HY590.45
Modern Topics in Scalable Storage Systems

Kostas Magoutis
magoutis@csd.uoc.gr
http://www.csd.uoc.gr/~hy590-45
Google network and infrastructure sites

CREATE TABLE Users {
    uid INT64 NOT NULL, email STRING
} PRIMARY KEY (uid), DIRECTORY;

CREATE TABLE Albums {
    uid INT64 NOT NULL, aid INT64 NOT NULL, name STRING
} PRIMARY KEY (uid, aid),
    INTERLEAVE IN PARENT Users ON DELETE CASCADE;

```
<table>
<thead>
<tr>
<th>Users(1)</th>
<th>Directory 3665</th>
</tr>
</thead>
<tbody>
<tr>
<td>Albums(1,1)</td>
<td></td>
</tr>
<tr>
<td>Albums(1,2)</td>
<td></td>
</tr>
<tr>
<td>Users(2)</td>
<td>Directory 453</td>
</tr>
<tr>
<td>Albums(2,1)</td>
<td></td>
</tr>
<tr>
<td>Albums(2,2)</td>
<td></td>
</tr>
<tr>
<td>Albums(2,3)</td>
<td></td>
</tr>
</tbody>
</table>
```

# Logical data layout

<table>
<thead>
<tr>
<th>Albums</th>
<th>Photos</th>
</tr>
</thead>
<tbody>
<tr>
<td>user_id</td>
<td>album_id</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Source: Sebastian Kanthak (Google), Strange Loop Conference, St. Louis, MO, 2013
Physical data layout: interleaved tables

Source: Sebastian Kanthak (Google), Strange Loop Conference, St. Louis, MO, 2013
Sharding

Cross-shard support for
• Transactions (read/write)
• Consistent (snapshot) reads

Source: Sebastian Kanthak (Google), Strange Loop Conference, St. Louis, MO, 2013
Spanner server organization
Replication

Source: Sebastian Kanthak (Google), Strange Loop Conference, St. Louis, MO, 2013
Spanner software stack
Directories ("buckets") can be moved across groups
Key techniques

• Paxos for data replication of each tablet across zones

• Two-phase locking (2PL) for serializability
  – Transactions should acquire all locks they need before starting
  – For performance: support read-only transactions without locks
  – Multi-value concurrency control (MVCC): timestamps/snapshots
  – Timestamps consistent with externally visible order

• Two-phase commit (2PC) for cross-table atomicity
Example: Ad System

<table>
<thead>
<tr>
<th>Campaig_n id</th>
<th>keyword</th>
<th>bid</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>strange loop</td>
<td>$2.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>region</th>
<th>time</th>
<th>campaign_id</th>
<th>cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>US</td>
<td>2013/09/20-07...</td>
<td>4</td>
<td>$1.50</td>
</tr>
<tr>
<td>US</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EU</td>
<td>2013/09/20-06...</td>
<td>4</td>
<td>$0.50</td>
</tr>
<tr>
<td>EU</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Source: Sebastian Kanthak (Google), Strange Loop Conference, St. Louis, MO, 2013
What can go wrong if using local timestamps

- Txn 1 creates a new ad on US server
- Ad serving system notified
- Ad server in Europe
- User clicks on ad
- Txn 2 logs click on EU server

Invariant: Any snapshot that contains txn 2 should also contain txn 1

Source: Sebastian Kanthak (Google), Strange Loop Conference, St. Louis, MO, 2013
External consistency

• Assume T1 commits before T2 starts according to global wall-clock
• T1 should be serialized before T2
• T2’s commit timestamp should be > T1’s commit timestamp
• Must apply even if T1 and T2 do not conflict

Source: Sebastian Kanthak (Google), Strange Loop Conference, St. Louis, MO, 2013
TrueTime API

<table>
<thead>
<tr>
<th>Method</th>
<th>Returns</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>TT.now()</code></td>
<td><code>TTinterval: [earliest, latest]</code></td>
</tr>
<tr>
<td><code>TT.after(t)</code></td>
<td>true if $t$ has definitely passed</td>
</tr>
<tr>
<td><code>TT.before(t)</code></td>
<td>true if $t$ has definitely not arrived</td>
</tr>
</tbody>
</table>
Picking commit timestamps

Acquired locks

\[ \text{Pick } s = \text{TT.now().latest} \]

Release locks

\[ s \text{ Wait until TT.now().earliest } > s \]

Commit wait

average \( \varepsilon \)  

Source: Wilson Hsieh, Spanner: Google’s Globally-Distributed Database, OSDI’12
Commit-wait and replication

Source: Wilson Hsieh, Spanner: Google’s Globally-Distributed Database, OSDI’12
Commit-wait and 2PC

Source: Wilson Hsieh, Spanner: Google’s Globally-Distributed Database, OSDI’12
Life of a read / write

TrueTime servers

TrueTime
(Tmin, Tmax)