www.qconferences.com
www.qconbeijing.com

QCon北京2014大会 4月17—19日

伦敦 | 北京 | 东京 | 纽约 | 圣保罗 | 上海 | 旧金山

London · Beijing · Tokyo · New York · Sao Paulo · Shanghai · San Francisco

QCon全球软件开发大会

International Software Development Conference

www.infoq.com/cn

InfoQueue

@InfoQ

infoqchina

软件
正在改变世界！

特别感谢
QCon上海合作伙伴

QCon Shanghai 2013

京东文件系统简介

刘海锋

JD.COM 京东

1. 为什么自主研发
2. 实现技术与经验
3. 正在进行的工作

问题与挑战

- 商品订单
  - 365 * 数亿 * ~10KB
- 商品图片
  - 几十亿 * (20 ~200KB)
- 库房记录
  - 365 * 十亿 * (KB ~ MB)

各种方案

- 关系数据库
  - Oracle Exadata, …
  - Pains – 没法扩容、定期删除
- 开源存储系统
  - HDFS、FastDFS、…
  - Pains – 难以选型、定制、维护

自主研发

- 核心软件系统可以自主研发
  - If you believe you can☺
- 注意事项
  - 紧扣业务需求
    - 高度定制
  - 分期开展
    - 第一期不做大而全
    - 缩短开发周期，提高ROI

系统定位

- V1: Scalable *SystemKey*-File Store
  - 特别针对海量小文件
  - 强可靠、强一致、高可用
  - Key由系统本身生成
    - 可满足很多业务需求 – Really?
- V2: support user-defined keys and more

1. 为什么自主研发
2. 实现技术与经验
3. 正在进行的工作

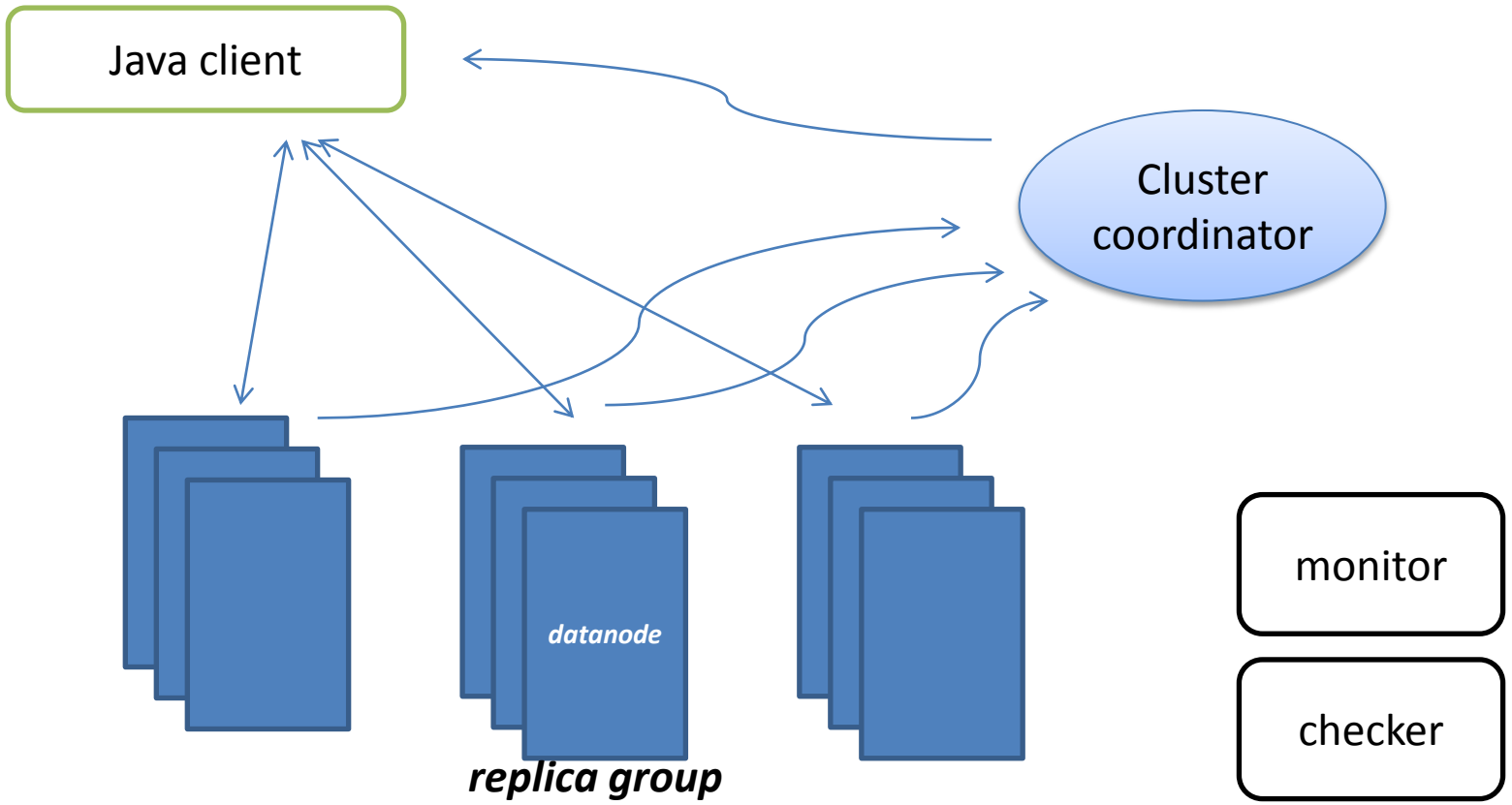
编程语言的选择

- Go写系统框架，C写单机存储引擎
  - 各取所长
- Why Go?
  - 良好性能，高开发效率
    - 适合小团队开发

```
xiaojing@Inspiron:~/tech/jfs/branches/haifeng/src$ wc -l tfnode/*.go nimblestore/store.go cmd/jfsd/jfsd.go util/config/config.go util/zkwrapper/zk.go
  66 tfnode/const.go
 204 tfnode/follower.go
 257 tfnode/proto.go
 329 tfnode/recovery.go
 725 tfnode/server.go
 132 tfnode/stats.go
 145 nimblestore/store.go
  78 cmd/jfsd/jfsd.go
  82 util/config/config.go
  92 util/zkwrapper/zk.go
2110 总用量
```

总体架构

- 客户端
  - Java, C/C++, nginx-based, Go, ..
- Cluster Coordinator
  - ZooKeeper
- Datanode
  - 3 datanodes form a ***replica group***
    - 1 leader + 2 followers
    - 一致、持久的存储单元
  - 通常每磁盘部署一个实例


Java client
Cluster coordinator
datanode
replica group
monitor
checker

集群视图

- replica-group Id -> member addresses & weights
  - Weight标识可读可写状态

/jfs-root/
datanodes/
replicagroup-1/
10.111.11.1:20130; 0
10.111.15.2:20130; 0
10.111.11.3:20130; -1
…
replicagroup-5/
10.111.12.1:20130; 1758
10.111.12.2:20130; 0
10.111.13.1:20130; 0
…

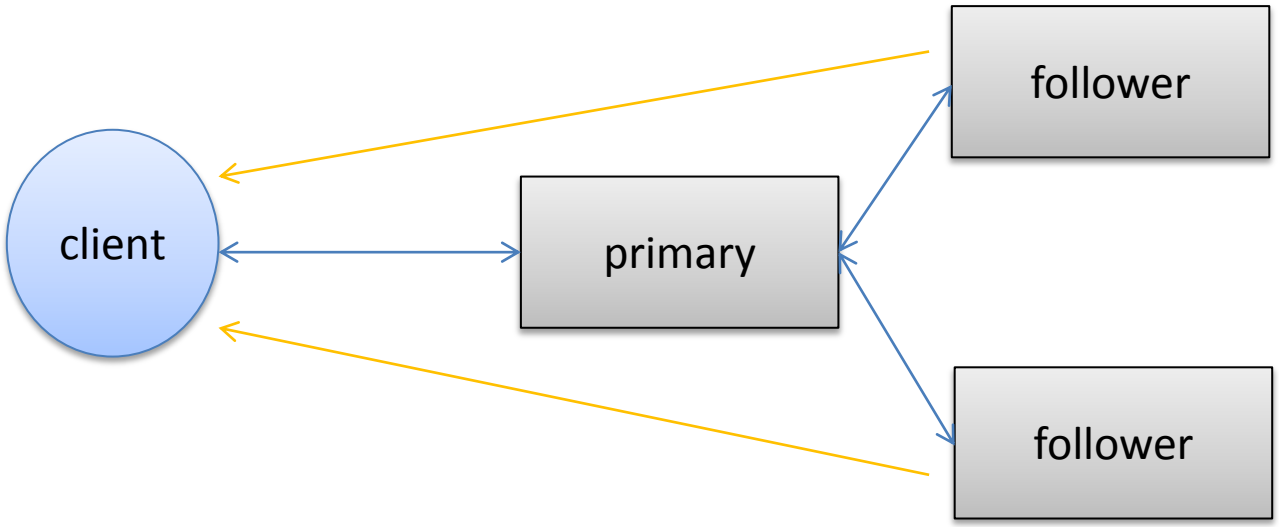
- 客户端实现读写路由等关键逻辑

JFS Key举例

- *jfs/t5/8/10240/10000/A5B8FC33-Y*
  - Replica Group - 5
  - Internal key - 8/10240/10000
    - Chunk Id/Offset/Size
  - CRC - A5B8FC33
  - Y - 已压缩

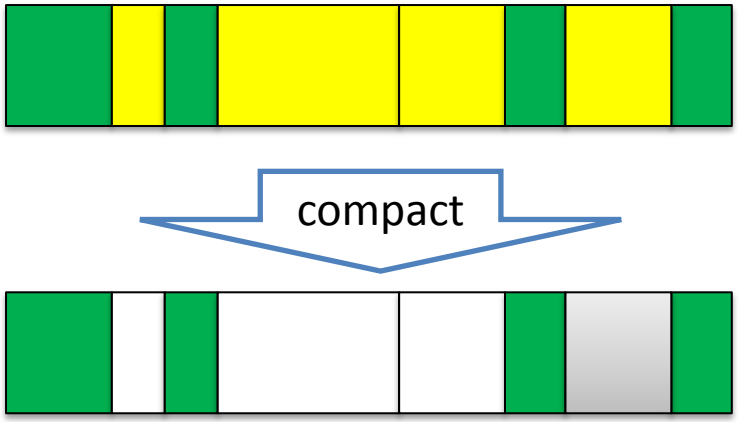
JFS复制协议

- Paxos算法的变体
  - 固定成员角色 – one primary + 2 followers
    - 不做majority-based leader election
  - Full-quorum replication
    - 二元状态机 - ReplGroupReady or ReplGroupSplit



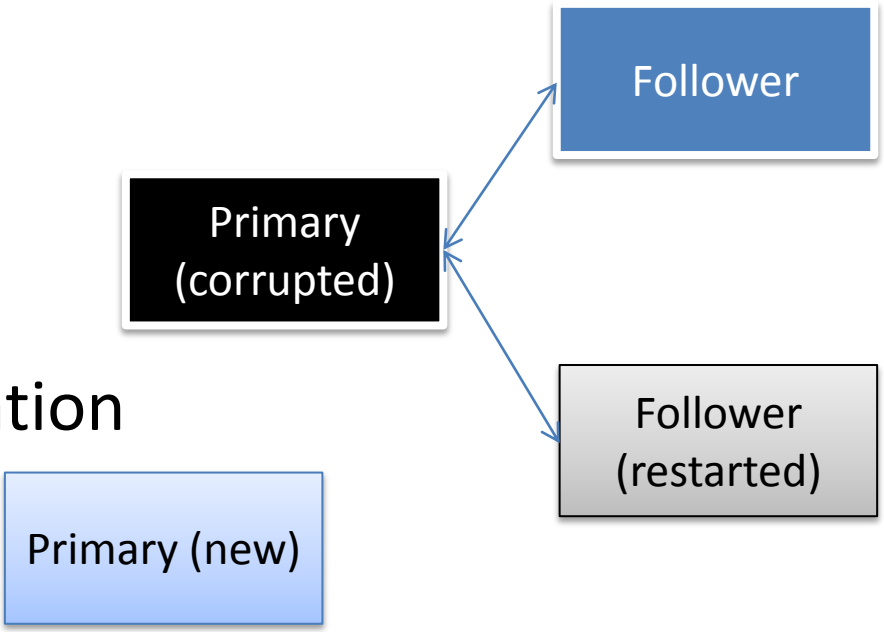
单机存储引擎

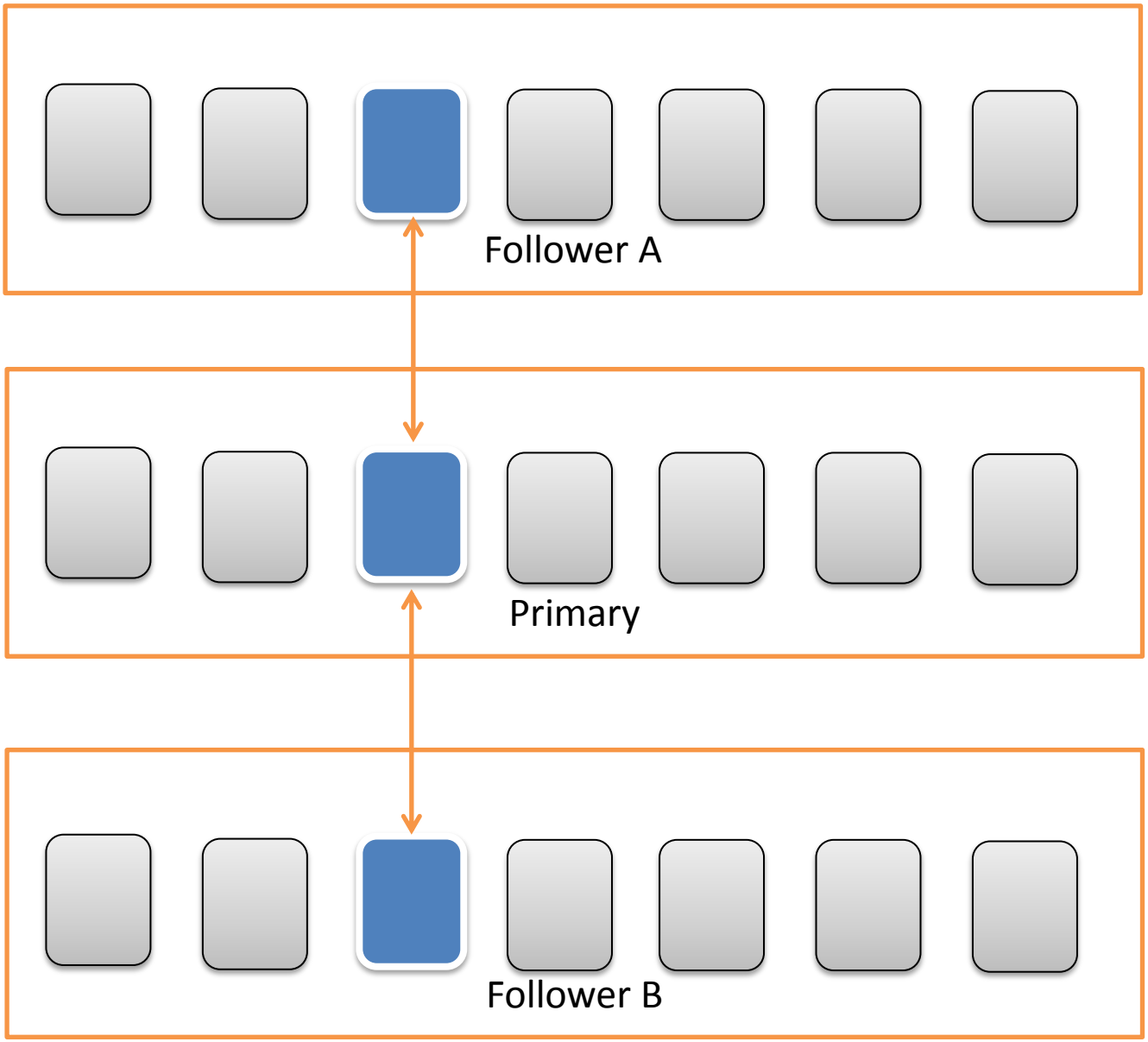
- 一组Append-Only文件，无内存索引
  - chunkId/offset/size as internal key
- 便于Crash-Recovery操作
  - 就是做文件同步
- 如何做Garbage-Collection？
  - 利用lseek()



故障处理与恢复

- 两种类型
  - Fail-Restart
  - Fail-Replace
- 统一处理
  - Chunk file synchronization




Follower A
Primary
Follower B

可靠性与一致性

- 强一致
  - 各个成员均写入才返回写成功
  - 同一复制组保持数据一致
    - **每个数据文件的每个字节**
- 强可靠
  - 集群宕机、磁盘损坏
  - 文件误删除或截断
    - 若某个或某两个数据文件被误删，系统自动恢复

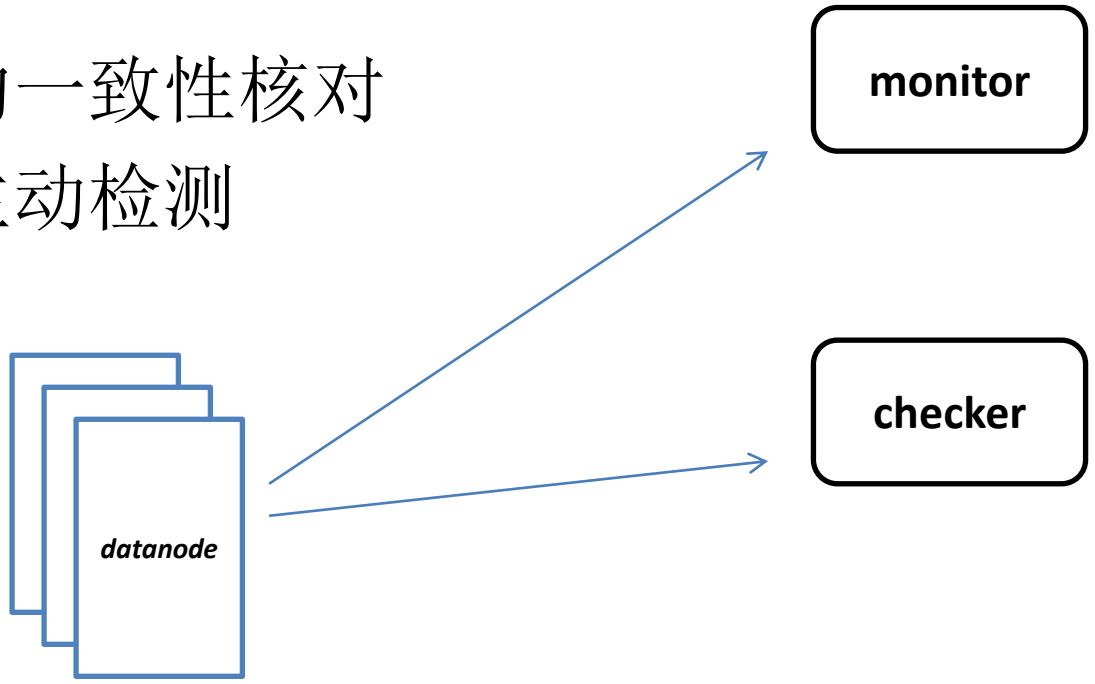
可用性定义

- 给定一个复制组
  - 若某成员在线则读可用
  - 仅当全体成员在线才可写
- 整个系统的写可用性
  - 至少一个复制组可写

- 除非IDC断网断电，JFS集群总是可写可读
  - 因为总是部署很多复制组在不同网段

离线模块

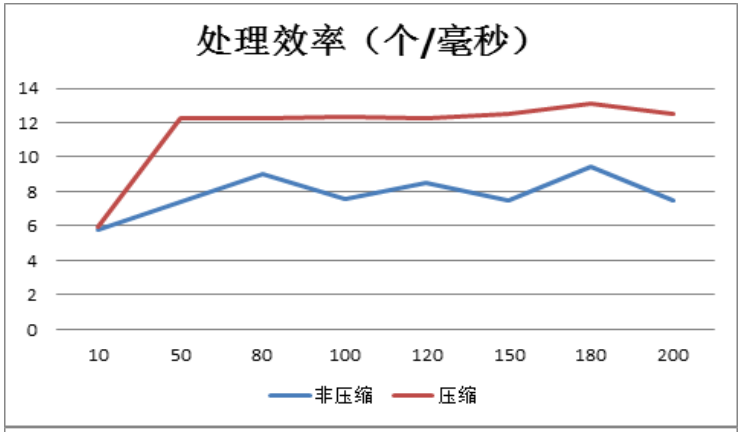
- Monitor
  - 各个datanode的运行时metrics
- Checker
  - 各个复制组的一致性核对
  - 磁盘故障的主动检测

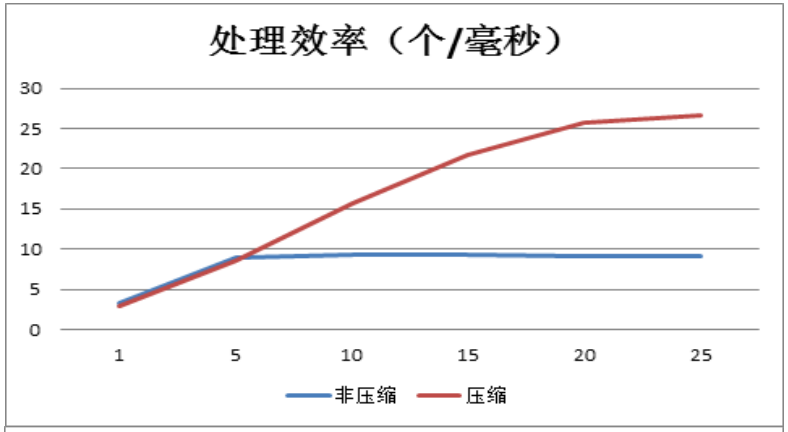


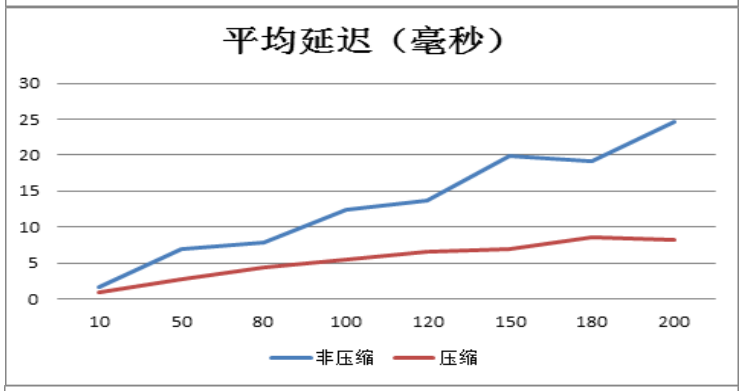
持续优化

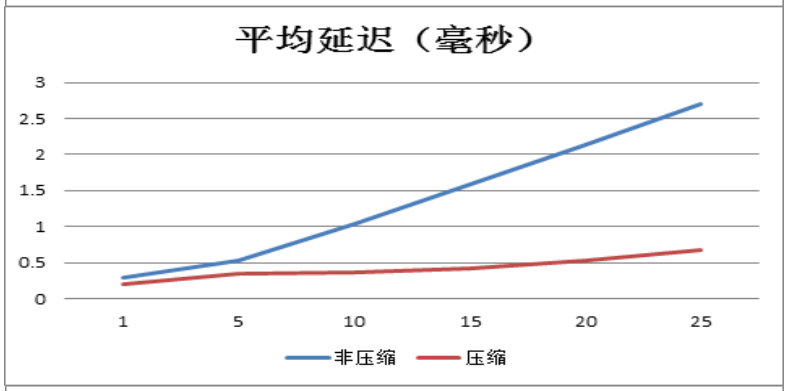
- 流水线写
  - 支持大文件
- 透明压缩
  - 针对文本对象，节省带宽与机器资源
  - 客户端压缩 vs 服务端压缩
- 多数据中心
  - 异步复制，客户端就近读取
- 通过Erasure Coding降低存储成本
  - 利用访问时效性

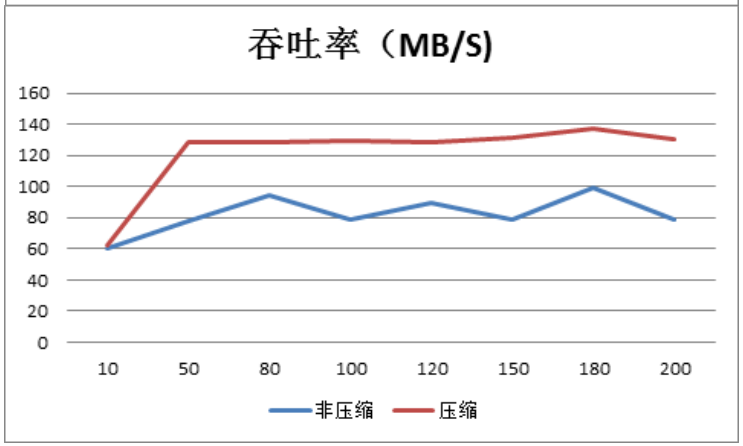
性能数字

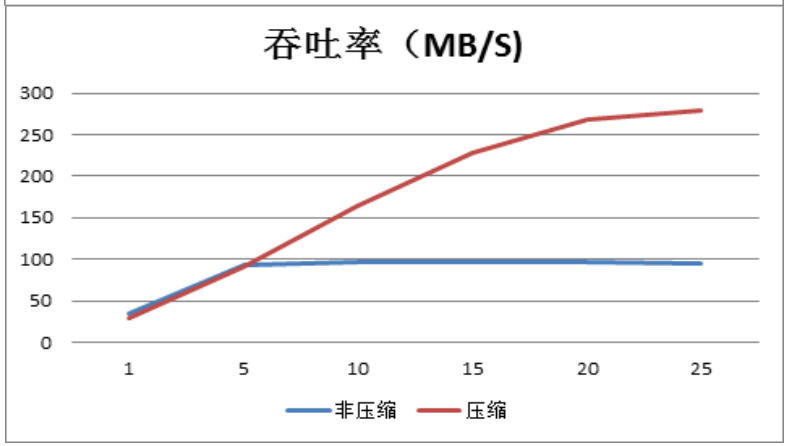

处理效率（个/毫秒）
14
12
10
8
6
4
2
0
10
50
80
100
120
150
180
200
非压缩
压缩


处理效率（个/毫秒）
30
25
20
15
10
5
0
1
5
10
15
20
25
非压缩
压缩


平均延迟（毫秒）
30
25
20
15
10
5
0
10
50
80
100
120
150
180
200
非压缩
压缩


平均延迟（毫秒）
3
2.5
2
1.5
1
0.5
0
1
5
10
15
20
25
非压缩
压缩


吞吐率（MB/S)
160
140
120
100
80
60
40
20
0
10
50
80
100
120
150
180
200
非压缩
压缩


吞吐率（MB/S)
300
250
200
150
100
50
0
1
5
10
15
20
25
非压缩
压缩

应用举例

- 商品订单
  - 每年500TB
- 库房流水记录
  - 每年超过1PB
- 商品图片
  - 近百TB，持续增长
- 消息队列服务
  - 开发中

相关工作

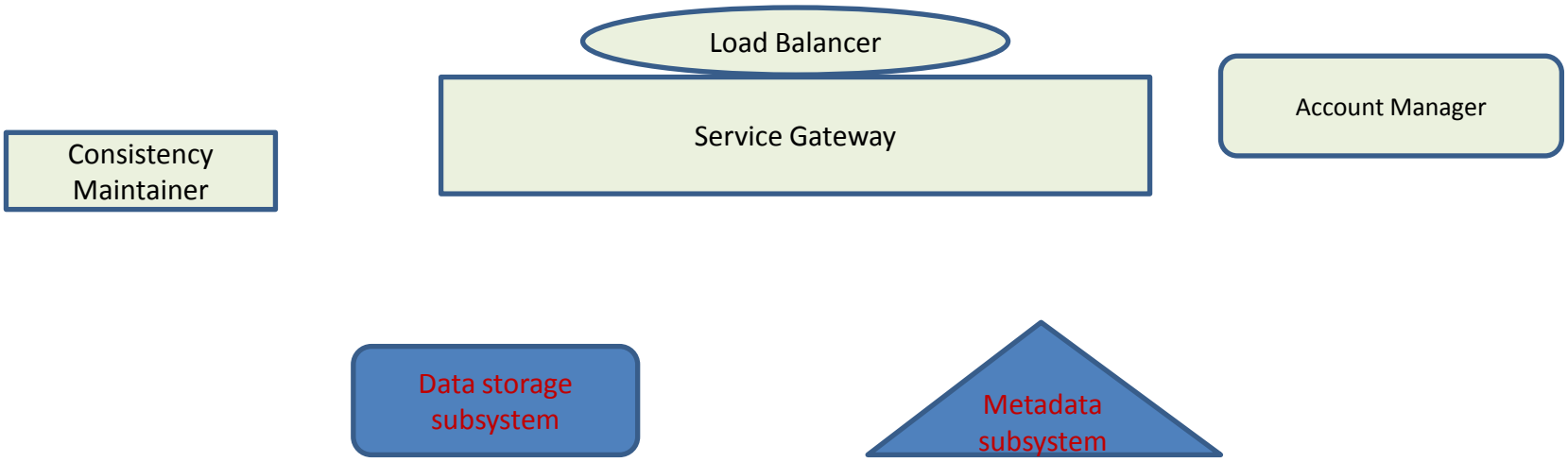
- 类似系统
  - Facebook’s Haystack
  - Taobao’s TFS
  - FastDFS、Weed-FS、…
- Jingdong Filesystem V1
  - **更重要的数据**
  - 强一致性
  - 无单点故障
  - 无内存索引
  - 透明压缩, et al.

1. 为什么自主研发
2. 实现技术与经验
3. 正在进行的工作

重新审视需求

- 核心业务的海量小文件
  - 交易订单、商品图片、库房记录、消息队列...
- 云存储服务
  - 面向私有/公有云的对象存储服务
  - 针对IaaS平台的持久块设备存储

对象存储基本架构

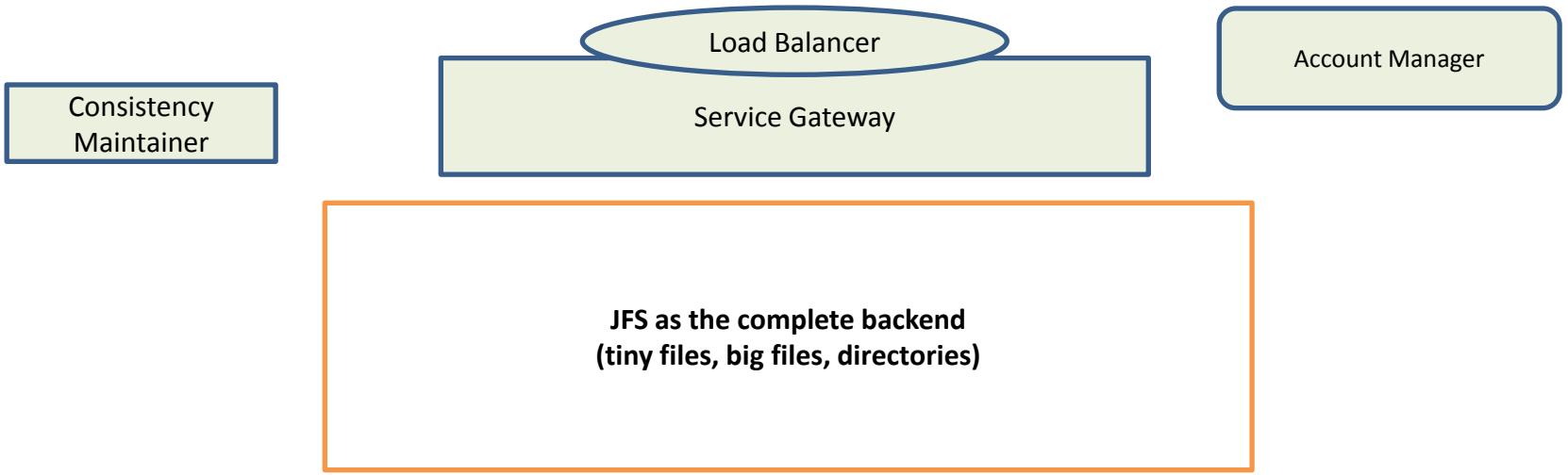

Load Balancer
Service Gateway
Consistency Maintainer
Account Manager
Data storage subsystem
Metadata subsystem

object storage evolution inside JD.com

- 之前版本
  - HDFS作为数据存储子系统
    - Pains：85%容量是小于1MB的对象
  - MySQL sharding负责元数据管理
    - Pains： manual partitioning
- 现在版本
  - JFS替换HDFS负责小文件存储
- 下一版本
  - JFS as the ONLY backend

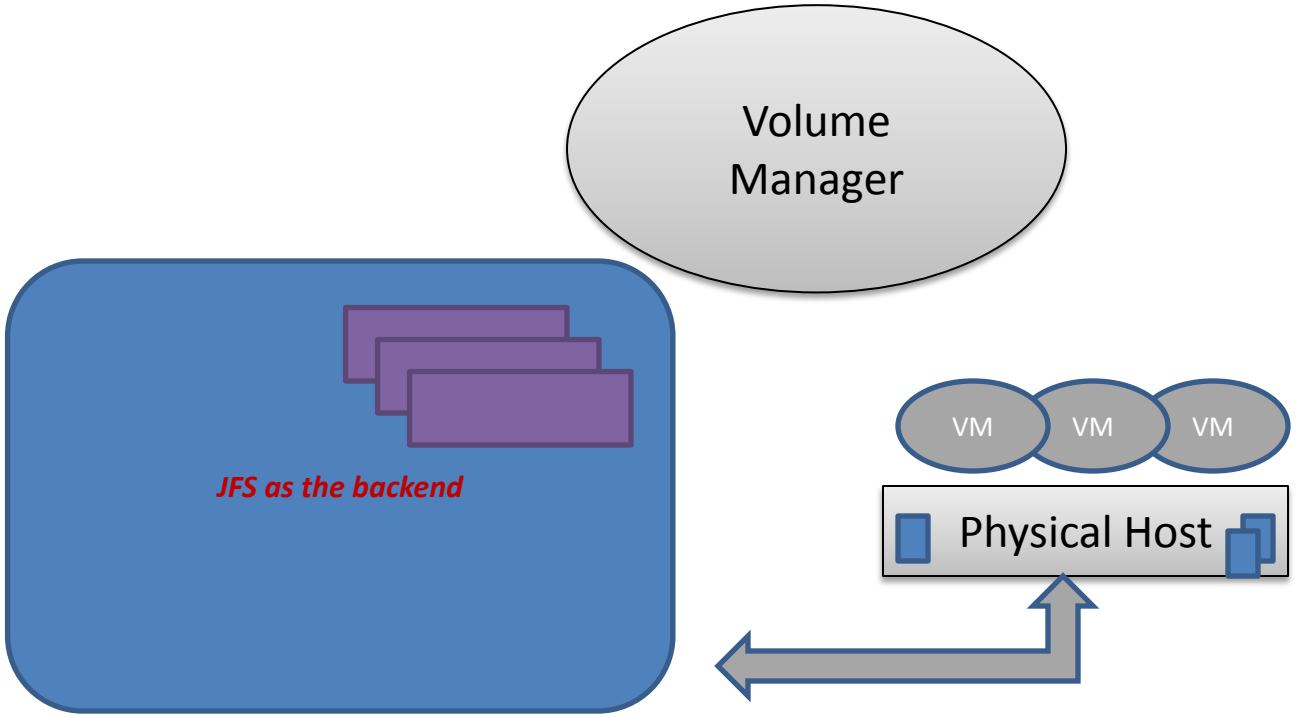
JFS-v2： Scalable Directories

- 作为对象存储服务的完整后端
  - Bucket -> Directory
- JFS Directory 特性
  - 单级目录，自定义文件名与属性
  - 单个目录内文件数不限
  - 支持目录内prefix/range查询、有序遍历



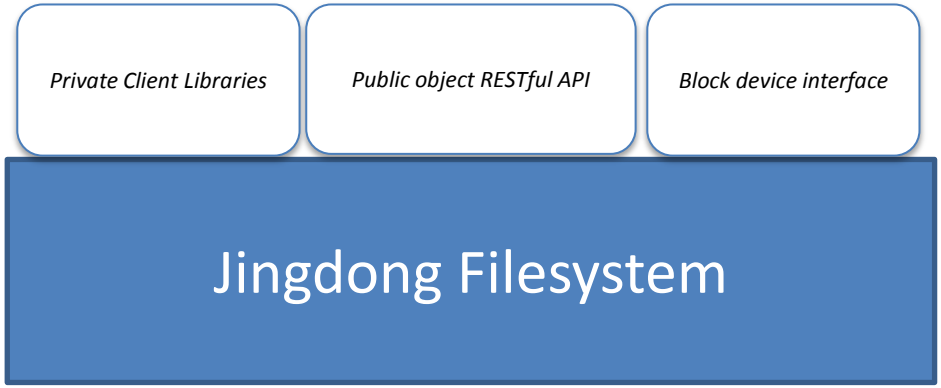
JFS-v3： Block-device Files

- 预分配、定长、独占写
- 在此基础上提供持久块存储服务



愿景

- 统一的存储后端，提供不同产品抽象
  - 内部各子系统有独立的复制协议与存储引擎



总结

- Jingdong Filesystem
  - 自主研发，分期推进，逐步扩展
  - 首先针对核心业务的海量在线数据
    - 商品订单、库房记录、图片 …
  - 将作为统一云存储服务的基础平台
    - objects, queues, tables, block devices

- Thank You!

- And stay tuned☺

JD.COM 京东