Are Databases of Any Use in Modern Geo Services?

Peter Baumann
Jacobs University
Campus Ring 1
D_28759 Bremen
+49-421-200-3178
p.baumann@jacobs-university.de

ABSTRACT

Database systems is the technology established for giving users flexible, fast access to large, complex data holdings. In geo services, however, uptake of database technology is mixed: while it is considered useful for metadata, geo service builders resort to special-built solutions – not the least because classic databases do not support the data types encountered in geo sciences.

However, driven by "Big Data" challenges, research in the various domains has made substantial progress: geo sciences better understand their complex data, and databases get enriched with new data types. Hence, it is time to revisit the status quo.

In this paper we inspect relevant database technology to assess its usefulness for geo services. Starting point are OGC standards, as they represent consolidated, yet practice-proven requirements on service implementations. We claim that there important new building blocks are available, and point out current limitations. This way, the paper attempts to bridge gaps between normally disparate research fields, hopefully fostering cross-fertilization.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Scientific Databases; Spatial databases and GIS. K.1 [**The Computer Industry**]: Standards.

General Terms

Design, Standardization, Languages.

Keywords

Geo services, Big Data, databases, NewSQL, standards.

1. INTRODUCTION

Database systems is the technology established for giving users flexible, fast access to large, complex data holdings. When it comes to geo services, however, uptake of database technology is mixed: while it is considered useful for metadata, geo service builders resort to special-built solutions – not the least because classic databases do not support the data types encountered in geo sciences.

However, driven by "Big Data" challenges, research in the various domains has made substantial progress: geo sciences better understand their complex data, and databases get enriched with new data types. Hence, it is time to revisit the status quo.

In this paper we inspect relevant database technology to assess its usefulness for geo services. As a basis for our investigation we resort to the relevant OGC standards, as they represent consolidated, yet practice-proven requirements on service implementations. In terms of Big Data, we find in particular three main contributors today: grids (both regular and irregular), point clouds, and general meshes (albeit the latter to a lesser extent). These information categories are subsumed by the *coverage* abstraction which, generally speaking, is a digital representation of a space/time varying phenomenon [2][23][25]. The corresponding service definition is given by the OGC Web Coverage Service (WCS) [4][1] and its associated spatio-temporal raster query language, Web Coverage Processing Service (WCPS) [3][6].

Our approach, then, is to scan database research to find technology matching the data structures – and corresponding services – pertaining to coverages. We find that there important new building blocks are available, and point out current limitations. This way, the paper attempts to bridge gaps between normally disparate research fields, hopefully fostering cross-fertilization.

The remainder of this paper is organized as follows. Section 2 introduces the OGC coverage data and service model. Based on this, data-intensive service requirements on technical level are spotted in Section 3. In Section 4, this is matched against the state of the art in databases. Finally, conclusions are drawn in Section 5.

2. COVERAGES

2.1 Data Model

The abstract definition of a coverage is provided in ISO 19123 [23], which is identical to OGC Abstract Topic 6 [25]. On principle, a coverage is a function that maps so-called direct positions to corresponding values. In the case of a grid, the direct positions are determined by the grid points; in point clouds (MultiPointCoverages), direct positions are arbitrary. Continuous coverages can be modeled by adding specific interpolation methods for obtaining values also between direct positions.

ISO 19123 is abstract and not interoperable – many diverging implementations are possible based on ISO 19123. The OGC "GML 3.2.1 Application Schema – Coverages" standard [2], nicknamed GMLCOV, refines this to an interoperable level; in fact, conformance tests allow validating down to single pixel level. Figure 1 illustrates the GMLCOV coverage types by way of examples.

Specifically for gridded data, several cases are distinguished; Figure 2 shows the most important ones. GML 3.3 [28], a compatible extension to GML 3.2.1 **Error! Reference source not found.**, refines GMLCOV accordingly.

Conference'10, Month 1–2, 2010, City, State, Country.

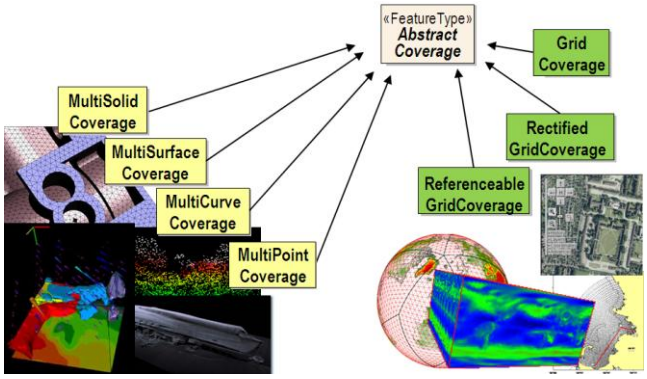

Figure 1: Coverage types as per GMLCOV

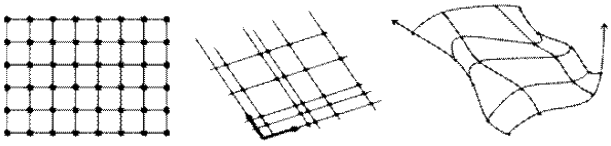

Figure 2: Important grid types in GMLCOV: regular grid (left), irregular grid (center), warped grid (right).

2.2 Service Model

While coverages can be served by any service suitable to deliver features (such as WMS, WFS, and SOS), the Web Coverage Service (WCS) offers dedicated functionality for the server-side filtering and processing of coverages. Since WCS 2.0 [4][1], the coverage data and service model are decoupled in OGC. This allows any service based on the generic OGC Feature data type to deal with coverages as well, as coverages are just a subtype of features. That said, the WCS suite provides the most comprehensive functionality support while other service types have various limitations.

WMS, for example, can only deliver 2-D coverages with three bands, interpreted as red/green/blue. WFS, on the other hand, can deliver any type of coverage, however only in completeness, without any subsetting functionality.

WCS is split into a mandatory WCS Core, which every implementation must support, and a set of WCS Extensions which are optional. This allows implementations to give an individual level of support, ranging from simple data access to complex analytics. Application profiles bundle the core with select extensions for particular purposes, usually also adding bespoke functionality. Currently, an Earth Observation (remote sensing) profile has been adopted, and a MetOcean (meteorology & oceanography profile is in the make. Figure 3 shows the overall organization of the WCS suite, with the data specifications on the left and the service components on the right hand side.

According to this modular approach, WCS offers functionality facets ranging from simple download of coverages or subsets thereof up to complex analytics. The latter is provided by the Web Coverage Processing Service (WCPS) standard described in detail in [3] and [6]; in a nutshell, this is a multi-dimensional raster query language aware of spatial and temporal coordinates. For example, the following query inspects MODIS scenes M1, M2, and M3 in turn and delivers those ones where a threshold is exceeded in the near-infrared (nir) band; from those coverages passing the filter, the difference between red and nir bands is computed and returned as GeoTIFF:

```
for $c in ( M1, M2, M3 )
where max_cells( $c.nir > 127 )
return encode( abs($c.red-$c.nir), "image/tiff" )
```

Big Data analytics engines like rasdaman can evaluate such requests in realtime using a combination of optimization, parallelization, and new hardware.

3. DATABASE SUPPORT

3.1 Vector Data

Vector data, including appropriate spatial coordinate handling, is present in databases since quite a while. Often they support the Simple Feature standard [9].

PostGIS [13] is an object-relational extension on top of the open-source PostgreSQL relational DBMS which is widely in use for geographic data, mainly for its comprehensive support for geometric data bearing geographic coordinates. Commercial vendors, such as Oracle, likewise support vectorial data handling.

Overall, this technology can be considered matured and adopted by the relevant communities.

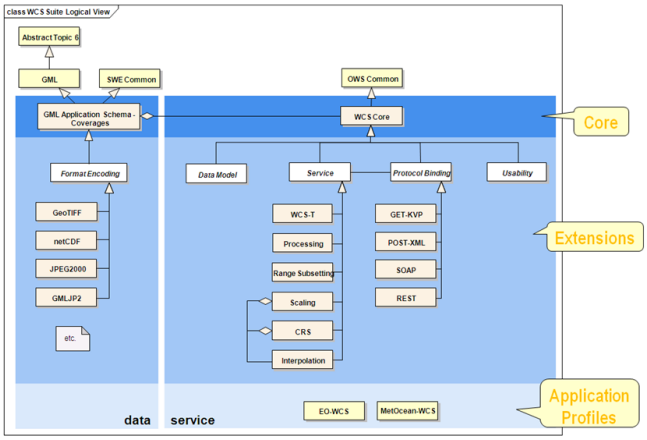

Figure 3: WCS standards suite overview

3.2 Raster Data

Regular and irregular grids appear manifold in the Earth sciences as spatio-temporal sensor, image, simulation, and statistics data. The common underlying data structure, massive multi-dimensional arrays, appear in manifold have long been neglected by the database community. An early system – and still the only completely implemented and fully operational one – is rasdaman [16][5]. Having coined the field of *array databases*, extends the standard relational model with a new attribute type, n-D arrays of unlimited size. As opposed to general-purpose methods like MapReduce [ref], array databases understand the specific semantics of gridded data and can exploit them for crisp, compact query formulation and highly effective automatic optimization and parallelization.

A geo semantics layer on top of the bare array engine supports all grid types defined in GMLCOV through OGC WMS, WCS, WCPS, and WPS interfaces. Rasdaman is OGC WCS Core Reference Implementation and WCPS Reference Implementation.

Internally, arrays are partitioned into sub-arrays, called *tiles*, which serve as unit of access. By processing arrays in a tile-by-tile fashion ("tile streaming"), the architecture is scalable to large objects. The query language, rasql, extends standard SQL natively

with n-D array operations; essentially, this adds linear algebra operations useful for signal, array, and statistics processing. The architecture is built to scale out from laptops to clouds, clusters, and federations of data centers.

PostGIS Raster is an extension to PostGIS which supports 2-D raster imagery through so-called map algebra functions; unlike in rasdaman, these are implemented as user-defined data types and, hence, not as tightly integrated and optimizable. PostGIS Raster generally is considered suitable for small and medium size rasters [12]. On commercial side, Oracle GeoRaster is a counterpart with similar properties: it likewise supports only 2-D rasters, and performs limited query optimization.

Recently, this domain has found increased interest in the database research community. Consequently, several projects have started, and lab prototypes are available, among others, for SciQL [17], SciDB [14], and Extascid [7].

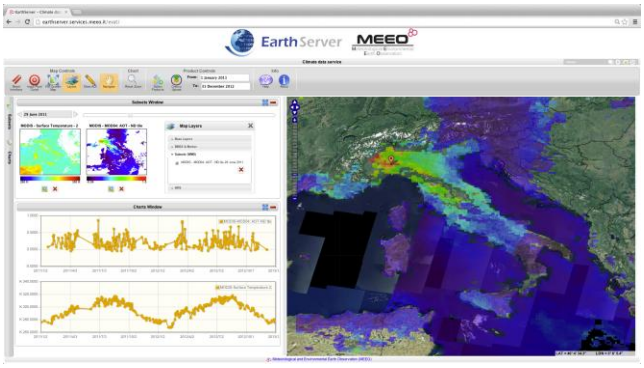

Figure 4: Atmospheric time series service using rasdaman [11].

3.3 Point Data

Point clouds are characterized by large numbers of points (going into the billions, and growing). In scaling out on point clouds relational databases possibly have a say again. For example, MonetDB shows promising handling of point clouds in its column-store architecture [21].

The rasdaman system utilizes PostgreSQL as a backend to support point clouds through its WCS interface, thereby unifying grid and point cloud access.

3.4 Mesh Data

Although mesh data are widely used in geometric modeling since decades (for example, in Computer-Aided Design, CAD) and although *geometric* data types have been addressed in depth by databases, *geographic* meshes have not enjoyed similar attention.

The relational model is not suited for such data, as it centers around independent tuples, thereby ignoring all the connectivity information. As a result, many relevant queries cannot be expressed at all (such as the intersection of two mesh objects, which typically will generate new nodes on the fly).

Notably, one of the first database models ever, CODASYL [24], allowed to represent general graphs. However, it did not have a query language, not to speak of the complex constraints a geometric object must fulfill. Decades later, two recent directions in database research have taken up on graphs.

RDF deals with arbitrary directed, labeled graphs based on triples representing an edge with its two endpoints. The corresponding query language, SPARQL, is crafted to integrate well with SQL, so there is a seamless coexistence of sets and graphs. Typically, RDF is used to model ontologies, i.e., descriptions object sets. No applications are known that stretch into the dimensions of geographic items where single object graphs may well contain millions of nodes and edges.

Graph databases, like neo4j [15], have set out to provide scalable graph retrieval. The corresponding query languages allow graph traversal and subsetting, so are suited for dealing with the topology of geographic objects. Support for geometric and geographic operations, though, has not been addressed yet. While separate from relational databases as of today the models seem compatible enough to get integrated in future.

Another option might be to represent graphs by their corresponding incidence matrices and utilize array databases for storage and retrieval [26]. It remains to be investigated whether this can be efficient and scalable.

Specifically for geo data, important contributions have been provided by Howe and Maier [10]. They introduce topology operations relevant for geo services in a manner suitable for query languages. The missing geometry support is being added by Mahdiraji [20] whose preliminary results look promising.

In summary, mesh support by databases is in its infancy, but there are promising approaches worth being followed further.

3.5 Integration

Looking back to the different coverage types and their degree of support by databases we find that several rather distinct database types come into play. This observation has been paraphrased by Michael Stonebraker as “no one size fits all”, heralding the NoSQL and NewSQL era in database research where conceptual models beyond tables are considered seriously.

This however, raises another issue: in practice it is common that different information categories appear in one and the same application. For example, the different coverage types of rasters, point clouds, and meshes often occur in combination. The challenge is to establish a common, coherent data management framework.

Conceptually, this is under way: point clouds fit with the standard relational model, SPARQL and SQL go together well, and an ISO initiative is under way [22] to extend the simple, incomplete 1-D array support into fully-fledge n-D arrays with powerful image / signal processing and statistics operations.

In terms of implementation this raises the question of efficient query evaluation across the information categories, such as joins between point clouds and rasters. Earlier in database research the concept of mediators has been established.

Multibase [31] aims at unified access to heterogeneous, distributed databases based on the Functional Data Model and its language, DAPLEX [30]. Products like IBM DB2 DataJoiner [8] and Garlic [18] followed. AMOS II [29] utilizes an object-oriented extension of DAPLEX with a mediator/wrapper architecture.

Originally, mediators were established to integrate different flavors and syntax variants of just one model, the relational one. Later, this was extended to files and other data sources, however, all compatible to tabular structures. In face of the radically different information categories listed in this paper, the field seems to be waiting for its revival in another dimension of complexity, namely orchestration of query processing across heterogeneous models. This is a wide open field today.

4. CONCLUSIONS

Database support for (geo) scientific data types means an important step forward for next-generation geo services aiming beyond ortho imagery and vectors. In particular, multi-dimensional (regular and irregular) grids, point clouds, and meshes are needed.

In the age of Big Data it is not sufficient any longer that each research domain pursues its own ways of finding solutions, often reinventing the wheel or, conversely, inventing inadequate wheels. Specifically, the geoinformatics domain and core computer science domains like databases, Web services, programming languages, and supercomputing, share challenges seen from different angles. It is not too infrequent that similar ideas appear in different fields. For example, array databases offer declarative query languages on large n-D arrays which internally are partitioned for efficient access to subsets. SciHadoop is an approach independent from databases where an array-tuned query language is put on top of Hadoop. Data formats like TIFF and NetCDF also support the concept of array partitioning. It is worthwhile, therefore, to extend this small, focused survey into a larger one incorporating more domains and also implementation aspects. Fostering exchange, therefore, seems promising.

From the manifold advantages which databases can offer three are of prime importance: The *flexibility* to ask any question any time, provided by the query language; the *scalability* provided by optimization, new hardware, parallelization, etc., without requiring any specific handling nor skills on the user's side; and *information integration* which significantly improves both quality of service and administration.

We have attempted to provide a synoptic view on the Big Geo Data problem from two particular angles: the geo service side is represented by the coverage standards which provide a concentrated amalgamation of experience in the field. This we have contrasted with relevant facets in current database research. Focus was on conceptual modeling, tentatively leaving out questions of architecture, optimization, parallelization, etc.

5. ACKNOWLEDGMENTS

This work has been supported by the European Commission under FP7-INFRA grant 283610 EarthServer.

6. REFERENCES

[1] Baumann, P.: Beyond Rasters: Introducing the New OGC Web Coverage Service 2.0. SIGSPATIAL, 2010, pp. 320-329

[2] Baumann, P.: GML 3.2.1 - Application Schema for Coverages. OGC 09-146, 2010

[3] Baumann, P.: OGC Web Coverage Processing Service (WCPS) Language Interface Standard. OGC document 08-068r2, 2009

[4] Baumann, P.: OGC Web Coverage Service (WCS) Interface Standard – Core. OGC document 09-110r3, 2010

[5] Baumann, P.: On the Management of Multidimensional Discrete Data. VLDB Journal 4(3)1994, pp. 401-444

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

[7] Cheng, Y., Rusu, F.: EXTASCID: An Extensible System for the Analysis of Scientific Data. XLDB-2012 poster

[8] Gupta, P., Lin, E.: DataJoiner: a practical approach to multi-database access. Proc. 3rd Int. Conf. on Parallel and Distributed Information Systems, 1994, pp. 264

[9] Herring, J.: OpenGIS Implementation Standard for Geographic Information - Simple Feature Access - Part 2: SQL option. OGC 06-104r4, 2010

[10] Howe, B., Maier, D.: Algebraic Manipulation of Scientific Datasets. VLDB 2004, pp. 924-935

[11] http://earthserver.services.meeo.it/mea/, accessed 2014-04-15

[12] http://lists.osgeo.org/pipermail/postgis-users/2014-April/039024.html, accessed 2014-04-15

[13] http://postgis.net/, accessed 2014-04-15

[14] http://scidb.org, accessed 2014-04-15

[15] http://www.neo4j.org, accessed 2014-04-15

[16] http://www.rasdaman.org, accessed 2014-04-15

[17] http://www.sciql.org/, accessed 2014-04-15

[18] Josifovski, V., Schwarz, P., Haas, L., Lin, E.: Garlic: A New Flavor of Federated Query Processing for DB2. ACM SIGMOD, New York, USA, 2002, pp. 524–532

[19] Kottman, C.: OpenGIS® Abstract Specification 6 - Topic 6: The Coverage Type and its Subtypes. OGC 00-106, 2000

[20] Mahdiraji, A.R.: Database Support for Unstructured Meshes. PVLDB 6(12)2013, pp. 1404-1409

[21] Martinez-Rubi, O., Kersten, M. L., Goncalves, R., Ivanova, M.: A Column-Store Meets the Point Clouds. FOSS4G-Europe Academic Track, 2014 (submitted)

[22] Misev, D., Baumann, P.: Extending the SQL Array Concept to Support Scientific Analytics. SSDBM 2014 (submitted)

[23] N.n.: Geographic Information - Coverage Geometry and Functions. ISO 19123:2005

[24] N.n.: Information processing system – Database languages – NDL. ISO 8907:1987

[25] N.n.: Topic 6: Schema for coverage geometry and functions. OGC 07-011, 2006

[26] Paoluzzi, A.: Algebraic Mining of Solid Models from Images. Image Information Mining, Bucuresti, Romania, 2014

[27] Portele, C.: Geography Markup Language (GML) Encoding Standard. OGC 07-036, 2007

[28] Portele, C.: GML - Extended schemas and encoding rules. OGC 10-129r1, 2012

[29] Risch, T., Josifovski, V., Katchaounov, T.: Functional Data Integration in a Distributed Mediator System. In: The Functional Approach to Data Management: Modeling, Analyzing and Integrating Heterogeneous Data, Springer 2004, p. 483

[30] Shipman, D.: The Functional Data Model and the Data Language Daplex. ACM SIGMOD, NY, USA, 1979, pp 59–59

[31] Smith, J.M., et al: Multibase: Integrating Heterogeneous Distributed Database Systems. AFIPS 1981, vol. 50, pp. 487–499

[32] Tomasic, A., Raschid, L., Valduriez, P.: Scaling Access to Heterogeneous Data Sources with DISCO. IEEE TKDE, 10(5):808–823, September 1998