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

SIGMOD 2012, Pavlo et al.

Hefu Chai
Credit

• Part of slides from Andy Pavlo
There is a saying...

• Girls are really only interested in two things. They want a guy that is good looking, or they want a guy that really knows a lot about databases.

Andy Pavlo
Existing database partitioning Techniques

• Notion of data declustering
  • Overhead of maintaining transaction consistency
  • Lock contention

Not applicable to OLTP systems!
H-Store

OLTP Transactions

Fast  Repetitive  Small
We need an approach that supports...

• Stored Procedure
• Load balancing in the presence of time-varying skew
• Complex schemas
• Deployments with larger number of partitions
Skew-Aware Automatic Database Partitioning in Shared-Nothing, Parallel OLTP Systems

SIGMOD 2012
What are the key issues

• Distributed transactions
• Temporal workload skew
Distributed transactions

TPC-C NewOrder

No Distributed Txns
20% Distributed Txns

Partitions

txn/s
What are the key issues

- Distributed transactions
- Temporal workload skew
Temporal workload skew

• Think about the example of Wikipedia
  • Even though the average load of the cluster for the entire day is uniform, the load across the cluster for any point is unbalanced

• Static Skew Vs. Temporal Skew
• Maintain the **tradeoff** between distributed transactions and temporal skew

• Extend design space to include replicated *secondary* indexes

• Organically handling stored procedure routing

---

**Large Neighborhood Search**

**Skew-Aware Cost Model**
What are the design options

For each table:

• Horizontally partition
• Replicate on all partitions
• Replicate a secondary index for a subset of its column
• Effectively route incoming transaction requests
Horizontal Partitioning

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>
Table Replication

### 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>
### CUSTOMER

<table>
<thead>
<tr>
<th>c_id</th>
<th>c_w_id</th>
<th>c_last</th>
</tr>
</thead>
<tbody>
<tr>
<td>1001</td>
<td></td>
<td>RZA</td>
</tr>
<tr>
<td>1002</td>
<td></td>
<td>GZA</td>
</tr>
<tr>
<td>1003</td>
<td></td>
<td>Raekwon</td>
</tr>
<tr>
<td>1004</td>
<td></td>
<td>Ice</td>
</tr>
<tr>
<td>1005</td>
<td></td>
<td>Killah</td>
</tr>
<tr>
<td>1006</td>
<td></td>
<td>ODB</td>
</tr>
</tbody>
</table>
Stored Procedure Routing

Client Application

NewOrder (5, “Method Man”, 1234)
What are the key technique contributions

- Large-Neighborhood Search
- Skew-Aware Cost Model
Large-Neighborhood Search

Initial Design

- Select the **most frequently** accessed column for horizontal partitioning
- Greedily replicate **read-only** tables until no space left
- Select next most frequently accessed, **read-only** column as secondary
- Index attribute
- Select the routing parameter for stored procedures
Relaxation

- Allow LNS to escape a local minimum and jump to a new neighborhood of potential solutions
- Horticulture must decide:
  - How many tables to relax
  - Which tables to relax
  - What design options will be examined for each relaxed table
Large-Neighborhood Search

Local Search

Best Design

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

Proc Candidate
- Parameter: #1
What are the key technique contributions

• Large-Neighborhood Search
• Skew-Aware Cost Model
Cost Model

Distributed Transactions + Workload Skew Factor
Skew-Aware Cost Model

• Accentuates the properties that are important in a DB
• Compute quickly
• Estimate the cost of an incomplete design
• The cost estimates must increase monotonically as more variables are set
Skew-Aware Cost Model

• Measure
  • How much workload executes as a single-partition transactions
  • How uniformly load is distributed across the cluster

\[
\text{cost}(\mathcal{D}, \mathcal{W}) = \frac{\alpha \times \text{CoordinationCost}(\mathcal{D}, \mathcal{W}) + \beta \times \text{SkewFactor}(\mathcal{D}, \mathcal{W})}{\alpha + \beta}
\]

Tradeoff!
Skew-Aware Cost Model

Coordinator Cost

\[
\left( \frac{\text{partitionCount}}{(\text{txnCount} \times \text{numPartitions})} \right) \times \left( 1.0 + \frac{\text{dtxnCount}}{\text{txnCount}} \right)
\]

Total number of partitions accessed divided by total number of partitions could have been accessed, and scale it up.
Skew-Aware Cost Model

Skew Factor

\[
\frac{\sum_{i=0}^{numIntervals} skew[i] \times txnCounts[i]}{\sum txnCounts}
\]

To avoid time varying skew, divide W into finite intervals.
Incomplete Designs

• Query that references a table with an unset attribute in a design as being unknown
• For each unknown query:
  • Coordinator Cost: Assume that any unknown query is single-partitioned
  • Skew Factor: Assume that unknown queries execute on all partitions in the cluster
• ‘Unknown’ change to ‘known’
• ‘Known’ cannot change to ‘Unknown’

monotonically increase!
Optimizations

• Access Graphs
• Workload Compression
### Access Graph

![Access Graph Diagram]

**Vertex:** Table

**Edge:** tables are co-accessed

**Weight of edges:** the number of times the queries forming the relationship

<table>
<thead>
<tr>
<th>Edge#</th>
<th>Columns</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>C.C_ID ↔ C.C_ID</td>
<td>200</td>
</tr>
<tr>
<td>(2)</td>
<td>C.C_ID ↔ O.O_C_ID</td>
<td>100</td>
</tr>
<tr>
<td>(3)</td>
<td>O.O_ID ↔ OL.OL_O_ID</td>
<td>100</td>
</tr>
<tr>
<td>(4)</td>
<td>O.O_ID ↔ OL.OL_O_ID</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>O.O_C_ID ↔ OL.OL_C_ID</td>
<td></td>
</tr>
</tbody>
</table>
Optimizations

- Access Graphs
- Workload Compression
Workload Compression

- combine sets of similar queries in individual transactions into fewer weighted records
- combine similar transactions into a smaller number of weighted records in the same manner
Throughput

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

% Single-Partitioned Transactions

Minutes Elapsed

0 1 2 3 4 5

0 20 40 60 80 100

0 2 4 6 8 10

0 24 48 72 96 120

0 10 20 30 40 50

0 24 48 72 96 120
Andy: it works!