Automatic Database Partitioning in Parallel OLTP Systems

@andy_pavlo
H-Store: A High-Performance, Distributed Main Memory Transaction Processing System

Client Application

Procedure Name
Input Parameters

Transaction Execution

Database Cluster
Client Application

Transaction Result

Database Cluster
Horticulture

Automatic Database Design Tool for Parallel Systems

Skew-Aware Automatic Database Partitioning in Shared-Nothing, Parallel OLTP Systems

SIGMOD 2012
SELECT * FROM WAREHOUSE WHERE W_ID = 10;

SELECT * FROM DISTRICT WHERE D_W_ID = 10 AND D_ID = 9;

INSERT INTO ORDERS (O_W_ID, O_D_ID, O_C_ID, ...) VALUES (10, 9, 12345, ...);

...
## Horticulture

### 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

| i_id  | i_name | i_price | ...
|-------|--------|---------|---------
| 603514| XXX    | 23.99   | -       |
| 267923| XXX    | 19.99   | -       |
| 475386| XXX    | 14.99   | -       |
| 578945| XXX    | 9.98    | -       |
| 476348| XXX    | 103.49  | -       |
| 784285| XXX    | 69.99   | -       |
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>
Client Application

NewOrder(5, “Method Man”, 1234)
Client Application

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

Input
- Schema
- Workload

Initial Design

Relaxation

Local Search

Best Design

Schema

Workload

Restart

STOP

DDL
Large-Neighborhood Search

Input

Best Design

Schema
DDT

Initial Design

Relaxation
Local Search

Restart

Large Neighborhood Search

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

**Proc Candidate**
- Parameter: #1
Cost Model

Distributed Transactions + Workload Skew Factor
Algorithm Comparison

(cost estimate) lower is better

Horticulture  State-of-the-Art

TATP

TPC-C

TPC-C Skewed
Throughput

(txn/sec)
higher is better

Horticulture  State-of-the-Art

TATP  +88%

TPC-C  +16%

TPC-C Skewed  +183%
Conclusion:
Dating scene is still difficult.
But partitioning your database is now easier.
H-Store

H-Store is an experimental main-memory, parallel database management system that is optimized for OLTP applications. It is a highly distributed, row-store based relational database that runs on a cluster on shared-nothing, main memory execution nodes.

The H-Store project is a collaboration between MIT, Brown University, Yale University, and HP Labs.

New Release (March 2012)

March 18th, 2012 at 3:30 pm

The H-Store project is releasing the latest version of its experimental OLTP database management system. This version contains a large number of speed and stability improvements. The next release will contain mostly new features.

Major highlights of this release:

1. Significant speed optimizations for single-partition transactions.
2. Fixed several memory leaks for aborted-only transactions.
3. Fixed several stability issues when transactions share lock-free data structures.
4. Fixed predicate aggregates pushdowns for distributed query plans.
5. Fixed the ability for transactions to return the proper release codes on failures that mimic VoltDB's protocol.
6. Fixed H-Store's fatigue, automatic deployment script for EC2 now supports multiple concurrent "virtual" clusters. This allows you to easily assign more slaves to an independent cluster under a single AWS account.
8. Fixed building on OS X Lion.

The source code for this release can be downloaded via GitHub.

http://hstore.cs.brown.edu

http://github.com/apavlo/h-store