

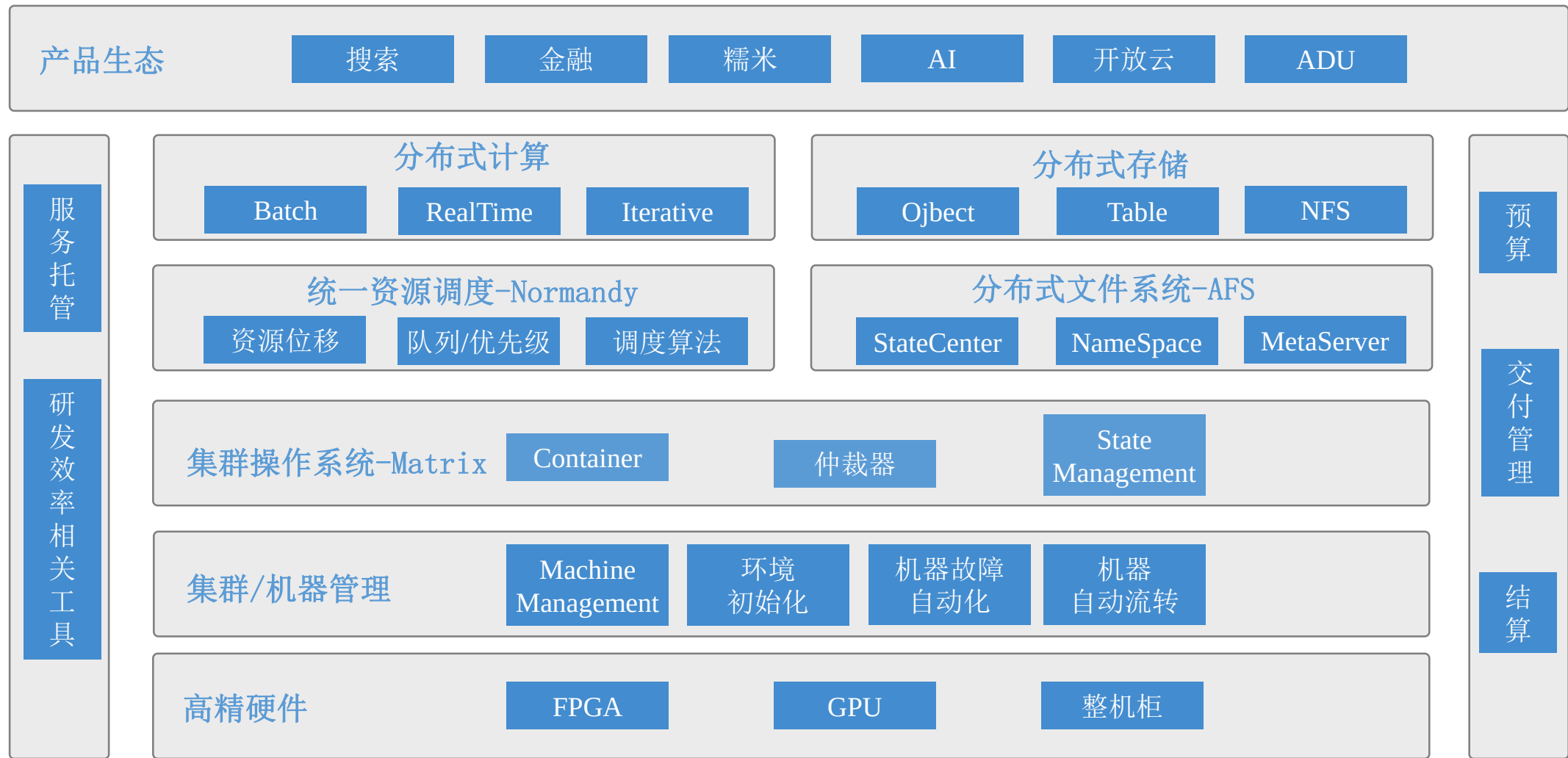


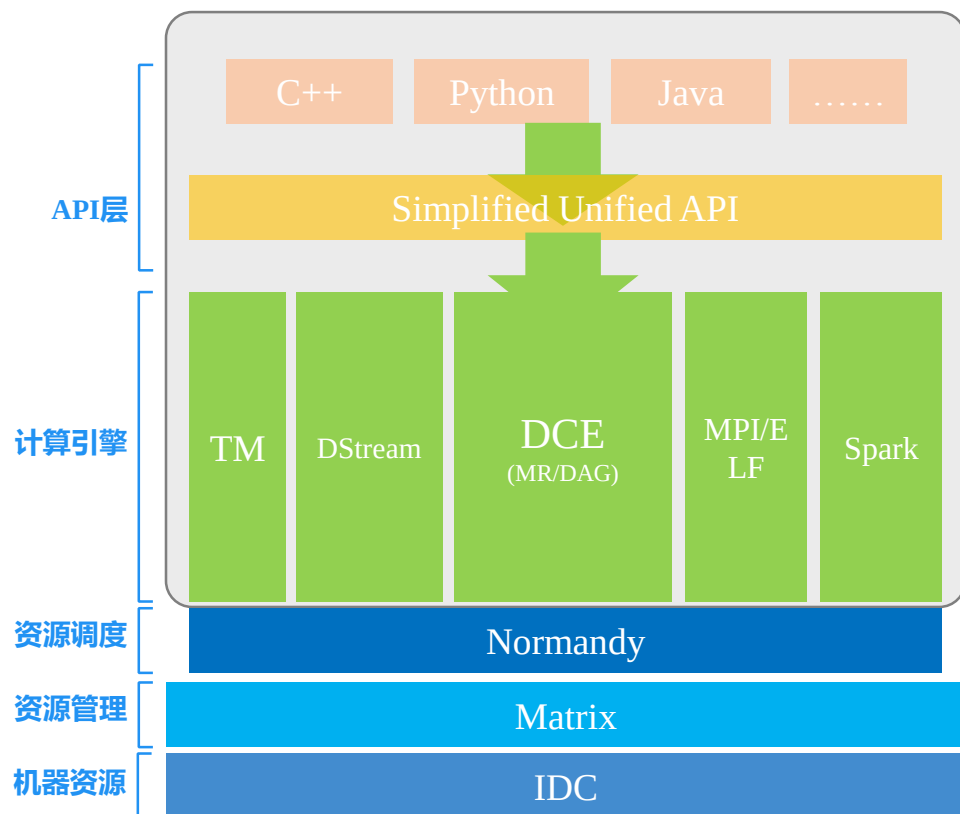
百度大数据离线计算平台流式Shuffle服务

百度 张建伟

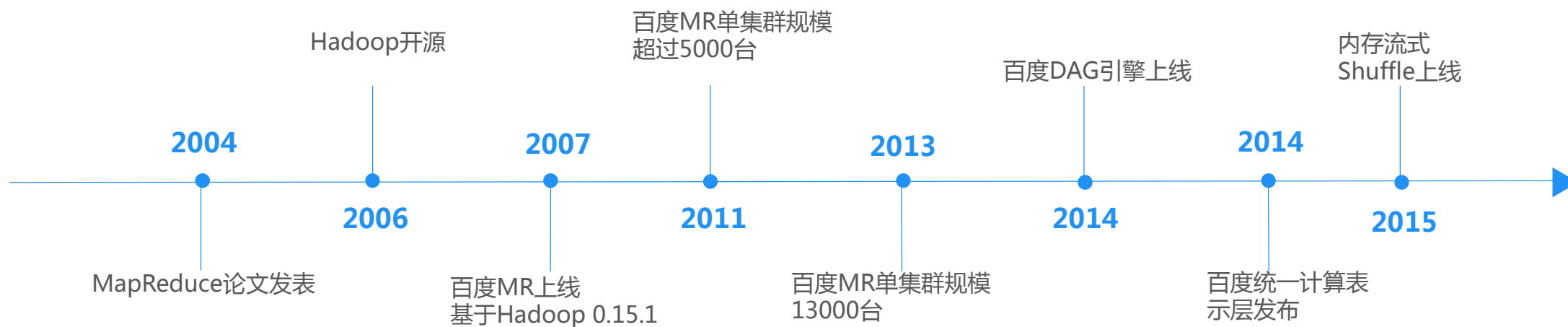
- 背景
- 架构
- 关键技术
- 收益与总结
- 下一步计划

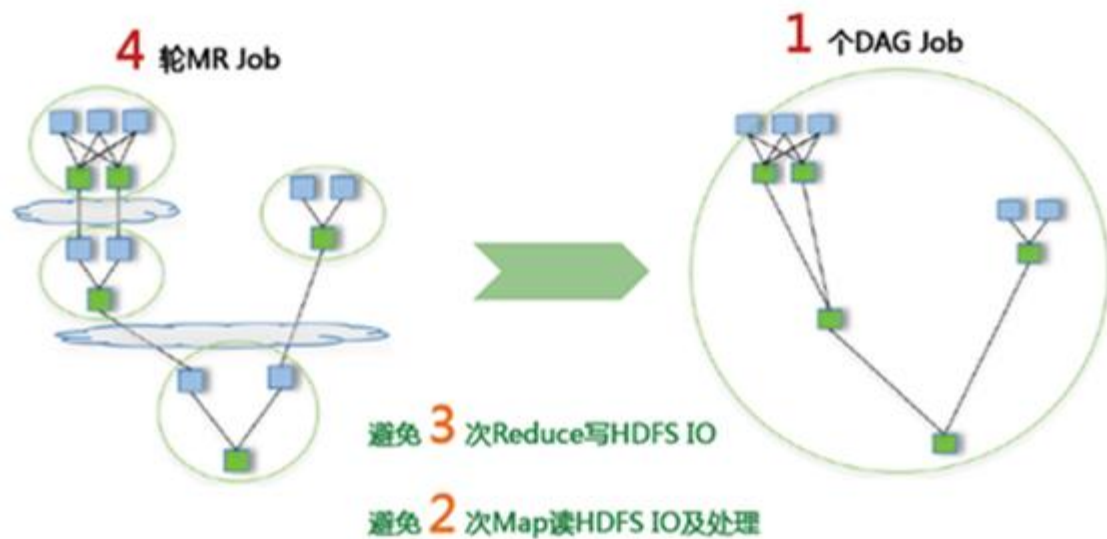
- 背景
- 架构
- 关键技术
- 收益与总结
- 下一步计划





背景-百度大数据离线计算平台发展历程





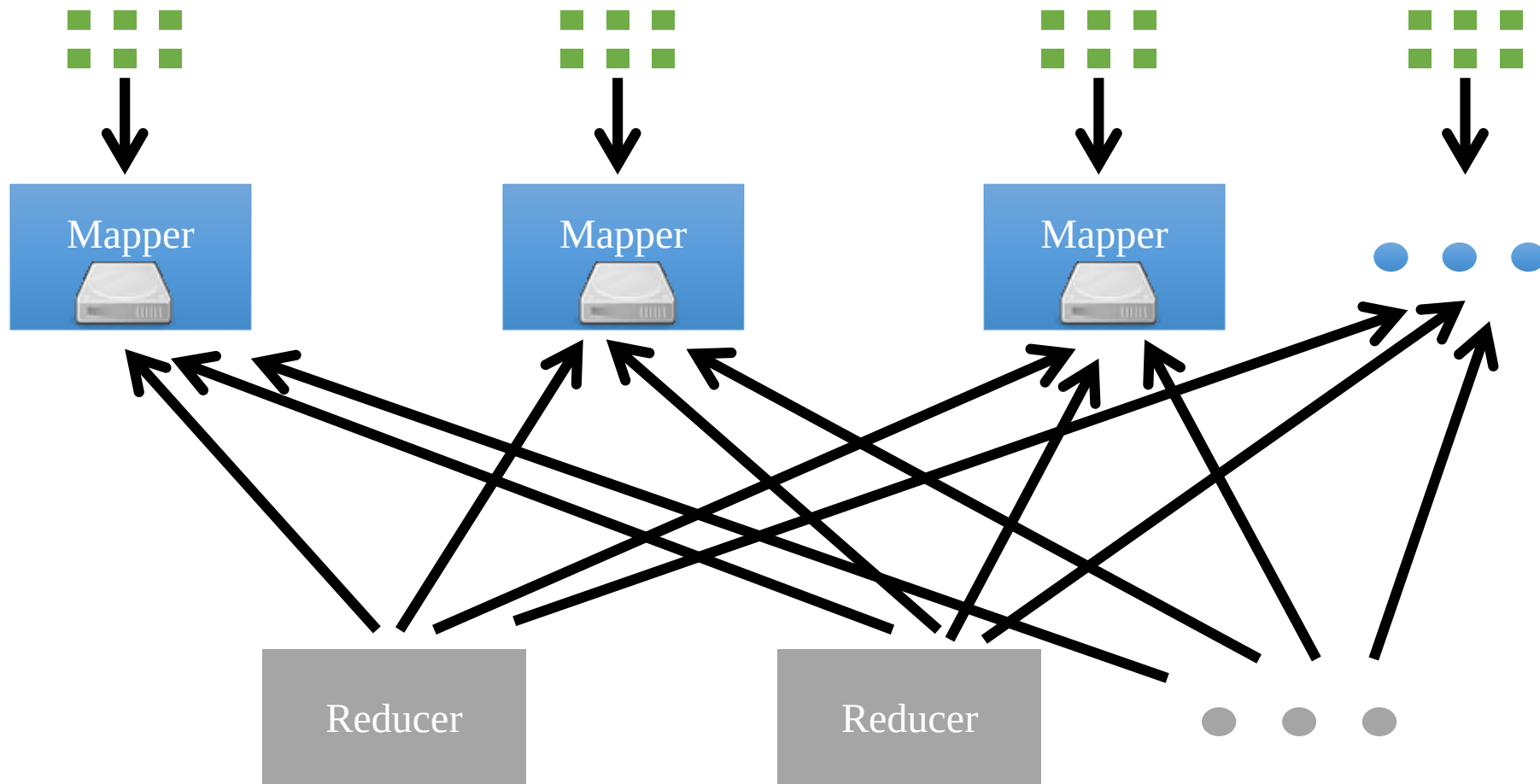
基于MR引擎，需要翻译成 **25个** MR Job

基于DAG引擎，只需要翻译成 **1个** DAG Job

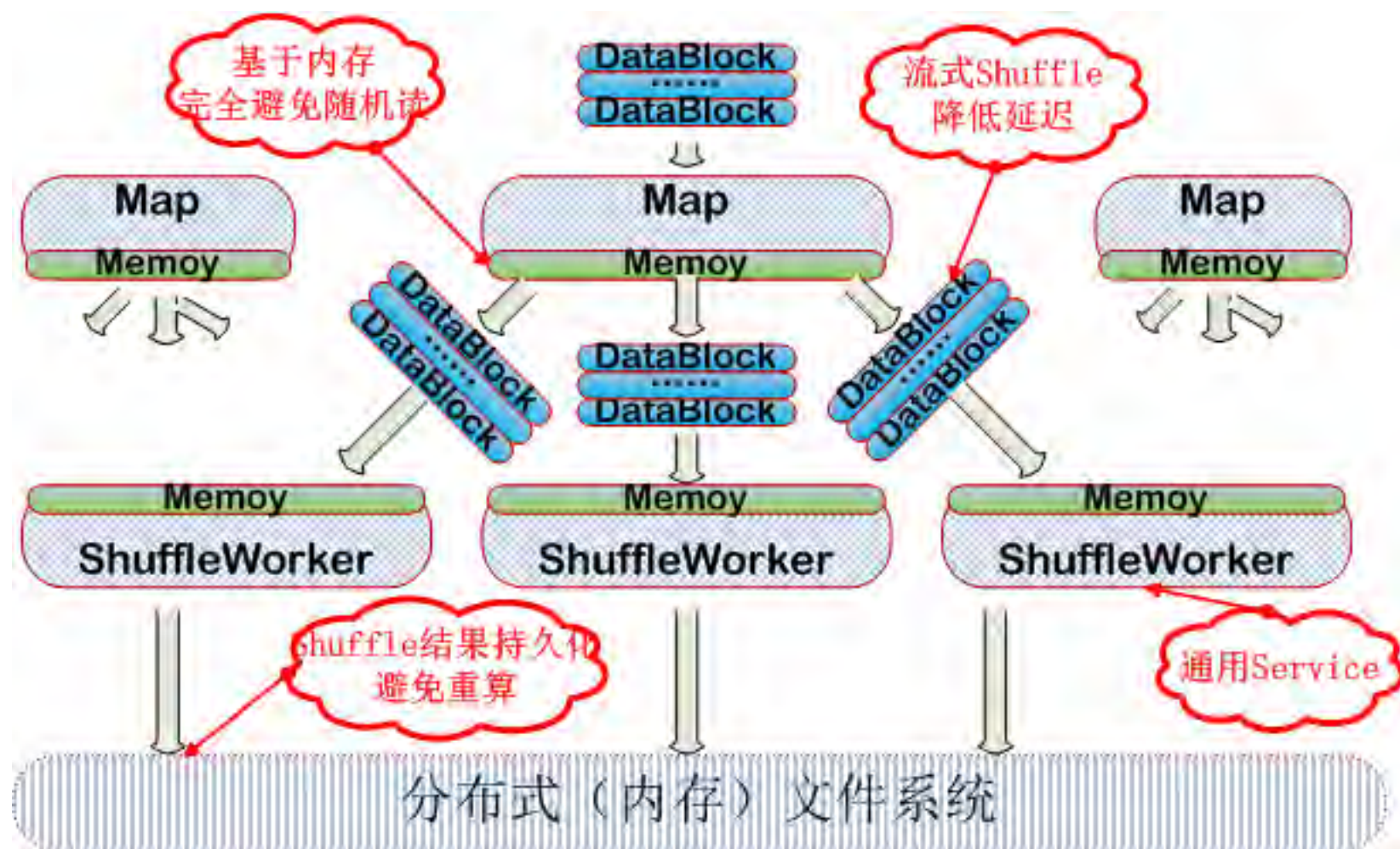
运行时间从 **5小时** 缩减到 **1小时**



背景-一般的Shuffle模式



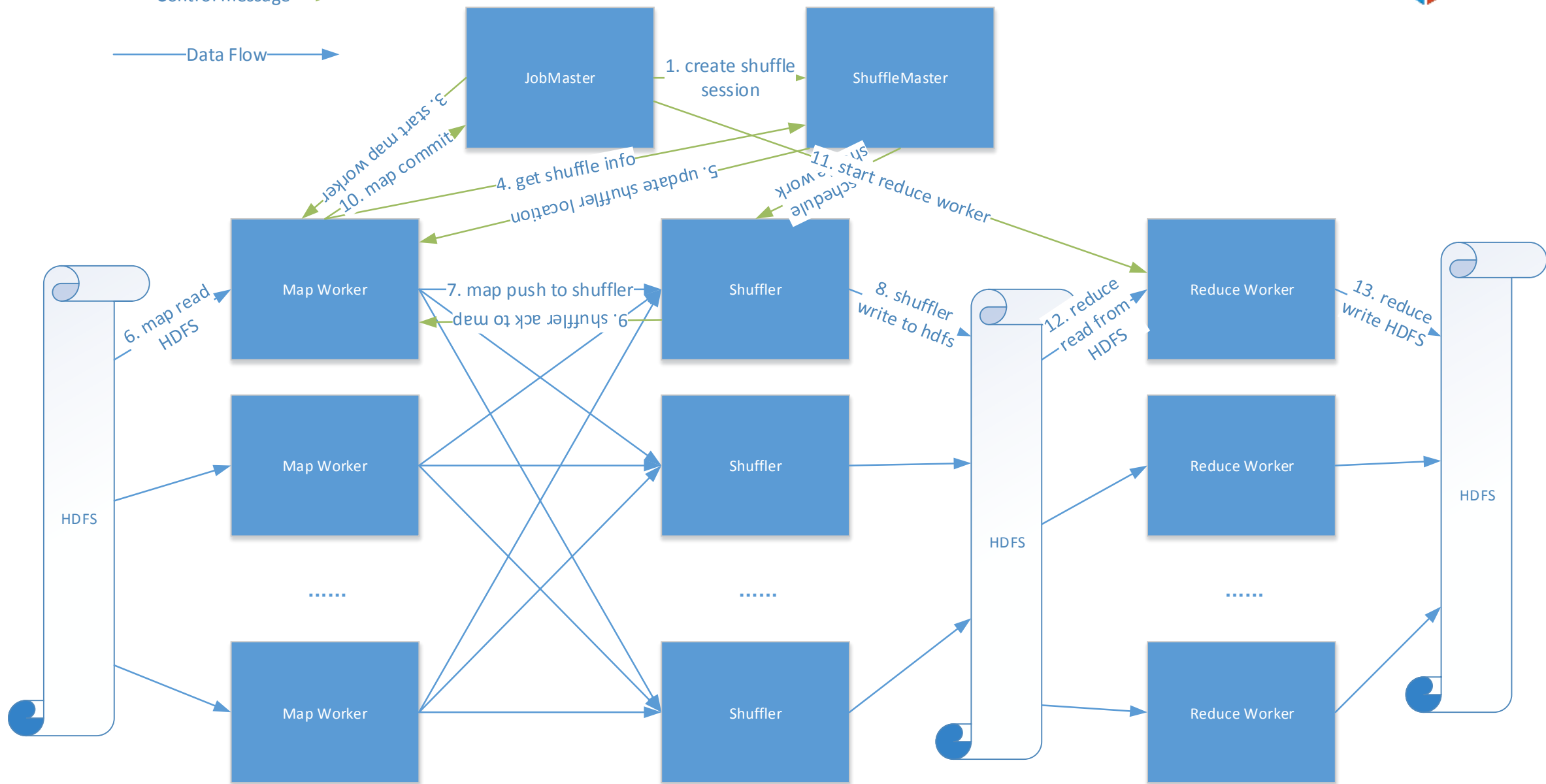
- 背景
- 架构
- 关键技术
- 收益与总结
- 下一步计划



架构

Control message

Data Flow



- ShuffleMaster
- Shuffler(Shuffle Worker)
- Writer
- Reader
- Session
- Shard

- 背景
- 架构
- 关键技术
- 收益与总结
- 下一步计划

关键技术-ShuffleMaster

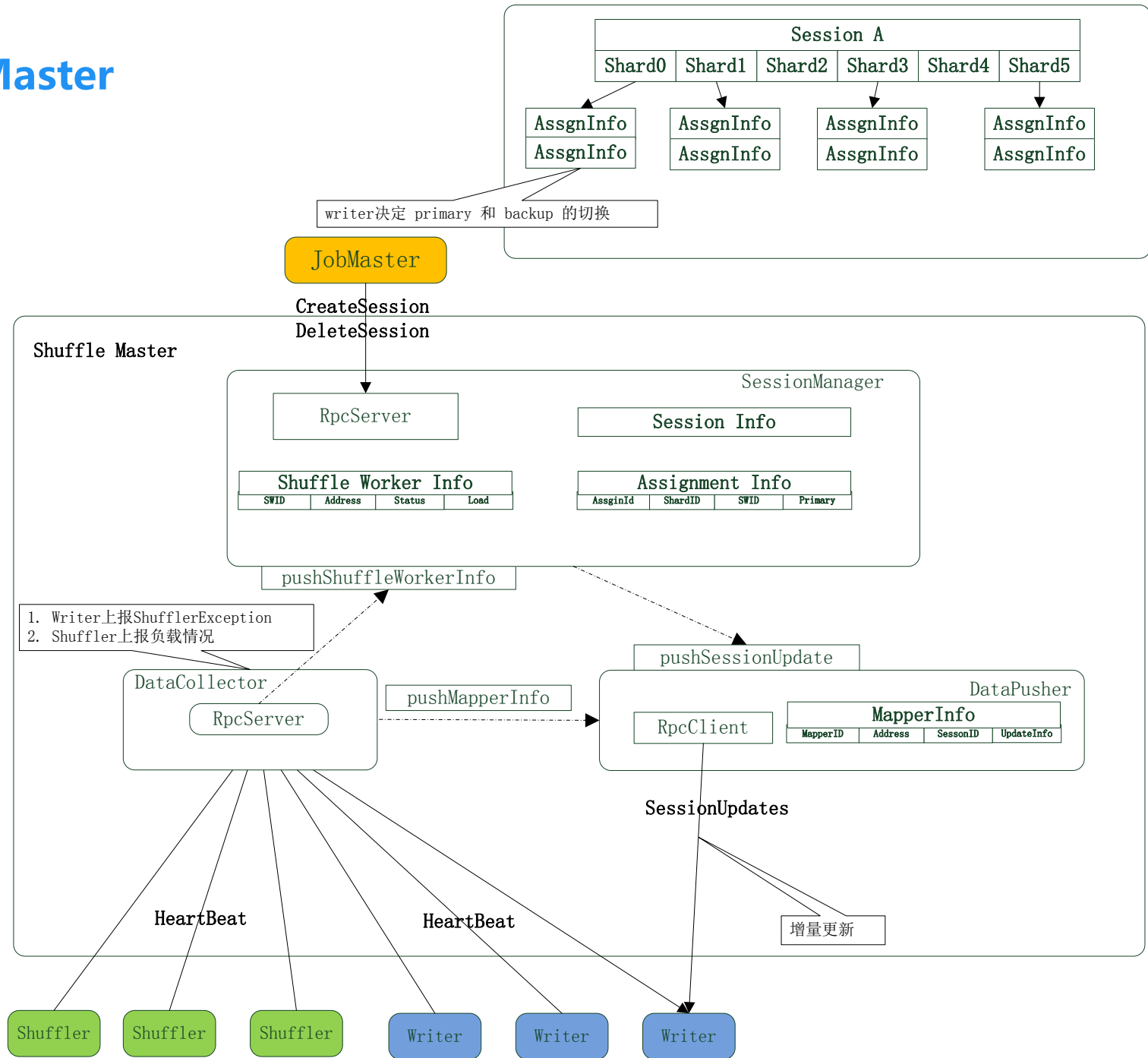


- 智能调度

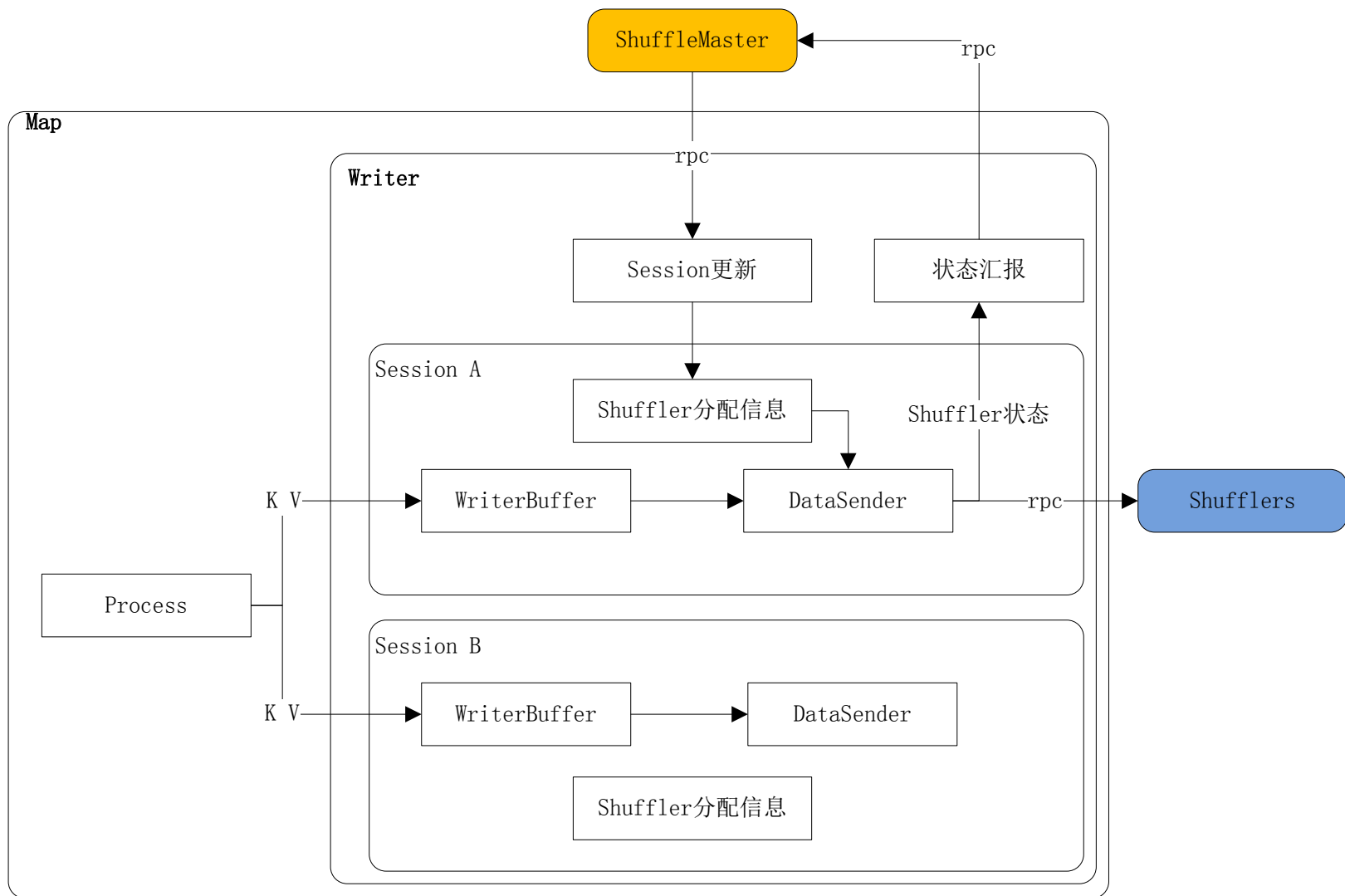
- 全局视图
- 异常检测
- 负载均衡

- 负载均衡

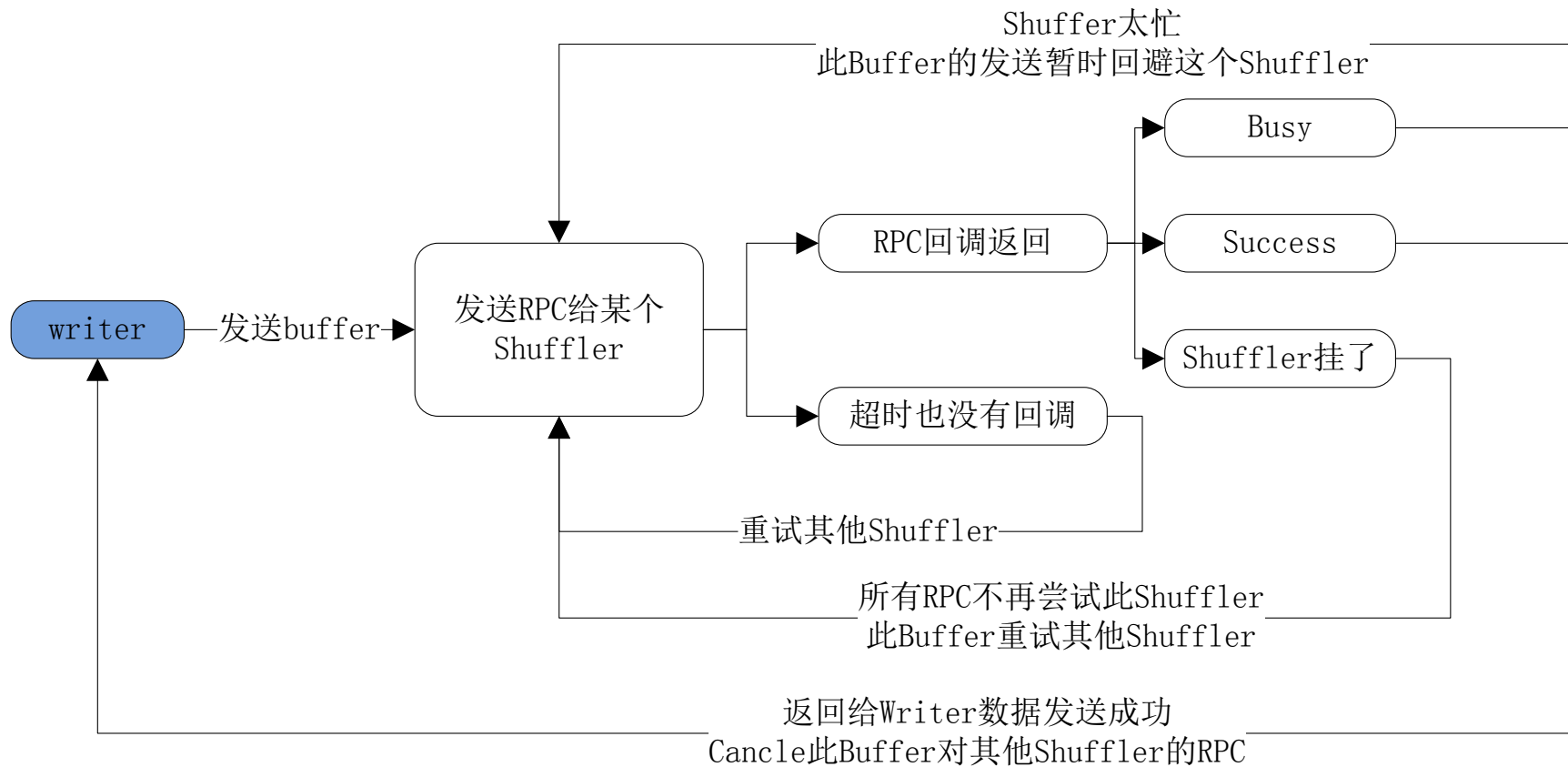
- Shuffler
- Shard



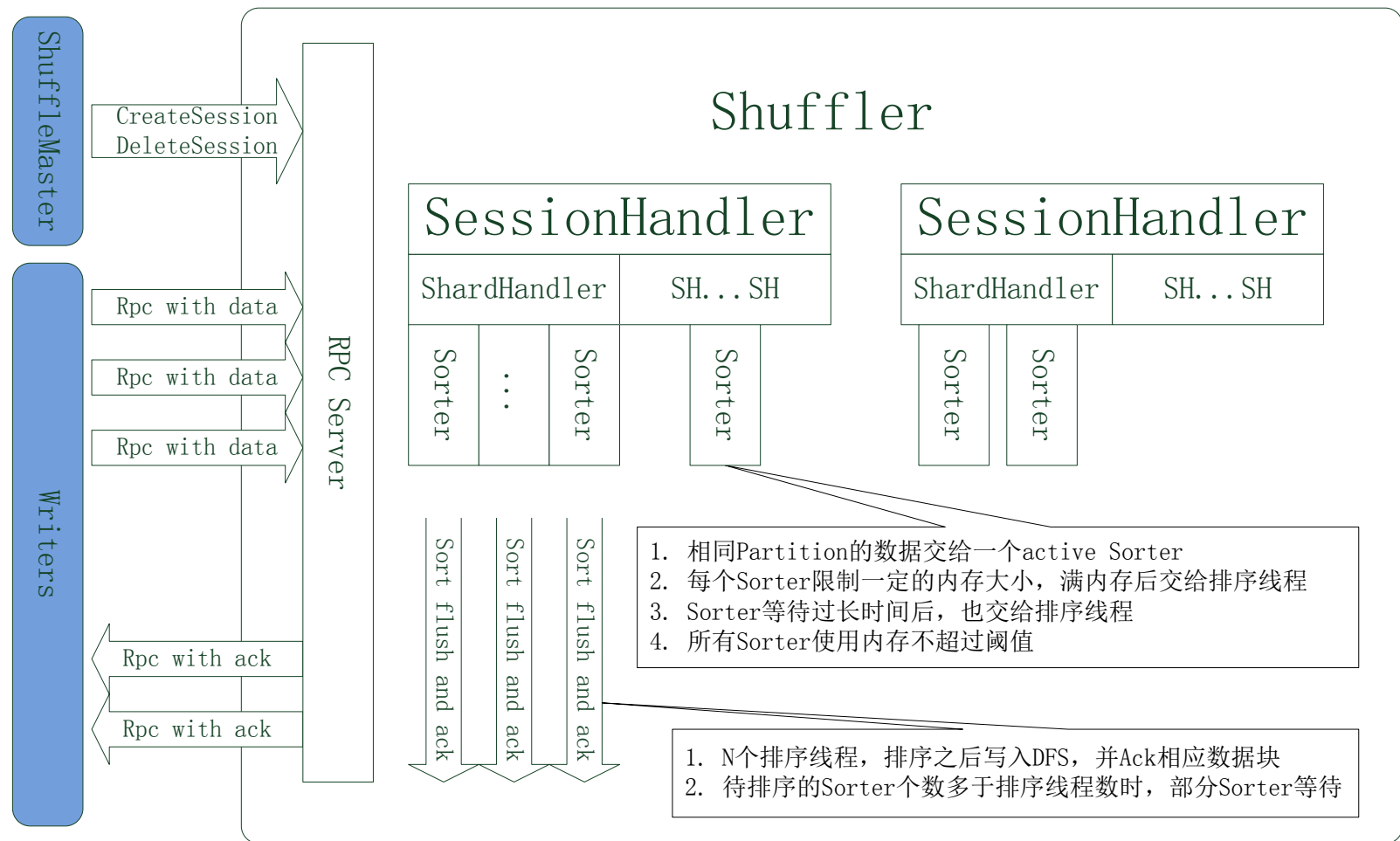
- 数据缓存与异步发送
- 异常处理



• 异常处理

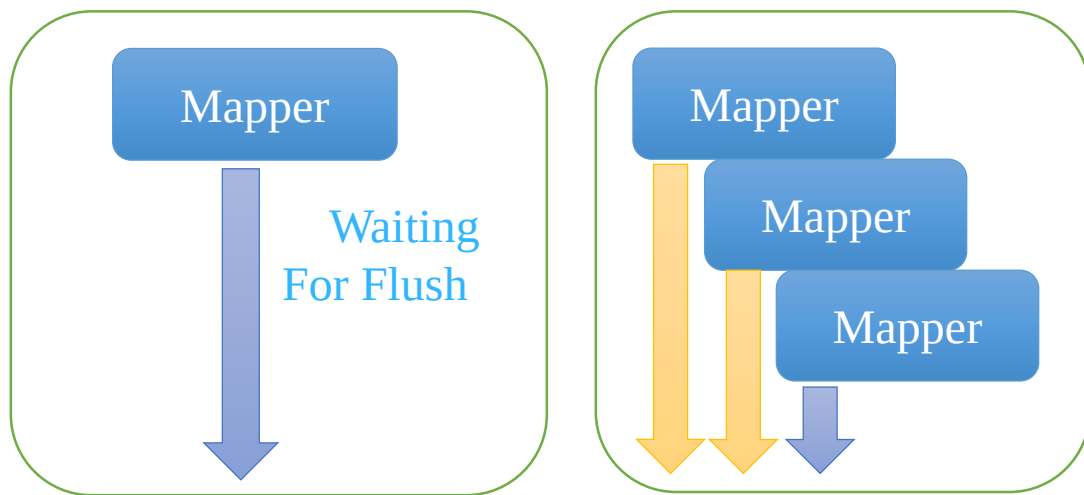


- 内存聚合
- 流控
- Sort&Flush

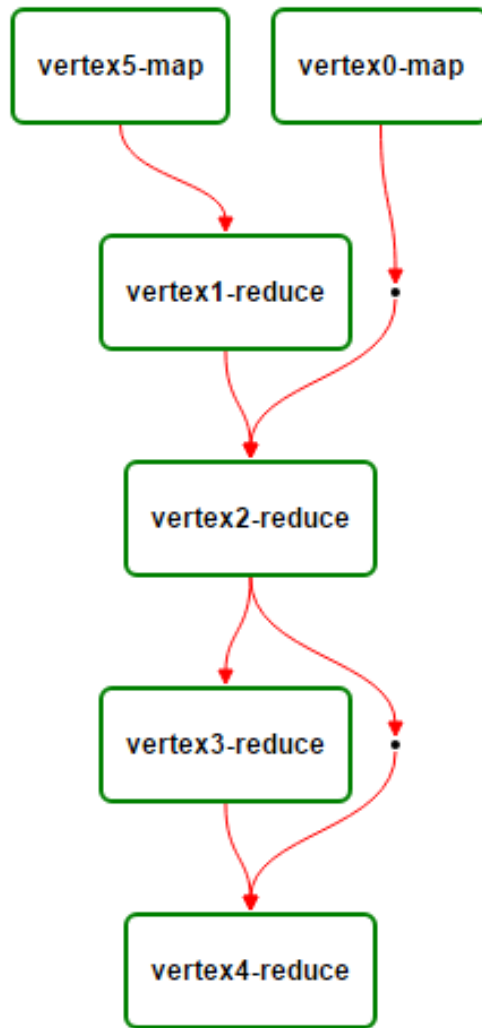


- 所有Map完成后，调度Reduce
- 直接读取DFS排好序的数据
- 去重&数据验证
- 多路归并排序

- Map端Writer，要等Shuffler将数据持久化到DFS后，才能将发送的rpc buffer释放
- 所有发送的数据被Shuffler持久化后，Map才能安全退出

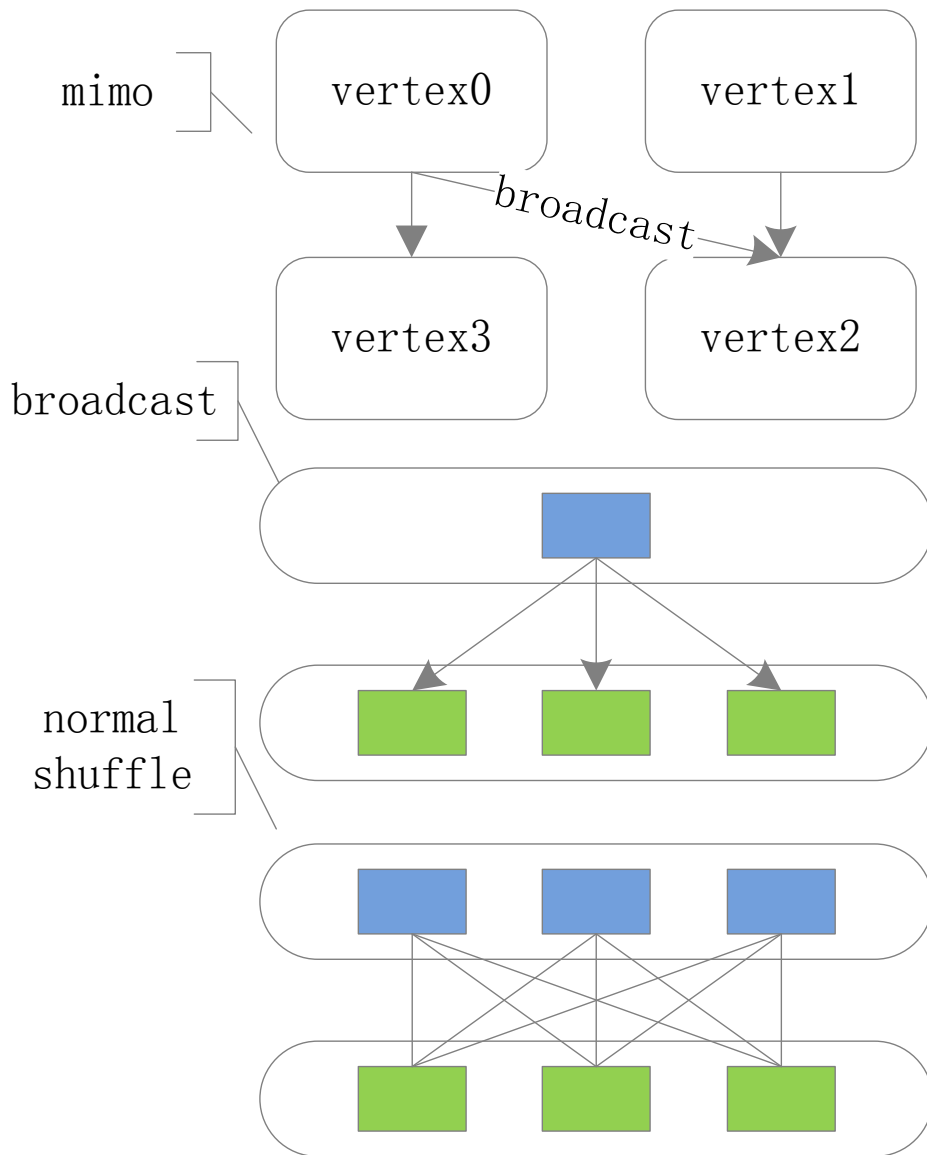


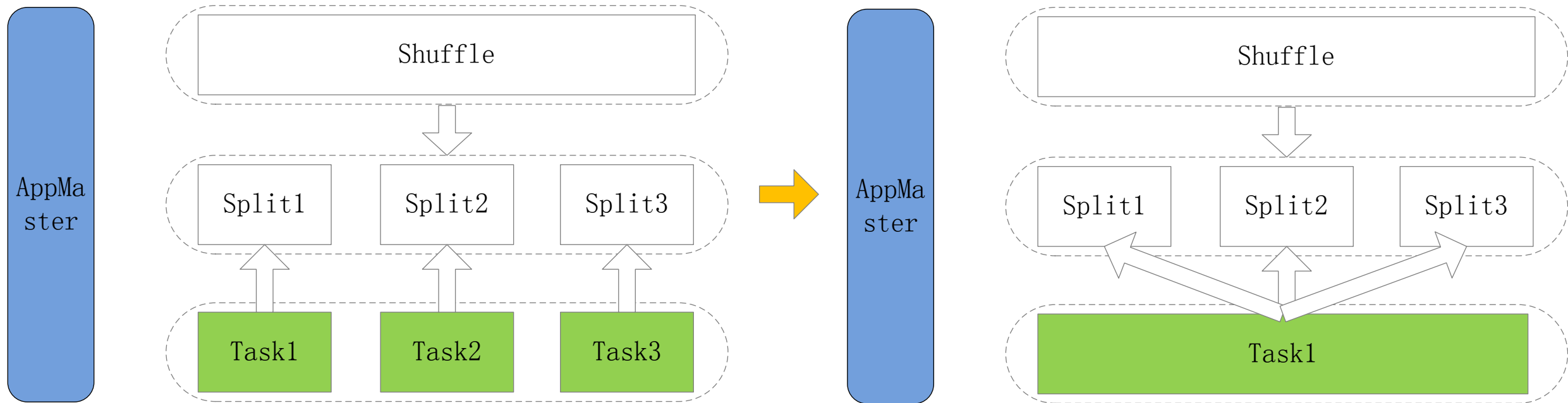
- 无MIMO时：
Vertex3和vertex4
收到vertex2的全
部两路数据再做
filter。此业务作
业多shuffle近10T
数据



• 方案

- 不同边可对应不同 session
- 不同 session 对应不同的 dfs 结果目录

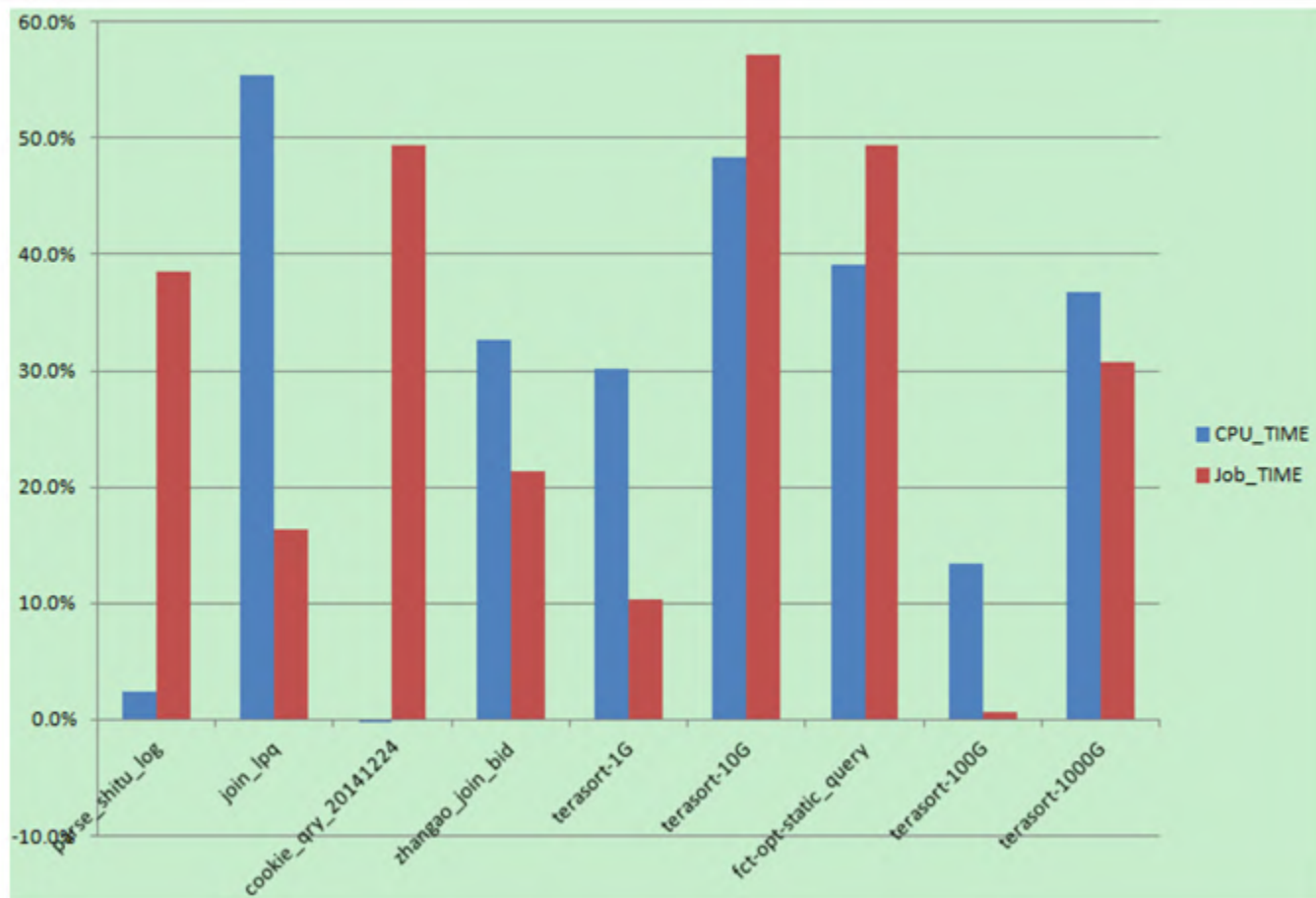




- 背景
- 架构
- 关键技术
- 收益与总结
- 下一步计划

• 收益

- 流式shuffle：减少旧shuffle map merge、reduce pull时间消耗
- 内存Push：map端不落盘
- Shuffler内存聚合：聚合度高，减少map端seek，减少reduce端merge路数，减少IO
- Pipeline：大大提高中小作业map端运行速度
- 中间数据持久化：避免重算（对dag作业尤为重要）



- 流式Shuffle服务

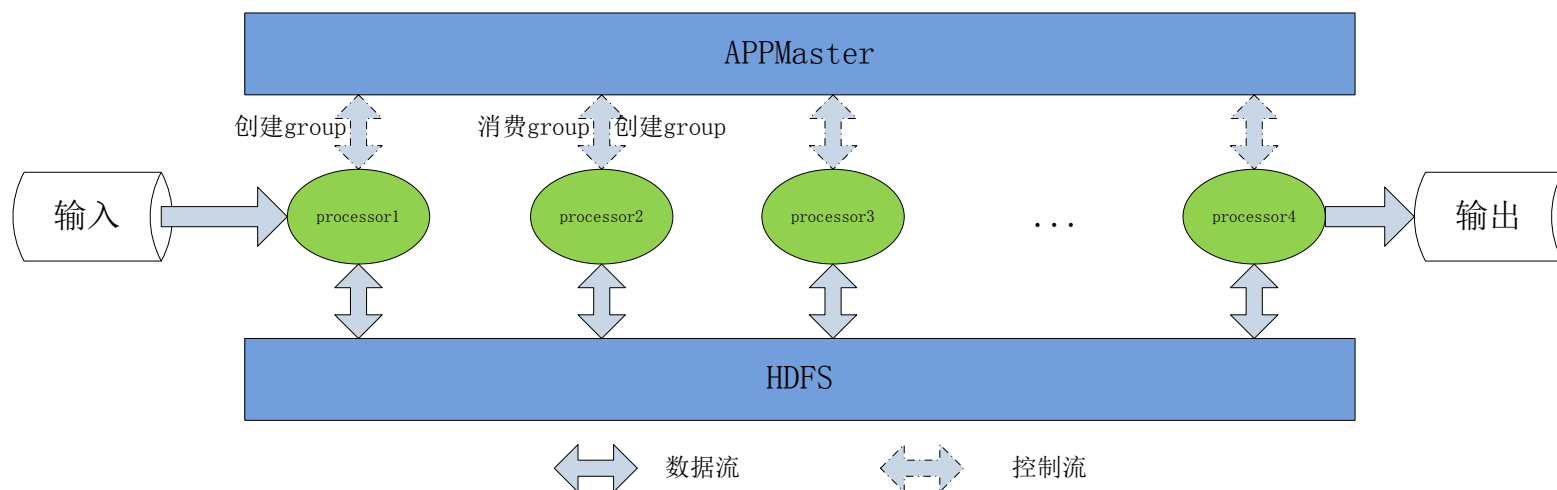
- push instead of pull
- Shuffle过程交给能拿到更多全局信息、更专业的shuffler来做
- Map、Shuffler、Reduce，每个都做自己最适合做且擅长做的事情
- Shuffle与Reduce解耦，简化Reduce，也为解决分桶不均问题提供可能
- Pipeline。无缝的流水线，减少无谓的等待

- 问题

- 更多的网络io（万兆网卡，网络不是瓶颈）
- Shuffler资源共享，作业间可能互相影响（让Shuffler资源非瓶颈）

- 背景
- 架构
- 关键技术
- 收益与总结
- 下一步计划

- Hash Shuffle(not only sort-avoid shuffle)



- 动态调整DAG拓扑

THANK YOU

cloud.baidu.com