CS244b - Aurora

Learning goals

- Chain replication
- Write-ahead logs
- Quorum systems
- Rethinking abstraction layers
3-tier architecture

Public

Application

Storage

Shared data

Client Requests

Page Rendering Components

Request Routing

Aggregator Services

Services

Dynamo instances

Amazon S3

Other datastores

Figure 1: Service-oriented architecture of Amazon’s platform
EBS

EC2 instance

2x replication
MySQL on EC2 vs. EBS

Bad for latency (median, tail)

Bad for net. b/w

InnoDB transaction commit

- WAL write record here
- Update data on disk
- Double-write pages
- Statement log, FRM
Figure 2: Network IO in mirrored MySQL
\( A2 + 1 \)

Survive AZ failure and failure of server not in AZ

MTTF, MTTR

MTTR

MTTF

Small segments (10GiB)
1. Apply log records to all storage servers.
2. Quorum system.

Figure 3: Network IO in Amazon Aurora
N = 6, W = 4, R = 3
Acronyms

PG - protection groups

LSN - log sequence number

VDDL - volume durable LSN

LAL - log allocation limit (10M)

CPL - consistency point LSN

MTR - mini transaction

SCL - segment complete LSN (per-server)

VCL - Volume complete list
MySQL-Aurora transaction commit

- Primary: create log records, assign a LSN, $VDL \leq LSN \leq UDL$
- Create commit record, flag as CPC
- Send commit record, wait for $VDL \geq LSN$
  Get W Acks from each PF
  use backlings to determine SCL
Storage API (RPCs)
- write log record
- read data page (assert LSN)
- truncate

Stored at storage server
Log, data pages, per-page log records
Reads

Read want : SCL ≥ VCL

Wait for DB server

DB Crash

- After DB instance reboot, wait for R
  from each PE, compute UDL
- Tell servers to truncate (UDL, ..., UDL+1n)
- Undo inflight transactions

Storage server crash
Read-only

Send log records to RO instances

Apply $\text{LSN} \leq \text{UDL}$

only complete MTRs
Table 1: Network IOs for Aurora vs MySQL

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Transactions</th>
<th>IOs/Transaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mirrored MySQL</td>
<td>780,000</td>
<td>7.4</td>
</tr>
<tr>
<td>Aurora with Replicas</td>
<td>27,378,000</td>
<td>0.95</td>
</tr>
</tbody>
</table>
### Table 3: SysBench OLTP (writes/sec)

| Connections | Amazon Aurora | MySQL  
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>40,000</td>
<td>10,000</td>
</tr>
<tr>
<td>500</td>
<td>71,000</td>
<td>21,000</td>
</tr>
<tr>
<td>5,000</td>
<td>110,000</td>
<td>13,000</td>
</tr>
</tbody>
</table>

### Table 5: Percona TPC-C Variant (tpmC)

| Connections/Size/Warehouses | Amazon Aurora | MySQL 5.6 | MySQL 5.7  
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>500/10GB/100</td>
<td>73,955</td>
<td>6,093</td>
<td>25,289</td>
</tr>
<tr>
<td>5000/10GB/100</td>
<td>42,181</td>
<td>1,671</td>
<td>2,592</td>
</tr>
<tr>
<td>500/100GB/1000</td>
<td>70,663</td>
<td>3,231</td>
<td>11,868</td>
</tr>
<tr>
<td>5000/100GB/1000</td>
<td>30,221</td>
<td>5,575</td>
<td>13,005</td>
</tr>
</tbody>
</table>