Hadoop集群在互联网企业的应用

- 京东商城
- 百度
- 阿里巴巴

京东商城

- 源起：为POP商家进行日志分析服务

瓶颈

- 性能瓶颈：采用Oracle RAC（2节点），IBM小型机，由于数据量极大，无法满足时效要求
- 成本瓶颈：小型机再进行高配和节点扩展，价格太贵

Hadoop集群作为解决方案

- 20多个节点的Hadoop集群
- 数据定时从收集服务器装载到Hadoop集群（周期为天级或小时级）
- 数据经过整理（预处理）后放进数据仓库系统，数据仓库是基于Hive架构的，使用Hive的主要原因是技术人员基本都是基于Oracle数据库的技能，由于Hive支持SQL查询，因而技能可以平稳过渡
- 数据仓库查询统计的结果会被导到hbase，然后和应用进行连接，应用不与hive直接连接的原因，是基于效率的考虑。导出数据到hbase由自行开发的一段C程序完成。
- 应用即portal通过API与hbase连接获取数据

遇到的挑战

- Hadoop集群比较顺利，反映Hadoop项目本身已经较有成熟度。但由于Hadoop系统考虑用户权限较少，而对于大规模公司，势必要实施多级权限控制。解决的方法是通过修改源代码加上权限机制
- Hbase极不稳定，反映在某些数据导入导出连接过程里会丢失数据。判断为源代码bug，通过修改源代码解决

心得体会

- 总体来说，Hadoop项目很成功，现在整个EDW（企业数据仓库系统）都基于Hadoop。集群已经发展到>200节点。之前传闻的购买Oracle Exadata实际是用于下单交易系统，并非Hadoop项目失败。
- 大型企业成功应用Hadoop，必须有源代码级别修改的技术力量。普通的程序员转型阅读修改Hadoop源代码并不困难。
- HiveSQL和Oracle的SQL有一些差异，大约花一周时间阅读Apache的Hive wiki基本能掌握

部门结构

- 运维团队（负责管理维护集群的正常运行）
- 数据仓库团队（根据业务部门的要求进行数据统计和查询）
- 成都研究院（负责底层，包括源代码修改和按上层部门要求开发Map-Reduce程序，比如一些UDF）

Hadoop在淘宝和支付宝的应用

- 从09年开始。用于对海量数据的离线处理，例如对日志的分析，也涉及内容部分，结构化数据
- 主要基于可扩展性的考虑
- 规模从当初的3-4百节点增长到今天单一集群3000节点以上，2-3个集群
- 支付宝的集群规模也达700台，使用Hbase，个人消费记录，key-value型

2013.01.23

对Hadoop源码的修改

- 改进Namenode单点问题
- 增加安全性
- 改善Hbase的稳定性
- 改进反哺Hadoop社区

管理模式

- 集团统一管理
- Hadoop运维团队
- Hadoop开发团队
- 数据仓库团队（Hive）

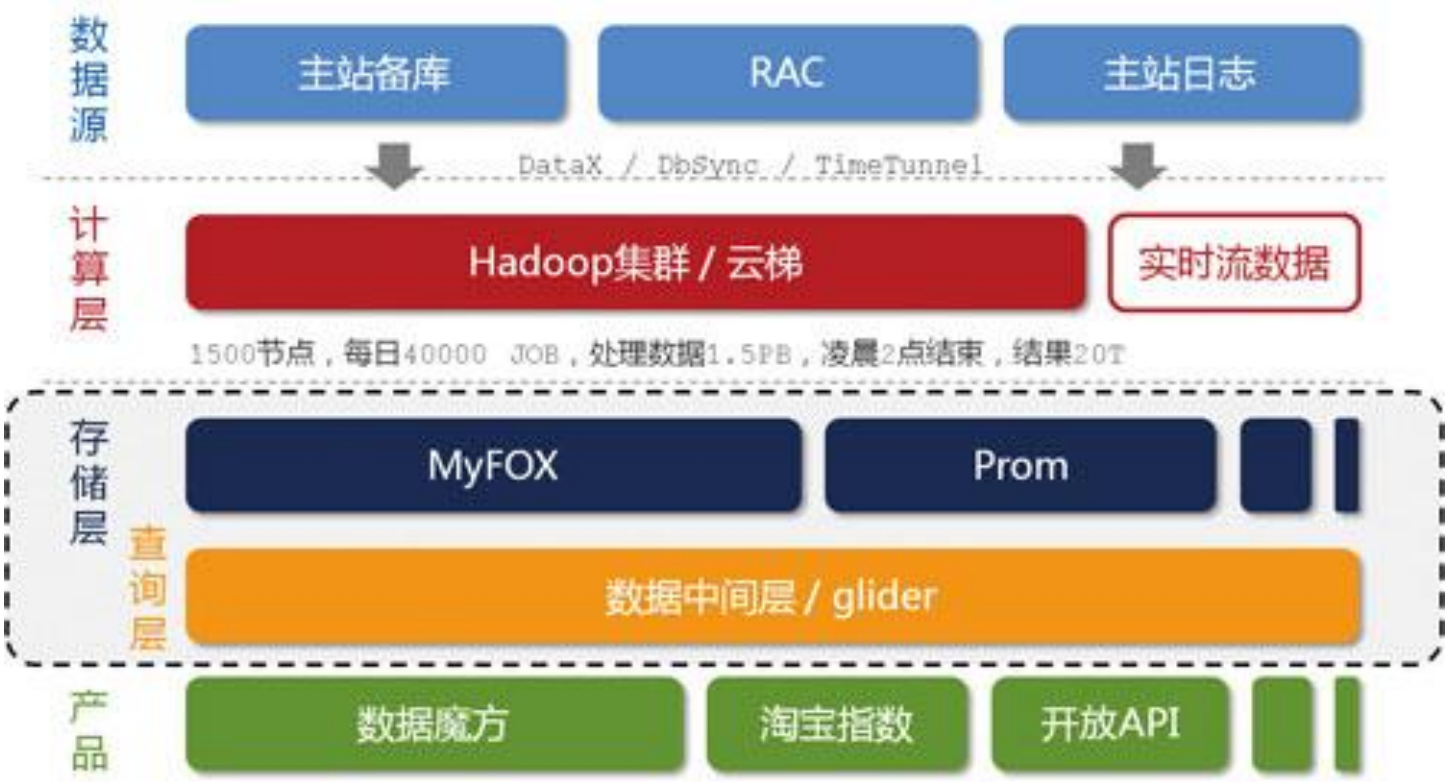
准实时的流数据处理技术

- 从Oracle, Mysql日志直接读取数据
- 部分数据源来自应用消息系统
- 以上数据经由Meta+Storm的流数据处理，写入HDFS，实现实时或准实时的数据分析
- 数据装载到Hive进行处理，结果写回Oracle和Mysql数据库

淘宝数据魔方

2013.01.23

架构图



架构图

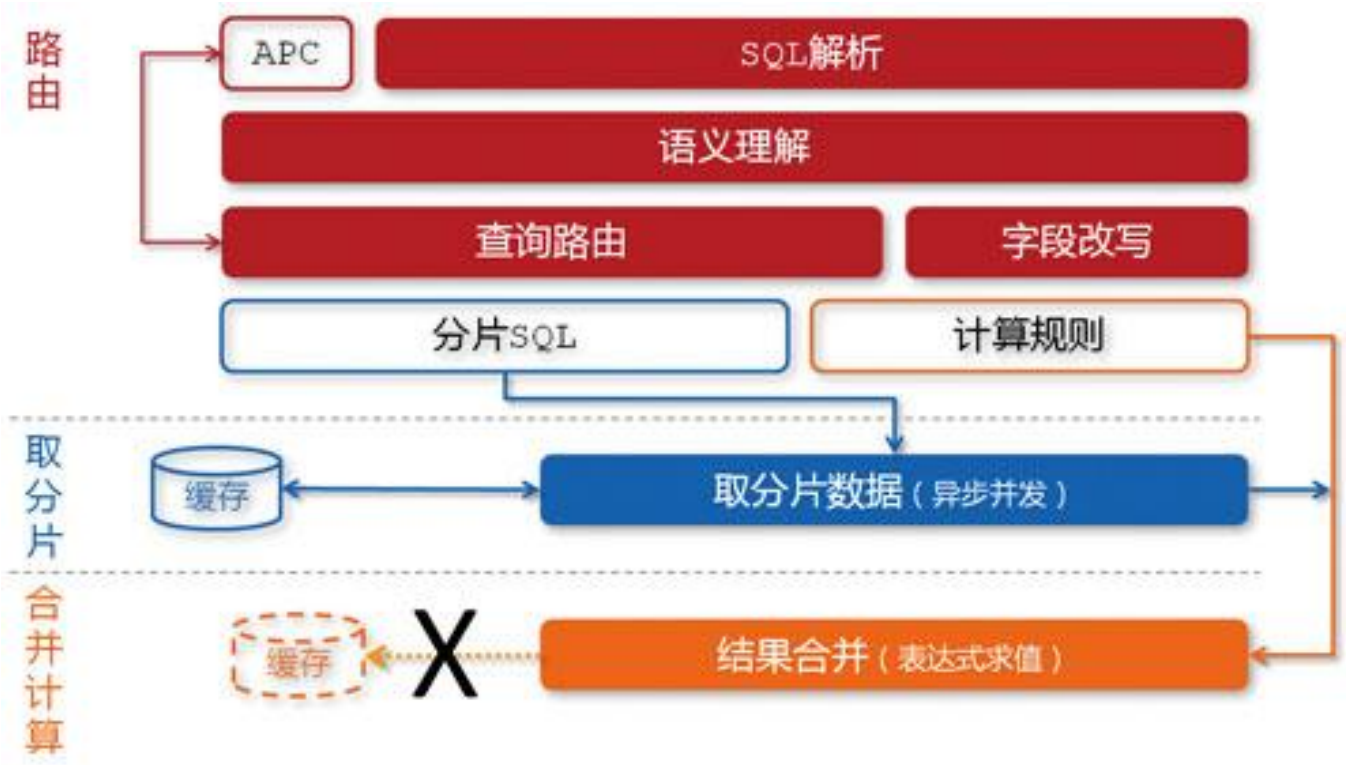
- 架构分为五层，分别是数据源、计算层、存储层、查询层和产品层。
- 数据来源层，这里有淘宝主站的用户、店铺、商品和交易等数据库，还有用户的浏览、搜索等行为日志等。这一系列的数据是数据产品最原始的生命力所在。
- 在数据源层实时产生的数据，通过淘宝主研发的数据传输组件DataX、DbSync和Timetunnel准实时地传输到Hadoop集群“云梯”，是计算层的主要组成部分。在“云梯”上，每天有大约40000个作业对1.5PB的原始数据按照产品需求进行不同的MapReduce计算。
- 一些对实效性要求很高的数据采用“云梯”来计算效率比较低，为此做了流式数据的实时计算平台，称之为“银河”。“银河”也是一个分布式系统，它接收来自TimeTunnel的实时消息，在内存中做实时计算，并把计算结果在尽可能短的时间内刷新到NoSQL存储设备中，供前端产品调用。

架构图

- “云梯”或者“银河”并不适合直接向产品提供实时的数据查询服务。这是因为，对于“云梯”来说，它的定位只是做离线计算的，无法支持较高的性能和并发需求；而对于“银河”而言，尽管所有的代码都掌握在我们手中，但要完整地将数据接收、实时计算、存储和查询等功能集成在一个分布式系统中，避免不了分层，最终仍然落到了目前的架构上。
- 针对前端产品设计了专门的存储层。在这一层，有基于MySQL的分布式关系型数据库集群MyFOX和基于HBase的NoSQL存储集群Prom。

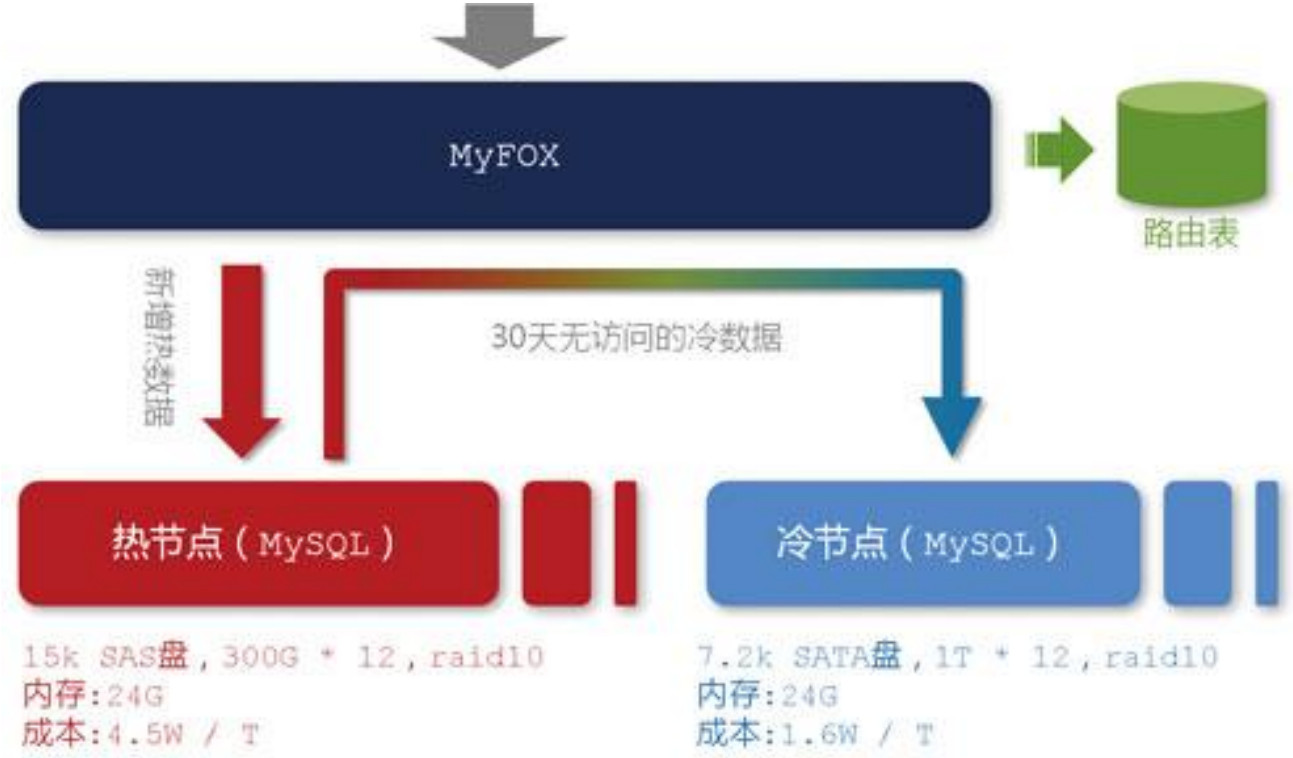
Myfox

- 数据查询过程



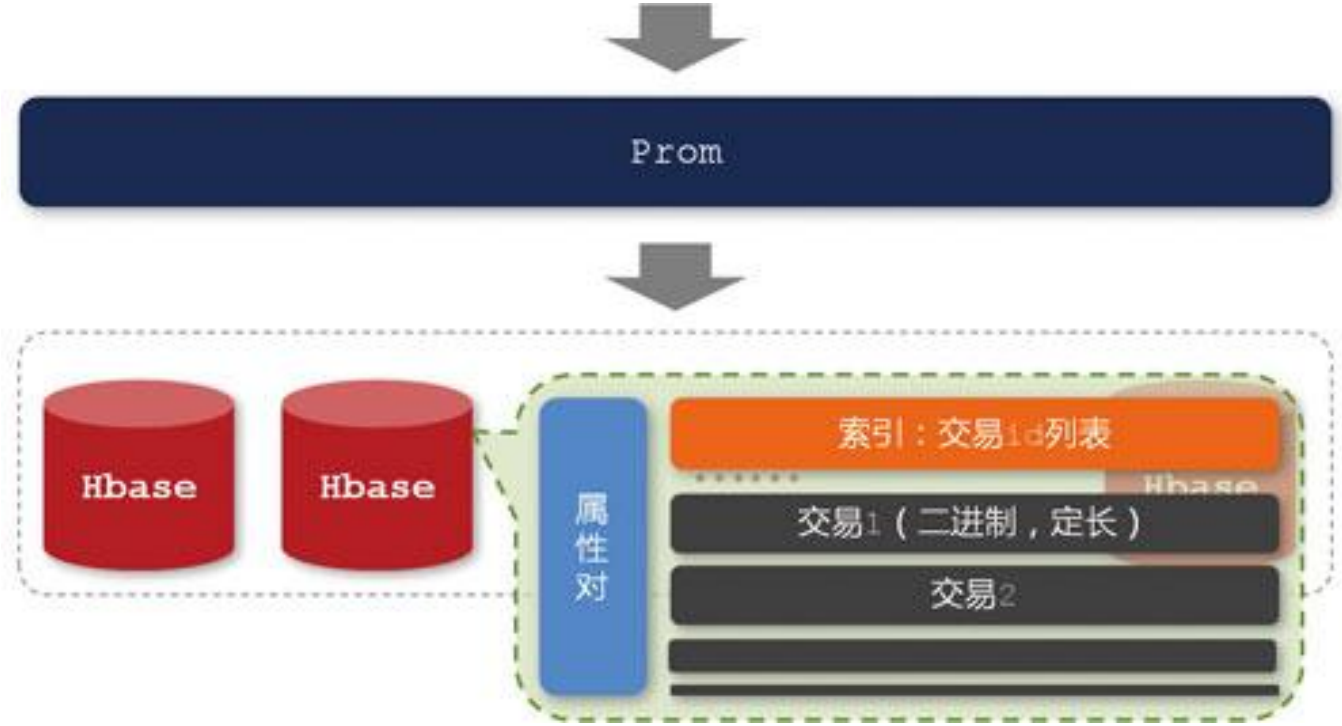
Myfox

- 节点结构



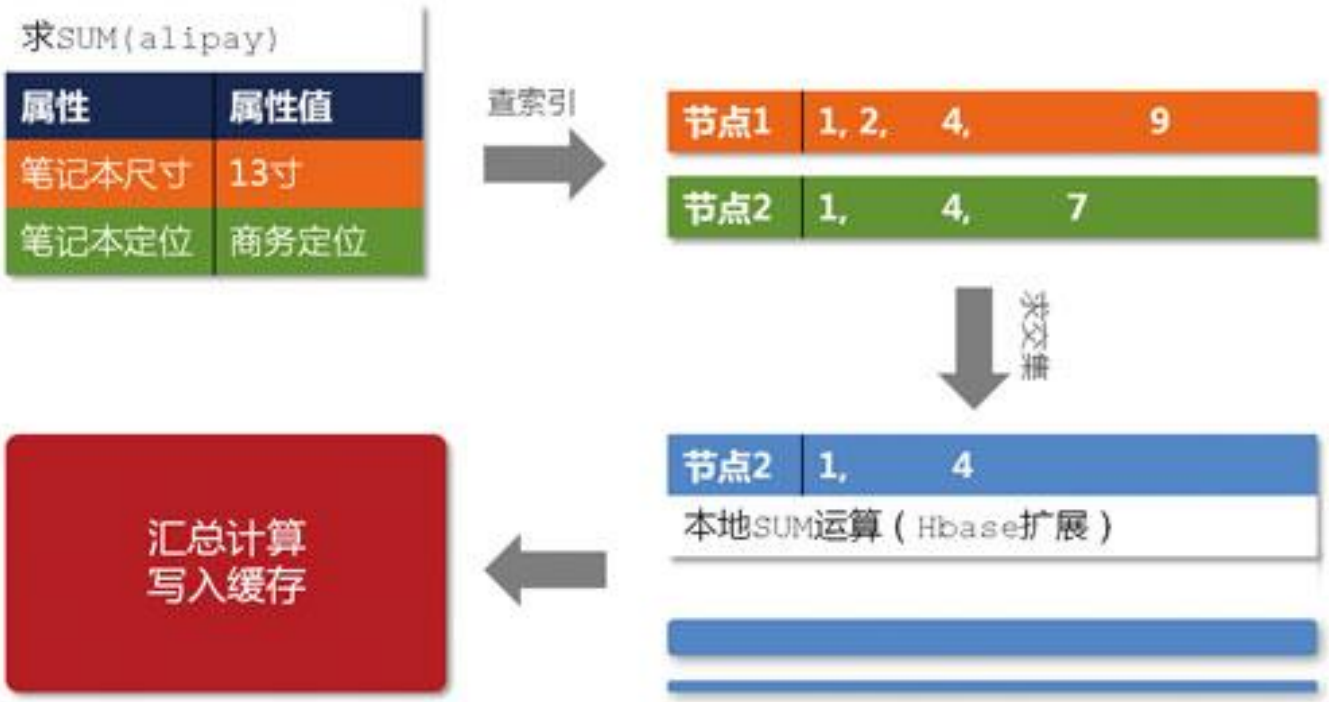
Prometheus

- Prom的存储结构



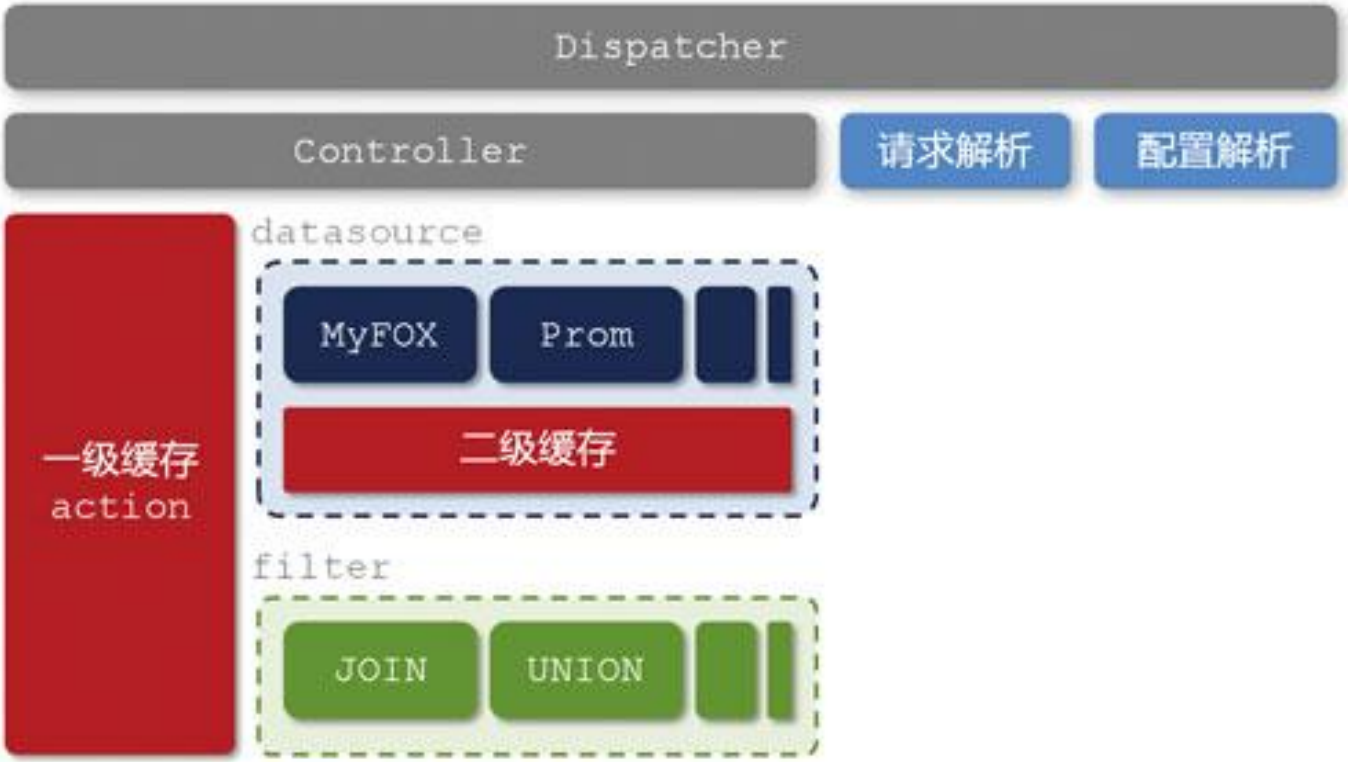
Prometheus

- Prom查询过程



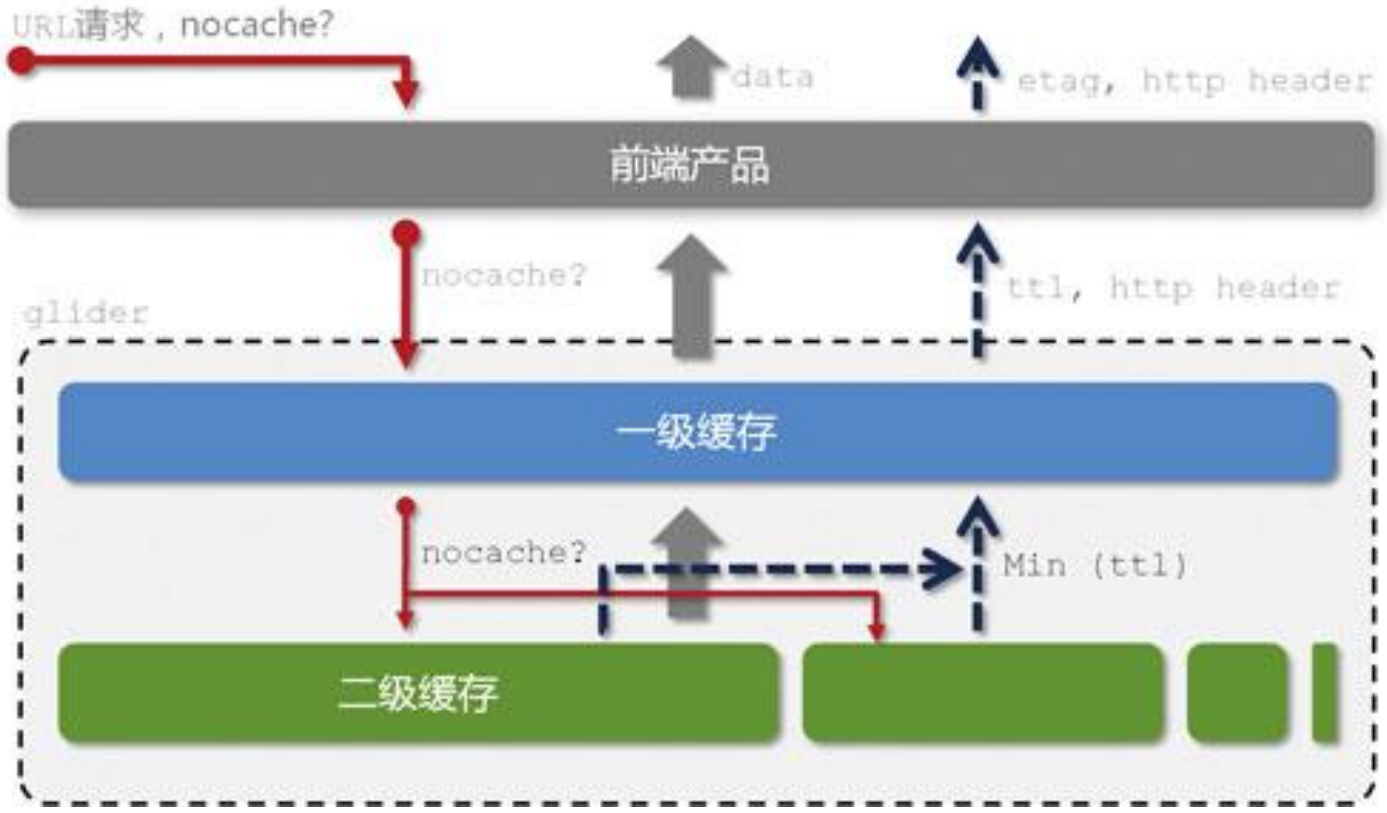
glider

- glider的技术架构



glider

- 缓存控制体系



量子恒道

2013.01.23

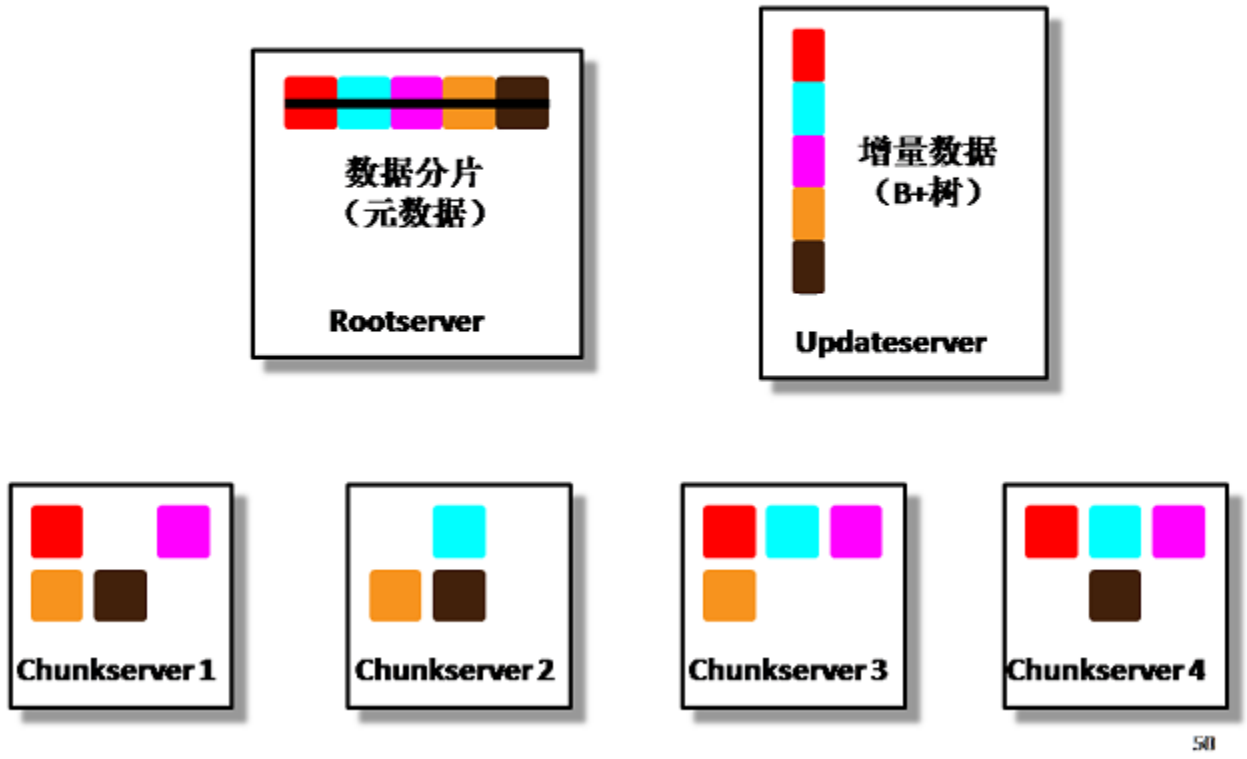
Oceanbase

2013.01.23

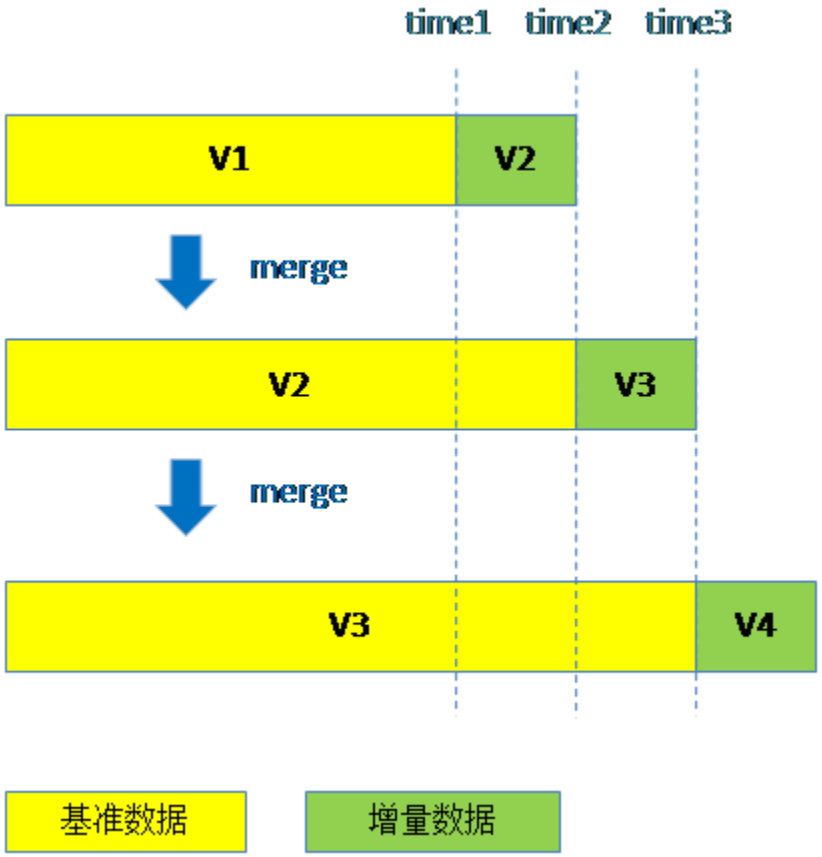
Oceanbase

- 分布式的结构化存储系统，采用强schema的形式，其数据是分布在多个数据节点上，并将读写数据做了完全的隔离。
- OB的数据节点分两种，一类是基准数据节点(!ChunkServer)，存储引擎是基于SSTABLE http://en.wikipedia.org/wiki/SSTable的。 一个是增量数据节点(!UpdateServer)，存储引擎是基于Btree(内存中的memtable)和SSTABLE(major-freeze-dump)的。
- **基准数据**：从开始至某个时间点的全量数据，是静态数据，在到下一个时间点合并之前，该部分数据不会发生变更。
- **增量数据**：是指从某个时间点至当前范围内新增的数据，增量数据会因为应用的各种修改操作(insert,update,delete)发生变更。

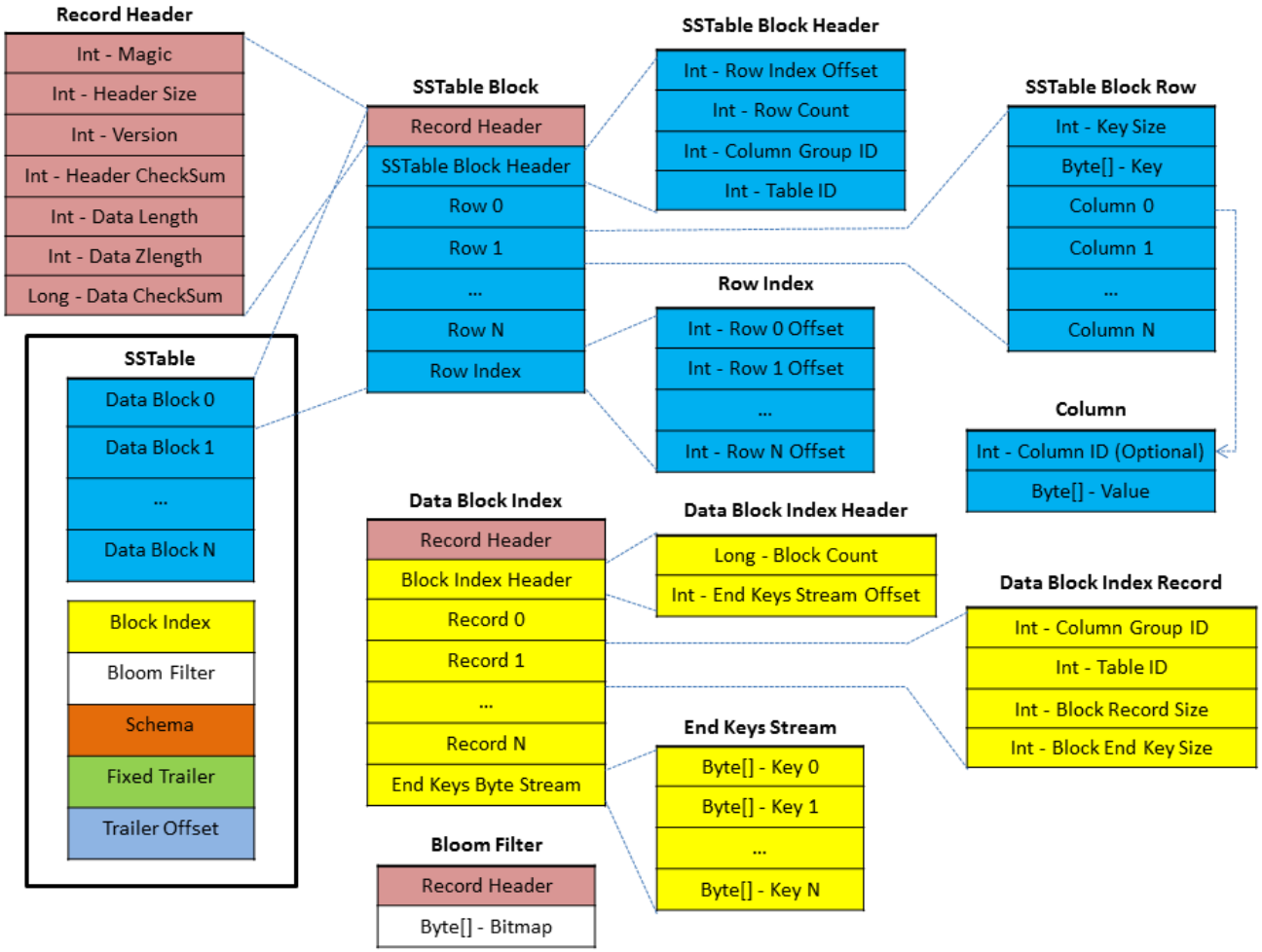
整体数据分布



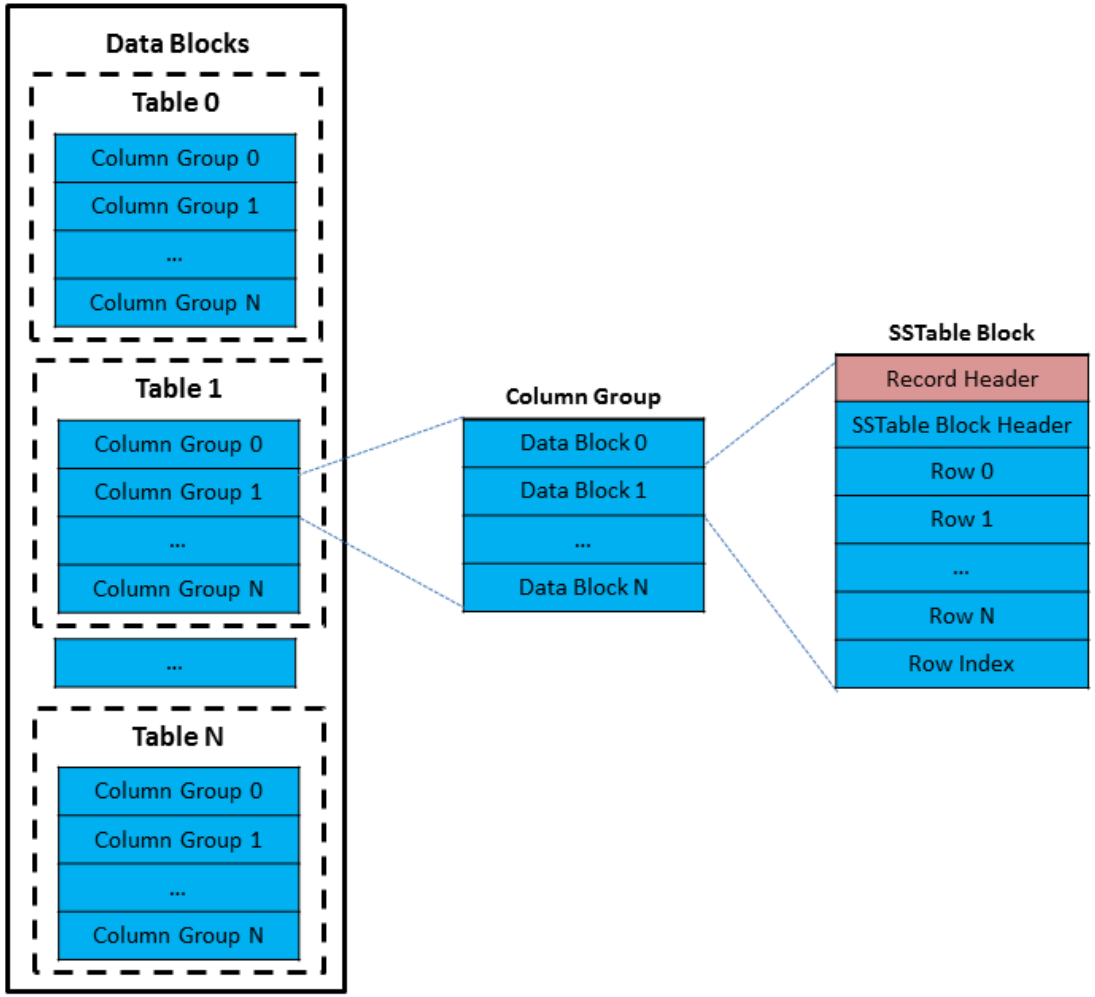
数据演进过程



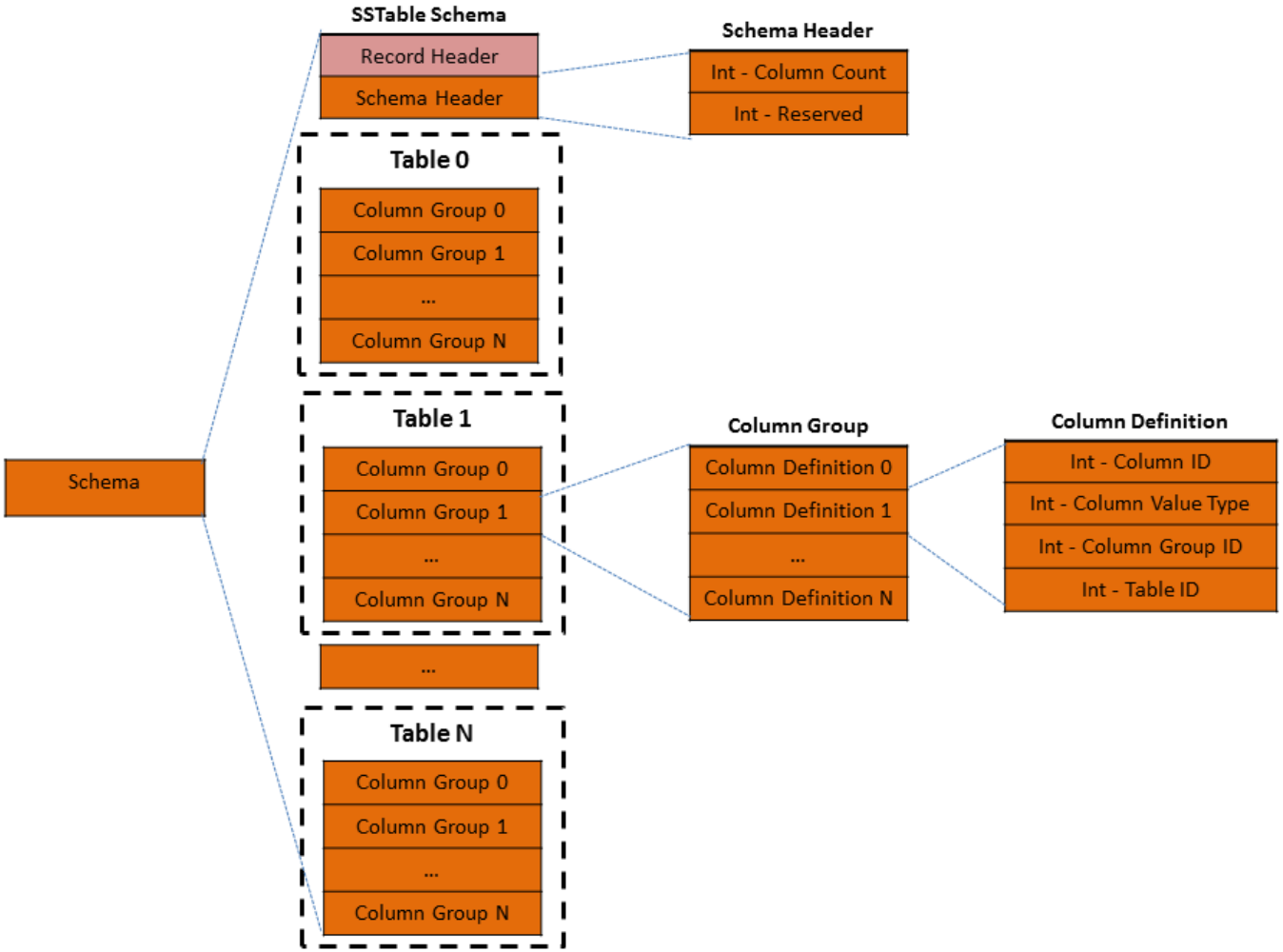
sstable的总体数据格式



sstable中data block的排列规则



sstable中的schema排列规则



2013.01.23

Hadoop@Baidu

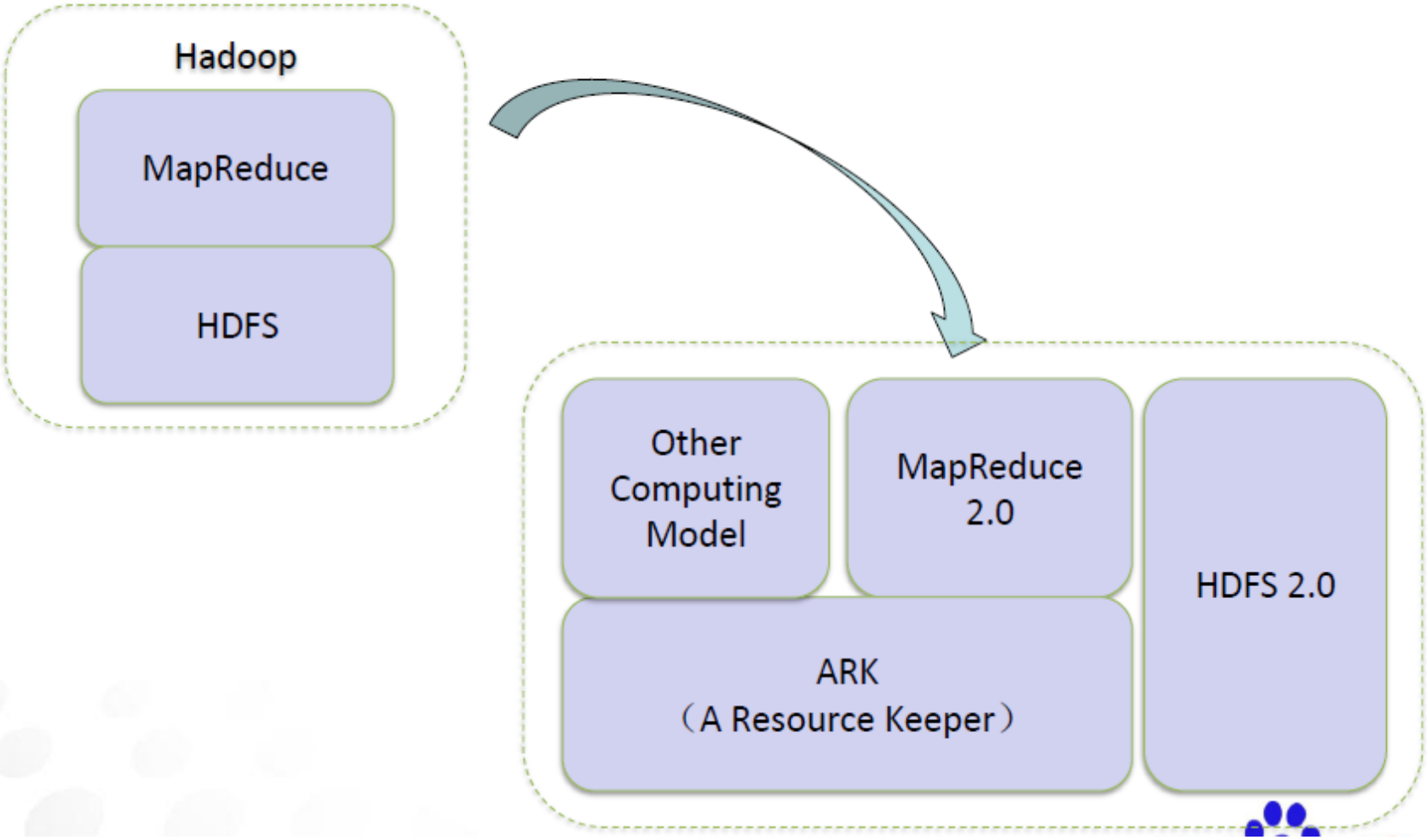
- 日志的存储和统计;
- 网页数据的分析和挖掘;
- 商业分析，如用户的行为和广告关注度等;
- 在线数据的反馈，及时得到在线广告的点击情况;
- 用户网页的聚类，分析用户的推荐度及用户之间的关联度。

- 2008
  - 开始于Hadoop v0.18/0.19
  - 300台机器，2个集群
- Now
  - 总规模2W以上
  - 最大集群接近4,000节点
  - 每日处理数据20PB+
  - 每日作业数120,000+

挑战

- 规模
  - 单集群1000→2000→3000→5000→10000
- 效率
  - 资源利用率（cpu/mem/io）—高峰vs平均
  - 存储利用—无压缩、冷数据
  - 存储与计算资源使用均衡问题
- 服务可用
  - 随着规模增大问题变得突出
    - 3K+节点升级或异常小时级中断
    - 用户影响面：在可用99.9%下用户容忍度变低

分布式计算技术2.0



HDFS2.0

- 1.0所面临的问题

集群规模大，Namenode响应变慢

Namenode单点，切换时间太长

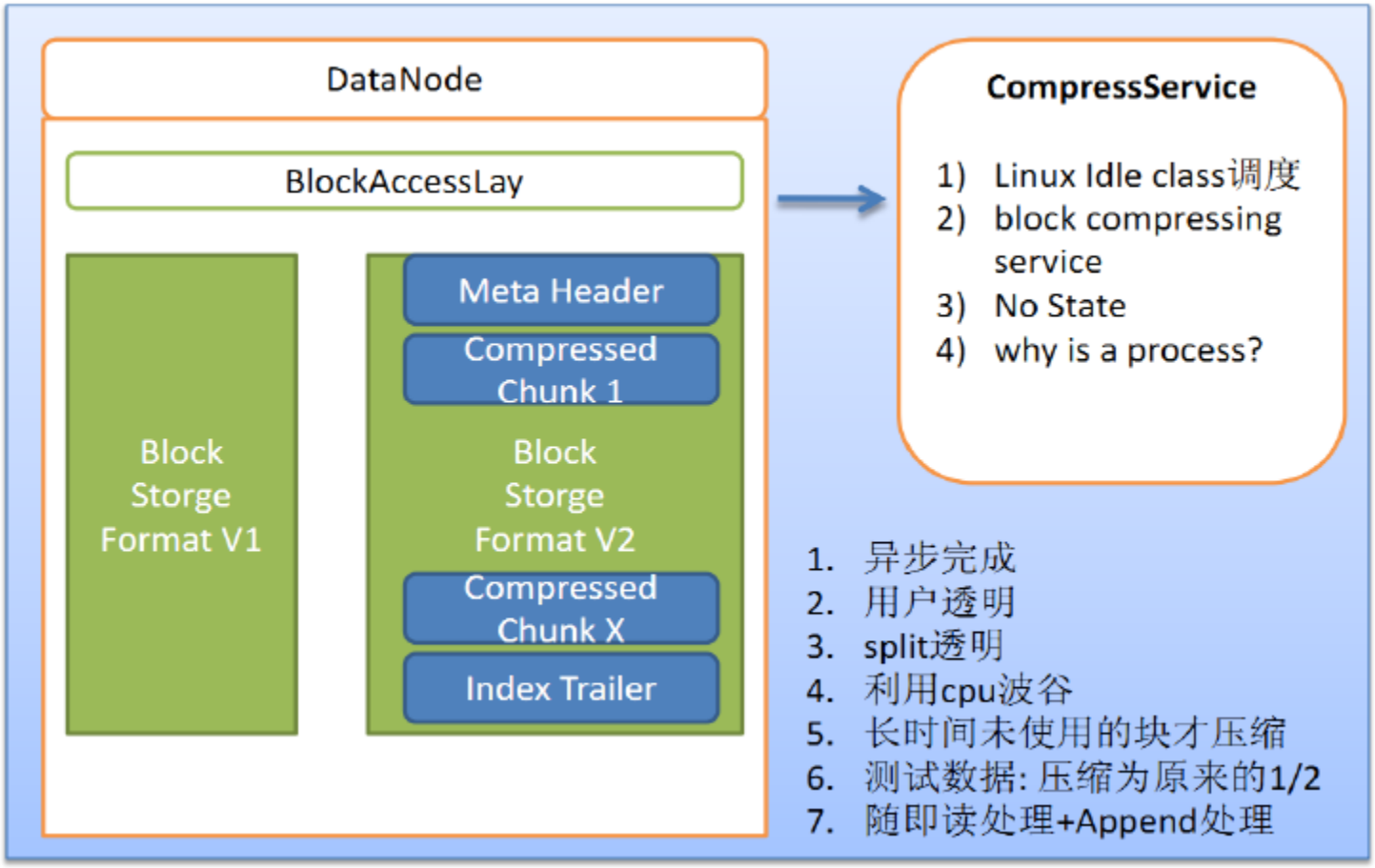
没有数据压缩

Namespace过于耗用资源

HDFS2.0可用性

- 热备支持
- 分钟级别切换
- 最坏情况，应用可能丢失1分钟级数据

HDFS2.0透明压缩



Map-Reduce2.0

- 1.0面临问题
  - JobTracker单点
    - 负载太重，扩展性受限→1W
    - 故障/升级中断服务重跑作业
  - 资源粒度过粗
    - slot（cpu、mem）
  - 资源利用不高
    - Shuffle+Reduce，空占slot

Map-Reduce2.0

- 可扩展性W台以上
- 架构松耦合，支持多种计算模型
- 可支持热升级
- 更精细的资源调度
- MR优化：Shuffle独立/Task同质调度