Making Fast Databases FASTER @andy_pavlo

Yale University Columbia University April 2012
Fast + Cheap
OLTP Through the Looking Glass, and What We Found There

SIGMOD 2008

Legacy Systems

TPC-C NewOrder

CPU Cycles

- Real Work
- Buffer Pool
- Latching
- Locking
- Logging
- B-Tree Keys

0%
20%
40%
60%
80%
100%

12.3%
29.6%
10.2%
18.7%
21.1%
8.1%
OLTP Transactions

Fast  Repetitive  Small
H-Store: A High-Performance, Distributed Main Memory Transaction Processing System

VLDB vol. 1, issue 2, 2008
Optimization #1: Partition database to reduce the number of distributed txns.
### CUSTOMER

<table>
<thead>
<tr>
<th>c_id</th>
<th>c_w_id</th>
<th>c_last</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>1001</td>
<td>5</td>
<td>RZA</td>
<td></td>
</tr>
<tr>
<td>1002</td>
<td>3</td>
<td>GZA</td>
<td></td>
</tr>
<tr>
<td>1003</td>
<td>12</td>
<td>Raekwon</td>
<td></td>
</tr>
<tr>
<td>1004</td>
<td>5</td>
<td>Deck</td>
<td></td>
</tr>
<tr>
<td>1005</td>
<td>6</td>
<td>Killah</td>
<td></td>
</tr>
<tr>
<td>1006</td>
<td>7</td>
<td>ODB</td>
<td></td>
</tr>
</tbody>
</table>

### ORDERS

<table>
<thead>
<tr>
<th>o_id</th>
<th>o_c_id</th>
<th>o_w_id</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>78703</td>
<td>1004</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>78704</td>
<td>1002</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>78705</td>
<td>1006</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>78706</td>
<td>1005</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>78707</td>
<td>1005</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>78708</td>
<td>1003</td>
<td>12</td>
<td></td>
</tr>
</tbody>
</table>
## ITEM

<table>
<thead>
<tr>
<th>i_id</th>
<th>i_name</th>
<th>i_price</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>603514</td>
<td>XXX</td>
<td>23.99</td>
<td>-</td>
</tr>
<tr>
<td>267923</td>
<td>XXX</td>
<td>19.99</td>
<td>-</td>
</tr>
<tr>
<td>475386</td>
<td>XXX</td>
<td>14.99</td>
<td>-</td>
</tr>
<tr>
<td>578945</td>
<td>XXX</td>
<td>9.98</td>
<td>-</td>
</tr>
<tr>
<td>476348</td>
<td>XXX</td>
<td>103.49</td>
<td>-</td>
</tr>
<tr>
<td>784285</td>
<td>XXX</td>
<td>69.99</td>
<td>-</td>
</tr>
</tbody>
</table>

### Diagram

```
+----------------+     +----------------+     +----------------+
| CUSTOMER       |     | CUSTOMER       |     | CUSTOMER       |
| ORDERS         |     | ORDERS         |     | ORDERS         |
| ITEM           |     | ITEM           |     | ITEM           |
|                |     | ITEM           |     | ITEM           |
|                |     |                |     |                |
```

- CUSTOMER
- ORDERS
- ITEM
### CUSTOMER

<table>
<thead>
<tr>
<th>c_id</th>
<th>c_w_id</th>
<th>c_last</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>1001</td>
<td>5</td>
<td>RZA</td>
<td>-</td>
</tr>
<tr>
<td>1002</td>
<td>3</td>
<td>GZA</td>
<td>-</td>
</tr>
<tr>
<td>1003</td>
<td>12</td>
<td>Raekwon</td>
<td>-</td>
</tr>
<tr>
<td>1004</td>
<td>5</td>
<td>Deck</td>
<td>-</td>
</tr>
<tr>
<td>1005</td>
<td>6</td>
<td>Killah</td>
<td>-</td>
</tr>
<tr>
<td>1006</td>
<td>7</td>
<td>ODB</td>
<td>-</td>
</tr>
</tbody>
</table>

---

**Diagram:**

- **CUSTOMER**
- **ORDERS**
- **ITEM**
Client Application

NewOrder(5, "Method Man", 1234)
Large-Neighborhood Search Algorithm
Large-Neighborhood Search

Initial Design

Relaxation

Local Search

Restart

Schema

Workload

DDL

Large Neighborhood Search
Large-Neighborhood Search

Table Candidate
- Horizontal: C_ID
- Replication: False
- 2ndry Index: {C_ID,C_NM}

Proc Candidate
- Parameter: #1
Throughput

(txn/s)

Horticulture  State-of-the-Art

TATP +88%

TPC-C +16%

TPC-C Skewed +183%
Search Times

- **TATP**
- **SEATS**
- **TPC-C**
- **TPC-C Skewed**
- **AuctionMark**
- **TPC-E**

Graphs showing search times in different scenarios.
Optimization #2: Predict what txns will do before they execute.
» Partitions Touched?
» Undo Log?
» Done with Partitions?
Current State:

```
begin
```

Input Parameters:
```
\begin{align*}
    w_{id} &= 0 \\
    i_{w\_ids} &= [0, 1] \\
    i_{ids} &= [1001, 1002] \\
\end{align*}
```

GetWarehouse:
```
\begin{align*}
    SELECT * FROM \text{WAREHOUSE} \\
    \text{WHERE } w_{ID} = ?
\end{align*}
```
Estimated Execution Path

Input Parameters:
\[ w_{id}=0 \]
\[ i_{w_{ids}}=[0,1] \]
\[ i_{ids}=[1001,1002] \]

Transaction Estimate:

- Confidence Coefficient: 0.96
- Best Partition: 0
- Partitions Accessed: \{0\}
- Use Undo Logging: Yes
Throughput

Transaction per second (txn/s)

Houdini vs Assume Single-Partitioned

TATP +57%

TPC-C +126%

AuctionMark +117%
Prediction Overhead

- Estimation
- Execution
- Planning
- Coordination
- Other

TATP
TPC-C
AuctionMark
Conclusion: Achieving fast performance is more than just using only RAM.

Future Work: Reduce distributed txn overhead through creative scheduling.
https://github.com/apavlo/h-store
Help is Available

+1-212-939-7064

Graduate Student Abuse Hotline
Available 24/7
Collect Calls Accepted