EarthServer

Scalable On-Demand Processing for the Earth Sciences

EuroGEOSS
Madrid, 2012-jan-27

Peter Baumann
Jacobs University | rasdaman GmbH
Motivation

- Serving *data* is not enough
- Upcoming trend:
  - *service quality* as differentiating criterion
    - Why do you shop at Amazon?
- transition from *data stewardship* to *service stewardship*
- Specifically with high-volume data, what can this mean?
Motivational Example: 
Sat Image Time Series Archive

[DFD-DLR, Diederich et al]
Emerging Research Field: Array Analytics

- Array := n-D sequence (cf programming languages) := raster
- **Array Analytics** := *Efficient analysis on multi-dimensional arrays of a size several orders of magnitude above evaluation engine’s main memory*

  - Typically in client/server setup
  - „Big Analytics“ on „Big Data“, both ad-hoc and long-tail
  - Cf 1st Workshop on Array Databases, Uppsala 2011
    www.rasdaman.com/ArrayDatabases_Workshop

- Research issues:
  - Concepts: modeling, access interfaces (particularly, query languages)
  - Architecture: storage, processing, optimization
  - Scalability, usability, applications, standards

- Initiatives: rasdaman, SciDB, SciQL, PostGIS Raster, ...
Standards Perspective:
OGC Web Coverage Service

- In Core: simple data access:
  \[ \text{subset} = \text{trim} \mid \text{slice} \]

- further functionality in extension packages
**OGC Web Coverage Processing Service (WCPS)**

“XQuery for Coverages”: navigation, extraction, aggregation, analysis

- **Time series**
- **Image processing**
- **Summary data**
  - current value is 322.00;
  - average over all values up to now currently is 745.17692307653205.
- **Sensor fusion & pattern mining**
WCPS By Example

- "From MODIS scenes M1, M2, and M3, the absolute of the difference between red and nir, in HDF-EOS"

```python
for $c$ in (M1, M2, M3)
  return encode(
    abs($c$.red - $c$.nir),
    "hdf"
  )
```

(hdf_A', hdf_B', hdf_C')
WCPS By Example

- "From MODIS scenes M1, M2, and M3, the absolute of the difference between red and nir, in HDF-EOS"
  - but only those where nir exceeds 127 somewhere

```python
for $c$ in ( M1, M2, M3 )
  where
    some( $c$.nir > 127 )
  return
    encode(
        abs( $c$.red - $c$.nir ),
        "hdf"
    )
```
WCPS By Example

- "From MODIS scenes M1, M2, and M3, the absolute of the difference between red and nir, in HDF-EOS"
  - ...but only those where nir exceeds 127 somewhere
  - ...inside region R

```python
for $c$ in ( M1, M2, M3 ),
    $r$ in ( R )
where
    some( $c.nir > 127 and $r )
return
encode(
    abs( $c.red - $c.nir ),
    "hdf"
)
```

(hdf_A)
What Is a Coverage, After All?

Coverage = n-D "space/time-varying phenomenon"

- [ISO 19123, OGC AT6]
What’s Missing?

- OGC services currently support nD rasters, but not all coverage types
  - Irregular grids, TINs, general meshes, point clouds, trajectories, ...
- Retrieval support for „new“ coverage types
- Lack of data / metadata service integration
- Scalability
  - Automated distributed processing of queries
  - Cloud
- Lack of clients
**EarthServer: Big Earth Data Analytics**

- **Mission:** to enable standards-based ad-hoc analytics on the Web for Earth science data
  - scalable to Petabyte/Exabyte volumes
  - directly manipulate, analyze & remix any-size geospatial data

- **Core idea:** integrated query language for all spatio-temporal coverage data
  - OGC standards based
  - Server + client

- **Started** Sep 1 2011, 3 years, 5.38m EUR budget
EarthServer Technical Approach

- OGC interface standards
  - WCS & WCPS + XQuery

- Server: extend rasdaman, a fully implemented n-D array DBMS
  - scalability: semantic-based automatic distribution using cascading services, parallel & distributed query evaluation, cloud virtualization, SMP

- Clients, clients, clients!
  - from mobile devices to Web GIS to high-end immersive VR

- iterative approach
  - early / basic / full / consolidated release
EarthServer Lighthouse Applications

- Each 100+ TB ultimately
- front-end to existing archives - no new archives
Conclusion

- **Vision:** barrier-free „mix & match“ access to multi-source, any-size geo data
  - “Complex Analytics on Big Data”

- **Goals:**
  - integrated query language; scalability & query distribution; GIS integration; clients; in-situ processing; standards

- **Platform:** rasdaman Array DBMS

- **Issues:**
  - Model extension & QL integration
  - Distributed processing, scalability (cloud, SMP, ...)
  - Flexible clients

- **Follow progress on** [www.earthserver.eu](http://www.earthserver.eu)
Large-Scale Data Services
Research @ Jacobs U

- Jacobs University:
  international, multi-cultural
  - 1,300+ students, 110 nations
  - English official language on campus

- Large-Scale Scientific Information Systems research group
  - www.jacobs-university.de/lisis
  - large-scale n-D raster services and beyond: theory, practice, application, standardization
  - Main outcomes:
    - rasdaman: n-D raster DBMS -> GmbH
    - several OGC standards

- Hiring PhDs, postdocs, programmers
enabling databases with large-scale raster data

- multidimensional SQL, Java, C++
- intelligent storage & query optimization
- interoperable: DB systems, OGC, data formats, GDAL, Map Server

**rasql = multidimensional expressions in SQL**

```
select img.green[x0:x1,y0:y1] > 130
from LandsatArchive as img
```

in operational use on dozen-Terabyte map objects

- Ortho image, thematic, elevation, climate, ...

**WCPS reference implementation**
Semantic Interoperability: Comparison

- WCPS: semantics in query

```xml
for $c$ in ( M1, M2, M3 )
return encode abs( $c$.red - $c$.nir ), "hdf"
```

- WPS: semantics in human-readable text

```xml
<ProcessDescription ...
<ProcessDescription processVersion="2" storeSupported="true" statusSupported="false">
  <ows:Identifier>Buffer</ows:Identifier>
  <ows:Title>Create a buffer around a polygon.</ows:Title>
  <ows:Abstract>Create a buffer around a single polygon. Accepts the polygon as GML and provides GML output for the buffered feature. </ows:Abstract>
  <ows:Metadata xlink:title="spatial" />
  <ows:Metadata xlink:title="geometry" />
  <ows:Metadata xlink:title="buffer" />
  <ows:Metadata xlink:title="GML" />
<DataInputs>
  <Input>
    <ows:Identifier>InputPolygon</ows:Identifier>
    <ows:Title>Polygon to be buffered</ows:Title>
    <ows:Abstract>URI to a set of GML that describes the polygon.</ows:Abstract>
    <ComplexData defaultFormat="text/XML" defaultEncoding="base64" defaultSchema="http://foo.bar/gml/3.1.0/polygon.xsd">
      <SupportedComplexData>
```

EuroGEOSS Madrid, 2012-Jan-27 EarthServer 21
### State of the Art

- **Oracle GeoRaster**
  - 2D; incomplete QL embedding;
  - no known optimization

- **PostGIS Raster**
  - Good QL integration
  - No storage layout tuning, no adaptive tile streaming,
    no raster query optimization
  - small tiles, ... - scalability???

- **SciDB (US)**
  - n-D arrays announced, only components demoed

- **MapReduce**
  - Too generic to be optimal

```sql
DECLARE
    g sdo_georaster;
    b blob;
BEGIN
    SELECT raster INTO g
    FROM uk_rasters
    WHERE id = 4;
    dbms_lob.createTemporary(b, TRUE);
    sdo_geor.getRasterSubset(
        georaster => g,
        pyramidlevel => 0,
        window => sdo_number_array(0, 0, 699, 899),
        bandnumbers => '0',
        rasterBlob => b)
END;
```

```sql
SELECT g.green[0:699, 0:899]
FROM uk_rasters AS g
WHERE oid(g) = 4
```