On the Efficient Evaluation of Array Joins

Big Data in the Geosciences Workshop
IEEE Big Data, Santa Clara, US, 2015-oct-29

Peter Baumann, Vlad Merticariu
Jacobs University | rasdaman GmbH
{p.baumann,v.merticariu}@jacobs-university.de
Data Homogenization With OGC Standards

sensor feeds

coverage server

sensor, image [timeseries], simulation, statistics data

simulation data
Web Coverage Service (WCS)

- **OGC Coverages** unifying regular & irregular grids, point clouds, meshes
  - OGC Coverage Implementation Schema

- **WCS Core**: access to spatio-temporal coverages & subsets
  - subset = trim | slice

- **WCS Extensions**: optional functionality facets
  - Scaling, CRS transformation, …

Large, growing implementation basis: rasdaman, GDAL, QGIS, OpenLayers, OPeNDAP, MapServer, GeoServer, NASA WorldWind, EOxServer; Pyxis, ERDAS, ArcGIS, …
OGC Web Coverage Processing Service (WCPS)

- = high-level spatio-temporal geo analytics language std

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

"From MODIS scenes M1, M2, M3: difference between red & nir, as TIFF"
- ...but only those where nir exceeds 127 somewhere, within LandMask
Agile Analytics on Earth & Planetary datacubes
- rasdaman + NASA WorldWind
- Rigorously standards: OGC WMS + WCS + WCPS
- 100s of TB online now, next: 1+ Petabyte per cube

Intercontinental initiative, 3+3 years:
EU + US + AUS
Agile Array Analytics: rasdaman

- "raster data manager": SQL + n-D arrays
- Scalable parallel “tile streaming” architecture
  - Joins!
- Supports R, QGIS, OpenLayers, MapServer, GDAL, EOxServer, Pyxis, ERDAS, ArcGIS, ...
- Blueprint for OGC WCPS, ISO Array SQL stds
Array Partitioning

- **Goal:** faster loading by adapting storage units to access patterns
- **Approach:** split n-D array into n-D partitions (“tiles“)
- **Tiling classification based on degree of alignment [ICDE 1999]**
  - all implemented in rasdaman

chunking [Sarawagi, Stonebraker, DeWitt, ...]
Why Irregular Partitioning?

[Centrella et al: scidacreviews.org]
Array Join: What Happens?

- "MODIS red band with cloud mask applied":
  \[ \text{$Modis.red \times CloudMask$} \]

- **Case 1**: same tile shapes, same position
  - Easy

- **Case 2**: same tile shapes, different position
  - Overlaps, not so easy

- **Case 3**: different tile shapes
  - Worse

- **Case 4**: different dimensions
  - Gimme a break!
Array Join: Problem Statement

- Goal: minimize tile reads when evaluating "A op B" in face of some arbitrary, independent partitioning of A and B

\[\text{Modis.red} \times \text{CloudMask}\]

\[
\begin{array}{ccc}
A1 & A3 & A5 \\
A2 & A4 & A6 \\
\end{array}
\]

\[
\begin{array}{ccc}
B1 & B4 \\
B2 & B3 \\
\end{array}
\]
Bipartite Traversal Graphs

```
<table>
<thead>
<tr>
<th>A1</th>
<th>A2</th>
</tr>
</thead>
<tbody>
<tr>
<td>A3</td>
<td>A4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>B1</th>
<th>B2</th>
</tr>
</thead>
<tbody>
<tr>
<td>B3</td>
<td>B2</td>
</tr>
</tbody>
</table>

```

```
<table>
<thead>
<tr>
<th>A1</th>
<th>B1</th>
</tr>
</thead>
<tbody>
<tr>
<td>A2</td>
<td>B2</td>
</tr>
</tbody>
</table>

```

```
<table>
<thead>
<tr>
<th>A3</th>
<th>B3</th>
</tr>
</thead>
<tbody>
<tr>
<td>A4</td>
<td>B4</td>
</tr>
</tbody>
</table>

```

```
<table>
<thead>
<tr>
<th>B3</th>
<th>A1</th>
</tr>
</thead>
<tbody>
<tr>
<td>B2</td>
<td>B1</td>
</tr>
</tbody>
</table>

```

```
<table>
<thead>
<tr>
<th>A2</th>
<th>B2</th>
</tr>
</thead>
<tbody>
<tr>
<td>A4</td>
<td>B5</td>
</tr>
</tbody>
</table>

```

```
<table>
<thead>
<tr>
<th>B4</th>
<th>A3</th>
</tr>
</thead>
<tbody>
<tr>
<td>B5</td>
<td></td>
</tr>
</tbody>
</table>

```
Finding Complete Paths

- Leonhard Euler: „complete, minimal path for even-degree nodes“

  ![Pregel river, Königsberg](image)

- Carl Hierholzer: „not even degree? Add aux edges!“

  ![Diagram](image)
Finding Shortest Paths

- Assumption: can hold 1 red, 1 blue tile
- How to find shortest path? See paper!

Start / end!

- \langle B4, A3, B2, A1, B1, B3, A3, B5, A4, B2, A2 \rangle
- \langle B4, A3, B3, A1, B1, A2, B2, A4, B5, A3, B2, A1 \rangle
Complexity?

- **Best case: full alignment**
  \[ |E_A| + |E_B| \]

- **Worst case: All tiles of A x all tiles of B**
  \[ |E_A| \times |E_B| \]
What else?

- We have: *tile traversal with minimum duplicate tile reads*

- Variation 1: *buffer size to avoid dup reads?*
  - query cost estimation

- Variation 2: *for buffer size given, how much improvement?*

- Variation 3: *parallelize disconnected subgraph*

- Variation 4: *how many tiles to ship between nodes for optimal parallelization?*
Wrap-Up

- Array Database queries offer new Big Data service level, including **datacube fusion**
  - Standardization: ISO Array SQL, OGC WCPS

- Need efficient algorithms
  - graph-based **Array Join** for arbitrary partitioning

- EarthServer: Petascale **Datacube Analytics**
  - rasdaman

[rasdaman screenshots]
Datacube Research @ Jacobs U

- Large-Scale Scientific Information Systems research group
  - Flexible, scalable n-D array services
  - www.jacobs-university.de/lsis

- Main results:
  - pioneer Array DBMS, rasdaman
  - standardization:
    - OGC Big Geo Data (also ISO, INSPIRE, W3C)
    - ISO Array SQL

Hiring PhD students, PostDocs