

MongoDB在字节跳动的应用及挑战

张雷

2020-01-04

01

字节跳动基础架构
介绍

02

MongoDB在字节
的应用场景及挑战

03

未来的优化

04

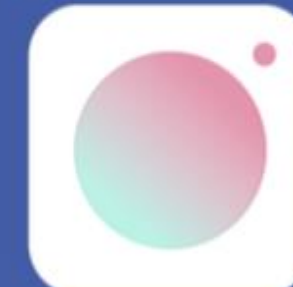
QA

01

字节跳动基础架构介绍

字节跳动-全球创作与交流平台

字节跳动成立于2012年3月，目前公司的产品和服务已覆盖全球150个国家和地区、75个语种，在40多个国家和地区位居应用商店总榜前列。

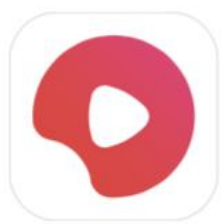


我们有很多优秀的APP



火山小视频

火山小视频是普通人拍摄视频、记录生活、分享兴趣的原创小视频社区。产品基于个性化推荐，可以让每个视频作品，都找到“志趣相投”的用户。火山小视频独有的“圈子”、“K歌”、“火山唱见”等功能，不断地满足着用户多样化的产品使用需求。



西瓜视频

西瓜视频是字节跳动旗下的一款视频 APP，作为聚合多元文化的综合视频平台，它通过个性化推荐，源源不断地为不同人群提供丰富的优质内容，同时鼓励多样化创作，帮助人们轻松地向全世界分享视频作品。



懂车帝

懂车帝是“看车、选车、买车”一站式汽车媒体和服务平台，产品基于个性化推荐引擎帮助用户发现感兴趣的汽车内容，同时配有车型库、360度全景看车等选车工具，首创短视频社区“车友圈”，为用户打造内容+社区+工具的多元生态。



皮皮虾

今日头条官方爆笑社区，一个只要打开就能让你笑到停不下来的 App。皮皮虾一直坚持「传播快乐，分享生活」的目标，以特有的「神评」玩法形成了独特的社区氛围。之后，皮皮虾将不断提供更多有趣的功能，帮助用户创造出更多优质的内容。



Faceu 激萌

一款更多年轻人都喜爱的相机 APP，海量酷炫贴纸、激萌表情包、实时美颜、趣味特效、视频跟拍让摄影社交更丰富更有趣，满足全方位拍摄需求。累计用户量过 4 亿，每天为用户提供 1.7 亿次服务



轻颜相机

轻颜相机是一款主打高级感的质感自拍相机，连续霸榜应用商店。拥有时下最流行的滤镜、美颜效果、和海量拍照“姿势”模板，一键就能 get 潮流自拍。



关于字节跳动的一些数字

7

字节跳动旗下产品
全球 DAU 超过7亿

15

字节跳动旗下产品
全球 MAU 超过15亿

3.2

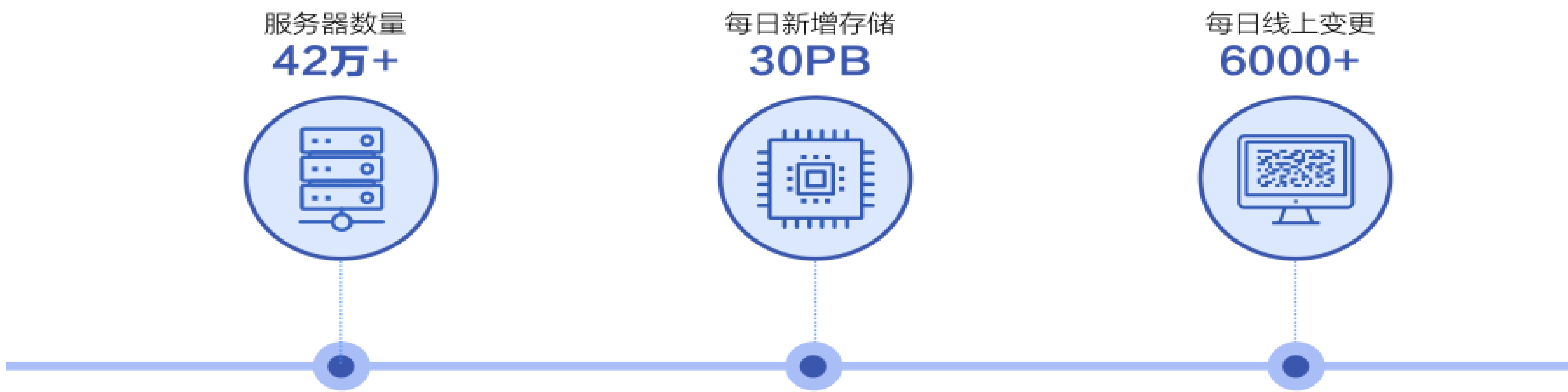
抖音 DAU 超过3.2亿

字节基础架构-提供世界级的基础设施和架构



字节基础架构团队是一支面向全球的基础设施研发团队，旨在提供世界级的基础设施和架构，为字节跳动业务发展保驾护航。

字节跳动基础架构面临的挑战



业务敏捷变更发放，超大集群/数据规模治理与急速的数据增长是字节跳动基础架构面临的三大挑战。

02

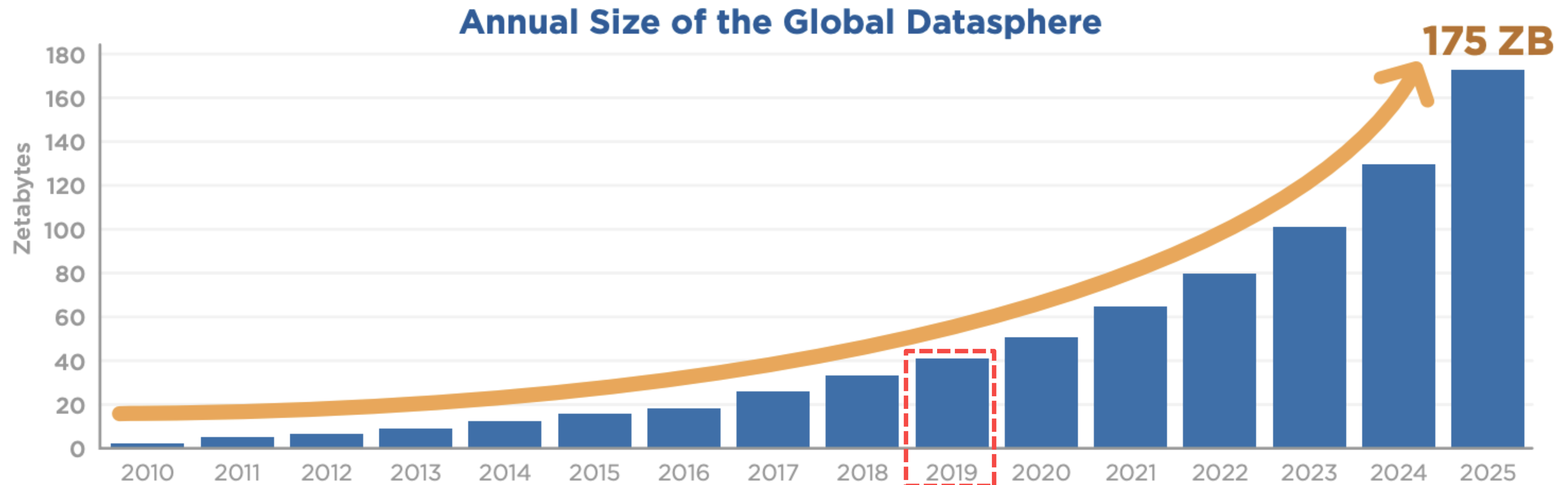
MongoDB在字节的应用场景及挑战



Why MongoDB

数据增长是未来5年的主题

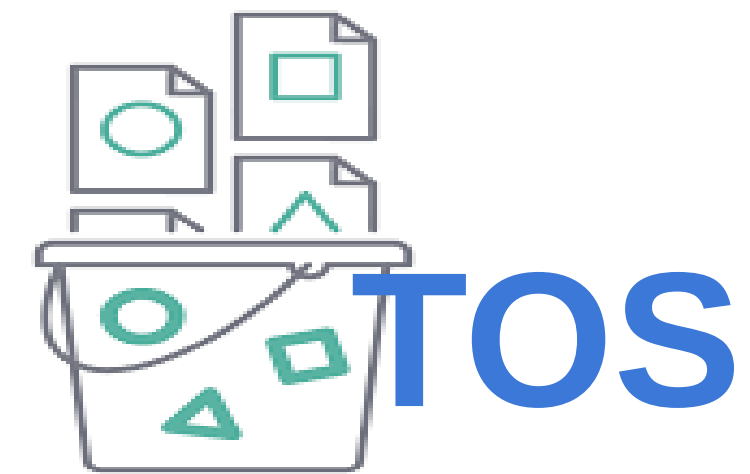
Figure 1 - Annual Size of the Global Datasphere



Source: Data Age 2025, sponsored by Seagate with data from IDC Global DataSphere, Nov 2018

字节跳动的管理和产生数据将还会持续高速增长，如何有效治理数据是一个巨大的挑战。

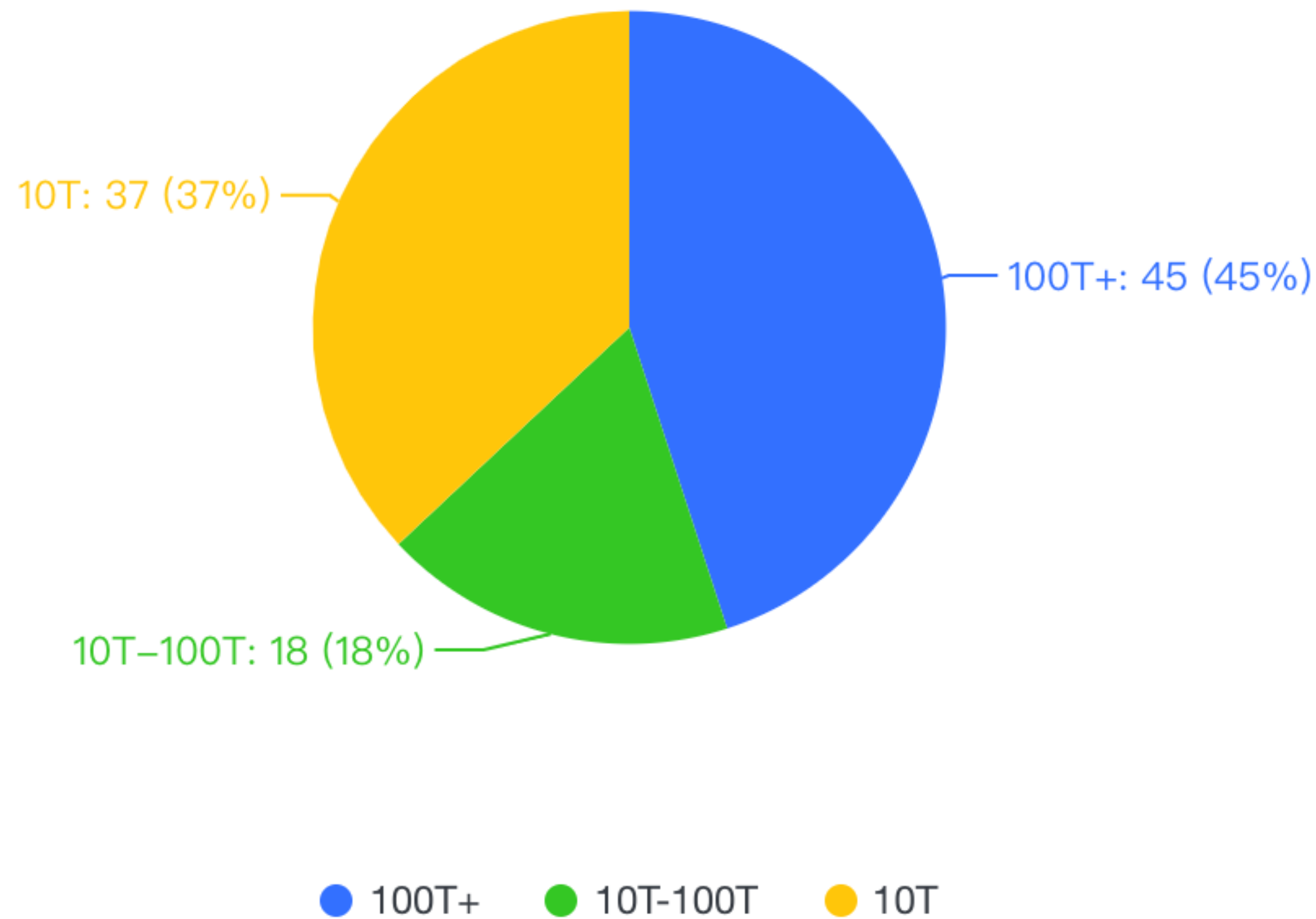
应用的数据管理标准三件套：Redis + MySQL + TOS



- 结构化与半结构化数据存储于MySQL中
- MySQL规模化增长，数万台服务器集群/数千万的QPS/数十PB的数据量
- 业务期望的灵活度以及MySQL的弹性扩展能力逐渐成为主要矛盾

结构化/半结构化数据的增长瓶颈：How to Scale

库容量分布

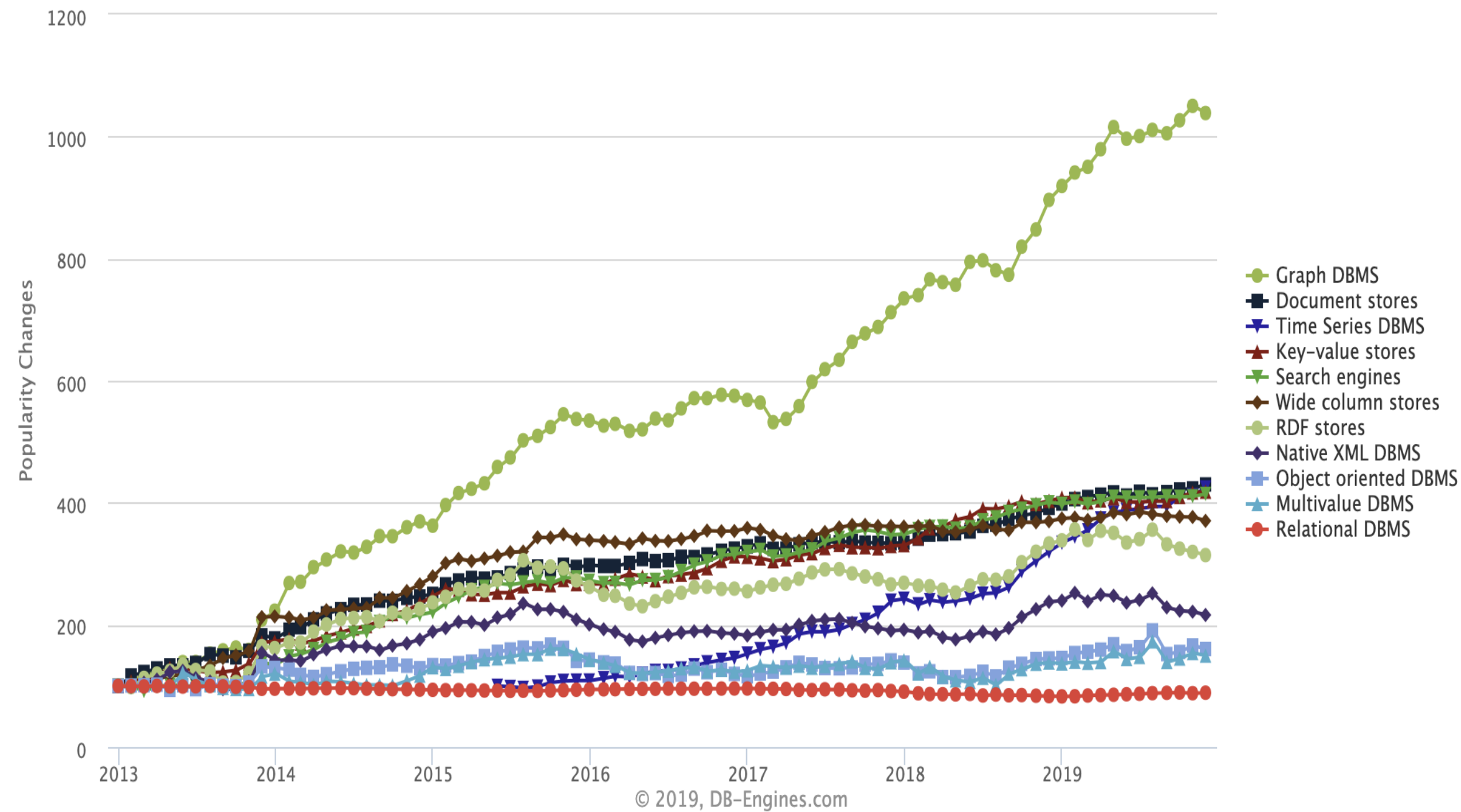


几大痛点：

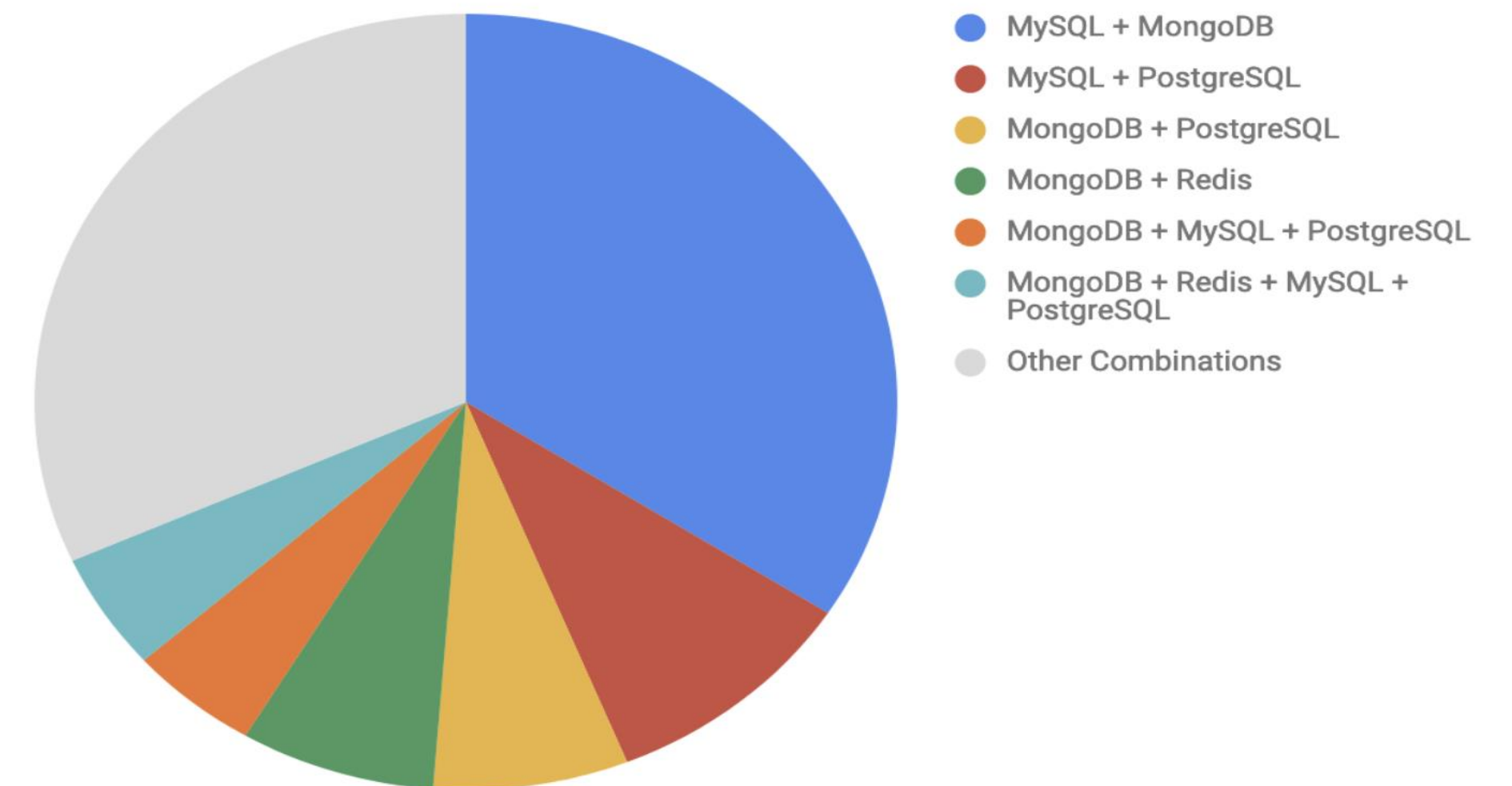
- 难以扩容
- 中间件的约束
- 性能瓶颈
- 低的利用率
- 业务迭代速度

业界的调研

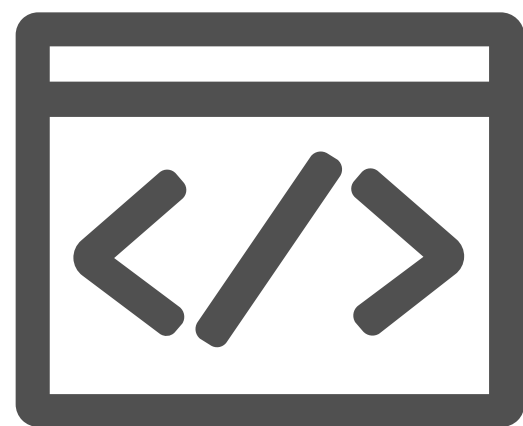
Complete trend, starting with January 2013



MySQL + MongoDB: 34.15%
MySQL + PostgreSQL: 9.76%
MongoDB + PostgreSQL: 7.32%
MongoDB + Redis: 7.32%
MySQL + MongoDB + PostgreSQL: 4.88%
MySQL + MongoDB + PostgreSQL + Redis: 4.88%

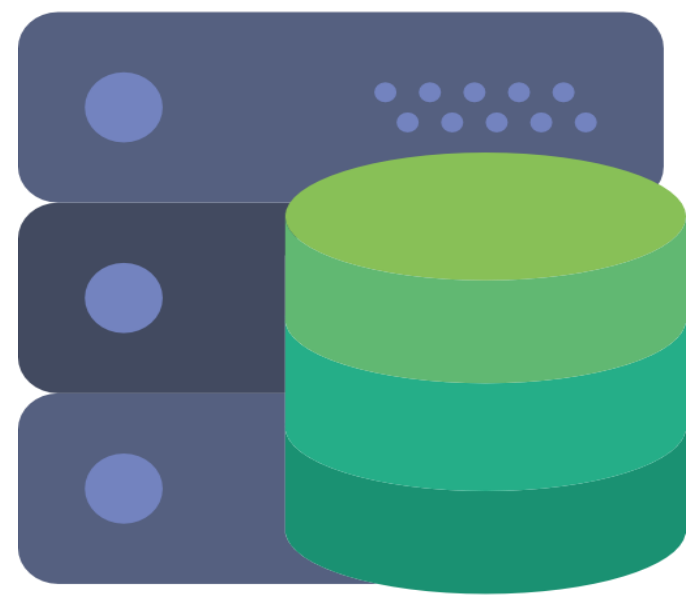


schema free + 成熟的社区 + 一致性与性能的高度可配置



MongoDB的应用情况

MongoDB当前使用概况



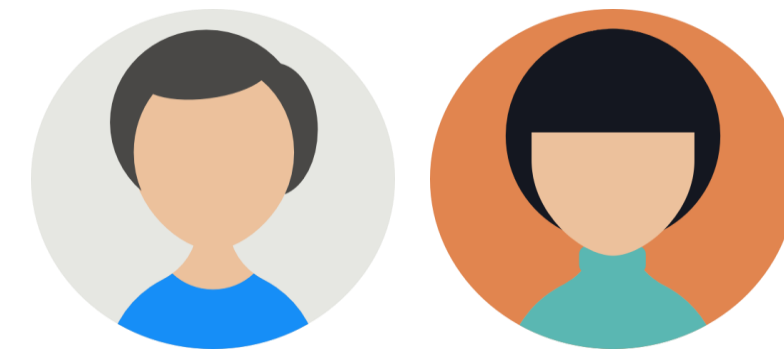
Servers

1800+



Clusters

150+

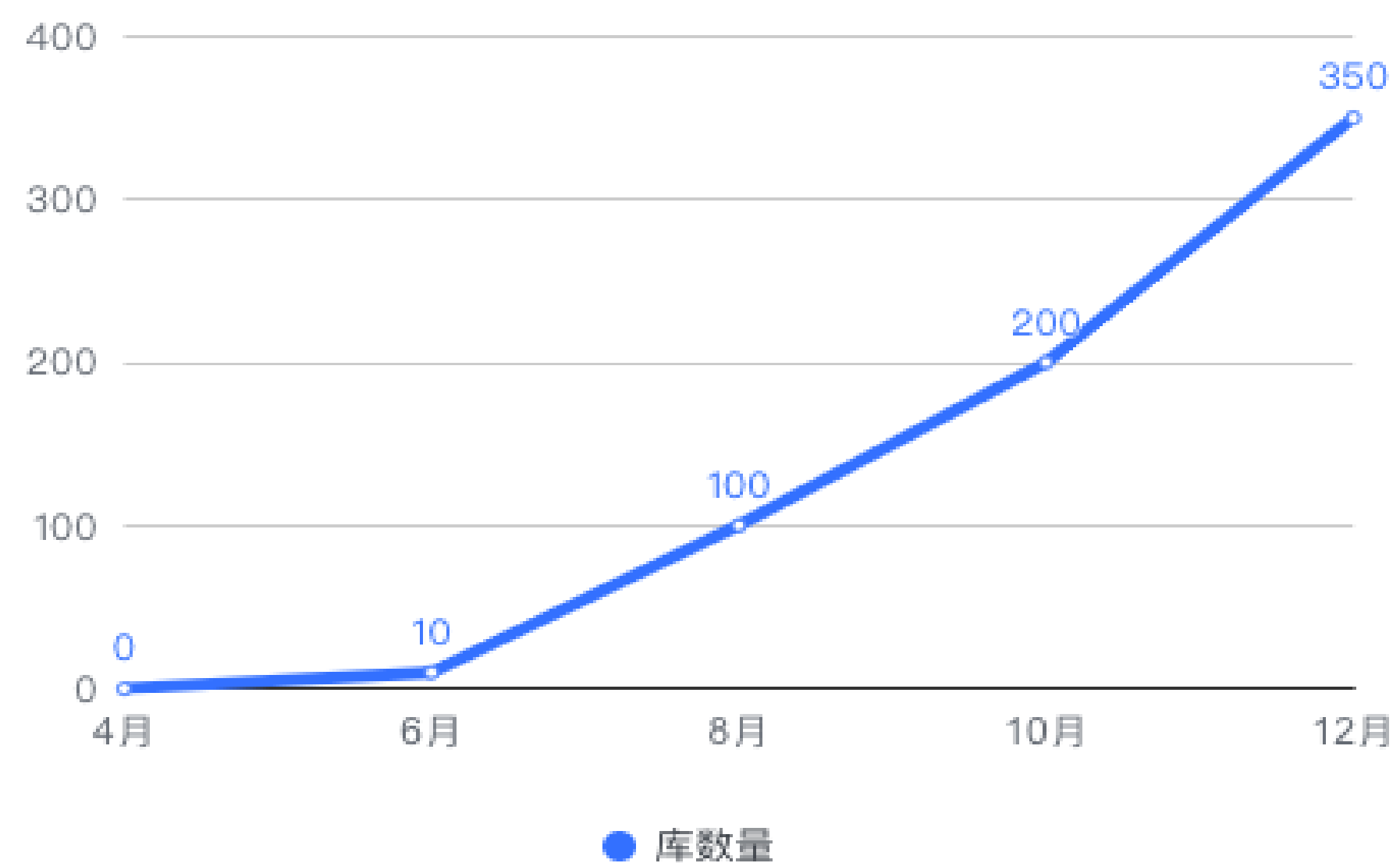


Users

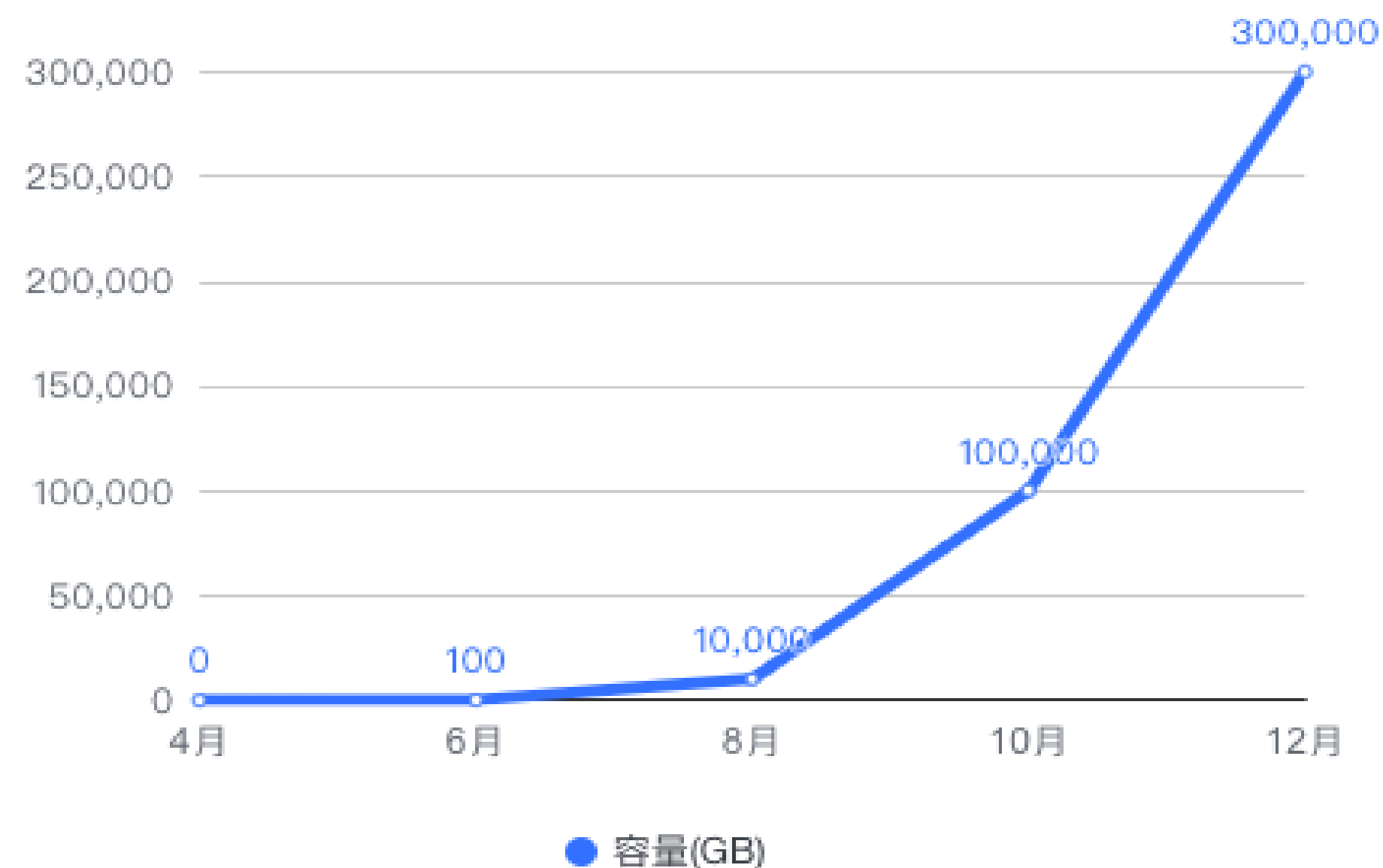
350+

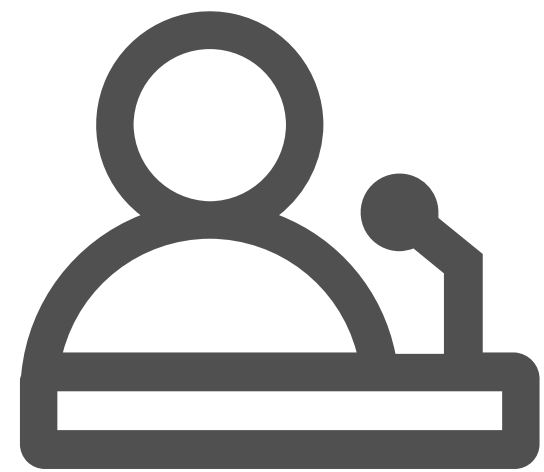
用户与存储量快速增长

用户库增长曲线



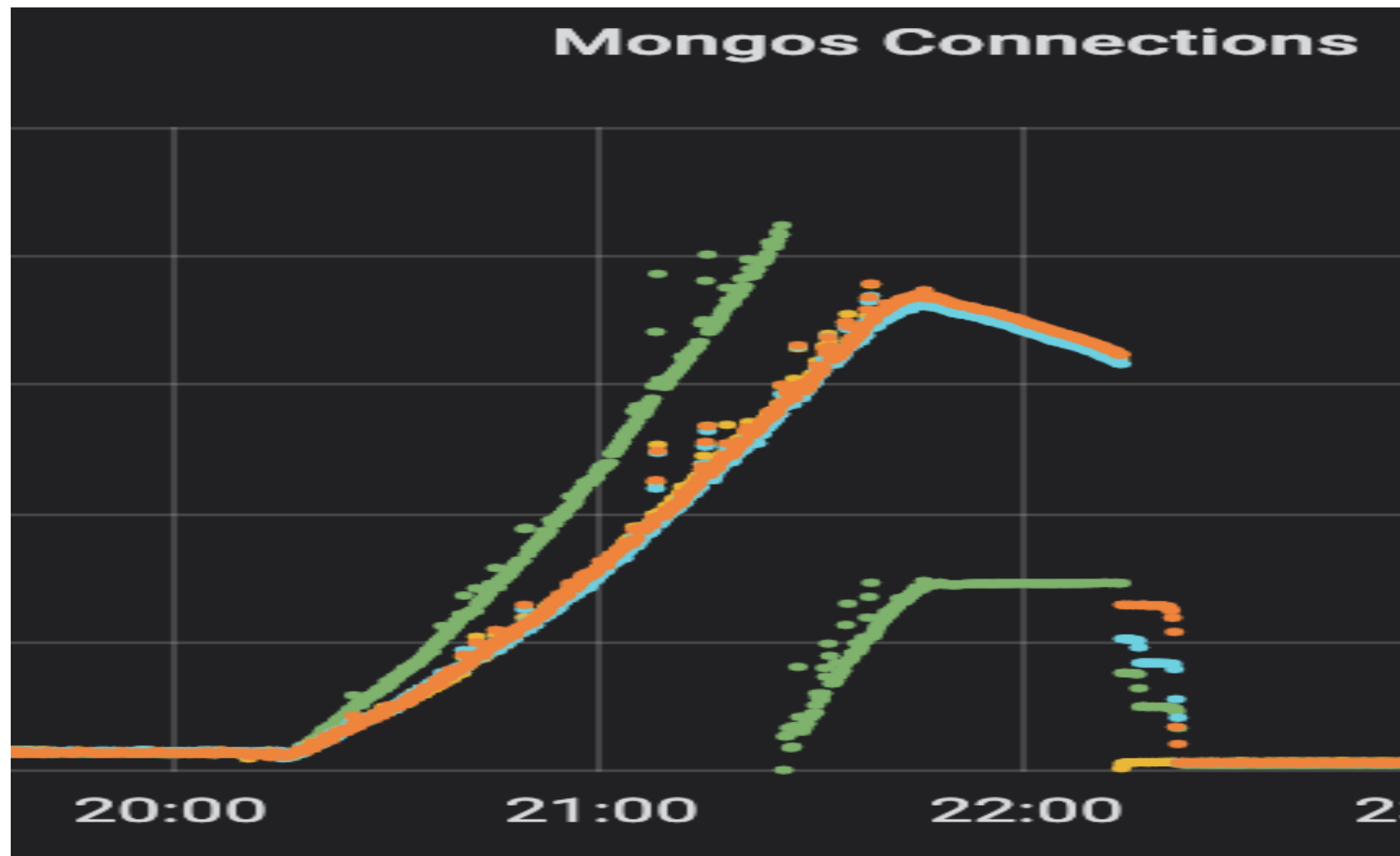
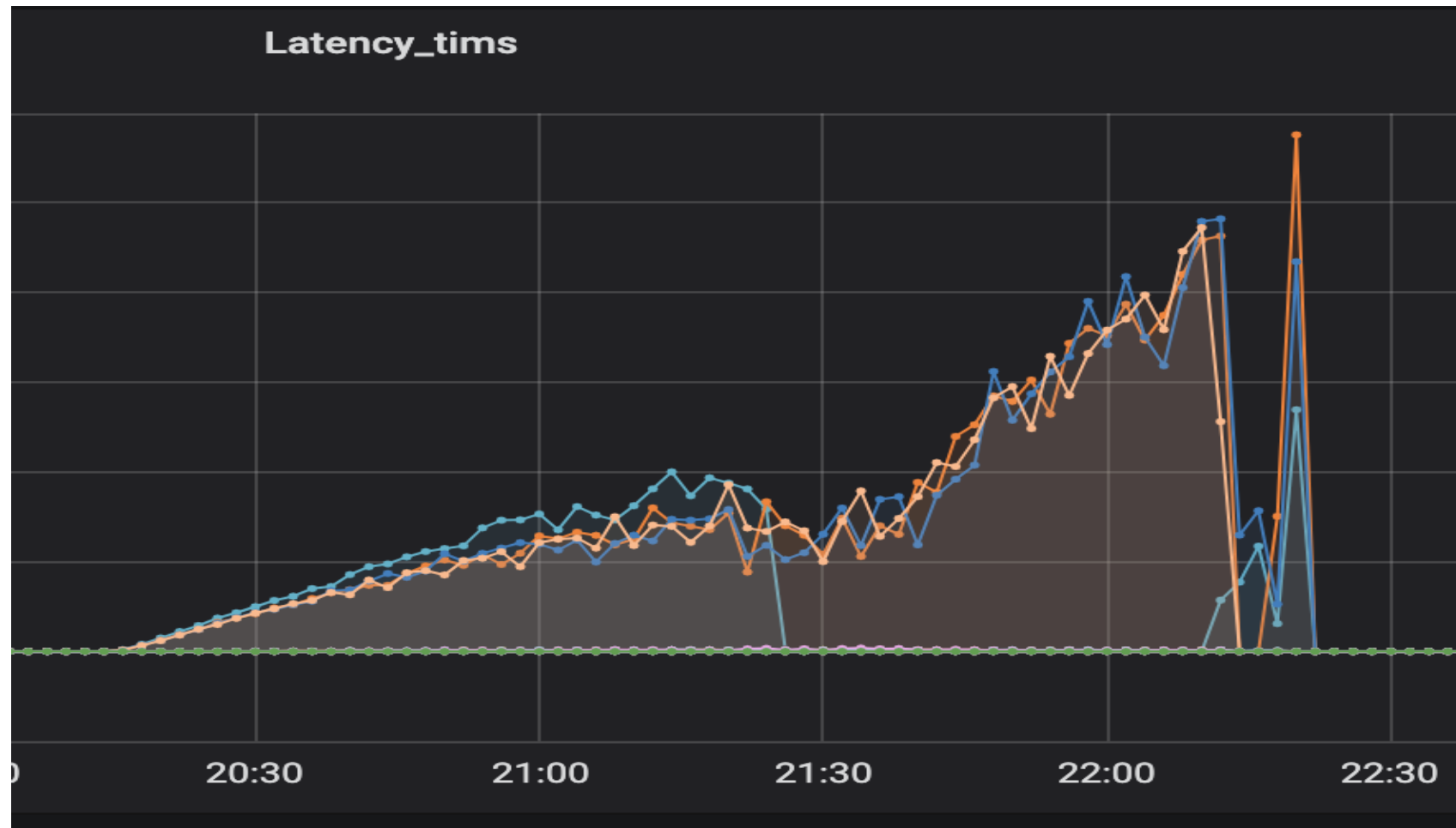
容量增长曲线





一些典型的Use cases

在线TP：一些在线数据处理服务



业务特点：

- 数据模型多变：新增Collections比较常见
- 时延和毛刺：要求低时延低毛刺
- 请求量：请求量大，并且一般带有重试逻辑

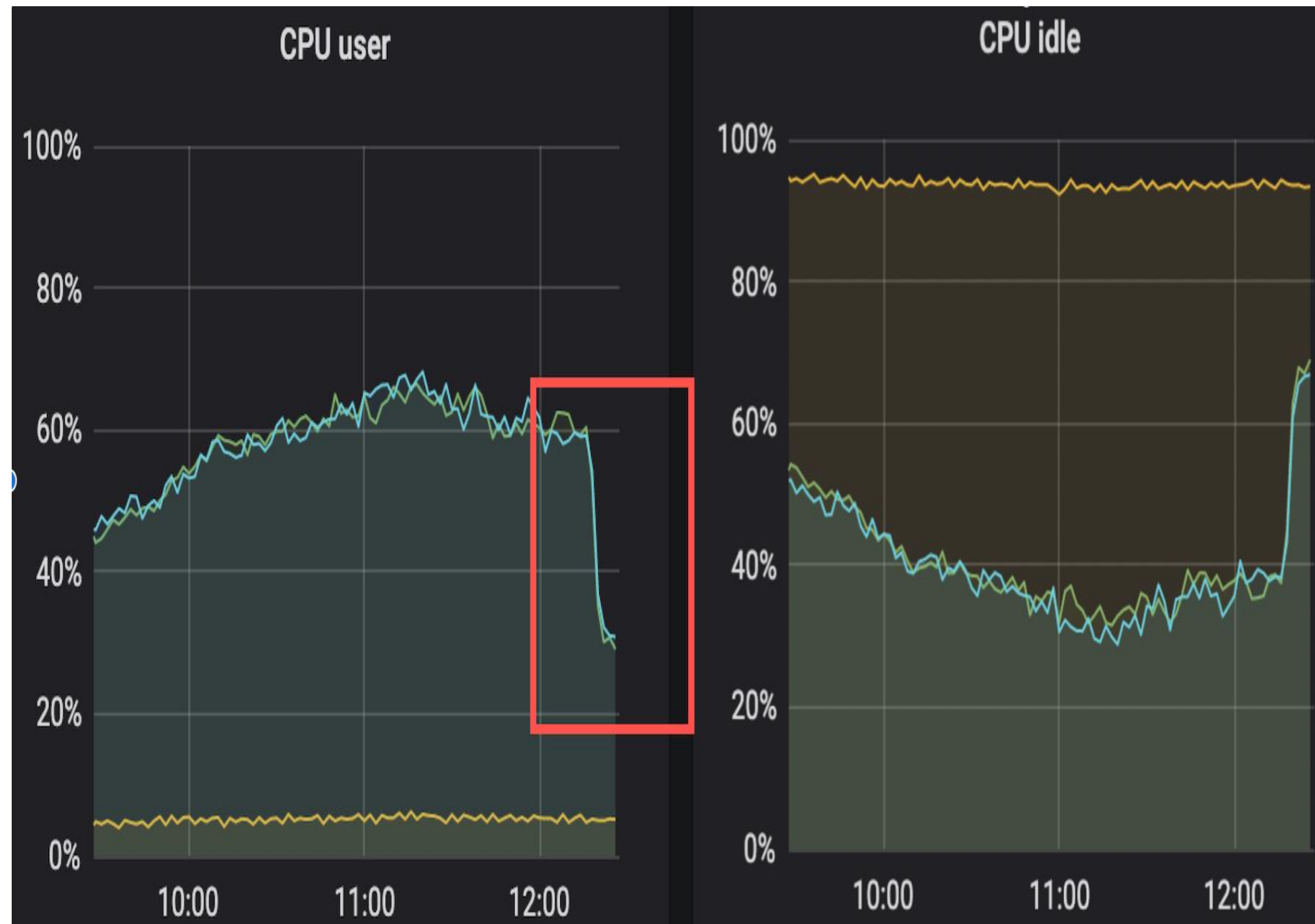
遇到问题：

- 业务新增Collection后，一开始数据量不大，不带索引查询影响不大
- 随着数据量增大，超过阈值后，延迟和连接数都暴涨，服务不可用

解决方案：

- 如果查询没有带maxTimeMS，设置一个默认值
- 增加慢查询监控告警，发现全表扫描并超过阈值后告警
- 端到端流控，驱动-mongos-mongod全链路流控，防止雪崩

LBS：地理位置信息查询服务



```
db.short_video.find({"geometry.coordinates": {
  $near: {
    $geometry: { type: "Point", coordinates: [113.86,
22.58]},
    $minDistance: 0,
    $maxDistance: 200,
    $hintCorrectNum: 200,
    $maxScanNum: 2000
  }}})
```

业务特点：

- 计算密集型：数据量小，计算量大
- 写入/更新负载低，但实时查询负载高，随着业务发展需要横向很多从库

遇到问题：

- 大范围、精准的位置查询非常消耗CPU
- 业务对LBS查询时延敏感，负载较高时大概率出现慢查询

解决方案：

- 支持用户根据自身业务需要指定精准/模糊查询，以及控制最大扫描点数

游戏：用于运营日志存储与分析



业务特点：

- 存储密级型：数据量大，写入量较大，查询量一般
- 偏在线分析：需对事件做各种分析

遇到问题：

- 单机磁盘无法满足存储需求
- 查询条件多变，无法建立有效索引

解决方案：

- 使用多分片集群
- 在从库上使用MapReduce分析

```
function() {
    emit(this.event.actor.id+',', {
        'serid':this._id+',',
        'time':this.time,
        'roleid':this.event.actor.id+',',
        'tag_list':this.event.data.tag_list,
    });
}
""")
myreduce = Code("""
    function(key,values) {
        var pos = values.length - 1;
        return values[pos];
    }
})
```


元信息：用于各类中台系统的元信息存储和分析



```
{
  "_id" : ObjectId("zzz"),
  "status" : "normal",
  "fnCount" : 0,
  "packageCount" : 0,
  "tableCount" : 0,
  "parterCount" : 0,
  "tags" : [ ],
  "name" : "zzzz",
  "summary" : "zzzz",
  "serviceld" : "zzzz",
  "uid" : "zzzz",
  "attrs": {...},
  "__v" : 0
}
```

业务特点：

- 中台系统：Schema可能需嵌入具体业务的Object
- 时延和毛刺：要求低时延和低毛刺
- 数据量和请求量：数据量大，以点查为主

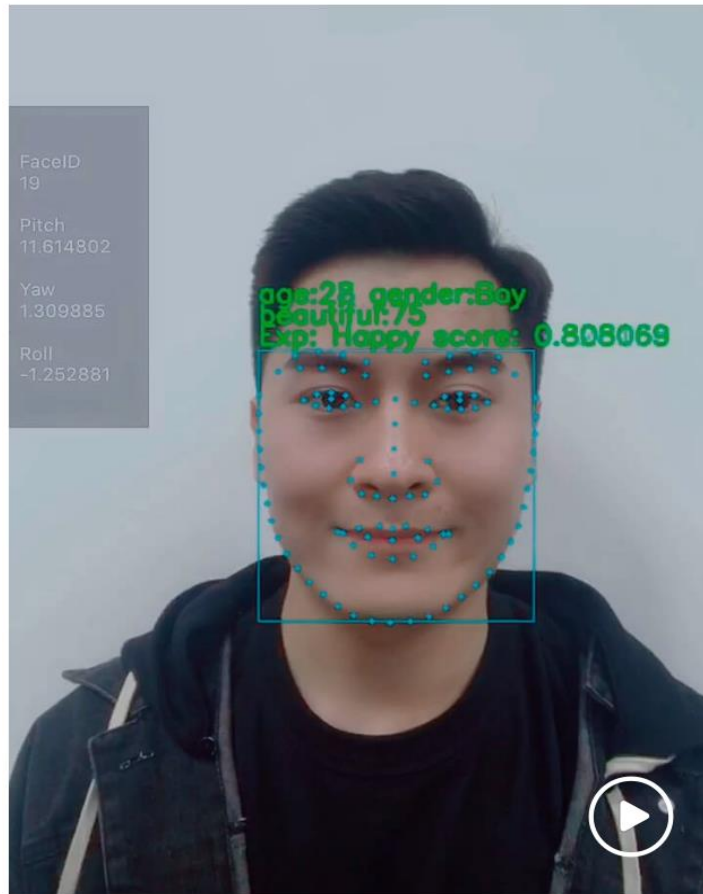
遇到问题：

- 请求量太大，单节点无法扛住
- 有因果一致性要求

解决方案：

- 多分片，并开启读分流
- 采用因果一致性session

AI：用于存储AI计算的中间及最终模型值



技术演示视频



特效应用视频

业务特点：

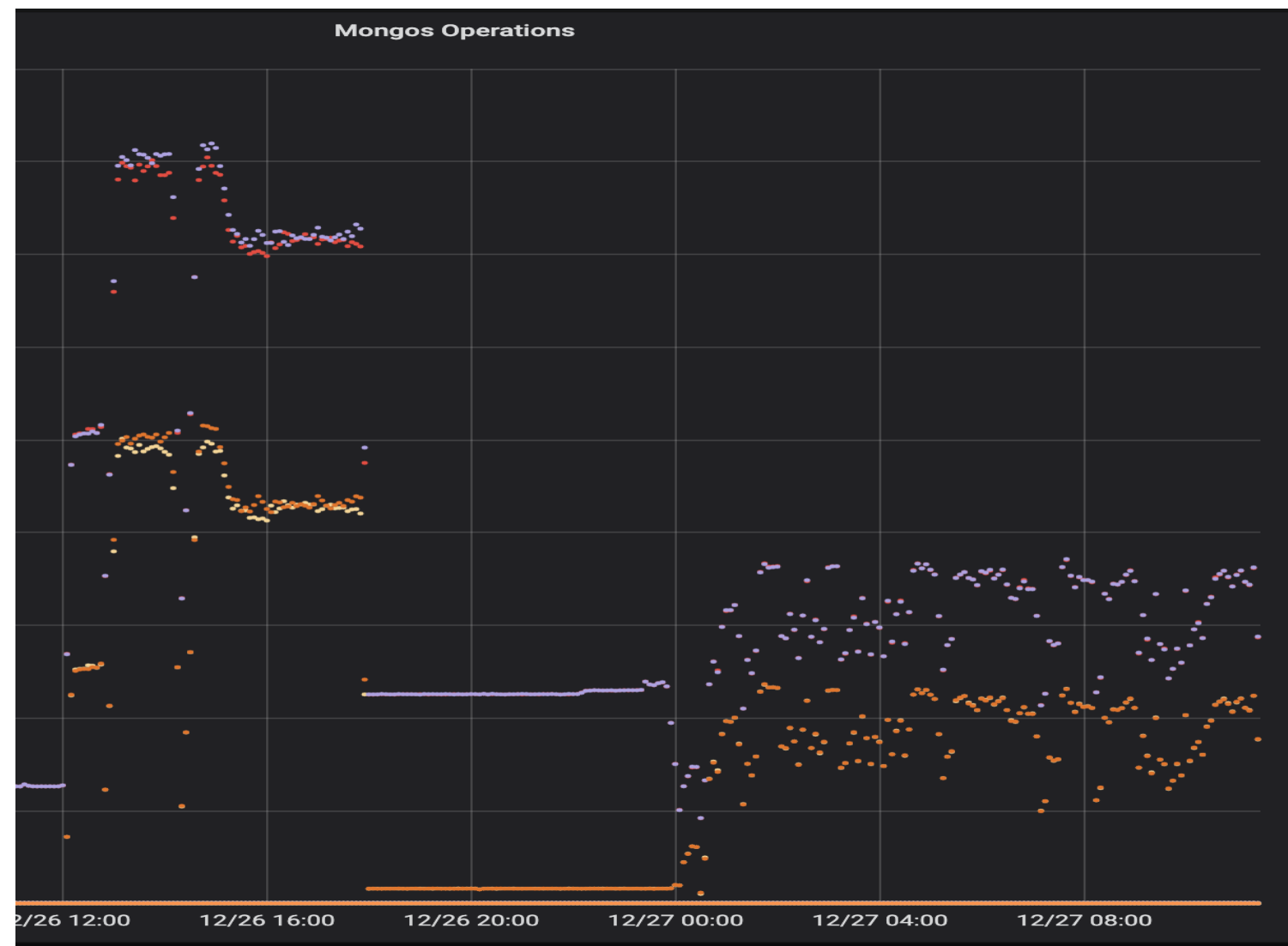
- Burst写入：某段时间集中写入大量数据
- 时延和毛刺：要求一般，容忍度较高
- 数据重要性：可重建，但花费很长时间

遇到问题：

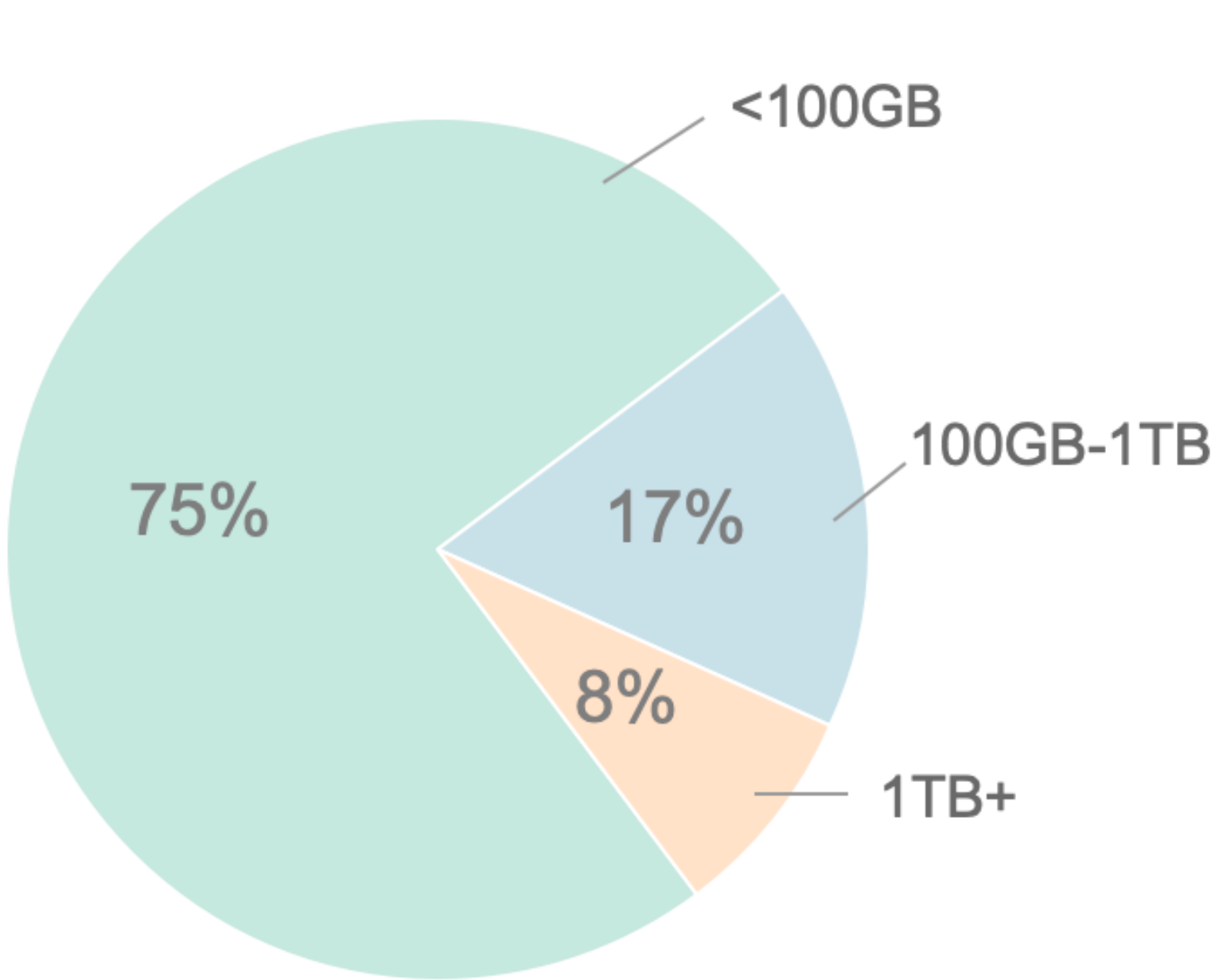
- 资源使用不均衡，时而高负载，时而低负载
- 数据量大，存储成本高

解决方案：

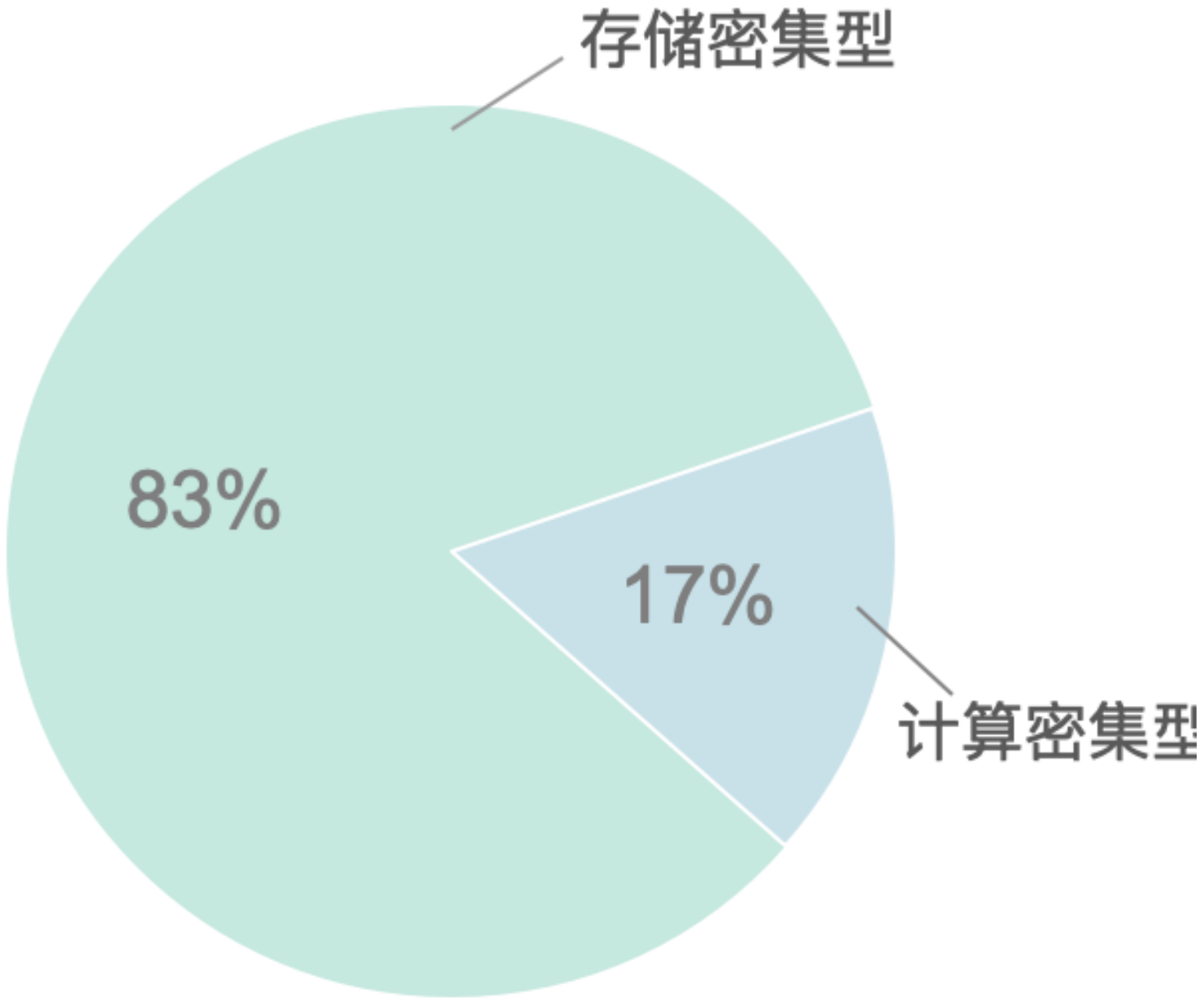
- 使用批量插入方式，优化插入性能
- 缩减副本数，定期冷备，降低成本



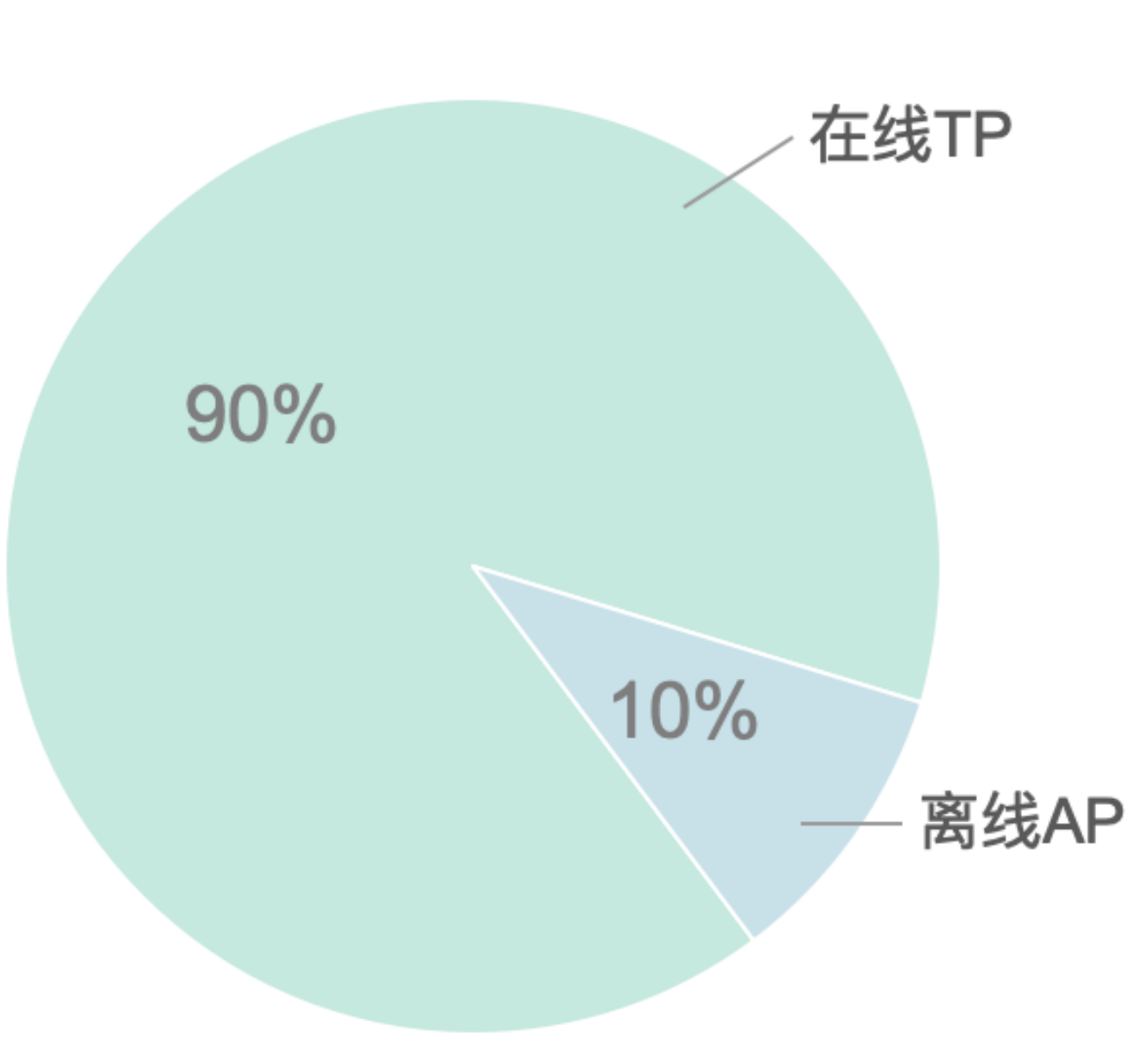
应用场景小结



按容量统计实例数



按计算/存储负载统计



在线/离线业务



未来的优化

未来的关注点与一些优化方向

弹性

计算&存储
按需扩展



- 计算存储分离
- 容器化

成本

规模化场景下，数据
价值挖掘与成本优化



- 冷热分离
- 数据智能调度/压缩

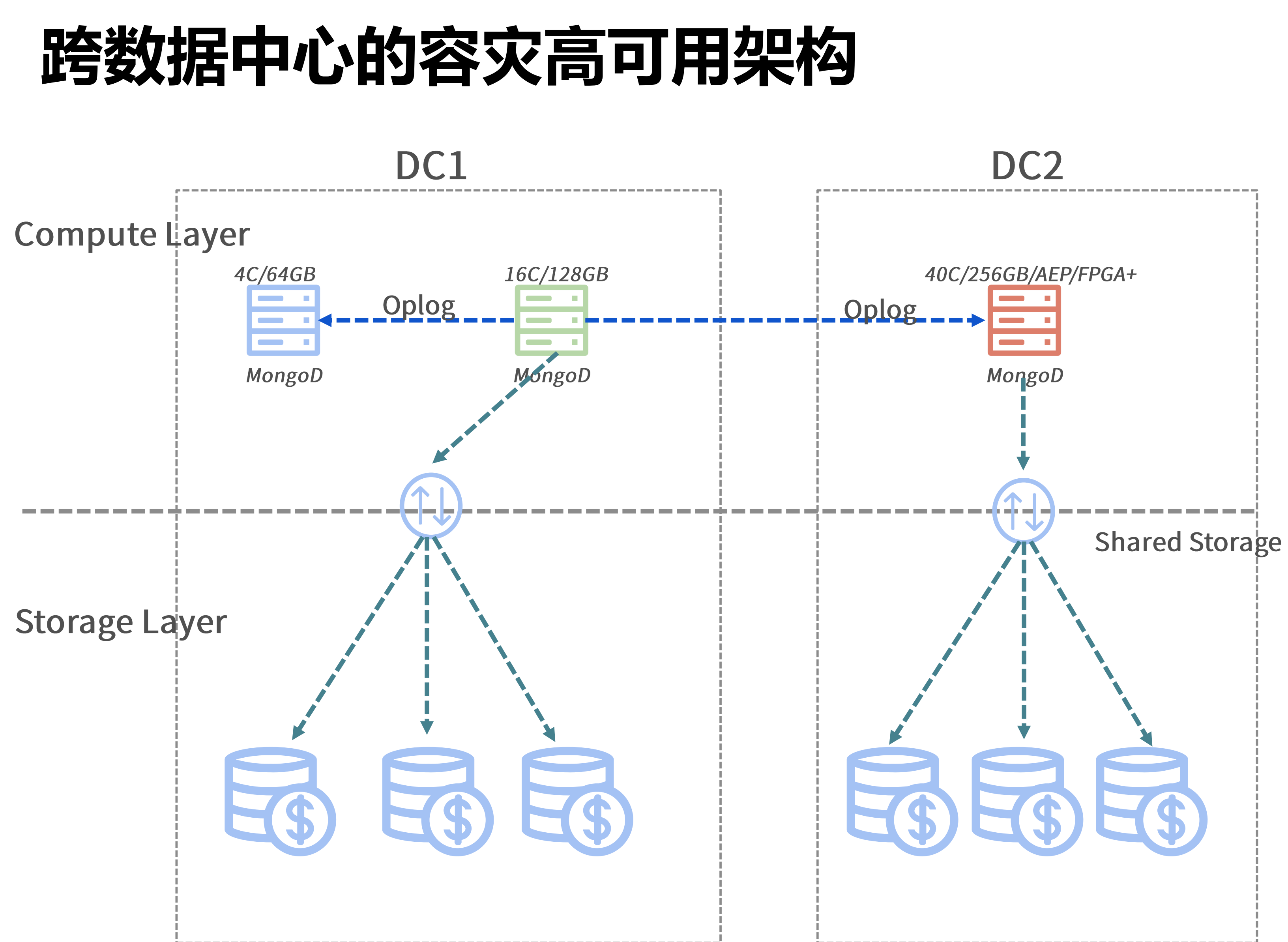
性能

有限资源下追求
极致时延和QPS



- 软硬协同加速
- 协程/FPGA/AEP

跨数据中心的容灾高可用架构



1. 计算&存储分离

1. 按需调度的计算层
2. AEP for hot data cache
3. FPGA for 压缩/加密/卸载
4. EC使能的分级存储层

2. 跨数据中心容灾架构

1. 数据中心间复制oplog
2. 一致性与性能可配置



QA

欢迎投递简历!!!

zhanglei.michael@bytedance.com





mongoing
中文社区

mongoing.com

欢迎关注MongoDB中文社区



MongoDB中文社区网址:

<http://www.mongoring.com/>

MongoDB中文社区技术交流群:

- 1.请关注微信服务号
- 2.点击 “社区互联”
- 3.选择 “联系我们”

MongoDB中文社区微信服务号+订阅号