COMMIT/

Spatiotemporal data warehouses for travelers

Martin L. Kersten
Milena Ivanova
Jennie Zhang

http://www.commit-nl.nl/
A PUBLIC-PRIVATE RESEARCH COMMUNITY

NEWS

12 JAN  
Internal kick-off COMMIT for all co-workers

17 NOV  
Lancering COMMIT groot succes

14 NOV  
Persbericht on uitnodiging COMMIT d.d. 7 november 2011

AGENDA

There are no COMMIT events planned at the moment.

PROJECTS

SOCIALY-ENRICHED ACCESS TO LINKED CULTURAL MEDIA

A collaboration between Delft University of Technology and Auslum.

SWELL: SMART REASONING SYSTEMS FOR WELL-BEING AT WORK AND AT HOME

A collaboration between Radboud University Nijmegen and TNO.

SENSOR NETWORKS FOR PUBLIC SAFETY

A collaboration between University of Twente and Ambient Systems.

VERY LARGE WIRELESS SENSOR NETWORKS FOR WELL-BEING

A collaboration between VU University Amsterdam and DevLab.
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Metis: Dependable Cooperative Systems for Public Safety</td>
<td>Trusted Healthcare Services</td>
<td>Spatiotemporal Data Warehouses for Travellers</td>
<td>Interaction for Universal Access</td>
<td>E-Infrastructure Virtualization for E-Science Applications</td>
</tr>
<tr>
<td>From Data to Semantics for Scientific Data Analysis</td>
<td>E-Biobanking with Imaging for Healthcare</td>
<td>E-FoodLab</td>
<td>Information Retrieval for Information Services</td>
<td>Virtual Worlds for Well-Being</td>
</tr>
</tbody>
</table>

*Note: The images represent various research areas and projects in the field of information technology and data science.*
Spatiotemporal data warehouses for travellers

Objectives
Develop a scientifically sound technological basis for harvesting knowledge in real-time from massive spatiotemporal event databases gathered from people, sensors and scientific observatories.

Deliverables
Development, deployment and dissemination of an open-source platform for efficient and scalable data management for trajectory and geo-tagged information.

Work packages
1. Trajectory mining
2. Time-trail warehouse architecture
3. Trail mining
4. Trajectory stream processing
5. Community tagging
6. User-centric geo-profiling
7. Multi-scale query processing
8. Databarehouse virtualization

Non-profit organizations:
Koninklijk Nederlands Meteorologisch Instituut (KNMI)

Partners involved:
Arcadis, EuroCottage, Byyes, MonetDB, TomTom

Project leader:
Martin Kersten

Research Institute(s):
Centrum Wiskunde & Informatica (CWI), University of Twente, Utrecht University

Expected deliverables
40 conferences
4 products
5 software
2 user studies
1 other result

Downloads
PDF P19 Spatiotemporal
Work packages

1. **Trajectory mining**
2. **Time-trail warehouse architecture**
3. **Trail mining**
4. **Trajectory stream processing**
5. **Community tagging**
6. **User-centric geo-profiling**
7. **Multi-scale query processing**
8. **Datawarehouse virtualization**
Objectives: Design and develop a reference architecture for database-centric analysis of large spatiotemporal data streams comprised of trajectory events.

Background: Efficient content-based analysis of time trails, i.e., streams of (Who, What, When, Where) events, poses new challenges for query optimization. It is the key component to achieve the required scalability into the hundreds of terabytes of events that should be mined and inspected at near real-time requirements.

Partners: KNMI and CWI
Objective: The goal of this project is to extend the Krimp data mining algorithms to mine the patterns in large Time-Trail Vault data warehouses efficiently, this is also known as trajectory mining.

The key subjective criteria are geographic cohesion and semantic (as denoted by the attributes) meaningfulness, e.g., patterns that describe geo-temporal unrelated events are deemed unimportant. Mining constraints are used to enforce the subjective application criteria. The objective criteria, based on Minimum Description Length principle (MDL), aim to ensure that the data analyst is confronted with statistically significant results, only.

Partners: KNMI and Utrecht University
Components of the Time-Trail warehouse

• MonetDB, a column store pioneer
• SQL and SciQL query languages
• SciLens computational platform
• Data vaults to integrate distributed repositories
Open-Source Development

- >10,000 downloads/month
- Used in 132 countries
- Feature releases: 3-4 per year
- Bug-fix releases: monthly
- QA
  - Automated nightly testing on >20 platforms
  - Ensure correctness & stability
  - Ensure portability
  - Bug reports become test cases
  - Semi-automatic performance monitoring
  - Passed static code verification by Coverity with only minor problems
The Software Stack

- **Front-ends**
  - SQL 03
  - Optimizers

- **Back-end(s)**
  - MonetDB 5

- **Kernel**
  - MonetDB kernel

- **Strategic optimization**
  - MAL

- **Tactical optimization: MAL -> MAL rewrites**
  - MAL

- **Runtime operational optimization**
  - MAL
function user.s2_1():void;
barrier _73 := language.dataflow();
_2:bat[oid:int] := sql.bind("sys","t","c",0);
_7:bat[oid:int] := sql.bind("sys","s","x",0);
_10 := bat.reverse(_7);
_11 := algebra.join(_2,_10);
_13 := algebra.markT(_11,0@0);
_14 := bat.reverse(_13);
_15:bat[oid:int] := sql.bind("sys","t","a",0);
_17 := algebra.leftjoin(_14,_15);
_18 := bat.reverse(_11);
_19 := algebra.markT(_18,0@0);
_20 := bat.reverse(_19);
_21:bat[oid:int] := sql.bind("sys","s","z",0);
_23 := algebra.leftjoin(_20,_21);
exit _73;
_24 := sql.resultSet(2,1,_17);
sql.rsColumn(_24,"sys.t","a","int",32,0,_17);
sql.rsColumn(_24,"sys.s","z","int",32,0,_23);
_33 := io.stdout();
sql.exportResult(_33,_24);
end s2_1;

EXPLAIN SELECT a, z FROM t, s WHERE t.c = s.x;
The Software Stack

Front-ends
- SQL 03
- SciQL
- JSON/JAQL

Back-end(s)
- MonetDB 5

Kernel
- MonetDB kernel
MonetDB vs Traditional DBMS Architecture

- **Architecture-Conscious Query Processing**
  - vs Magnetic disk I/O conscious processing
    - Data layout, algorithms, cost models

- **RISC Relational Algebra (operator-at-a-time)**
  - vs Tuple-at-a-time Iterator Model
    - Faster through simplicity: no tuple expression interpreter

- **Multi-Model: SQL, XML/XQuery, ..., JSON/JAQL, RDF/SPARQL**
  - vs Relational with Bolt-on Subsystems
    - Columns as the building block for complex data structures

- **Decoupling of Transactions from Execution/Buffering**
  - vs ARIES integrated into Execution/Buffering/Indexing
    - ACID, but not ARIES. Pay as you need transaction overhead.

- **Run-Time Indexing and Query Optimization**
  - vs Static DBA/Workload-driven Optimization & Indexing
    - Extensible Optimizer Framework;
    - cracking, recycling, sampling-based runtime optimization
MonetDB SciQL

- SciQL (*pronounced ‘cycle’*)
- A backward compatible extension of SQL’03
- Symbiosis of relational and array paradigm
- Flexible structure-based grouping
- Capitalizes the MonetDB array storage
  - Recycling, an adaptive ‘materialized view’
  - Zero-cost attachment contract for cooperative clients
SciQL examples

- CREATE ARRAY matrix (  
  x integer DIMENSION[0:4],  
  y integer DIMENSION[0:4],  
  w float DEFAULT 0 );

- CREATE ARRAY vmatrix (  
  x integer DIMENSION[-1:5],  
  y integer DIMENSION[-1:5],  
  w float DEFAULT 0 );

- -- embedding a matrix[4][4] into a zero enlarged one
- INSERT INTO vmatrix SELECT i, j, val FROM matrix;
- INSERT INTO vmatrix SELECT j, i, val FROM matrix;

Figure 1: Alternative Array Storage Schemes
--- structural aggregation

SELECT [x], [y], avg(v) FROM matrix
GROUP BY matrix[x:x+2][y:y+2];

SELECT [x], [y], avg(v) FROM matrix
GROUP BY DISTINCT matrix[x:x+2][y:y+2];

SELECT [x], [y], avg(v) FROM vmatrix
GROUP BY matrix[x-1:x+1][y-1:y+1];

SELECT x, y, ... FROM vmatrix
GROUP BY matrix[x][y],
    matrix[x-1][y], matrix[x+1][y],
    matrix[x][y-1], matrix[x][y+1];
SciQL is developed with the requirements of Astronomy, Seismology, Remote Sensing, ...

SciQL implementation is used for forest fire detection based on remote sensing images.
   – TELEIOS project

Jennie Zhang is key architect of its implementation
A dream machine

1 Diamond
64 core CPU, 1TB

16 Rocks dual 8-core, 64GB RAM, 8TB SSD

64 Bricks dual quad core, 16GB Ram, 2TB SSD

256 Pebbles Intel Atoms, 4GB, 0.5 TB SSD

GB Ethernet fabric
<table>
<thead>
<tr>
<th>Configuration</th>
<th>Quantity</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>144 boxes AMD bobcat</td>
<td>144</td>
<td>8 GB RAM</td>
</tr>
<tr>
<td>1 GB ethernet</td>
<td></td>
<td>2 TB HDD</td>
</tr>
<tr>
<td>8 TB HDD RAID</td>
<td></td>
<td>8 TB HDD RAID</td>
</tr>
<tr>
<td>144 boxes I7 2600K 3.4Ghz</td>
<td>144</td>
<td>16 GB RAM</td>
</tr>
<tr>
<td>1 GB ethernet</td>
<td></td>
<td>40 Gb INFINIBAND</td>
</tr>
<tr>
<td>2 TB HDD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40 Gb INFINIBAND</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8 boxes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&gt; 32 cores</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&gt; 128 GB RAM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 GB ethernet</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40 Gb INFINIBAND</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 PB SAN</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Scientific Repositories

- High volumes and thousands/millions files
- Domain-specific standard formats (SEED)
- Locating data of interest hard
- Limitations: flexibility, scalability, performance
DBMS Solutions

- Declarative query languages, scalability, performance, concurrent access
- Up-front data load
- Impedance mismatch
- Limited access to science libraries

Database

Metadata

Raw data

Application

SQL

APIs

DBMS

Science
Libs

mSeed
Repository
MonetDB Data Vault

- Symbiosis between file repositories and databases
- Transparent just-in-time access to external data of interest
MonetDB Data Vault

Strawman’s integrated heterogeneous distributed solution

New Application

SQL
APIs

MonetDB Data Vault

FTP
Curl

Database

Old Application

GFZ SEED Repository

ORFEUS SEED Repository

IRIS SEED Repository

Old Application

Strawman’s integrated heterogeneous distributed solution
Data Vault Added Value

- Extended functionality
- Low start-up cost
- Performance and scalability
- Open-source solution
Challenges

• Find open-source libraries
• Wrap functionality of external libraries
• Symbiotic query processing
• Cache management for data vaults
• Access control
COMMIT for Seismology

- Develop a reference infrastructure
- Provide a sound implementation of SciQL
- Realize the potentials of the data vault
- Gain insight in mass-scale experimentation