Benchmarks for Mobile Database Access

Niraj Tolia
Carnegie Mellon University
Need for Mobile DB Benchmarks
Need for Mobile DB Benchmarks

Mobility

Wireless WANs
Need for Mobile DB Benchmarks

Mobility

Wireless WANs

Database-enabled Applications
Need for Mobile DB Benchmarks

Mobility

Wireless WANs

Database-enabled Applications

How can you measure a mobile client’s performance?
Types of Benchmarks

Trace Replay
Types of Benchmarks

- Trace Replay
- Macro Benchmarks
Types of Benchmarks

- Trace Replay
- Macro Benchmarks
- Micro Benchmarks
Types of Benchmarks

- Trace Replay
- Macro Benchmarks
- Micro Benchmarks
MobileSales

- Used to evaluate the Cedar system [MobiSys 2007]
  - Database access over low-bandwidth networks

- MobileSales
  - Design Choices
  - Benchmark Description
  - Conclusion
Ideal Benchmark Properties

- Relevant
- Simple
  - Understandable
  - Metrics
- Portable
- Scalable
- Acceptance

Source: Jim Gray, March, 1997
Commonly Used Benchmarks

- TPC - \{App, C, E, H\}
- RUBBoS, RUBiS (originally from Rice Univ.)
- Plus host of other benchmarks…

- Some obviously not a good fit (DSS benchmarks)
- Almost all of these are server oriented
New Benchmark Options

Modify Existing Benchmark

Create New Benchmark
Origin of MobileSales

- **TPC-App**
  - Application server and web services benchmark
  - Simulates a B2B transactional environment
  - Designed to stress the Application Server

- **Workload**
  - Retail distributor on the Internet
  - “… supports user online ordering and browsing”
Mobile Scenario Adaptation

Clients <-> Application Server <-> Database Server

XML <-> JDBC/ODBC
Mobile Scenario Adaptation

- Insight: 1:1 mapping between XML and JDBC
Mobile Scenario Adaptation

- Insight: 1:1 mapping between XML and JDBC
Mobile Scenario Adaptation

- Insight: 1:1 mapping between XML and JDBC
- Shifted focus from Application Server to Client
Mobile Scenario Adaptation

- Insight: 1:1 mapping between XML and JDBC
- Shifted focus from Application Server to Client
Mobile Scenario Adaptation

- Insight: 1:1 mapping between XML and JDBC
- Shifted focus from Application Server to Client
- Background clients model concurrent access
Mobile Scenario Adaptation

- Insight: 1:1 mapping between XML and JDBC
- Shifted focus from Application Server to Client
- Background clients model concurrent access
- Retains workload, interactions, dataset, etc.
Benchmark Properties

- Online distributor system
- Clients perform *Interactions*

<table>
<thead>
<tr>
<th>Interaction Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Customer</td>
</tr>
<tr>
<td>Change Payment Method</td>
</tr>
<tr>
<td>Create Order</td>
</tr>
<tr>
<td>Order Status</td>
</tr>
<tr>
<td>View New Products</td>
</tr>
<tr>
<td>Product Detail</td>
</tr>
<tr>
<td>Change Item</td>
</tr>
</tbody>
</table>
Benchmark Properties

- Online distributor system
- Clients perform *Interactions*

<table>
<thead>
<tr>
<th>Interaction Type</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Customer</td>
<td>1 %</td>
</tr>
<tr>
<td>Change Payment Method</td>
<td>5 %</td>
</tr>
<tr>
<td>Create Order</td>
<td>50 %</td>
</tr>
<tr>
<td>Order Status</td>
<td>5 %</td>
</tr>
<tr>
<td>View New Products</td>
<td>7 %</td>
</tr>
<tr>
<td>Product Detail</td>
<td>30 %</td>
</tr>
<tr>
<td>Change Item</td>
<td>2 %</td>
</tr>
</tbody>
</table>
Database Properties

- Database contains information on
  - Customers, Orders, Items, Current Stock, etc.
  - Dataset scales in relation to # test clients
Database Schema

CUSTOMER
192 × Clients

Legend
TABLE NAME
<Cardinality>
Database Schema

CUSTOMER
192 × Clients

ADDRESS
1.4 × CUSTOMER

COUNTRY
92

Legend

TABLE NAME
<Cardinality>
Database Schema

CUSTOMER 192 × Clients

ADDRESS 1.4 × CUSTOMER

COUNTRY 92

AUTHOR 25,000

ITEM 100,000

STOCK 100,000

Legend

TABLE NAME
<Cardinality>
Database Schema

CUSTOMER
192 × Clients

ADDRESS
1.4 × CUSTOMER

ORDERS
10 × CUSTOMER

COUNTRY
92

AUTHOR
25,000

ITEM
100,000

STOCK
100,000

Legend
TABLE NAME
<Cardinality>
Database Schema

CUSTOMER
192 \times \text{Clients}

ADDRESS
1.4 \times \text{CUSTOMER}

COUNTRY
92

ORDERS
10 \times \text{CUSTOMER}

ORDER_LINE
5.5 \times \text{ORDERS}

AUTHOR
25,000

ITEM
100,000

STOCK
100,000

Legend
TABLE NAME
\langle \text{Cardinality} \rangle
View Products Interaction

Input:

Product IDs

Query:

SELECT <product description> FROM item, author WHERE item_id IN (Product IDs)
Create Order Interaction

Input:

Customer ID, Shipping Address, Item List, Payment Type

Query:

SELECT <cc info> FROM customer WHERE c_id
SELECT <add info> FROM customer, address, country
INSERT INTO address <add info> (if new)
SELECT <item info, stock> FROM item, stock
INSERT INTO orders <order info>
INSERT INTO order_line <item info>
Metrics

- Performance metrics
  - Throughput (Total No. of Interactions)
  - Latency (Average Interaction completion time)
Axes of freedom

- Interaction Frequency
  - Currently ~ 40:60 Read:Write ratio
  - Can be modified
- Think Time
- Benchmark Duration (5 minutes works well)
- Number of Concurrent Clients
Back to Benchmark Properties

- Relevant – Resembles a real application
- Simple – Reasonably so
  - Understandable
  - Metrics
- Portable – Runs on PDAs and Servers
- Scalable – Dataset scales with requirements
- Acceptance – Up to community
Next steps

- Examine other applications
  - Customer Relationship Management
  - Disaster Recovery (Sahana Mobile)

- Look at benchmarks based on trace-replay
Conclusions

- MobileSales
  - Benchmark for mobile access to databases
  - More details in paper and TPC-App documentation

Source code available at [http://www.cs.cmu.edu/~ntolia](http://www.cs.cmu.edu/~ntolia)
“What has the power to generate heated controversy, hard feelings, and bold accusations? A sex scandal? Litigation? Nope – try benchmarks.”

— J. Stokes
Ars Technica, 1999