



传说中，MongoDB会丢数据

— 高可用分布式集群的数据安全





1

MongoDB数据安全架构

2

数据丢失场景及分析

3

MongoDB 3.2 新功能预览





关于我

TJ Tang 唐建法

MongoDB 大中华区技术顾问

原联邦快递首席架构师

开源项目anoogose作者





mongodb 丢数据



百度一下

网页

新闻

贴吧

知道

音乐

图片

视频

地图

文库

更多»

百度为您找到相关结果约82,600个

搜索工具

[为什么MongoDB会丢数据 - longxibendi的专栏 - 博客频道 - CSDN.NET](#)

2014年10月8日 - MongoDB 丢数据的说法已经出现很久很久了,传言甚多。这里简单总结下场
景。 1.在MongoDB很早的版本,2.0之前,没有journal,加上默认不是安全写,系统一宕...

[blog.csdn.net/longxibe...](#) - 百度快照 - 88%好评

[别再用MongoDB了!](#)

2015年7月22日 - 近日,他在个人博客上发表了一篇博文《为什么你应该永远、永远、永远不要
再使用MongoDB》。在文中,他列举了如下理由:丢失数据(见1、2); 默认忽略...

[www.infoq.com/cn/news/...](#) - 百度快照 - 89%好评

[千万别用MongoDB?真的吗? - 博客 - 伯乐在线](#)

2011年11月10日 - 某人发了一篇《Don't use MongoDB》的血泪控诉,我把原文翻译如下,你可
以看看...3.主从结点间的数据复制有缺口,导致从结点丢失主结点有的数据。是的,...

[blog.jobbole.com/5701/](#) - 百度快照 - 评价

[MongoDB数据丢失问题-CSDN论坛-CSDN.NET-中国最大的IT技术社区](#)

4条回复 - 发帖时间: 2014年12月17日

2014年12月17日 - MongoDB数据丢失 我是做前端的,这两天碰到一个问题:我发送数据给后台,
后台给我的提示是成功的(查看连接请求的状态是成功的、网页返回也是成功的),因...

[bbs.csdn.net/topics/39...](#) - 百度快照 - 88%好评



Atomicity
Consistency
Isolation
DURABILITY



WIKIPEDIA
The Free Encyclopedia

... guarantees that transactions that have committed will survive permanently...



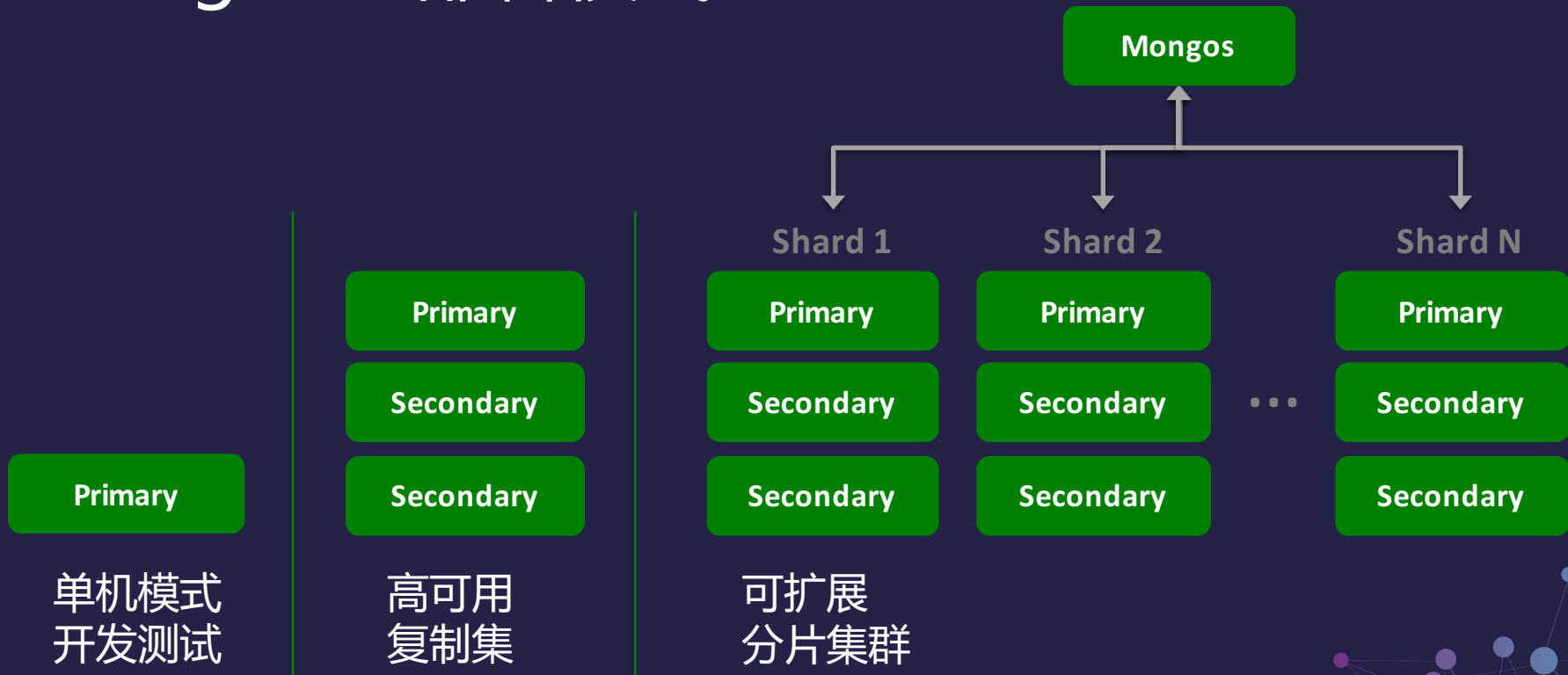


MongoDB 数据持久化设计



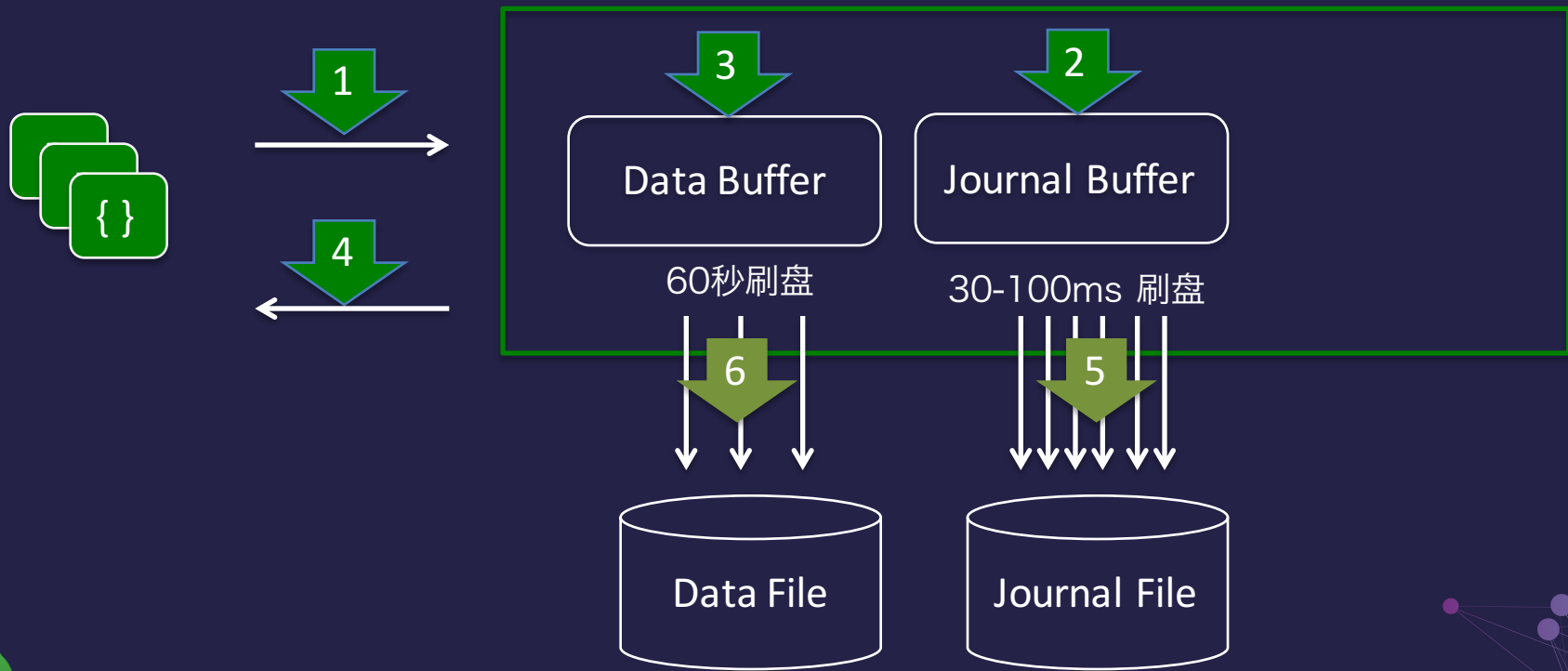


MongoDB 部署模式



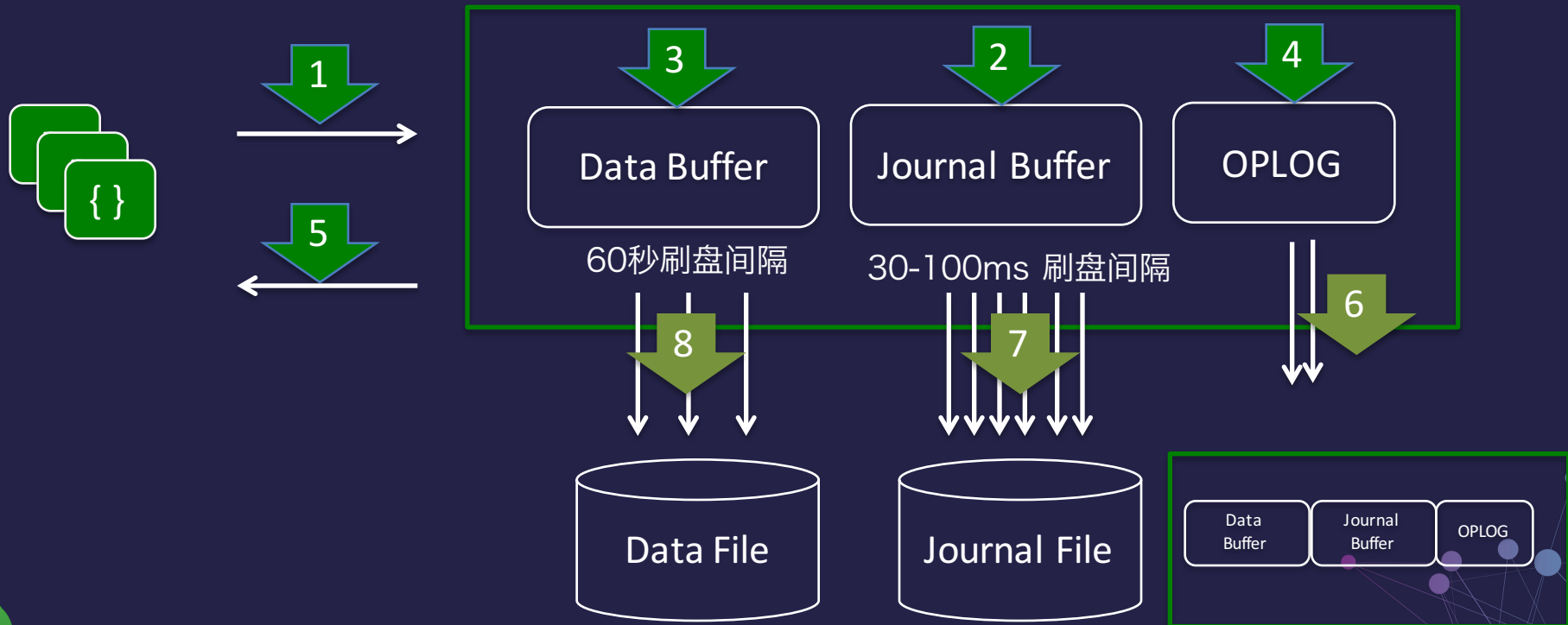


MongoDB 的单节点写操作





MongoDB 的复制集写操作





恢复日志 Journal

- 用于系统宕机时恢复内存数据
- 默认为异步刷盘
- 刷盘间隔：
 - MMAP : 100ms, 30ms if {j:1}
 - WiredTiger : 100MB or Checkpoint
- 可使用 j:1 来强制刷日志



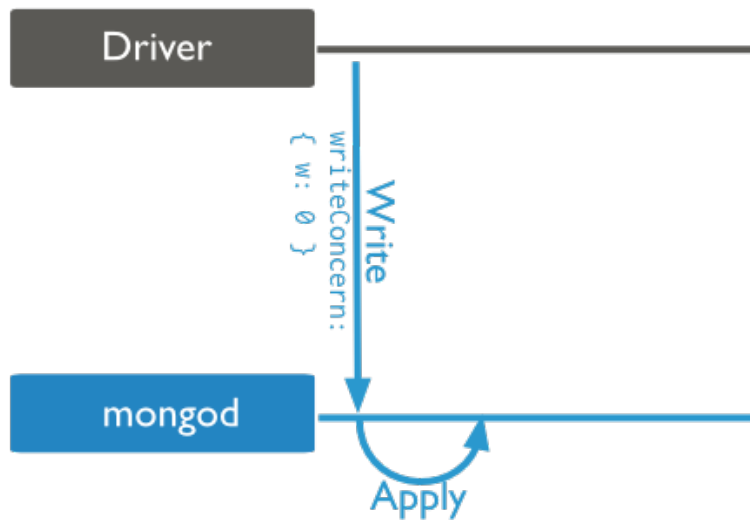


写关注机制 Write Concern

- 用来指定mongod对写操作的回执行为
- 可在connection level或者写操作level指定
- 支持以下值：
 - w: 0 | 1 | n | majority | tag
 - j: 1
 - wtimeout: millis

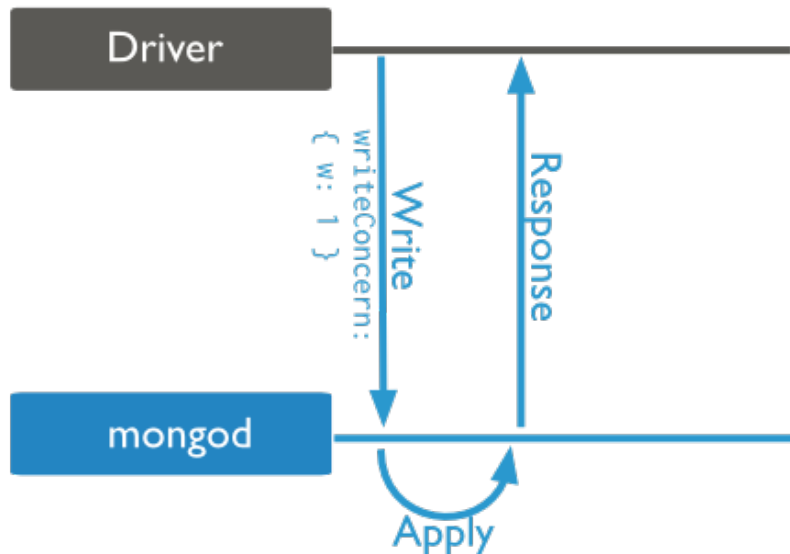


w:0 Unacknowledged



- 无任何回执
- 2.2 及之前版本默认行为
- 网络丢包、系统崩溃、无效数据
- 早期版本丢数据之罪魁祸首

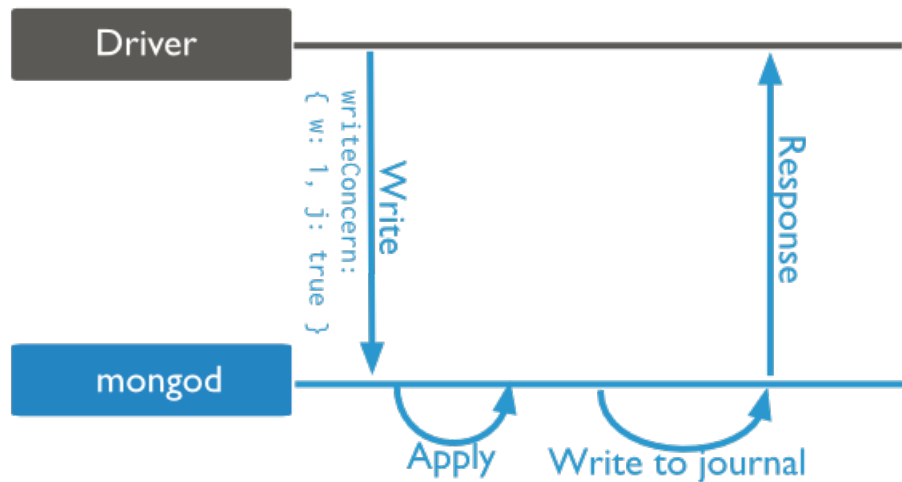
w:1 Acknowledged



- Mongod在写完内存后发送确认
- 2.4之后默认行为
- 能够处理网络故障、无效数据等错误状态
- 系统崩溃时可能会丢失最多100ms数据



j:1 Journalled

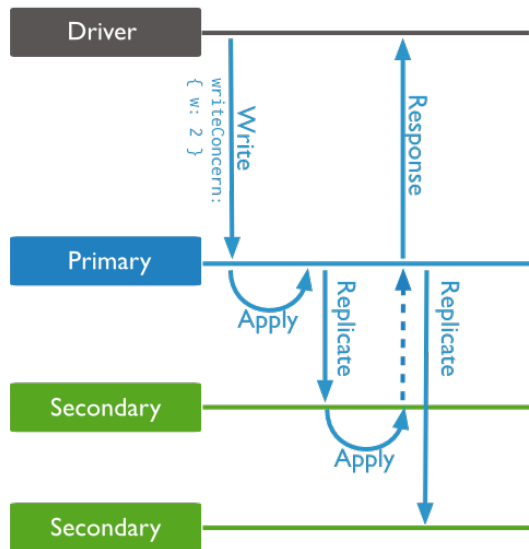


- Journal刷盘之后再发送写回执
- 30ms 间隔Group Commit – 单个请求可能会等最多30ms才返回

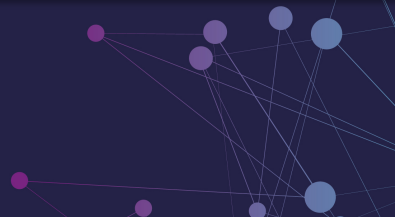
w:2/n/majority Replica Acknowledged



2015 杭州·云栖大会
COMPUTING CONFERENCE



- 等待数据复制到n个节点再发送回执





数据丢失场景分析





场景A：未使用写关注

重现

使用MongoDB 2.2，或者，
指定 {w:0} 写关注

插入一些无效数据，如10个文档
同一个 _id

检查实际插入数据数目

解决方案

升级到2.4+
指定 {w:1} 写关注



场景B：未使用j:1 系统崩溃

重现

使用MongoDB 3.0

W:1 高速持续写入数据

Kill -9 mongod

检查程序汇报写入的数据和
实际插入的数据

解决方案

使用 j:1



场景C：未使用 w: majority

重现

使用MongoDB 3.0 复制集

w:1, j:1 高速持续写入数据

Kill -9 mongod 主节点

检查程序汇报写入的数据和
实际插入的数据

解决方案

使用 w: majority



MongoDB 3.2 功能预览



Misc Agg

Read Committed

dump/restore compression

Data Replication Refactoring

Oplog Replication Refactor

Encryption at Rest

Bitwise query ops

V8 -> SpiderMonkey

Networking Layer

open source evergreen

Raft

Partial Index

General Purpose Auth

clang-format

Read Committed



3.2



In Memory SE

dump/restore archiving

LDAP

driver handshake

Collection Sampling

Doc Validation



WT as Default

Per index stats

Read after Optime

Config Servers as Replica Set

"Black Box" data file

Journaling Perf in WT

Oplog perf



Q&A



MongoDB中文社区 <http://www.mongoinc.com>

MongoDB在线课程 <http://university.mongodb.com>

jianfa.tang@mongodb.com