Filtered Replication POC

Skunkworks October 2016

Contributors

- Kevin Pulo
  - Senior Technical Services Engineer, Diagnostics and Defect Resolution
  - Sydney Office
  - At MongoDB for 3 years
- Sulabh Mahajan
  - Database Server Engineer (Storage/WiredTiger team)
  - Sydney Office
  - At MongoDB for 5 months

What

Filtered Replication / Partial Replication / Subset Replication

=

Secondary can replicate only some databases/collections

- Oplog is always whole
- No-op operations which are on excluded namespaces
  - Initial sync, oplog tailing, and rollback
- Writes never count for write concern
- No initial sync / rollback from filtered members
  - But tailing oplog from filtered members is fine

Why

- Consensus "arbiters" (arbiter with oplog)
  - Helps to avoid rollbacks
- Authable "arbiters"
  - Just replicate admin db and nothing else

- Node with massive oplog (but no data) for extra/emergency outage coverage
- Seed staging environment from prod, but without sensitive data
- Analytics secondary which is smaller/lower-powered
- Non-disjoint data partitioning (unlike sharding)
- Fancy stuff like replset/shard mitosis
- Help to nuke master-slave from orbit

```
{   "_id" : 1,
    "host" : "basique:12346",
    "arbiterOnly" : false,
    "buildIndexes" : true,
    "hidden" : true,
    "priority" : 0,
    "tags" : { },
    "slaveDelay" : NumberLong(0),
    "votes" : 1,
    "filter" : [   // namespace whitelist
        "admin",
        "wholedb",
        "partialdb.somecollection"
    ]
}
```

```
{   "_id" : 1,
    "host" : "basique:12346",
    "arbiterOnly" : false,    // MANDATORY
    "buildIndexes" : true,
    "hidden" : true,          // MANDATORY (for now)
    "priority" : 0,           // MANDATORY
    "tags" : { },
    "slaveDelay" : NumberLong(0),
    "votes" : 1,
    "filter" : [   // namespace whitelist
        "admin",   // MANDATORY
        "wholedb",
        "partialdb.somecollection"
    ]
}
```

```
{
    "_id" : "replset",
    "version" : 1,
    "configsvr" : false,                  // MANDATORY
    "protocolVersion" : NumberLong(1),    // PROBABLY MANDATORY
    "members" : [
        ...
```

- No changing filter rules on reconfig (except on new member)
- No system collections
- No local db

Demo

After Rollback

Documents on unfiltered member:

```
{ "ns" : "excluded.excluded", "n" : 0 },
{ "ns" : "excluded.excluded", "n" : 2 }

{ "ns" : "partial.excluded", "n" : 0 },
{ "ns" : "partial.excluded", "n" : 2 }

{ "ns" : "included.included", "n" : 0 },
{ "ns" : "included.included", "n" : 2 }

{ "ns" : "partial.included", "n" : 0 },
{ "ns" : "partial.included", "n" : 2 }
```

Documents on filtered member:

```
{ "ns" : "included.included", "n" : 0 },
{ "ns" : "included.included", "n" : 2 }

{ "ns" : "partial.included", "n" : 0 },
{ "ns" : "partial.included", "n" : 2 }
```

Lessons learnt

- Lots of edge cases
- Judicious

  ```
  if (!isNamespaceReplicated(ns))
      continue;
  ```

  will get you a long way
- jstests are a double-edged sword for skunkworks

Where to from here?

- Many possible future scopes of various degrees
    - Proper filtering rules (ordered mixed white/black list, regexes)
    - Don't even give ops on excluded namespaces to the repl writer worker threads
    - Allow hidden: false (advertise filters in isMaster, drivers use when directing queries)
    - Allow changing filters on reconfig (drop newly excluded namespaces, mini-initial sync to get newly included namespaces
    - Allow initial sync from filtered member which has a superset of my namespaces
    - Allow initial sync from several filtered members which together have all namespaces

- Possible project for MongoDB 3.6?
- https://github.com/devkev/mongo/tree/filter1
- These slides