PERCONA

MongoDB Backups, All Grown up!

David Murphy

David Murphy

MongoDB Practice Manager for Percona

Past highlights

- MySQL & NoSQL Architect for Electronic Arts
- Original and Lead DBA for Objectrocket, the performance DBaaS for MongoDB
- Mongo Master Alumni and Core contributor
- 15+ previous years or experience in RDBMS' including write patches and plugins for MySQL.

Agenda

Typical types of backups

Complications when sharding

Consistent backups are easier in 3.2

New backup tool from Percona Labs

PSMDB providing free and open source hot backups

PERCONA

Typical Types of Backups

Looking at single node and replica set backups. What works, what doesn't and in what cases?

The Logical Backup

Almost always this will have the word **dump** in the tool name.

Mongodump

- Will connect to
  - The node you're connected to
  - If replset, will connect to random secondary
  - If mongos connects to **PRIMARIES** more on this later around sharding
- Single mongod nodes are impossible to get a non-blocking consistent backup!
- Restores take a long time as they must reconstitute the data files and build index structures

Binary Backups: LVM Snapshots

iSCSI/NFS take the LVM method and improve things:

- Backup are 100% size, but de-duplication may help
- No hydrations or copying
- MongoDB historically performed badly on network storage
  - Should not use MMAP with NFS, others may be OK with tuning

Netapp's work wonders with:

- Single request ACK's to NVRAM
- Cluster groups to manage disk QOS
- Hourly incremental snapshots

Binary Backups: iSCSI/NFS Snapshots

LVM comes with some considerations:

- Backup will be 100% the size of the real data
- No need to re-hydrate

Snapshot can be made instantly but have some requirements:

- Space must exist on the Volume Group
- COW will reduce performance while active

Restores are fast and consistent for a replSet or single node

MongoDB Ops Manager (MOM)

Automated backups with recovery from a closed source cloud provider platform.

Optionally can be installed in dedicated environments.

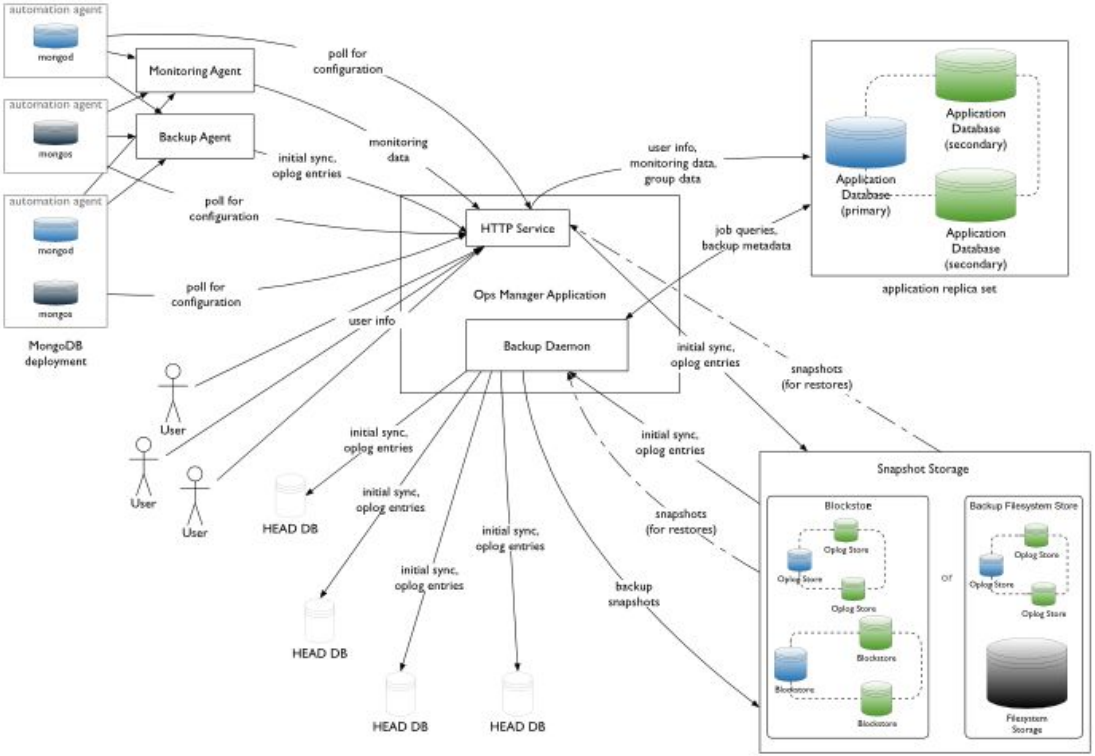
Part of the Ops Manager platform for Monitoring, configurations and backups

Types of backups:

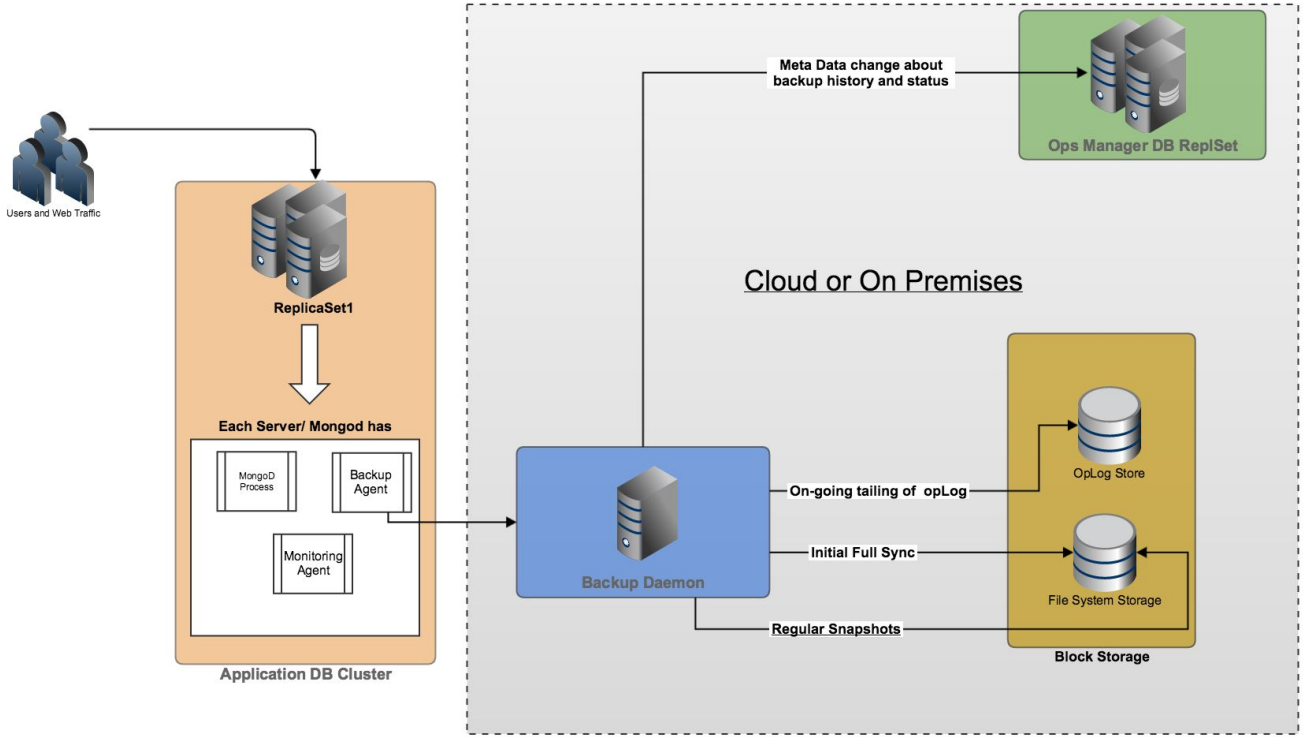
- Initial
- Incremental
- Snapshots

Backups cost 2.50$ USD per GB

Backups::MOM - Official Architecture



Backups::MOM - Simplified Architecture



Backups::MOM - Initial

The first step in all backup methods is the initial sync

PIT window: Cloud = 24 hours, On-Prem= up to a few days

- System start to tail to the oplog immediately
- It will take a 10MB slice of data at a time from each replset (if sharded), and ship it to the storage layer
- When all data is copied, the oplog applies occur to ensure the data is consistent to that point.
- Oplog applying is then paused, and the DB (which was cloned locally) is broken into smaller files and compressed.

Backups::MOM - PIT and Snapshots

Once the initial sync is done, these can start occurring.

Continuation of oplog applying starts again after initial finished.
Every 6 hours this is stopped for the incremental snapshot to occur:

- DB files are broken into smaller pieces just like the initial sync
- These chunks are then compressed and compared to the sync's
- If they are the same they are not kept, allowing for deduplication
- As mentioned before, oplog is kept from 24 hours to several days depending on the deployment
  - You can only do point in time recovery for a snapshot and time that is included in the oplog range limit

Backups::MOM - Sharding Special Notes

When doing a backup with a sharded collection, special things occur to help the system be consistent.

- When a snapshot is needed, the balancer is paused and a no-op entry is added to each replication stream, so the backup knows where to stop replicating to:
  - Reason: prior to 3.2, the config servers were not a replica-set and could not be treated like the shards.
- Point in time recovery is not supported for sharded setups, instead you need to enable checkpoints
  - Restore uses oplogs to the checkpoint to restore from the snapshot, however configs need to be fully copied at each checkpoint

PERCONA

Sharding Complications Recap

Sharding Complications: Consistency

Mongo shards will not all finish backups at the same time. This means some shards will finish before others. Without transactions, finding a consistent point is an issue.

Logical backups:

- Mongodumps via a mongos do not allow --oplog, and you can't get a consistent backup
- Make sure balancing is not occurring during the backup

Binary/Snapshot backups:

- Great in a single replica set, but missing oplog prevents consistency

A new tool is needed to handle and wrap both of the situations.

Sharding Complications: Tool Support

Quick break down of consistent sharded support limits per tool type:

- Logical backups:
  - No native support for multi-shard consistent backups
- Binary/Snapshot backups:
  - No support for multi-shard consistent backups
- MongoDB Ops Manager (MOM):
  - Only snapshot support is allowed, recovery of sharded clusters is NOT point in time

PERCONA

Mongo 3.2 Improvements

New replica set configs and what they mean for better backups

What's New to 3.2?

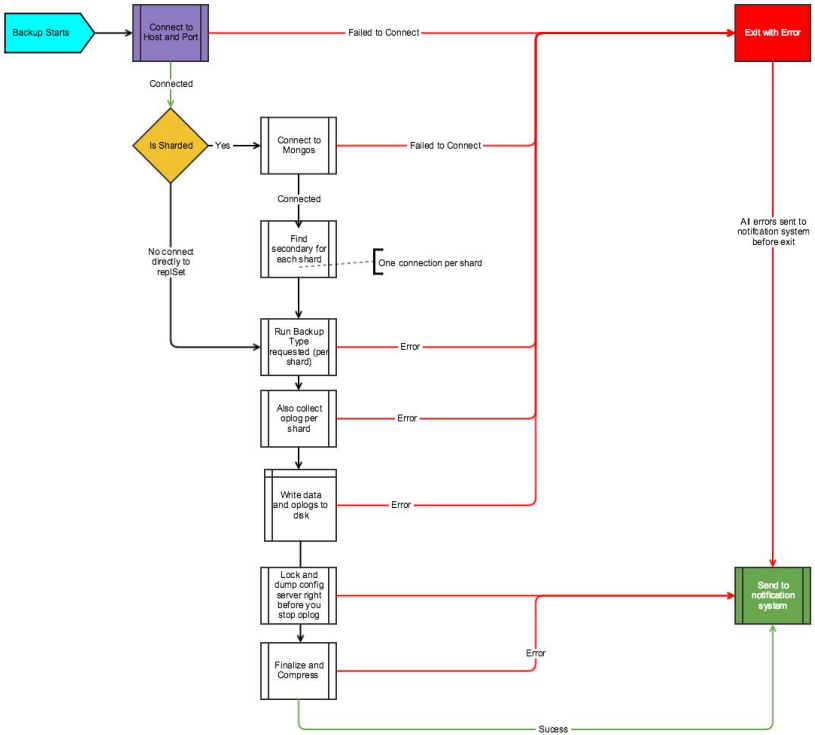
Mongo 3.2 was **HUGE** for backup and sharding consistency!

Some of the previous challenges:

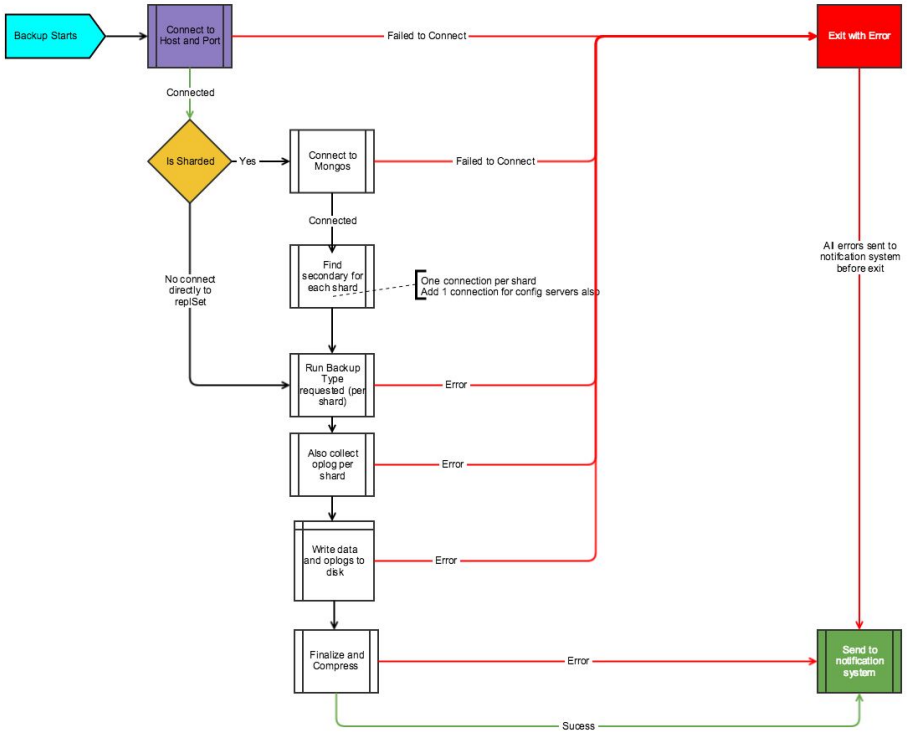
- Micro time variations when locking and backing up the configs servers could result in intermittent bad backups
- Any new database, sharding action or balancing action would error when config server was locked for a backup
- There was no good way to play the config server from a backup forward with cluster oplogs, so your recovery was limited by the last balancing actions.
- As a side note we can now trust config servers to not get out of sync like they used to!!!

PERCONA

Backups: Flow before 3.2



Backups: Flow after 3.2



PERCONA

Percona-Labs's Consistent Backups (MCB)

Why did we need it? What does it do? Where is it going?

MCB: Why?

MongoDB Ops Manager is great, but it's not free or open source.
We need a tool that is free to everyone and full open source

Even still, it had some limits

- Paid per GB of backup storage used
- Only keeps 24 hours of oplogs, then you can only restore the last backup
- Unable to tell it where you want the backup's to live
- Unable to tell it how to notify your monitoring systems on success/fail
- Unable to select the backup style you want
    - EBS/LVM/ISCSI snapshots and logical dumps

MCB: What does it do?

NOT officially supported by Percona (why it's Percona-Labs)

- Single Python binary (just needs Python 2.7 installed)
- Detects if talking to mongos/replicaset, errors if single mongod
- Mongos will run in recursive mode and keep all shards in sync
- Currently only supports dumps, but work being done for modular
- Understands both pre/post 3.2 config server types and will have them both in the right way
- Balancer is disabled and secondary used is health checked

MCB: Compared to MOM

MOM still does more are MCB matures over time

Currently things MCB does not handle directly:

- Continuous Oplog collection for Incremental
- Breaking data file into smaller files
- De-duplication between backups

These are things that it may or may not get, as there are many ways to provide deduplication and we choose compression.

Continuous Oplog is something we have in mind, however it should be bound not on cost but on space a user of it wants to give it.

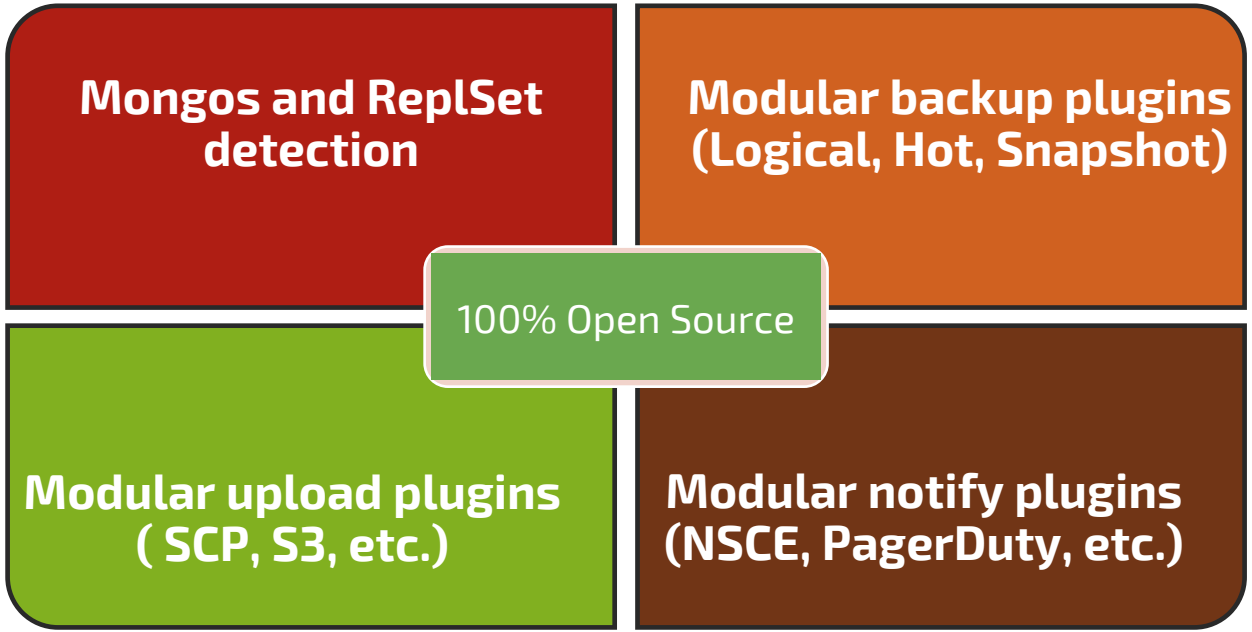
MCB: Where is it going?

Once mature, this tool will be folded into a more official Percona offering

Things we know it needs to do

1. Modularize and support
   a. Backup method
   b. Uploader method
   c. Notifier method
2. Have restore partner tool
3. Oplog Spooler daemon
4. Encryption
5. Filters (limit which DBs/collections)
6. Support for WT and RockDB hot-backups

Mongo Consistent Backup - Vision



MCB: Where is it found?

Just browse over to:

https://github.com/Percon-Lab/mongodb_consistent_backup

There you can clone or download the repo, the tool comes with a build script to build the single binary.

Simply build the tool on a similar machine (rhel7 x64 for example) and the sysadmins can put it on the final systems. (They won't need anything installed except Python 2.7, that will make them happy!)

It's GPL, so feel free to use it and contribute. The more interest we see in any issues, the higher it will be prioritized.

PERCONA

WiredTiger Open Source Hot Backup

PERCONA

Hot-Backup in Percona Server MongoDB

In Percona Server we have two methods for hot backups

For either engine you can now run:

```
db.adminCommand({createBackup: 1, backupDir: backupPath})
```

We kept the classic MongoRocks command for now, but the plan is to remove it in 3.4

Run:

```
db.adminCommand({setParameter:1, rocksdbBackup: backupPath})
```

PERCONA

Mongo Consistent Backup

Consistent Backup Github: http://bit.ly/28InDul
Questions?

What features do you think are missing to make backups more mature?

Other tools you think the team should focus on?

Contact Info:

Twitter	@dmurphy_data @percona
Github	dbmurphy percona_lab percona