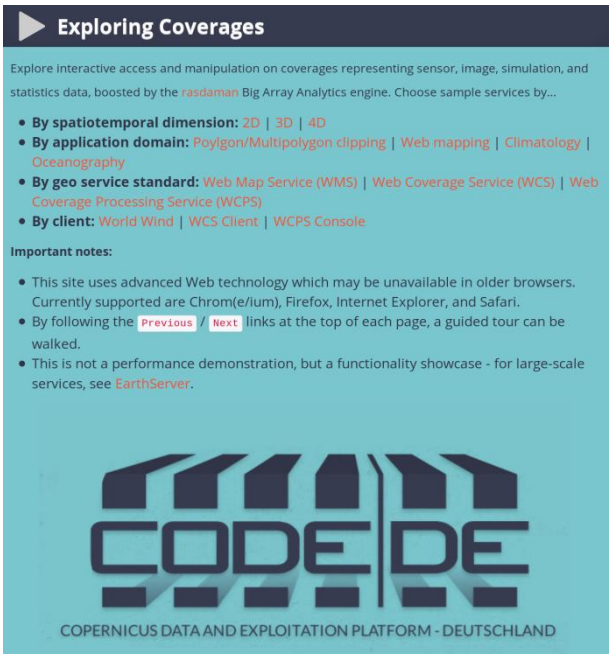

Figure 8: Overview of the initial-stage CODE-DE Datacube Service demonstrator

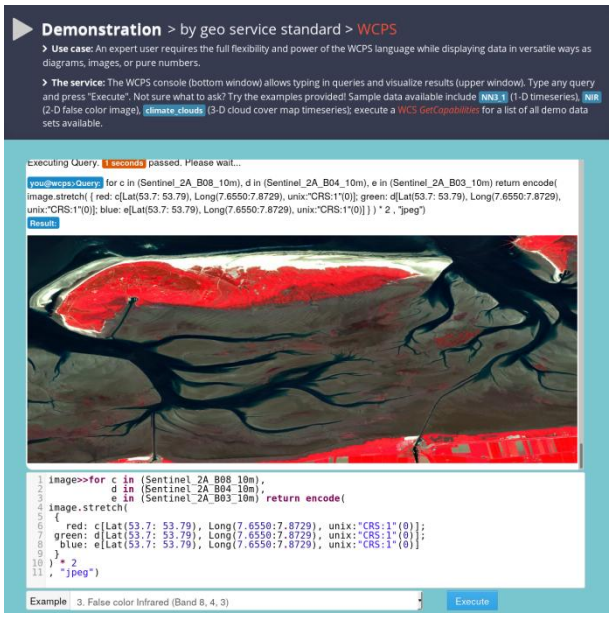

Figure 9: WCPS example: false color band combination on Sentinel-2 data

6. Conclusion

Datacubes today are well accepted in the Earth science community as a central means for achieving analysis-ready data, i.e.: provide data in a way that users can achieve their analysis and visualization tasks with minimal effort and technical expertise. The rasdaman team has first proposed the concept, has implemented it over a substantial period into a flexible, scalable, Peta-scale proven framework, and – as an unanticipated side effect – has become a leader in datacube standardization, not only in the arth Sciences with OGC, ISO, and INSPIRE geo datacube standards, but also by extending the ISO SQL language with datacube functionality [15].

In the BigDataCube project, the bridge between public and private datacubes is closed, thereby finally bringing Earth datacubes into industrial use.

Acknowledgements

This work is partially funded by the German Ministry of Economy and Energy (BMWi).

References

[1] P. Baumann et al.: Big Data Analytics for Earth Sciences: the EarthServer Approach. International Journal of Digital Earth (2015)

[2] P. Baumann, D. Misev, V. Merticariu, B. Pham Huu: Datacubes: Towards Space/Time Analysis-Ready Data. In: J. Doellner, M. Jobst, P. Schmitz (eds.): Service Oriented Mapping - Changing Paradigm in Map Production and Geoinformation Management, Springer Lecture Notes in Geoinformation and Cartography, 2018

[3] P. Baumann, E. Hirschorn, J. Maso, A. Dumitru, V. Merticariu: Taming Twisted Cubes. Proc. ACM SIGMOD Workshop on Managing and Mining Enriched Geo-Spatial Data (GeoRich), San Francisco, USA, June 26 - July 01, 2016

[4] P. Baumann, A. M. Dumitru, V. Merticariu: The array database that is not a database: file based array query answering in rasdaman. Advances in Spatial and Temporal Databases, Springer Berlin Heidelberg, 2013. 478-483

[5] P. Baumann, S. Feyzabadi, C. Jucovschi: Putting Pixels in Place: A Storage Layout Language for Scientific Data. Proc. IEEE ICDM Workshop on Spatial and Spatiotemporal Data Mining (SSTDM'10), December 14, 2010, Sydney, Australia

[6] P. Baumann, A.P. Rossi, B. Bell, O. Clements, B. Evans, H. Hoenig, P. Hogan, G. Kakaletris, P. Koltsida, S. Mantovani, R. Marco Figuera, V. Merticariu, D. Misev, B. Pham Huu, S. Siemen, J. Wagemann: 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), 2017, pp. 91 - 119

[7] P. Baumann. A Database Array Algebra for Spatio-Temporal Data and Beyond. Proc. NGITS99, LNCS 1649, pp. 76–93, Springer, 1999

[8] P. Baumann: On the Management of Multidimensional Discrete Data. VLDB Journal 4(3)1994, Special Issue on Spatial Database Systems, pp. 401 – 444

[9] P. Baumann: The Datacube Manifesto, http://earthserver.eu/tech/datacube-manifesto, seen 2018-09-06

[10] P. Baumann: Web Coverage Service Standard – Core, version 2.0.1. OGC 09-110r4, http://www.opengeospatial.org/standards/wcs; seen 2018-09-06

[11] P. Baumann: Coverages Uncovered: Agile Analytics on Spatio-Temporal Data Cubes. In: S. Morain (editor-in-chief): ASPRS Manual on Remote Sensing, version 4, 2017

[12] P. Baumann: Web Coverage Service Interface Standard – Transaction Extension. OGC 13-057r1, https://portal.opengeospatial.org/files/13-057r1, seen 2018-09-06

[13] P. Baumann: The OGC Web Coverage Processing Service (WCPS) Standard. Geoinformatica, 14(4)2010, pp. 447 – 449

[14] P. Baumann: Language Support for Raster Image Manipulation in Databases. Proc. Int. Workshop on Graphics Modeling, Visualization in Science & Technology, Darmstadt/Germany, April 13 - 14, 1992, Springer 1993, pp. 236 – 245

[15] D. Misev, P. Baumann: Enhancing Science Support in SQL. Proc. Workshop Data and Computational Science Technologies for Earth Science Research (co-located with IEEE Big Data), Santa Clara, US, October 29, 2015

[16] J. de la Beaujardiere: Web Map Service (WMS) Implementation Specification. OGC 06-042, http://www.opengeospatial.org/standards/wms, seen 2018-09-06

[17] P. Furtado, P. Baumann: Storage of Multidimensional Arrays Based on Arbitrary Tiling. Proc. 15th IEEE Int. Conf. on Data Eng., pp. 480–489

[18] S. Lehner, A. Pleskachevsky, M. Bruck: High resolution satellite measurements of coastal wind field and sea state. Int J Remote Sensing, 33(23)2012, pp. 7337-7360

[19] N.n.: rasdaman v9.7. http://doc.rasdaman.org/, seen 2018-09-06

[20] n.n: BigDataCube – Big Earth Datacube Analytics Made Easy. http://www.bigdatacube.org/, seen 2018-09-06

[21] n.n: CODE-DE Datacube Services. http://code-de.bigdatacube.org/, seen 2018-09-06

[22] A. Pleskachevsky, W. Rosenthal, S. Lehner: Meteo-Marine Parameters for Highly Variable Environment in Coastal Regions from Satellite Radar Images. ISPRS J Photogramm Remote Sensing, 119, 464-484, 2016

[23] A. Pleskachevsky, S. Jacobsen, B. Tings, E. Schwarz: Sea State from Sentinel-1 Synthetic Aperture Radar Imagery for Maritime Situation Awareness. Int J Remote Sens (accepted), 2018

[24] P. Strobl, P. Baumann, A. Lewis, Z. Szantoi, B. Killough, M. Purss, M. Craglia, S. Nativi, A. Held, D. Trevor: The Six Faces of The Datacube. Proc. Conf. on Big Data from Space (BiDS'17), 28-30 November 2017, Toulouse, France

[25] A. Tzotsos, A. Karmas, V. Merticariu, D. Misev, P. Baumann: A Datacube Approach to Agro-Geoinformatics. Proc. 6th Intl. Conf. on Agro-Geoinformatics, Fairfax, USA, 7 August 2017