FROM SENSOR-CENTRIC TO USER-CENTRIC - WHEN ARE DATA ANALYSIS-READY?

Peter Baumann

Jacobs University, p.baumann@jacobs-university.de

Abstract – Despite the current wave of providing data analysis-ready we claim that some essential properties for easy, non-EO-expert and non-programmer exploitation of EO data are not usually considered in service design. These properties relate to the quality of service (human or machine) users experience, and conversely to the burden that is imposed when accessing archives. We spot some critical features and propose solutions.

Index Terms— Analysis-ready data, datacubes, standards, coverages, Web Coverage Service (WCS), Web Coverage Processing Service (WCPS), OGC

INTRODUCTION

The term Analysis-Ready Data (ARD), originally coined by the USGS Landsat team in 2017 [14], has seen a rapid uptake in the Earth Observation (EO) community. Not surprisingly, we encounter a variety of different interpretations which, however, all agree that EO data need to be offered in a way better suitable for consumption in particular by non-programmers and non-EO experts.

CEOS recently started to propagate CEOS Analysis Ready Data for Land (CARD4L) as data processed to allow "immediate analysis with a minimum of additional user effort and interoperability both through time and with other datasets" [9]. Among some metadata requirements CARD-4L implies radiometric and geometric calibration plus solar and view angle correction and atmospheric correction (optical) and topography and incidence angle correction (radar).

Obviously, among the core features of such data is to offer EO data in a homogenized, aggregated manner which abstracts away from particular storage organizations and encodings which traditionally pose problems to users – sometimes described as going "from files to pixels" to indicate the different semantic level of EO offerings. Standards are helpful here if they establish an abstraction not based on files and scenes, but on higher-level objects, such as the OGC Web Coverage Service (WCS) suite [2][1][6][7].

As temporal analysis constitutes today's killer application in EO it is indispensable that analysis readiness does not only address horizontal spatial extent (as has been achieved with seamless maps) but also time axis. Ultimately, all spatio-temporal axes should be included thereby having elevation and bathymetry, too. In the end, spatio-temporal analysis readiness inevitably leads to the concept of multi-dimensional datacubes, first presented in [8]; see also [1].

However, while the advantages of such a data organization for access (i.e., simple download) of data are imminent. Even ftp download, however, constitutes a service API, albeit with rudimentary functionality – and this is what we observe many organizations still focusing on. However, users today want to get away from a service philosophy of "go take the data and do the analysis yourself" but rather expect server-side analysis capabilities. Obviously, the quality of service is of crucial relevance for user uptake. We claim, therefore, that in parallel to analysis-ready data we need to consider *analysis-ready services*. In this contribution we first inspect the state of the art, based on the ESA Sentinel archives. By doing so we spot several shortcomings which allow us to propose corresponding steps towards better EO service quality. To demonstrate feasibility of these ideas we present their realization in the European Datacube Engine, rasdaman.

The remainder of this paper is organized as follows. In Section 2 we exemplarily describe EO archive structures which complicate access. In Section 3, we introduce steps for improvement, and in Section 4 we describe a sample implementation of such steps. Section 5 concludes the paper.

EO SERVICES: STATUS AND IMPEDIMENTS

In contrast to many discussions about analysis-ready data we adopt a holistic approach and consider consequences of design decisions for the user experience. The central question guiding us is: How much knowledge and work is needed by the client in order to perform a particular task in some server? Knowledge includes aspects such programming skills required for performing a given task; work refers to the number of steps to be performed by the client, their complexity, and their resource needs.

We are of the opinion that such questions are applicable to both human users – typically accessing a service through some visual point-and-click client – and machine users where some algorithm – possibly deeply hidden in some service mash-up – when accessing a service. For the service as such this does not make any difference as it invariably "sees" the client through protocols and API invocations. Therefore, we prefer talking about clients than users in the sequel. Based on these considerations we establish some sample service request situations which will form our test cases subsequently.

978-1-5386-9154-0/19/$31.00 ©2019 IEEE

1.1. EO Archive Access Use Cases

The service features commonly discussed go substantially beyond downloading of objects or parts of it ("subsetting"), but include various aspects of server-side processing in the widest sense (not that already reformatting into another encoding involves CPU cycles). We find the following classification useful:

Data access: complete download of a particular object which has been identified through some search, link, or metadata reference. Implementations emphasizing simplicity of the server code often require that the object be returned in its exact original byte steam representation, such as the data format in which the object is stored in the server. A typical example is OGC Web Map Tiling Service (WMTS).

Data extraction: download of a part of an object identified. As this means drilling into the object anyway this use case is often combined with re-encoding into some client-selected data format. A typical example is OGC Web Coverage Service (WCS) Core.

Data filtering: prior to downloading find out whether some data object is fit for your purpose. This can require inspection of both data and metadata. OGC Web Coverage Processing (WCPS) can do this on sets of datacubes [5].

Data processing: apply some computational steps to an object in the server (following the Big Data paradigm of "ship code to server") and ship the resulting (new) object to the client. This can be a fixed, predefined process (such as through an OGC Web Processing Service (WPS) process) or an ad-hoc, flexible query (such as through an OGC WCPS processing request). For reasons of differentiation we assume processing of always one object, as the case of combination is addressed separately, coming next.

Data fusion: recombine a result object from two or more server-side objects. In the most general case these objects can reside on different servers, obviously subject to different, independent regimes of data presentation in terms of extent, resolution, Coordinate Reference System (CRS), etc.

Data maintenance: modify the offering of a remote service by either creating a new object, update all or part of an existing object, or delete an object from this server. Such updates must be possible concurrently to other client access and therefore need to adhere to the well-known ACID transaction properties.

1.2. ESA EO Archive Data Provisioning Case Study

For the Sentinel archives ESA suggests the SAFE format for uniform access to data offered. EO data are preprocessed into so-called granules which can be seen as tiles. From our perspective, some properties of SAFE are in particular practically relevant; we discuss these in turn.

A granule covers an area of 100 x 100 km. This leads to file sizes of typically 600 – 800 MB meaning that users must download files of these sizes for any processing. Moreover, it also means that any service working on such granules must load units of this size into main memory before processing of any request can start, be it a simple WCS GetCoverage or a complex WCPS analytics request. Detailed benchmarks have shown that an optimal tile size for general-purpose extraction and processing is in the area of 3 – 5 MB [10]. Hence, the units of storage access are by about two orders of magnitude too coarse for being efficient.

Further, a SAFE file is a zip archive containing the pixel payload plus a series of metadata. In terms of storage access this means that the zip file needs to be opened and the image file(s) need to be extracted. Depending on the implementation of the zip decoder this may mean significant extra processing which slows down server-side result generation.

Finally, image files are provided in the JPEG format following a lossless encoding regime. Using JPEG – despite its lossless storage – has several relevant consequences. As JPEG applies a transformation from time to spectral space, reconstructing a pixel from a JPEG stream requires (i) accessing several memory locations and (ii) significant CPU cycles. Altogether, albeit such data will already be in RAM this means extra overhead slowing down response generation in the server.

Yet another consequence of the wavelet-based storage of Sentinel products is the inability to optimize spatio-temporal subsetting, one of the most basic and widely used access operations at all. Some formats, like TIFF and NetCDF, support internal tiling which an intelligent server may exploit to load less than the whole file for subsetting requests. Obviously, considering the hundreds of Megabytes of file sizes, this can mean a significant difference in data loading. In contrast, with JPEG such a data load optimization is not possible as data are structured in a completely different manner on disk. In passing we note that all these computational steps may require intermediate representations in main memory or, even worse, on disk which additionally impacts request response time and server resource consumption.

All these considerations also hold for updates, in particular partial updates which are common when building and maintain a datacube.

RECOMMENDATIONS TOWARDS ANALYSIS-READY SERVICES

In this Section we set up a set of requirements aiming at making services more analysis-ready. Our guidance is simple: *how much effort – again, in terms of knowledge and resource requirements – does it take for a client to access and process a particular pixel set in the course of decision making?* Based on the observations we propose the following set of recommendations for EO service providers in order to achieve a high level of service quality. We differentiate between data and service modeling aspects, bearing in mind though that both are tightly intertwined.

Requirement 1: Provide data access in a granularity suitable for efficient storage access across all spatio-temporal dimensions, i.e.: x/y/z/t. This can be achieved by either retiling of data into a scheme that best fits client access patterns or at least utilizing some file format that supports internal tiling, such as NetCDF. Tile shape and size must be adjustable for the server architecture and workload – there is no “one size fits all” tiling for spatio-temporal data. Particular algorithms (like convolutions) and user scenarios (like disaster mitigation) will lead to different most suitable tilings. As normally more than just one application should be supported there will regularly be conflicting optimal tiling schemes, in which case a tradeoff will have to be found. For datacubes this applies to all x/y/z/t dimensions, hence traditional 2D GeoTIFF archives will show degraded performance.

Requirement 2: Minimize the number of CPU cycles and storage / memory access required for reconstructing a given pixel set in main memory. This rules out wavelet-based encoding options.

Requirement 3: Store data analysis-ready. Reconstruction of analysis-ready data on the fly is not only inefficient (if almost every query will require the same processing) but may introduce numerical inconsistencies. The authoritative values should be readily available in the database / archive. In terms of the usual processing levels, this excludes Level 1a and 1b; analysis-readiness in the sense of “we can logically aggregate into user-centric units such as datacubes without any loss of precision” starts with Level 1c (error corrected, radiometrically corrected, orthorectified).

Requirement 4: Ship code to data. Surprisingly, this well-known Big Data principle is not always implemented today. Low-level ftp, RESTful subsetting APIs, etc., do not allow server-side processing, but leave that to the client. However, also many python-based APIs, as well as WPS-based approaches, require application code to run on the client with just procedural calls to fixed server-side functionality. Instead, clients should be able to ship their processing requests for execution on the server, close to the data to avoid expensive data shipping round trips.

Requirement 5: High-level server-side filtering and processing language. While “ship code to data” is a must implementations vary widely in the API quality. Sometimes procedural source code, such as python, is shipped to the server for execution – obviously, a major security hole. Instead, a high-level, declarative language should be provided at the abstraction level of, say, SQL with its tremendous success. An equivalent is given by the WCPS Earth datacube analytics language [5] which is declarative, has a well-defined semantics, and is adopted OGC standard.

Requirement 6: Transparent federation. Data fusion often requires combination of objects sitting in different data centers. Ideally, the task of extraction, download, homogenization, and combination should not be with the client, but on the server. This requires intelligent ad-hoc orchestration of arbitrary servers, including optimization of data exchange and processing distribution. Obviously, federation has a potential for massively boosting ease-of-use and performance.

Requirement 7: Open standards. In the spirit of interoperability data and service APIs should adhere to well defined and curated standards. Looking at the rigor of maintenance required this calls for standards, e.g., by OGC, ISO, and OASIS Open; in contrast, e.g., the W3C Spatial Data on the Web group has disbanded after releasing its documents. For EO data, specifically, the OGC and ISO standards apply which have the additional advantage of being kept in lock-step synchronization (e.g., OGC CIS [2] is identical to ISO 19123-2). Notably, the WCS suite allows both ingest and retrieval based on the same conceptual model, OGC coverages [2].

IMPLEMENTATION FEASIBILITY

In this Section we demonstrate feasibility by inspecting a service tool which, among others, offers the features recommended for efficient, client-friendly access. This is the European Datacube Engine and OGC datacube reference implementation, rasdaman (“raster data manager”), which has been developed over two decades into a cross-domain datacube engine [12][3][10]. A general survey of datacube tools has been published by RDA [13].

The rasdaman engine resembles a complete software stack, implemented from scratch to support fastest management and retrieval on massive multi-dimensional arrays in a domain agnostic way. Its array query language, rasql, meantime is adopted as the ISO SQL Multi-Dimensional Arrays (MDA) standard [11]. For EO datacubes rasdaman supports the declarative spatio-temporal datacube analytics language standard, OGC WCPS [5] (Requirements 4 & 5).

The overall system architecture centers around the multi-parallel rasdaman worker processes which operate on arbitrarily tiled arrays (Requirements 1 & 2, see Figure 1) stored in a database or read from some legacy archive (hence avoiding copies). When ingesting data they can be stored in a number of formats, including the CPU’s main memory array format (Requirement 2), through a WCS-T based ETL layer (Requirement 3) which homogenizes data and metadata, provides defaults, as well as the target tiling strategy [10]. Further tuning parameters include compression, indexing, cache sizing, etc. The resulting OGC compliant coverages represent analysis-ready space-time EO objects.

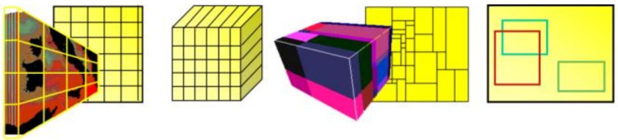
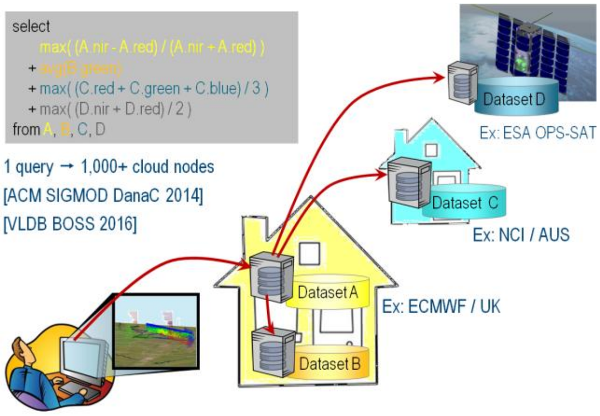


Figure 1
SAMPLE RASDAMAN DATACUBE PARTITIONING STRATEGIES (SOURCE: RASDAMAN).

In a rasdaman federation (Requirement 6), worker processes can fork subqueries to other cloud nodes or other data centers for load sharing and data transport minimization [4] (Figure 2). Figure 3 shows a visualization of actual federated query processing between the European Centre for Medium-Range Weather Forecast (ECMWF) in the UK and

National Computational Infrastructure (NCI) in Australia - both running rasdaman - for determining heavy rainfall risk areas from precipitation data at ECMWF and Landsat8 imagery at NCI [3]. The two query paths all lead to the same result for the user, thereby achieving location transparency.



```
select
    max( (A.nir - A.red) / (A.nir + A.red) )
  + avg(B.green)
  + max( (C.red + C.green + C.blue) / 3 )
  + max( (D.nir + D.red) / 2 )
from A, B, C, D
```

Figure 2
RASDAMAN TRANSPARENT DISTRIBUTED QUERY PROCESSING (SOURCE: RASDAMAN).

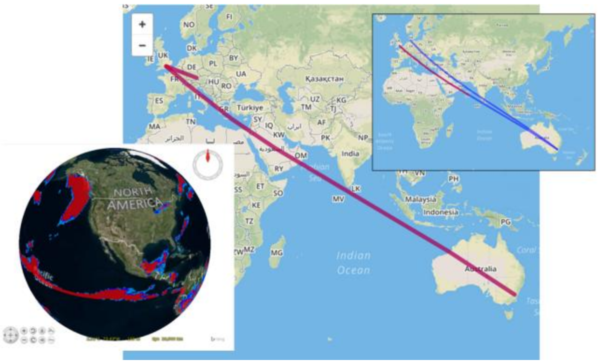


Figure 3
SAMPLE RASDAMAN INTERCONTINENTAL FEDERATION QUERY [13]

Being official OGC WCS Reference Implementation, rasdaman at the same time, to the best of our knowledge, is the most comprehensive WCS suite implementation and the only tool supporting all WCS extensions (Requirement 7).

CONCLUSION

In this contribution we motivate a less data-centric and more service-centric view, acknowledging that both are just two sides of the same coin. In a nutshell, data are ready for analysis when common math can be applied on the data without tweaking it for sensor or archive characteristics. This is no rocket science as we have shown; rather, the resulting requirements are available in implementation, thus underlining technical feasibility.

We do not claim that our list of EO service requirements is final, rather it is likely that more service quality facets will come up in future. However, from our experience with multi-Petabyte datacube service federations we feel that the requirements listed are all essential. As such, it is the hope that this contribution stimulates further discussion, shedding more light on service aspects than has been done up to now.

ACKNOWLEDGEMENT

This work has been supported partially by EU H2020 LandSupport, EU H2020 EOSC-hub, German BMWi BigDataCube, and German BMEL BigPicture.

REFERENCES

[1] 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.

[2] P. Baumann, E. Hirschorn, J. Maso: “Coverage Implementation Schema, version 1.1”. OGC document 09-146r6, www.opengeospatial.org/standards/wcs.

[3] 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

[4] P. Baumann, V. Merticariu: “On the Efficient Evaluation of Array Joins”. Proc. IEEE Big Data Workshop Big Data in the Geo Sciences, Santa Clara, US, October 29, 2015

[5] P. Baumann: “The OGC Web Coverage Processing Service (WCPS) Standard”. Geoinformatica, 14(4)2010, pp 447-479.

[6] P. Baumann: “Web Coverage Service (WCS) Interface Standard – Core, version 2.0”. OGC document 09-110r4, www.opengeospatial.org/standards/wcs.

[7] P. Baumann: “OGC Coverages Domain Working Group Public Wiki”. http://myogc.org/coveragesDWG.

[8] 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.

[9] CEOS: CEOS Analysis Ready Data for Land (CARD4L) Description Document. http://ceos.org/ard

[10] P. Furtado, P. Baumann: “Storage of Multidimensional Arrays Based on Arbitrary Tiling”. Proc. ICDE'99, March 23-26, 1999, Sydney, Australia.

[11] D. Misev, P. Baumann: “Enhancing Science Support in SQL”. Proc. IEEE Big Data Workshop Data and Computational Science Technologies for Earth Science Research, Santa Clara, US, October 29, 2015

[12] hidden rasdaman team: “rasdaman: Datacubes on Steroids”. Proc. ACM SIGSPATIAL, Seattle, USA, November 06, 2018

[13] RDA: “Array Database Assessment Working Group Report”. https://www.rd-alliance.org/groups/array-database-working-group.html.

[14] USGS: “U.S. Landsat Analysis Ready Data (ARD)“. https://landsat.usgs.gov/ard