PostgreSQL Durability & Performance
PostgreSQL Durability

- The ACID test
- Important data should be saved to disk when we COMMIT
- Transaction Log
## Hard Drive Latency

<table>
<thead>
<tr>
<th>Type</th>
<th>Latency (ms)</th>
<th>Transactions/Second</th>
</tr>
</thead>
<tbody>
<tr>
<td>5400 RPM</td>
<td>11.1</td>
<td>90</td>
</tr>
<tr>
<td>7200 RPM</td>
<td>8.3</td>
<td>120</td>
</tr>
<tr>
<td>10K RPM</td>
<td>6.0</td>
<td>167</td>
</tr>
<tr>
<td>15K RPM</td>
<td>4.0</td>
<td>250</td>
</tr>
<tr>
<td>Battery-Backed Write Cache</td>
<td>0.2</td>
<td>5000</td>
</tr>
</tbody>
</table>
Latency impact on throughput

The graph shows the pgbench transactions per second (TPS) for different numbers of clients. The x-axis represents the number of clients, ranging from 0 to 100, and the y-axis represents the TPS, ranging from 0 to 5000. Two lines are plotted: one for "Without BBU" and the other for "With working BBU."
Relaxing guarantee

• If we relax the guarantee
  – Databases much faster
  – Transaction data can be lost
PostgreSQL Flexible Durability

- synchronous_commit
- =on gives DURABILITY
- =off gives PERFORMANCE
Transaction Control

- synchronous_commit can be set
  - For the whole database
  - For an individual user
  - For an individual transaction

- Safe and Fast Transactions can co-exist without loss of performance or risk to data

- All of this has been available since 2007 (8.3)
Synchronous Replication

- New in PostgreSQL 9.1
- Zero Data Loss replication
- Efficient – thousands of TPS in tests
Sync Replication Durability
High Availability Concerns

- Commit waits for acknowledgement
- Commits on master could wait forever
- Server is down when all sync standbys gone
- Reduced availability with only two servers
- Need 3 servers for equal HA and sync rep
Target Cluster Architecture

- Master
- Many Standby Nodes
- synchronous_standby_names
- One active sync node
synchronous_standby_names

- First active standby on list becomes the sync node
- If that standby fails, moves to next name
- Standby name is application_name of standby
- Configuration same on all nodes
- synchronous_standby_names = "*"
Design for Performance

- Full duplex communication
- Reply messages have only write location
- Limited by network plus WAL write time
- Internet is approximately $\frac{1}{2}$ speed of light
## Measured Network Latency

<table>
<thead>
<tr>
<th>Type</th>
<th>Latency (ms)</th>
<th>Transactions/Second</th>
</tr>
</thead>
<tbody>
<tr>
<td>1Gbps</td>
<td>0.07</td>
<td>14286</td>
</tr>
<tr>
<td>100Mbps</td>
<td>0.3</td>
<td>3333</td>
</tr>
<tr>
<td>Baltimore-&gt;NY</td>
<td>15</td>
<td>57</td>
</tr>
<tr>
<td>Baltimore-&gt;SF</td>
<td>83</td>
<td>12</td>
</tr>
<tr>
<td>Baltimore-&gt;Netherlands</td>
<td>100</td>
<td>10</td>
</tr>
</tbody>
</table>
Scaling benchmark

- Master in Baltimore
  - BBWC to limit its overhead
- Standby at Casa 400, Amsterdam
- Commit rate measured with INSERT statements
- Measured ping time $\geq 100\text{ms}$
- Typical sync commit time $\geq 112\text{ms}$
- Theoretical single client max = 10 TPS
- Measured single client rate = 7 to 8 TPS
- How does it scale?
Efficient scaling
Sync Rep Performance

- Single sessions much slower than normal
- Overall server can be scale to high performance
- Applications using sync rep will be safe but slow
User Selectable Durability

• Set via synchronous_commit
• Two existing modes control master fsync
• Three new modes control sync rep
• World-first from PostgreSQL and 2ndQuadrant
  - Users can control the durability of each transaction
  - All durability levels can co-exist in one application
Log Shipping Developments

- 8.0 – Point in Time Recovery, Full WAL info
- 8.2 – Restartable Recovery, Log Switching
- 8.3 – Full page optimization, pg_standby
- 8.4 – BgWriter during Recovery
- 9.0 – Streaming Replication
  Hot Standby
- 9.1 – Synchronous Replication
- 9.2 – Cascading Replication
High Availability Replication

- Master-Slave clusters
- High Availability
- Read scalability
Multiple High Available Masters
Minimally Efficient Data Flow
Add Secondary Connections
Add extra read slaves

CABAL
Bi-Directional Replication

- OK, some people call it multi-master
- Read Anywhere
- Update Anywhere
- Conflict Resolution
- Conflict Avoidance
- Selectable (Local-only, Replicated, Sharded)
- Filtered, Deferrable
- Major Release Upgrades
PostgreSQL

• Durability

AND

• Performance

• Mixed to *your* requirements...