

B站上传调优实践

自我介绍

1. 唐君行 - Json
2. Bilibili视频云工程师
3. 负责上传，存储，点播CDN的建设和架构演进
4. 擅长系统架构设计和数据驱动研发

Agenda

1. 系统架构
2. 建立指标
3. 质量优化
4. 总结和TODO

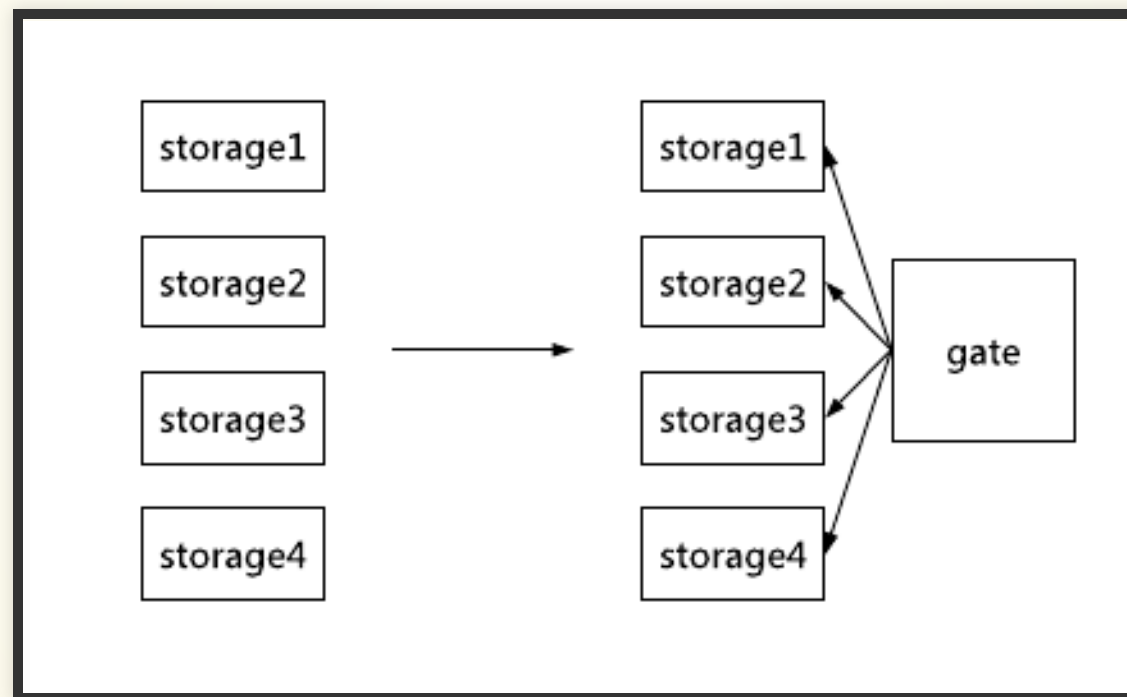


创作中心

上传业务概览

1. Up主需要答题才能获取上传资格
2. 多为优质长视频，平均体积400MB
3. 日上传量15w，保持年100%增长
4. 支持网页，iOS，Android，PC客户端

UpOS



专用上传服务=>通用对象存储的重构

UpOS

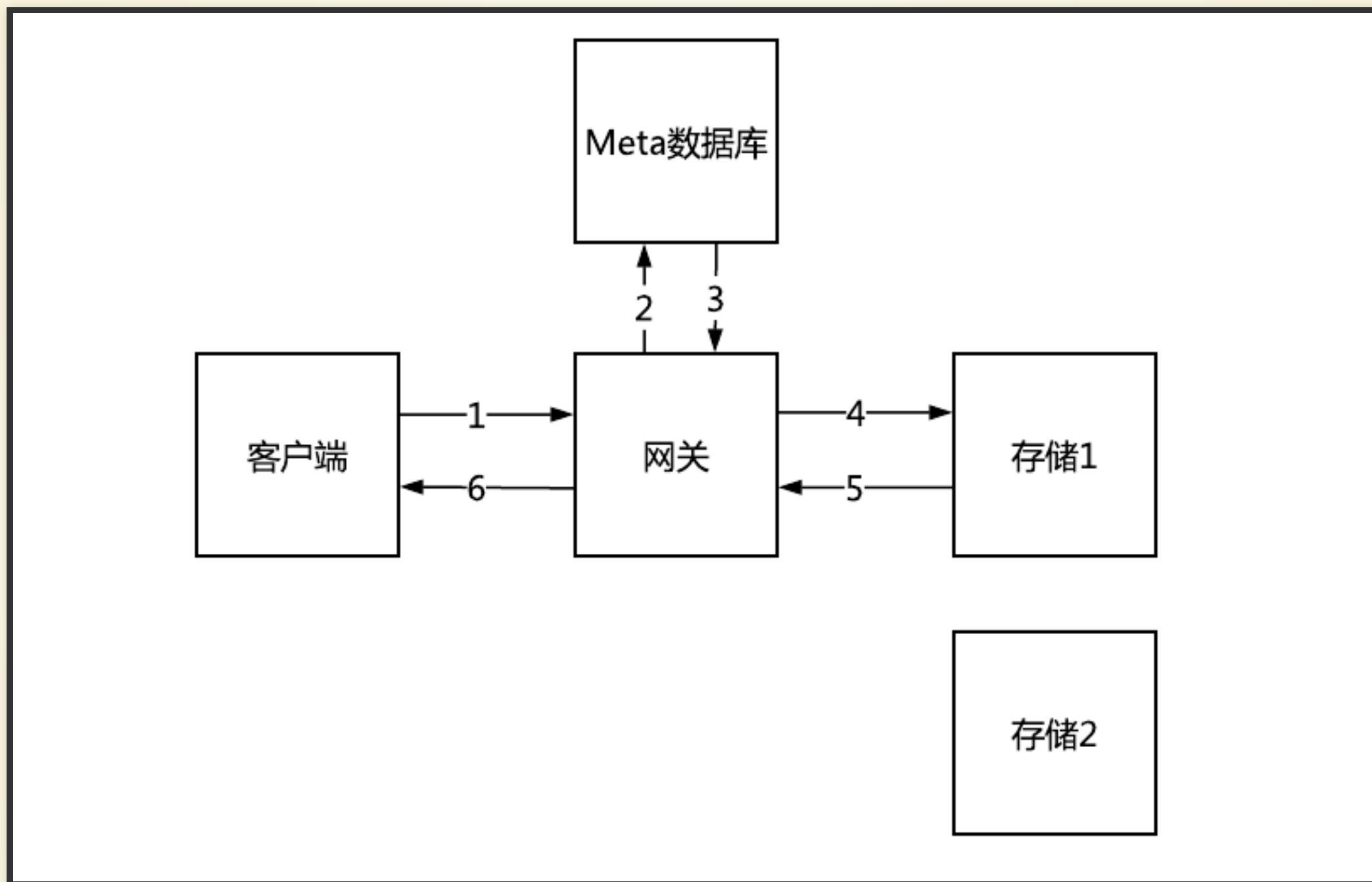
1. 设计成于业务无关的对象存储架构，可以分Bucket配置不同的参数
2. 承载B站UGC，PGC，音频，短视频的上传，处理，回源
3. 上传，存储体系统一，减少文件在系统间转移产生的网络IO

	OpenResty	Node.js	Go	PHP
Time(seconds)	1.328	4.133	2.171	3.368
Transferred(bytes)	1750000	1130000	1290000	1790000
RPS(#/s)	7529.50	2419.58	4605.53	2969.21
TPS(ms)	13.281	41.330	21.713	33.679
Transfer rate(kb/s)	1286.78	267.00	580.19	519.03

技术选型

OpenResty

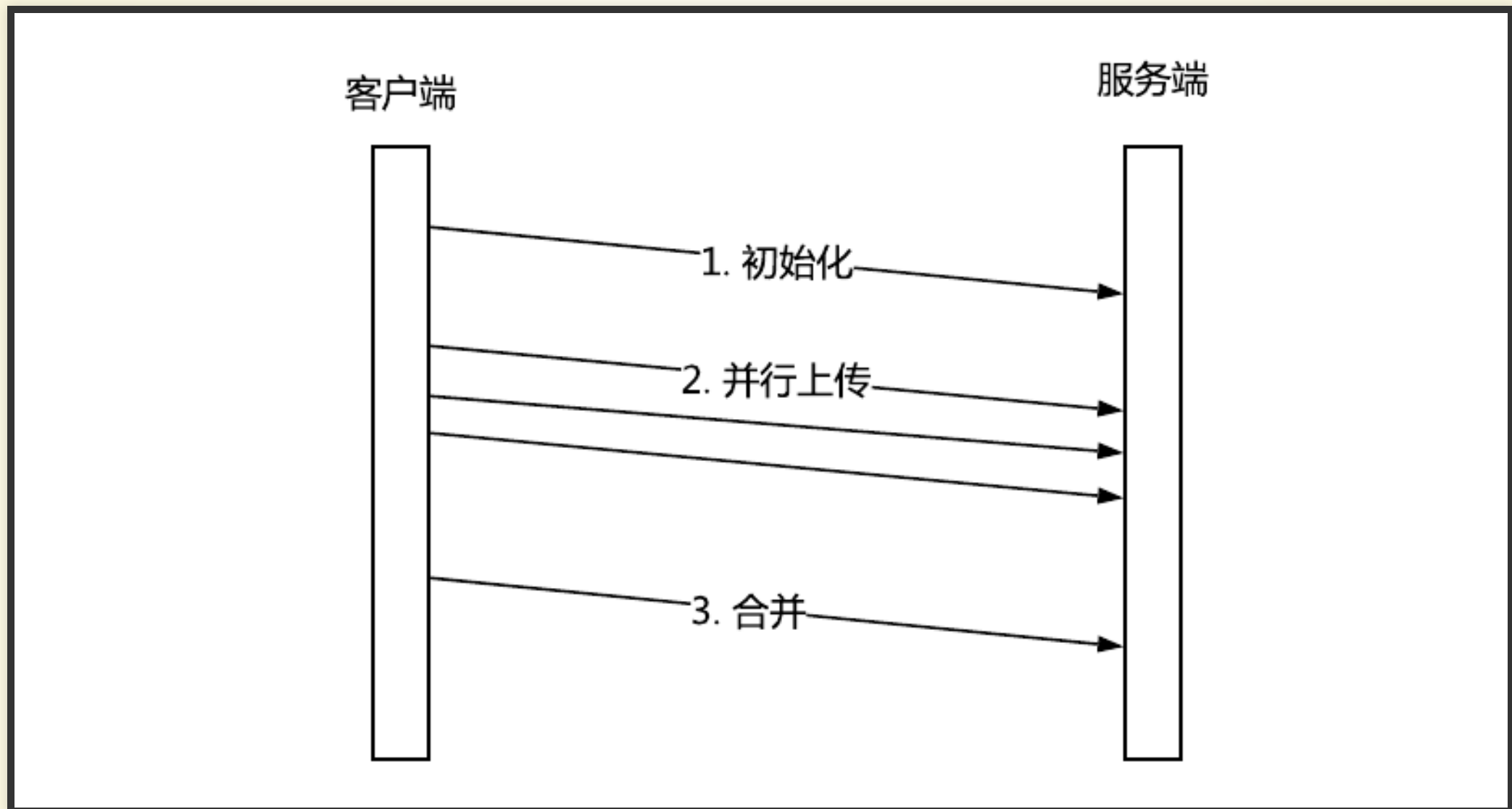
1. 同步代码实现异步IO
2. Nginx本身扮演网关和业务服务，简化架构
3. 直接访问Nginx的ClientBody，无语言框架的损耗
4. Lua开发效率高



存储架构

上传协议

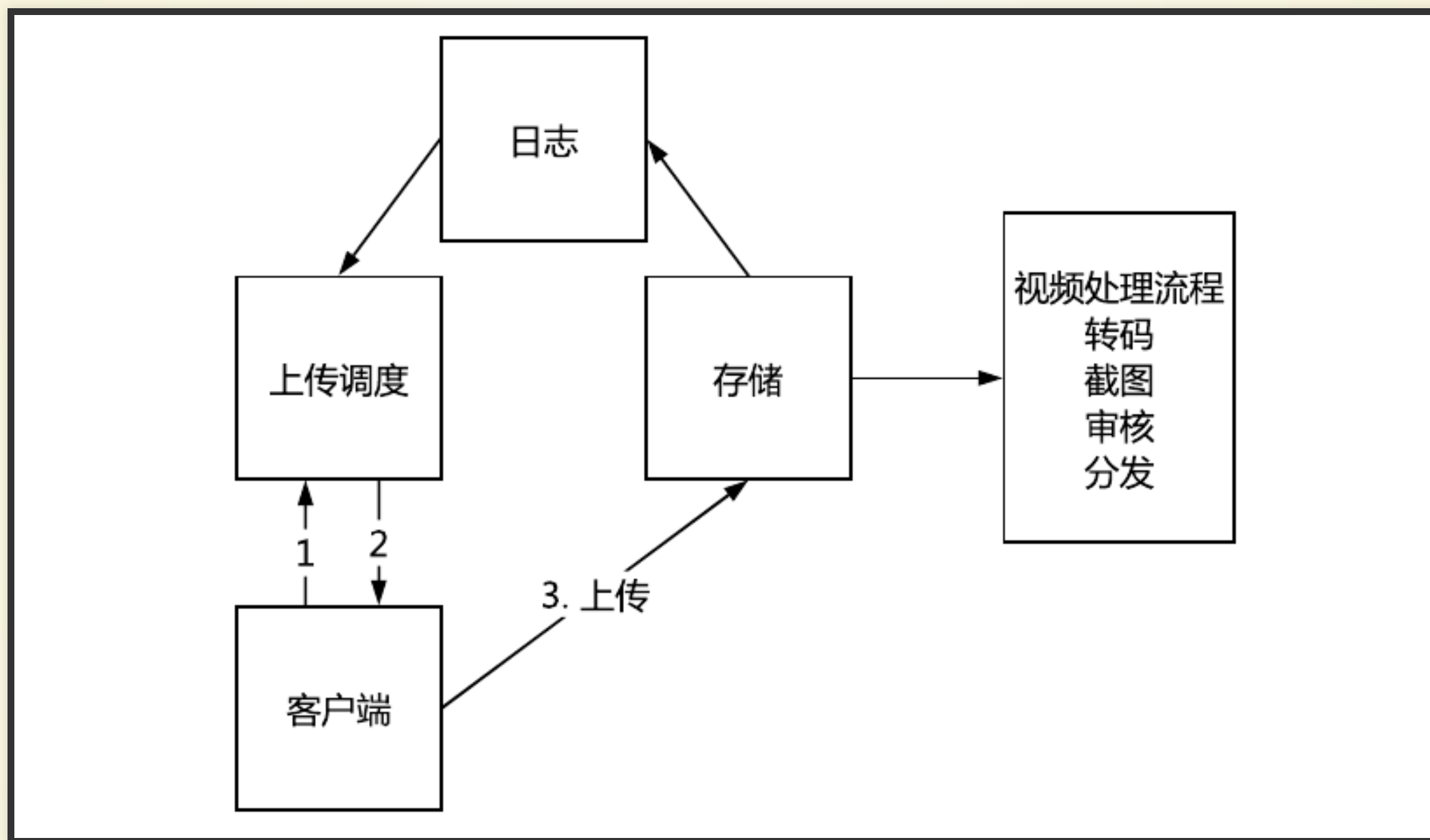
1. 实现S3上传协议
2. 可以复用S3生态中的组件
3. 分片上传，大文件上传过程中的重试单位为分片，提高整体上传成功率
4. 可以调节分片大小，并行数量，为日后调优创造条件



S3上传协议

S3上传协议

1. `uploads` 分配上传空间
2. `put` 并行上传，分片大小可以自己定义，通过 `partNumber` 来标识顺序
3. `post-complete` 按照 `partNumber` 合并成完整文件并且校验



上传架构

建立指标体系

指标	线性相关	计算简单	达到目标
慢速比	No	Yes	No
瞬时速度	Yes	No	No
全局均速	Yes	Yes	Yes
成功率	Yes	Yes	Yes

abtest, 数据闭环

1. 比较第三方cdn的上传质量
2. 比较分片体积, 并发数, 重试次数的影响
3. 比较不同平台的上传质量

链路层面优化

1. bgp直传vs节点加速
2. 接入多家上传cdn
3. ab对比上传指标
4. 节点数量决定质量

客户端选线vs服务端选线

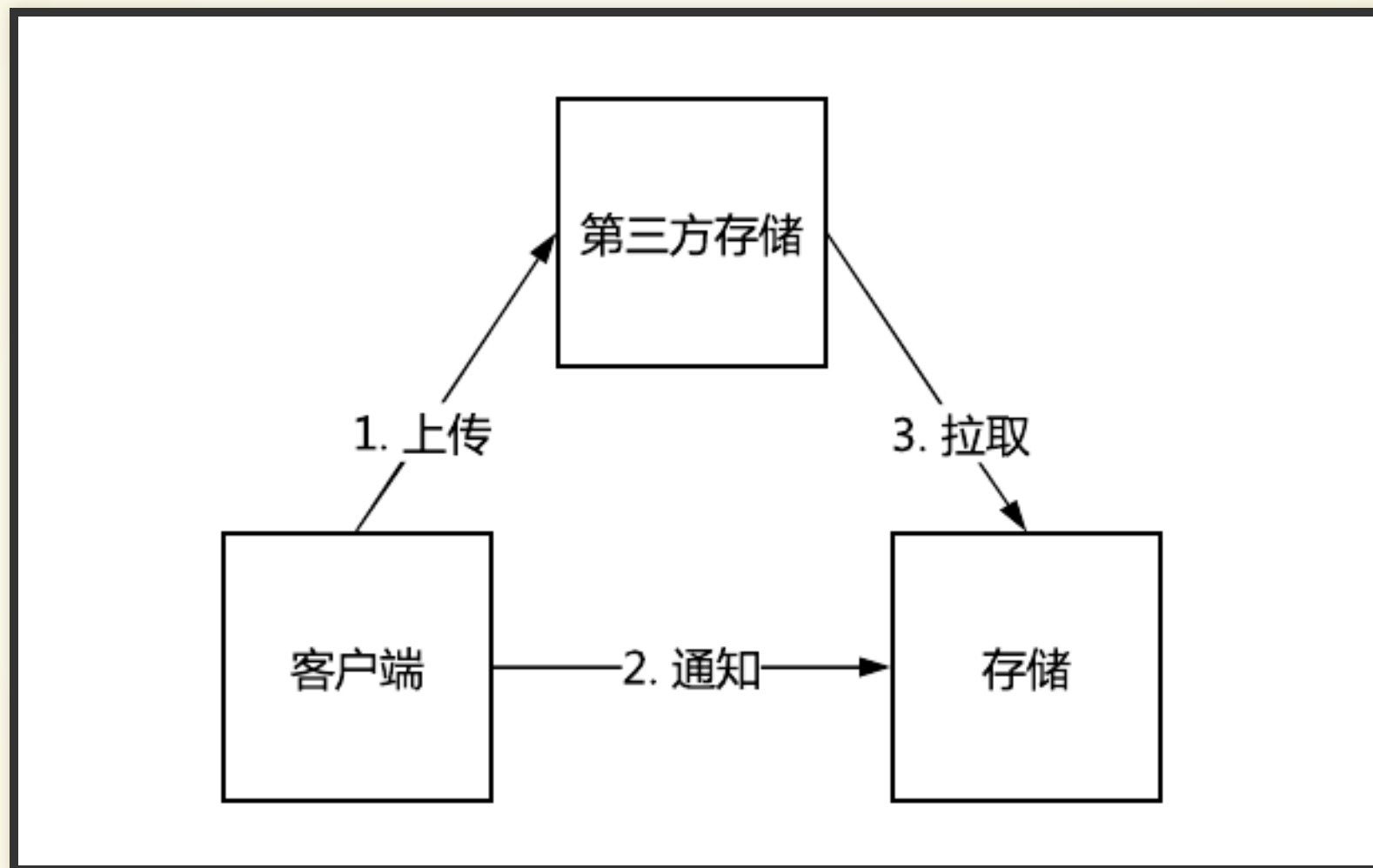
1. 服务端：ISP+地域线路调度=》效果不好
2. 服务端：优选线路，只保留成功率和速度最好的线路
3. 客户端：上传前探测

解决电信上下行带宽不对等

1. 通过IP库检测电信用户
2. 打开电信上传限制
3. 电信用户可以提高10倍

jssdk优化

1. 多文件并发=》文件内分片并发
2. 支持上传到第三方存储(阿里，腾讯，百度，七牛)



成本低，效果好

跨会话上传控制

1. 发现周末上传成功率下降
2. 分析周末Up主人均上传数量多
3. 使用js的localStorage实现锁，提高成功率1%

移动端上传优化

1. 服务端下发参数，便于数据驱动优化
2. 上报网络环境，主机型号等
3. 优选并发数,分片体积,重试次数,重试间隔

总结和成果

1. 重构系统=>建立指标体系=>优化指标=>达成目标
2. 成功率85% => 94%
3. 速度 400KBps => 1800KBps
4. 投诉 => 0

TODO

1. Quic上传优化移动端上传体验
2. 动态切换上传线路

引用

- Go在网易广域网上传加速系统中的应用
- 性能对比



Thanks