Jackpine: A Benchmark To Evaluate Spatial Database Performance

Suprio Ray, Bogdan Simion, Angela Demke Brown
Computer Science Department
University of Toronto
Toronto, Canada

ICDE 2011, Hannover, Germany
Motivation

• Explosion of spatial data
• Web-mapping and Location-based services are very popular

• How do we evaluate spatial database performance?
TPC-Spatial?

- Where is TPC-Spatial?

- No industry standard spatial database benchmark

- Related work
  - Sequoia 2000 (SIGMOD ‘93)

- Jackpine - a database benchmark for spatial workloads
Goals for the spatial benchmark

• Comprehensive coverage of spatial features

• Real-world workloads

• Extensible

• Portable
Overview

- Motivation
- Background on spatial database
- Micro benchmark
- Macro benchmark
- Challenges
- Using the benchmark
- Conclusion
Spatial query execution

• Spatial query
  – allow for the use of geometry data types
    • points, lines and polygons
  – topological relations
    • Intersects, touches, overlaps

• Two step query evaluation process
  – Filter
  – Refine
Topological relations

- **Dimensionally Extended Nine-intersection model**
  - Considers the max dimension of the intersections of two objects
  - Proposes 8 relations:

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Disjoint" /></td>
<td><img src="image" alt="Equals" /></td>
<td><img src="image" alt="Contains" /></td>
<td><img src="image" alt="Within" /></td>
</tr>
<tr>
<td>Disjoint</td>
<td>Equals</td>
<td>Contains</td>
<td>Within</td>
</tr>
<tr>
<td><img src="image" alt="Touches" /></td>
<td><img src="image" alt="Intersects" /></td>
<td><img src="image" alt="Crosses" /></td>
<td><img src="image" alt="Overlaps" /></td>
</tr>
<tr>
<td>Touches</td>
<td>Intersects</td>
<td>Crosses</td>
<td>Overlaps</td>
</tr>
</tbody>
</table>
Filter and Refine

- Spatial query: which US states were affected by Katrina
Filter and Refine

- Spatial query: which US states were affected by Katrina
Filter and Refine

- Spatial query: which US states were affected by Katrina

polygons

Filter

Candidate objects
Filter and Refine

- Spatial query: which US states were affected by Katrina
Overview

- Motivation
- Background on spatial database
- Micro benchmark
- Macro benchmark
- Challenges
- Using the benchmark
- Conclusion
Jackpine

- **Micro benchmark**
  
  **Goal:** Comprehensive coverage of spatial features
  
  - Spatial join queries involving topological relations *(based on Dimensionally Extended 9 Intersection model)*
  
  - Queries with spatial analysis and aggregation functions
  
  - Inserting records with geometric objects

- **Macro benchmark**
  
  **Goal:** Model real-world spatial applications
  
  - Six spatial applications
Jackpine - Micro benchmark

- **Spatial join queries**
  - Each row and each column is included at least once
  - Total number of queries 15

<table>
<thead>
<tr>
<th></th>
<th>Polygon and Polygon</th>
<th>Line and Line</th>
<th>Line and Polygon</th>
<th>Point and Polygon</th>
<th>Point and Line</th>
<th>Point and Point</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>✓</td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Disjoint</strong></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Intersects</strong></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td><strong>Touches</strong></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td><strong>Crosses</strong></td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td><strong>Overlaps</strong></td>
<td>✓</td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td><strong>Within</strong></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td><strong>Contains</strong></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
</tr>
</tbody>
</table>
Micro benchmark

- **Spatial analysis - analytic functions**
  - distance, dimension, envelope, buffer, convex hull

- **Spatial analysis - aggregation operations**
  - longest line, largest area, total length and area
Overview

• Motivation
• Background on spatial database
• Micro benchmark
• Macro benchmark
• Challenges
• Using the benchmark
• Conclusion
Jackpine - Macro benchmark

- Attempts to model real-world applications
- Scenarios: consist of a series of queries executed in sequence
  - Geocoding
  - Reverse Geocoding
  - Map search and browsing
  - Land Information Management
  - Flood risk analysis
  - Toxic spill
Macro benchmark - Geocoding

- Determine latitude, longitude from street address
- Common use case is locating addresses of people and organizations in the map
- Scenario queries
  - Finds the matching street segment given street address or zipcode
  - Latitude, longitude of the location is obtained from the street segment
Macro benchmark – Reverse Geocoding

- Obtain textual street address from latitude, longitude
- Common use case is producing trip activity report in location-based services

Scenario queries
- Find the closest city name
- Find the closest street name

Source: tracknet
Macro benchmark – Map search & browsing

• Simulates searches for points of interest in the map
• Simulates the display of the bounding box around POI in the map
• Two visit cases: Student visit to a campus & Tourist visit to a place

• Scenario queries
  – search by matching keyword
  – series of queries to fetch objects inside the bounding box
Macro benchmark – Land Info Management

- Maintain precise land parcel location and ownership info
- Used for tax assessment, valuation and mortgage

- Six queries including
  - Avg. value of single family residential properties
  - Commercial properties on unpermitted landfills
Macro benchmark – Toxic spill

- Toxic chemicals spilled in a waterway may spread miles
- Recursively determines all downstream segments from initial spill point

- Two queries including
  - If spill point is on any waterway segment
Macro benchmark – Toxic spill

• Toxic chemicals spilled in a waterway may spread miles
• Recursively determines all downstream segments from initial spill point

• Two queries
  – If spill point is on any waterway segment
  – Waterway segments within 20 mile downstream of spill point
Macro benchmark – Toxic spill

- Toxic chemicals spilled in a waterway may spread miles
- Recursively determines all downstream segments from initial spill point

- Two queries
  - If spill point is on any waterway segment
  - Waterway segments within 20 mile downstream of spill point
Macro benchmark – Flood Risk Analysis

- Flood Insurance Rate Map depicts flood hazard areas
- DFIRM database is used to determine flood insurance rate

Map of Travis County, TX
Macro benchmark – Flood Risk Analysis

- Flood Insurance Rate Map depicts flood hazard areas
- DFIRM database is used to determine flood insurance rate

- Four queries including
  - Residential property owners required to carry flood insurance
  - Industrial complexes in high risk areas

Flood Insurance Rate Map of Travis County, TX showing high risk areas
Goals for the spatial benchmark

- Comprehensive coverage of spatial features
- Real-world workloads
- Extensible
- Portable
Implementation

- **Spatial Scenario**: initialization, invocation of the queries and cleanup
- **Database Dialect**: contain actual SQL queries
Implementation

- Supported runtime parameters
  - number of warm-up runs
  - iterations
  - number of threads
  - database to run against

- Output report

<table>
<thead>
<tr>
<th>Avg Duration</th>
<th>Avg Ops Sec</th>
<th>Count Of ResultSet</th>
<th>Duration</th>
<th>Iterations</th>
<th>Other Exceptions</th>
<th>SQL Exceptions</th>
<th>Warmup Count</th>
<th>Warmup Duration</th>
<th>DBType</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.236</td>
<td>4.237288</td>
<td>310.0</td>
<td>0.708</td>
<td>3.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.972</td>
<td>MySQL 5.0.91</td>
</tr>
<tr>
<td>0.945333</td>
<td>1.057827</td>
<td>80.0</td>
<td>2.836</td>
<td>3.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>2.415</td>
<td>PostgreSQL 8.4.2</td>
</tr>
<tr>
<td>1.954666</td>
<td>0.511596</td>
<td>80.0</td>
<td>5.864</td>
<td>3.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>2.759</td>
<td>Informix</td>
</tr>
</tbody>
</table>
Experimental setup

• **Database**
  – MySQL 5.0.91, PostgreSQL 8.4.2 and Informix 11.50

• **Dataset** – TIGER® for Texas and Travis County dataset

<table>
<thead>
<tr>
<th></th>
<th>MySQL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total data set size (GB)</td>
<td>4.6</td>
</tr>
<tr>
<td>Number of tables</td>
<td>15</td>
</tr>
<tr>
<td>Size of the largest data table (MB)</td>
<td>1651.6</td>
</tr>
<tr>
<td>Size of the largest table index (MB)</td>
<td>416.9</td>
</tr>
<tr>
<td>Cardinality of the largest table</td>
<td>5,895,060</td>
</tr>
</tbody>
</table>

• **Machine** – Pentium 4 CPU, 512 MB RAM, 240 GB disk
Challenges and some observations

• Database idiosyncrasies
  – MySQL: no refine step, only filter step
  – MySQL: table order matters

• OGC standards compliance
  – Some spatial functions unsupported

• Configuration issues

• Tuning
Challenges and some observations

- Query execution phase
  - MySQL performs MBR-based filter, but does not execute the refinement phase

<table>
<thead>
<tr>
<th></th>
<th>Intersects</th>
<th>Contains</th>
<th>Equals</th>
<th>Overlaps</th>
<th>Touches</th>
<th>Within</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySQL</td>
<td>7141</td>
<td>6246</td>
<td>5973</td>
<td>573</td>
<td>37</td>
<td>6246</td>
</tr>
<tr>
<td>PostgreSQL</td>
<td>6621</td>
<td>6019</td>
<td>5971</td>
<td>22</td>
<td>532</td>
<td>6019</td>
</tr>
<tr>
<td>Informix</td>
<td>6621</td>
<td>6018</td>
<td>5971</td>
<td>23</td>
<td>532</td>
<td>6018</td>
</tr>
</tbody>
</table>

Area and Area queries: # of records returned
Challenges and some observations

- **Query execution phase**
  - MySQL performs MBR-based filer, but does not execute the refinement phase

<table>
<thead>
<tr>
<th></th>
<th>Intersects</th>
<th>Contains</th>
<th>Equals</th>
<th>Overlaps</th>
<th>Touches</th>
<th>Within</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySQL</td>
<td>7141</td>
<td>6246</td>
<td>5973</td>
<td>573</td>
<td>37</td>
<td>6246</td>
</tr>
<tr>
<td>PostgreSQL</td>
<td>6621</td>
<td>6019</td>
<td>5971</td>
<td>22</td>
<td>532</td>
<td>6019</td>
</tr>
<tr>
<td>Informix</td>
<td>6621</td>
<td>6018</td>
<td>5971</td>
<td>23</td>
<td>532</td>
<td>6018</td>
</tr>
</tbody>
</table>

Area and Area queries: # of records returned

Actual geometries touch Goose Island State Park and Aransas Bay
Challenges and some observations

- Query execution phase
  - MySQL performs MBR-based filer, but does not execute the refinement phase

<table>
<thead>
<tr>
<th></th>
<th>Intersects</th>
<th>Contains</th>
<th>Equals</th>
<th>Overlaps</th>
<th>Touches</th>
<th>Within</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySQL</td>
<td>7141</td>
<td>6246</td>
<td>5973</td>
<td>573</td>
<td>37</td>
<td>6246</td>
</tr>
<tr>
<td>PostgreSQL</td>
<td>6621</td>
<td>6019</td>
<td>5971</td>
<td>22</td>
<td>532</td>
<td>6019</td>
</tr>
<tr>
<td>Informix</td>
<td>6621</td>
<td>6018</td>
<td>5971</td>
<td>23</td>
<td>532</td>
<td>6018</td>
</tr>
</tbody>
</table>

Area and Area queries: # of records returned

But, MBRs overlap
Challenges and some observations

- **Query execution phase**
  - MySQL performs MBR-based filter, but does not execute the refinement phase

<table>
<thead>
<tr>
<th></th>
<th>Intersects</th>
<th>Contains</th>
<th>Equals</th>
<th>Overlaps</th>
<th>Touches</th>
<th>Within</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySQL</td>
<td>7141</td>
<td>6246</td>
<td>5973</td>
<td>573</td>
<td>37</td>
<td>6246</td>
</tr>
<tr>
<td>PostgreSQL</td>
<td>6621</td>
<td>6019</td>
<td>5971</td>
<td>22</td>
<td>532</td>
<td>6019</td>
</tr>
<tr>
<td>Informix</td>
<td>6621</td>
<td>6018</td>
<td>5971</td>
<td>23</td>
<td>532</td>
<td>6018</td>
</tr>
</tbody>
</table>

Area and Area queries: # of records returned

Actual geometries are disjoint
Challenges and some observations

- Query execution phase
  - MySQL performs MBR-based filer, but does not execute the refinement phase

<table>
<thead>
<tr>
<th></th>
<th>Intersects</th>
<th>Contains</th>
<th>Equals</th>
<th>Overlaps</th>
<th>Touches</th>
<th>Within</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySQL</td>
<td>7141</td>
<td>6246</td>
<td>5973</td>
<td>573</td>
<td>37</td>
<td>6246</td>
</tr>
<tr>
<td>PostgreSQL</td>
<td>6621</td>
<td>6019</td>
<td>5971</td>
<td>22</td>
<td>532</td>
<td>6019</td>
</tr>
<tr>
<td>Informix</td>
<td>6621</td>
<td>6018</td>
<td>5971</td>
<td>23</td>
<td>532</td>
<td>6018</td>
</tr>
</tbody>
</table>

Area and Area queries: # of records returned

But, MBRs overlap
Challenges and some observations

• Support for spatial functions not complete
  – MySQL does not support Distance, Dwithin, Buffer, ConvexHull
  – Informix does not support Dwithin

• Simulated functions
  – MySQL: Distance, Dwithin
  – Informix: Dwithin

• Runtime issues
  – Informix had runtime issues with Buffer, StartPoint, EndPoint
Challenges and some observations

- Table order in the spatial predicate is important for MySQL

- select count(*) from arealm a, edges e where

<table>
<thead>
<tr>
<th></th>
<th>intersects(e.shape, a.shape)</th>
<th>intersects(a.shape, e.shape)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use spatial index</td>
<td>X</td>
<td>✓</td>
</tr>
<tr>
<td>Exec time</td>
<td>23 hours</td>
<td>3 minutes</td>
</tr>
</tbody>
</table>

- select count(*) from arealm a, edges e where

<table>
<thead>
<tr>
<th></th>
<th>intersects(e.shape, a.shape) and a.ogr_fid= 3332</th>
<th>intersects(a.shape, e.shape) and a.ogr_fid= 3332</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use spatial index</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>Exec time</td>
<td>0.08 seconds</td>
<td>16 minutes</td>
</tr>
</tbody>
</table>

- select count(*) from arealm a1, arealm a2 where

<table>
<thead>
<tr>
<th></th>
<th>intersects(a1.shape, a2.shape)</th>
<th>intersects(a2.shape, a1.shape)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use spatial index</td>
<td>X</td>
<td>✓</td>
</tr>
<tr>
<td>Exec time</td>
<td>2 minutes</td>
<td>0.35 seconds</td>
</tr>
</tbody>
</table>
Challenges and some observations

- Table order in the spatial predicate doesn’t matter for

- PostgreSQL
  
  ```sql
  select count(*) from arealm_merge a1, arealm_merge a2 where
  ```

<table>
<thead>
<tr>
<th>Use spatial index</th>
<th>ST_Intersects(a1.the_geom, a2.the_geom)</th>
<th>ST_Intersects(a2. the_geom, a1. the_geom)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Exec time (sec)</td>
<td>15.09</td>
<td>19.22</td>
</tr>
</tbody>
</table>

- Informix
  
  ```sql
  select count(*) from arealm_merge a1, arealm_merge a2 where
  ```

<table>
<thead>
<tr>
<th>Use spatial index</th>
<th>ST_Intersects(a1.shape,a2.shape)</th>
<th>ST_Intersects(a2.shape,a1.shape)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Exec time (sec)</td>
<td>3.2</td>
<td>3.1</td>
</tr>
</tbody>
</table>
Overview

- Motivation
- Background on spatial database
- Micro benchmark
- Macro benchmark
- Challenges
- Using the benchmark
- Conclusion
Using the benchmark

• Upload the dataset from shape files
• Configure
• Run the benchmark
• View results

• Example:
  – Compare with PostgreSQL, MySQL and Informix
    • Not a fair comparison - no refine step in MySQL
Results - Record insertion

![Diagram showing time comparison for different operations]

- WritePoint: Informix - 12.4, Postgres - 11.3, MySQL - 15.0
- WritePolygon: Informix - 14.3, Postgres - 11.5, MySQL - 15.0
- WriteLine: Informix - 15.0, Postgres - 11.7, MySQL - 17.0

Times are shown on a log scale.
Results - Spatial join

![Bar chart showing performance times for different spatial join operations with Informix, Postgres, and MySQL. The chart compares the time in seconds for operations such as AreaContainsArea, AreaEqualsArea, AreaOverlapsArea, AreaTouchesArea, AreaWithinArea, and AreaDisjointArea.]
Results – Spatial join with an object
Results - Spatial analysis

The bar chart illustrates the time taken (in seconds) for various spatial analysis operations across different databases: Informix, Postgres, and MySQL. The operations include:

- DistanceFromPoint
- BoundingBox
- DimensionPolygon
- EnvelopeLine
- LongestLine
- LargestArea
- TotalLength
- TotalArea
- BufferPolygon
- ConvexHullPolygon

The y-axis represents the time in a logarithmic scale, allowing for a clear comparison of the different operations and databases.
Results - Macro benchmark

[Graph showing benchmark results for Informix, Postgres, and MySQL with specific times for tasks such as LandInfoMgt, ToxicSpill, FloodRisk, MapBrowsing, ReverseGeocoding, and Geocoding.]
Score Metric

- **Methodology**
  - Geometric mean over all queries, each exec time is normalized over a reference DBMS

- **Metric**

\[
Score_{DBMS_x} = \sqrt[N]{\prod_{q=1}^{N} \frac{Time_{q}^{DBMS_{ref}}}{Time_{q}^{DBMS_x}}}
\]
Score Metric

- Which queries to be included in the score?
  - Ideally all of them
  - Not all are supported by all databases

<table>
<thead>
<tr>
<th></th>
<th>Micro benchmark</th>
<th>Macro benchmark</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySQL</td>
<td>Incomplete</td>
<td>Incomplete</td>
</tr>
<tr>
<td>PostgreSQL</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Informix</td>
<td>Incomplete</td>
<td>Incomplete</td>
</tr>
</tbody>
</table>
• Scores
  – Included only those queries supported by all databases

<table>
<thead>
<tr>
<th></th>
<th>Micro benchmark</th>
<th>Macro benchmark</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySQL</td>
<td>5.17</td>
<td>1.06</td>
</tr>
<tr>
<td>PostgreSQL</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Informix</td>
<td>1.48</td>
<td>0.29</td>
</tr>
</tbody>
</table>
Conclusion

• Comprehensive coverage in the micro benchmark workloads

• Macro benchmarks includes representative real-world applications

• Portable

• Extensible

• Flexible
Future work

• Dataset scalability

• Spatio-temporal workloads

• Invite suggestions from the community
Thank you!