MongoDB surviving after unclean shutdowns
About Me

• Senior Database Engineer
• 15 Years of experience with data
• Started with Relational Databases in Sql Server 7.0
• Today I work in Brazil helping DBACorp’ customers get the best results.
• Over the Years have worked with:
  • MongoDB
  • SQL Server
  • Cassandra
  • Redis
  • Kafka

Alexandre Araujo
alexandre.araujo@dbacorp.com.br
@aleraraujo16
linkedin.com/in/alexandrearaujo16/
Agenda

• MongoDB internally has a powerful mechanism to face unexpected interruptions.

• We will talk about:
  • What WAL protocols means and how it was implemented to the MongoDB
  • How works the journal process
  • What kind of disasters impacts the MongoDB database
  • Why we have must concern about points of failure
  • When recover process takes action
  • And some configurations and strategies that you should use in your deploy.
After World War II

Magnetic Tape

Magnetic Disk

After Cold War

Oracle Database

Microsoft SQL Server

MySQL

PostgreSQL

Cloud and Data Explosion

mongoDB

aws

kafka

kubernetes
MongoDB Storage Engines

- Storage Engine is responsible for managing how data is stored.
- MongoDB provides a variety of Storage engines
- You can choose the appropriate Storage engine according to your deploy.
- WiredTiger is the default Storage engine since MongoDB 3.2.
WiredTiger Storage Engine

- WiredTiger uses snappy compression by default with up to 80% compression
- The WiredTiger in MongoDB uses C-API although exists WiredTiger in Python and Java.
- WT takes advantage of modern hardware and more performance between threads
- Eliminate blocking due to concurrency control using MVCC
- Between 7 and 10x better write throughput from older storages
- Documentation: https://source.wiredtiger.com/10.0.0/index.html
WiredTiger - Architecture
# WiredTiger Storage Engine

**Durability Mode**  
Three Modes

<table>
<thead>
<tr>
<th>Durability Mode</th>
<th>MongoDB Crash</th>
<th>OS Crash</th>
</tr>
</thead>
<tbody>
<tr>
<td>In-Memory</td>
<td>Potential data Loss</td>
<td>Potential data Loss</td>
</tr>
<tr>
<td>Write-No-Sync</td>
<td>Data Always Recoverable</td>
<td>Potential data Loss</td>
</tr>
<tr>
<td>FULL SYNC</td>
<td>Data Always Recoverable</td>
<td>Data Always Recoverable</td>
</tr>
</tbody>
</table>

Two modes of durability in MongoDB
WAL Protocol

- Write Ahead Logging protocol, most well-know recovery method, is the standard industry ensures that a record of every change to the database is available while attempting to recover from a crash.
- When a data is changed and committed it is forced do stable Storage.
- Traditional databases use a write-ahead log for recovery.
- In the ACID systems WAL generally refers to Durability.
A data journey from user writes until commit

1. Send code from apps
2. MongoDB Query Language
3. MongoDB Data Model
4. WiredTiger
5. Checkpoint + Journaling
WAL – Write Ahead Logging

1. Updates e Inserts
2. Flush to disk
3. Write Commit
4. Checkpoint

Volatile Memory

Non Volatile Memory

Write Operations

Write Buffering Memory

Journal Files

Collection Files
• WT creates one journal record for each write and index operation
• Max size limit is 100 megabytes and the minimum journal record size is 128 bytes
• This is not the Replication OpLog also it is not user-level transactions
• Automatically removes old journal files to maintain only the files needed to recover from last checkpoint
Journalism - Structure

- Location at: dbpath/jornal/WiredTigerLog.00000000001
- A record per each client initiated write operation.
- WiredTiger creates a single jornal record that includes both the update operation and its associated index modifications.
- Each record has a unique identifier.
Checkpoints

- Internal WiredTiger process
- Checkpoints flush dirty data at every 60 seconds creating stable data
- The consistent point-in-time snapshot of data writes all in memory to disk and ensures there is no point in time where data might be lost.
- Checkpoint will mark inside the journal indicating the last checkpoint
Journalism occurs at every 100 ms

1. Everytime a transaction commit
2. Wired Tiger Checkpoint API
3. WT_LOGGING Process
4. WT API Call Storage Function
5. WT Memory Process
6. Write to Journal Files
Checkpoints occur at every 60 ms

1. Occurs Internal Threads – 60ms
2. Wired Tiger Checkpoint API
3. Finding Dirty Data in WT_MEMORY
4. Open Snapshot Transaction
5. Write to collect and index
Failure Scenarios

- What happens after a crash:
  - The data is lost but the file system is consistent
  - The log has exactly the right information to fix the problem

- Kinds of failure:
  - Hardware Failure
  - Power Failure
  - Storage Volume Failure
  - Kill unexpected mongod process
  - Unexpected restart S.O.
Recovering – How mongod Recovers from crash

• Faster recovery = checkpoint + journal
• First check data files to find the id of the last checkpoint
• Then check the journal files for the record that matches of last checkpoint
• And finally applies the journal files since the last checkpoint
• MongoDB does not guarantee storage failure after a checkpoint.
Recovery Process

WRITE OPERATION WT_MEMORY
- Seahawks
- Panthers
- Packers
- Steelers
- Eagles

COMMIT DATA JOURNAL
- Seahawks
- Panthers
- Packers
- Checkpoint
- Eagles
- Crash
- Recovery

COLLECTION FILES
- Seahawks
- Panthers
- Packers
- Checkpoint
- Steelers
- Eagles
• Performance is a good indicator for use MongoDB in Memory
• It become durable when filling the memory buffer
• Become durable when a new journal is created
• Idle systems can trigger the write journal every 50ms
• Crashes does not guarantee data after last checkpoint
Tuning

- Change configuration option “commitIntervalMs” according to your workloads.
- The journal default value is 100 ms and the range can be between 1 and 500 ms.
- Use symlink for journal to a different hub increasing the concurrency of your write deploy.
- Big checkpoints can cause painful performance and you can control the WiredTiger CacheSize limiting the amount of dirty pages and sizing eviction threads.
- TRADE-OFFS

  - More frequent checkpoints mean less record that you reply and more faster your recovery although more intense use of your storage.
  - Less frequent checkpoints mean more record that you reply and more longer will be your recovery process.
Open Source is the new oil.
Thanks!

Any questions?

You can find me at:

- @alerarauido16 - Twitter
- alexandre.araujo@dbacorp.com.br
THANK YOU!