Agenda

1. Terminology
2. Lifecycle of a MySQL instance
3. How do we migrate MySQL instance
4. How do we migrate shard
5. Balancing
6. Testing Infrastructure for automations
Terminology
Terminology

What’s a instance?
What’s a shard?
What’s a replicaset?
Instance

foobar.prn:3307
foobar.prn:3309
foobar.prn:3306
Shard

db.helloworld

db.12345

db.44365
## Service Discovery

<table>
<thead>
<tr>
<th>Shard ID</th>
<th>Replicaset</th>
<th>Master</th>
<th>Slave</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-99</td>
<td>Replicaset 1</td>
<td>db1234.prn1:3306</td>
<td>db1234.frc1:3306</td>
</tr>
<tr>
<td>100-199</td>
<td>Replicaset 2</td>
<td>db4567.ftw1:3306</td>
<td>db4567.prn1:3307</td>
</tr>
<tr>
<td>200-299</td>
<td>Replicaset 3</td>
<td>db1234.atn1:3306</td>
<td>db1234.frc2:3308</td>
</tr>
</tbody>
</table>
## Service Discovery

<table>
<thead>
<tr>
<th>Shard ID</th>
<th>Replicaset</th>
<th>Master</th>
<th>Slave</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-99</td>
<td>Replicaset 1</td>
<td>db1234.prn1:3306</td>
<td>db1234.frc1:3306</td>
</tr>
<tr>
<td>100-199</td>
<td>Replicaset 2</td>
<td>db4567.ftw1:3306</td>
<td>db1234.prn1:3309</td>
</tr>
<tr>
<td>200-299</td>
<td>Replicaset 3</td>
<td>db1234.frc2:3308</td>
<td>db1234.atn1:3309</td>
</tr>
</tbody>
</table>
Lifecycle of an instance
Lifecycle of an instance

- States are production, spare, spare allocated, spare deallocated, reimage, drained
- Metadata includes instance properties like name, port, mysql rpm version, state etc
- A mysql shard hosts metadata of all instances in the fleet
Lifecycle of an instance

- Reimage
- Pre-prod checks success
- Production
- Drained
- Spare deallocated
- Prod checks fail
- Undrain
- Spare allocated
- Allocated for copy
- Spare
- Copy success
- Copy failed
- Copy success
- Production
- Copy success
Lifecycle of an instance

- Each state has its own processor to do the work
- Each state has a queue where work is queued
- Runs constantly scanning the fleet
Instance Migration
Clone an instance

Use case of cloning a production MySQL instance
- Replace a broken instance/host
- Move data around for maintenance
- Balancing host utilization
A workflow system that manages the requests for cloning MySQL instances

- spare allocation
- set up MySQL config
- copy data
- replication
- validation
- bring it online & remove the old instance if necessary
MPS Copy - Allocation

Choose the best slot for the instance based on its footprint

- Disk usage
- CPU utilization
- Failure domain
MPS Copy - Setup

Turn up an empty instance using the right configuration
- Install the right RPM version
- Bootstrap the correct directory
- Generate the right my.cnf based on its use case
- Make sure the empty instance is connectable
We support three different ways of cloning a production instance

- Physical copy: xtrabackup and myrocks_hotbackup
- Logical copy:
  - mysqldump
  - Restore from backup
MPS Copy - Replication

- Setup replication
  - From current production master
  - From Binlog Server
- Catchup
MPS Copy - Validation

If the data migration is a logical one, we will use snapshot based checksum to verify the correctness of data by comparing to its current master.
Register the new instance in our service discovery system so that the MySQL users will be able to notice this new instance that has been recently turned up.
Online Shard Migration
Online Shard Migration

Another fundamental piece of our infra to control the growth of each MySQL instance

- Instance can grow beyond the host level limit
  - Too big
  - Too hot
Online Shard Migration (OLM)

Key concept: Move the data of a shard into other smaller/cooler instance through logical migration and register the new address into the service discovery system
OLM

replicaset

instance

host
OLM

<table>
<thead>
<tr>
<th>Shard</th>
<th>Replicaset</th>
</tr>
</thead>
<tbody>
<tr>
<td>db.1</td>
<td>mysql.replicaset.2</td>
</tr>
<tr>
<td>db.2</td>
<td>mysql.replicaset.1</td>
</tr>
<tr>
<td>db.3</td>
<td>mysql.replicaset.3</td>
</tr>
</tbody>
</table>
OLM Processor

Workflow management for massive OLM operations
- Conflict solver
- Picking the best destination replicaset
- Kickoff the actual move
- Proper retry and cleanup
Balancing
Balancing

Find the right place for the workload in order to achieve maximum sustainable resource utilization
Poor Stacking

Total: 1000G
Poor Stacking

- RS1: 400G
- Free: 100G

- Host 2
  - Free: 500G

- Host 3
  - Free: 500G

- RS1: 400G
Poor Stacking

- **Host 1**: Free: 500G
- **Host 2**: Free: 500G
- **Host 3**: Free: 500G
- **RS1**: 400G
- **RS2**: 300G
- **RS2**: 300G
- **Host 3**: Free: 500G
- **RS2**: 300G
Poor Stacking

Host 1
Free: 500G
RS1: 400G
RS2: 300G
RS3: 100G

Host 2
Free: 500G
RS3: 100G

Host 3
Free: 500G
Poor Stacking

RS1: 400G
RS2: 300G
RS3: 100G
RS4: 200G
Free: 100G
Free: 100G
Free: 300G
RS4: 200G
Proper Stacking

- RS3: 100G
- RS1: 400G
- RS2: 300G
- Host 3: Free: 500G
- Host 1: Free: 500G
- Host 2: Free: 500G
- RS3: 100G
Proper Stacking

- **RS3**: 100G
- **RS1**: 400G
- **RS4**: 200G
- **RS2**: 300G
- **Host 3**: Free: 500G
Carve the Shape

RS1: 400G
RS2: 200G
RS3: 200G
RS4: 200G

Total: 1000G

Host: 500G
Poor Shape

- RS1: 400G
  - Free: 100G
  - RS2: 200G
  - RS3: 200G
- RS4: 200G
  - Free: 300G

X3
Carve the Shape

Total: 1000G

RS1: 400G
RS2: 300G
RS3: 200G
RS4: 100G
Rebalancer

Goal: Find the best slot for hosting the given workload profile and reduce the imbalance score across the fleet to be minimum
Rebalancer - Challenges

Multiple balancing factors
- CPU/Memory/Disk usage
- Fault domain spreading
- MySQL vs LBU anti affinity
Testing Infrastructure
Testing Infra for Automations

- Lots of Automation code handling critical components of Infra
- UnitTests are good but mock backend connection
- Need to test end to end
Testing Infra Goals

- build and canary packages based on the change
- provide signals at diff time for developer
- production like setup, but isolated environment
- iterate quickly with confidence
Run the test code

Testing Infra Service

deallocate

Virtual test assets

metadata

dev

metadata access

deallocate & refresh

allocate/deallocate

Diff

test signal

developer

trigger test

prod

snapshot
Q&A