MongoDB 中文社区

IT大咖说 知识分享平台

MongoDB在腾讯网媒产品中的应用

腾讯网 腾讯视频 天天快报

周奇

2017 05/20

腾讯网 qq.com

目录

腾讯网媒产品





2017一季度中国视频网站类App排行榜

排名
1
2
3
4
5
6
7
8
9

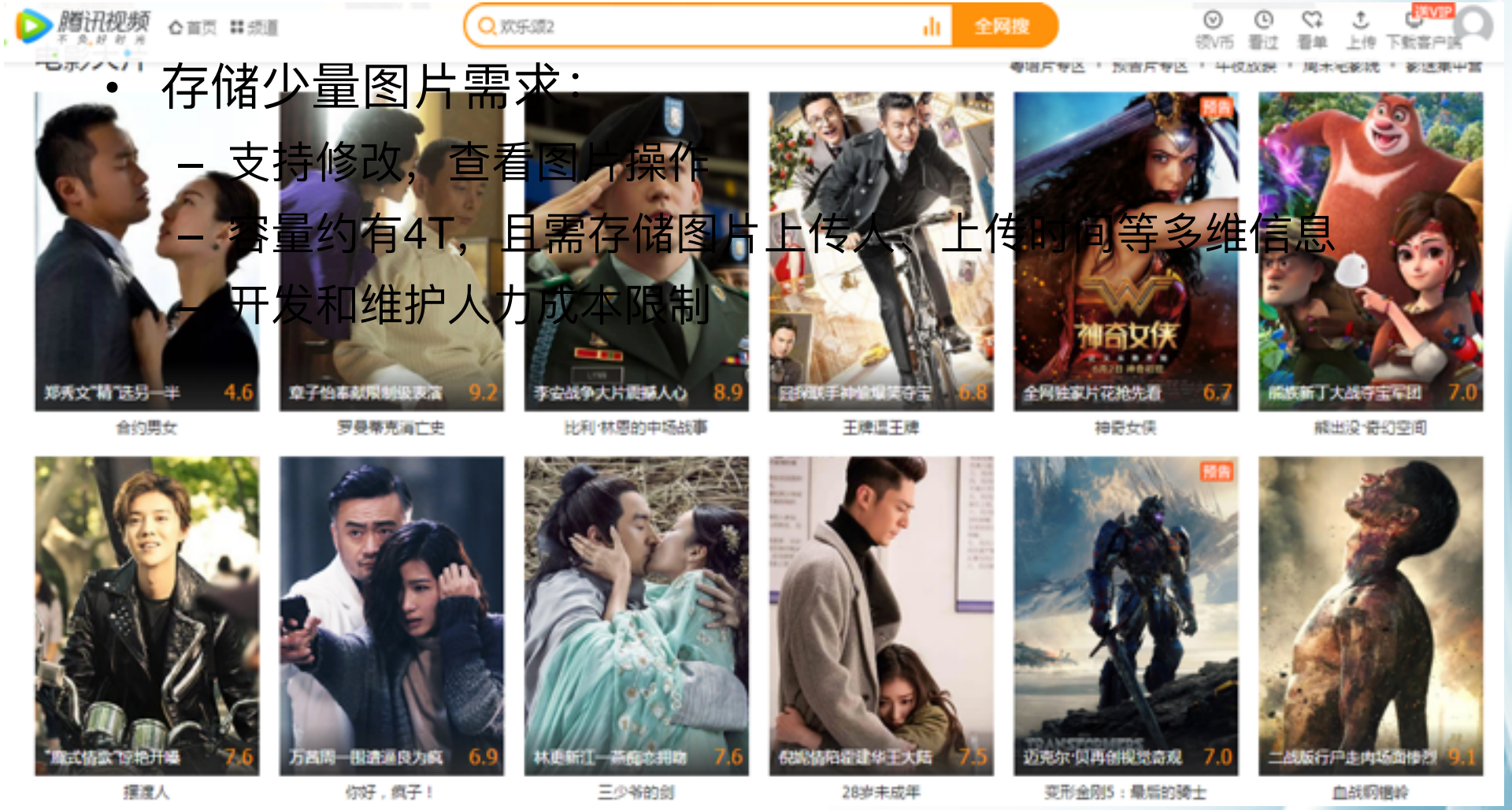
中国新闻AppTop10分类

排名	应用	类别	活跃渗透率
1	今日头条	聚合	15.13%
2	腾讯新闻	门户	10.11%
3	天天快报	聚合	4.61%
4	一点资讯	聚合	3.47%
5	搜狐新闻	门户	1.82%
6	网易新闻	门户	1.49%
7	凤凰新闻	门户	0.72%
8	新浪新闻	门户	0.64%

1 消息推送场景

- 消息推送场景
  - 需尽快完成推送，存储写入并发高，峰值30wqps
  - 数据量比较大，日均新增600G
  - 运营统计分析需求，查询纬度较多

2 简单的图片存储系统



- 存储少量图片需求：
  - 支持修改，查看图片操作
  - 容量约有4T，且需存储图片上传人、上传时间等多维信息
  - 开发和维护人力成本限制

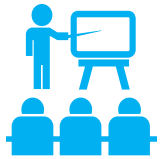
3 大容量存储场景

- 用户评论数据存储：
  - 存储量大，月均400G，存储成本高
  - 之前业务在MySQL按月分库分表，开发代码成本高

4 模式变更频繁场景

- 抓取文章数据存储
  - 频繁变更库表结构

MongoDB托管现状-运营



产品接入

视频-好莱坞消息系统
视频-图片平台
新闻-PUSH系统
客厅-用户标签

运营质量

数据量 31T
日请求量 120亿
集群数 78套
实例数 234

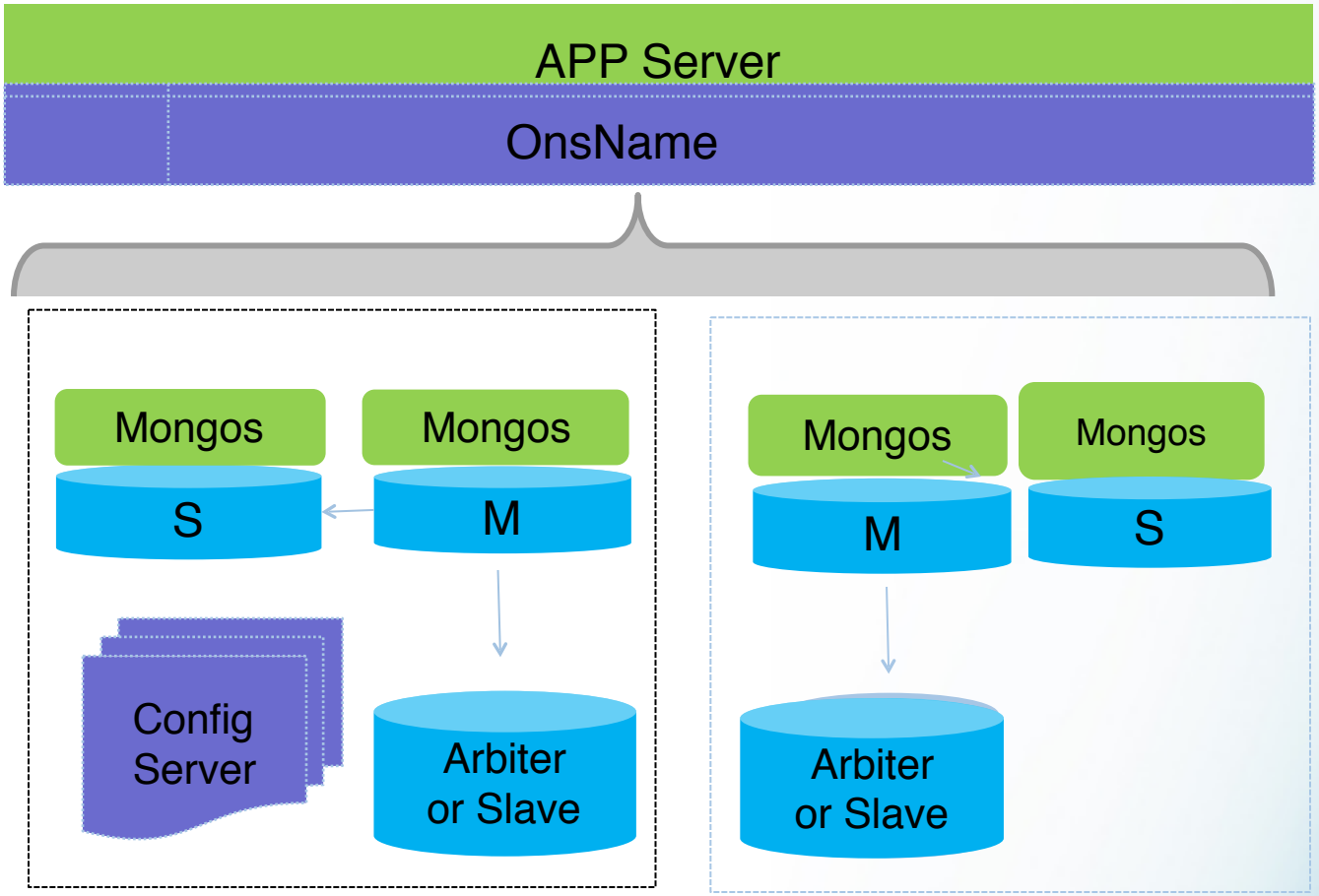
平台建设

名字服务接入
一键部署
实时监控
异常告警
数据备份

MongoDB托管现状-平台

- 高可用：
  - 通过名字服务接入，保证mongos层变更透明；mongod层依赖复制集自身高可用
- 监控告警：
  - 分别mongos和server层有30s粒度监控，慢查询统一上报pt工具分析。
- 备份恢复：
  - 通过mongodump全量和oplog增量备份保证复制集数据备份安全，分片环境暂不支持。

MongoDB托管现状-拓扑



Mongo“修炼之路”

- Mongo适用的场景
- Mongo使用的限制

- 如何正确访问Mongo集群？Mongo驱动使用注意点
- 如何搭建可靠的Mongo集群？系统和server参数调优

- Mongo性能管理
- 谨慎使用Sharding
- 数据备份恢复

1．Mongo适用的场景

- 存储产品选型
  - 需求分析、存储格式、访问分析需求、架构（高可用易扩展）、成本（硬件、开发，服务维护）
- MongoDB是高性能、高可用、易扩展的分布式文档型数据库
- 在线高性能吞吐，大数据量，松散数据结构的场景
  - 用户基础信息存储、业务流水记录、物联网监控信息存储
  - 小型图片服务器、GIS数据库、分布式文件系统

2．Mongo使用的限制

- 命名空间
  - dbname不区分大小写，collname区分大小写
  - 下划线和字母开头，不能以数字、$、空字符串
- 索引
  - 单集合索引最多64（注意数组子文档和多列存储方式）
  - 复合索引最多31列
  - 索引名字不能超过128字符包括. 复合索引要注意
- 数据
  - bson单个文档最大16M
  - 单文档嵌套最多100层
  - WT单库内coll默认无限制（元数据管理、文件句柄）

3．Mongo驱动使用

- writeconcert设置为 1

WriteConcern	含义
{ w : 0 }	写入不需要server端确认 适合批量写入，不关心正确性场景

readPreference	含义
primary	默认行为，都从主读取
primaryPreferred	优先从主读取，无主时候从Secondary读取
secondary	读操作只在从节点， 如果从节点不可用，报错或者抛出异常
secondaryPreferred	优先从Secondary读取，无从节点可用时候从主节点读取
nearest	从临近节点读取（主或者从）

4．系统和Server参数调优

- 系统：
  - raid 5 or raid 10，no raid
  - 使用ext4或者xfs文件系统，关闭atime
  - 关闭numa、关闭hugepage（进程需重启）
  - 设置最大openfile限制（进程需重启）

- server：
  - 控制cacheSize（系统层面设置vm.overcomit_memory=2，控制内存）
  - 进程用非root用户启动
  - 控制最大连接数5000
  - 关闭索引重建选项indexBuildRetry:false

我们的配置文件

#mongod.cnf

```
systemLog:
   destination: file
   path: "/data1/mongod/27000/log/mongod.log"
   logAppend: true
   logRotate: "rename"
   timeStampFormat: "iso8601-local"
   verbosity: 0
   #component:
   #   query:
   #      verbosity: 1
   #   storage:
   #      verbosity: 1
   #      journal:
   #         verbosity: 1
storage:
   dbPath: "/data1/mongod/27000/data"
   directoryPerDB: true
   syncPeriodSecs: 60
   journal:
      enabled: true
      commitIntervalMs: 100
   engine: wiredTiger
   wiredTiger:
      engineConfig:
         cacheSizeGB: 12
processManagement:
   fork: true
   pidFilePath: "/data1/mongod/27000/var/mongod.pid"
net:
   #bindIp: 0.0.0.0
   port: 27000
   http:
      enabled: false
   maxIncomingConnections: 5000
operationProfiling:
   slowOpThresholdMs: 100
   mode: "slowOp"
replication:
   oplogSizeMB: 51200
   replSetName: "80000095"
sharding:
   clusterRole: "shardsvr"

security:
   keyFile: "/data1/mongod/27000/mongod.key"
   clusterAuthMode: "keyFile"
   authorization: "enabled"
```

5．Mongo性能管理

- 动态设置慢查询size
  - 先drop，再create
  - db.createCollection("system.profile",{capped:true, size: 100000000})
- pt工具分析满查询
  - pt-mongodb-query-digest
- 定期rotatelog
  - db.runCommand({logRotate:1})
- killOp满查询
  - db.currentOp().inprog.forEach(function(item){if(item.secs_running > 200)db.killOp(item.opid)})

6．谨慎使用sharding

- 为什么shard?
  - 磁盘空间问题
  - 内存不足
  - 扩展写入

```
344+0800 I SHARDING [Balancer] moveChunk result: { chunkTooBig: true, estimatedChunkSize: 49877982, ok: 0.0, errmsg: "chunk too big to move" }
376+0800 I ASIO [NetworkInterfaceASIO-ShardRegistry-0] Successfully connected to 10.240.111.92:26000
379+0800 I SHARDING [Balancer] ChunkManager: time to load chunks for PushHistory.push_date_20170513: 34ms sequenceNumber: 52 version: 59|6267
3a8451d based on: 59|6153||5915d4a38c312cb913a8451d
381+0800 I SHARDING [Balancer] balancer move failed: { chunkTooBig: true, estimatedChunkSize: 49877982, ok: 0.0, errmsg: "chunk too big to move" }
000095_2 chunk: min: { devid: -7478064656446626857 } max: { devid: -7473671815992574501 }
382+0800 I SHARDING [Balancer] performing a split because migrate failed for size reasons
396+0800 I ASIO [NetworkInterfaceASIO-ShardRegistry-0] Successfully connected to 10.240.111.92:26000
398+0800 I NETWORK [mongosMain] connection accepted from 10.133.39.200:37302 #60714 (3 connections now open)
398+0800 I NETWORK [conn60714] end connection 10.133.39.200:37302 (2 connections now open)
415+0800 I ASIO [NetworkInterfaceASIO-ShardRegistry-0] Successfully connected to 10.49.83.33:27000
554+0800 I SHARDING [Balancer] split results: CannotSplit: chunk not full enough to trigger auto-split
554+0800 I SHARDING [Balancer] marking chunk as jumbo: ns: PushHistory.push_date_20170513, shard: 80000095, lastmod: 59|2997||
a8451d, min: { devid: -7478064656446626857 }, max: { devid: -7473671815992574501 }
```

谨慎使用sharding

- 若磁盘和内存不足：
  - 清理无效数据，重建repairDatabase
    - 需要一倍的磁盘空间
  - 先库级别拆分
    - db.runCommand({ movePrimary: , to: })
    - 务必刷新路由
  - 然后历史collection级别tag标签迁移
    - 做tag标记：sh.addShardTag(shard_name,tag_name)
    - 进行迁移：sh.addTagRange(ns, min, max, tag)
  - 最后业务collection开启sharding
    - sh.enableSharding()

7．Mongo备份恢复

- mongodump
- mongorestore

Thank You !

腾讯OMG-周奇

2017/05/20