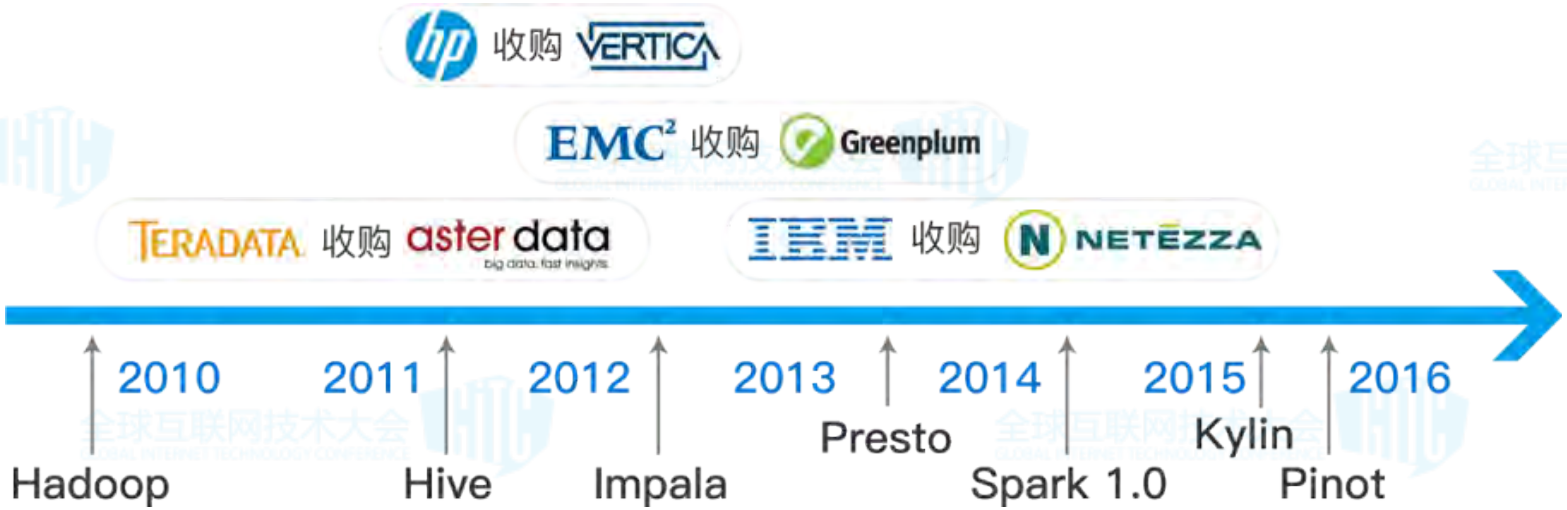
Bonree

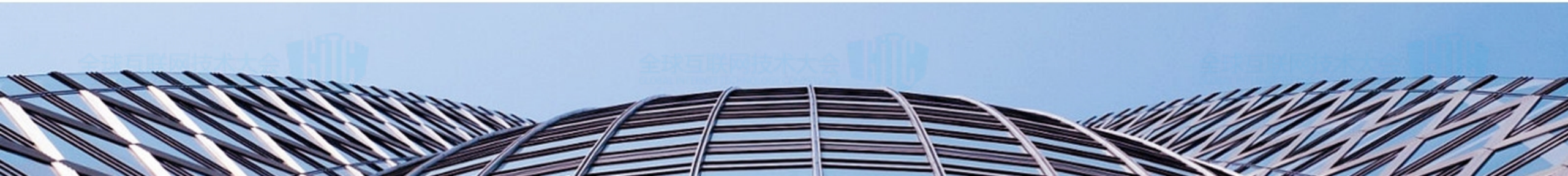
海量数据在线分析技术剖析

北京博睿宏远数据科技股份有限公司　　程捷

前言

Hadoop将我们带入了大数据时代，使得处理TB级别乃至PB级别的数据成为一种可能。但众所周知，依赖于MapReduce计算框架，导致实时性方面一直是Hadoop的一个硬伤。因此，如何实现对海量数据的秒级在线分析成为了不少大数据分析软件的核心目标。





好的数据存储和分析方案应该满足的标准

- 保证数据的原始列信息完整，即数据无损失
- 数据实时性，数据从产生到可查询不应有太大延迟
- 支持SQL-like查询方式，可灵活且快速响应业务需求
- 支持超大数据集的在线聚合秒级响应
- 数据存储设计灵活通用，可便利进行业务扩展和兼容其他计算引擎
- 数据支持编码和压缩存储，不存在明显的数据膨胀现象

业内常见大数据存储和分析方案对比

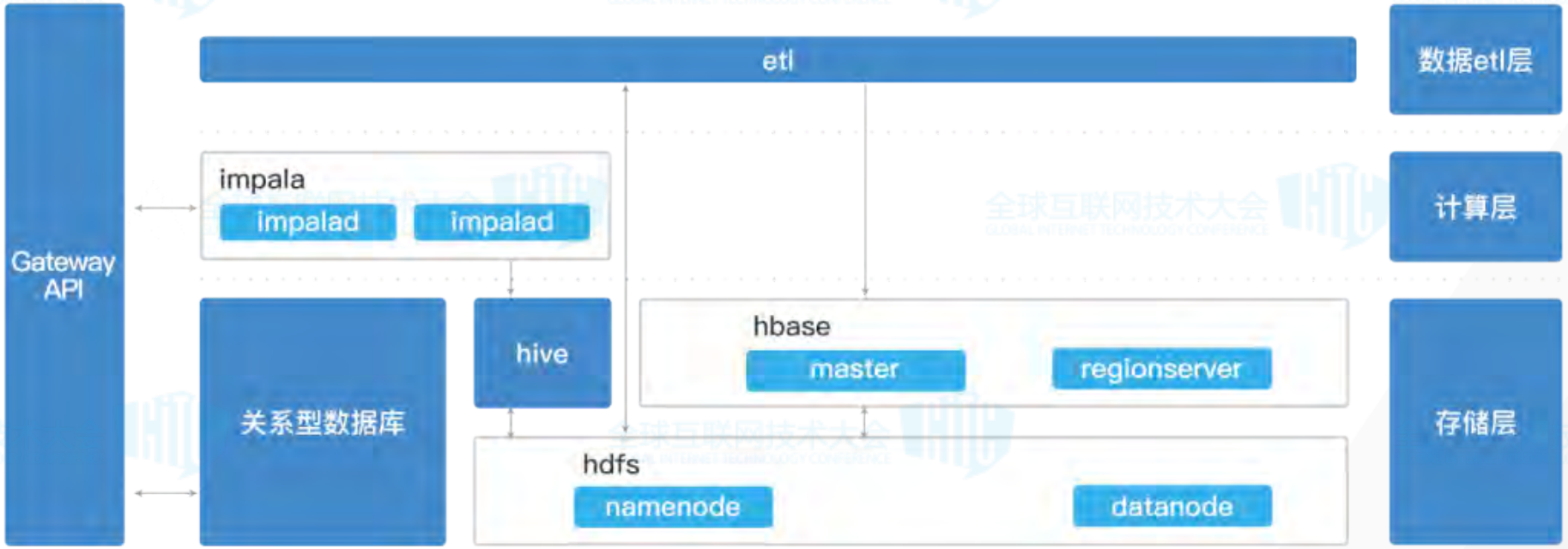
	原始数据无损失	数据实时性	支持SQL-like查询	在线聚合秒级响应	数据存储灵活性	复杂维度数据膨胀
HBase系 （OpenTSDB）	满足	准实时	不支持	优 （**非时序数据聚合**， 将可能导致全表scan而性能较差）	中 （rowkey预先设计， 聚合维度难以变更）	中 （依赖于hbase存储，不区分列 value类型编码，整体压缩）
Dremel系 （Hive、Impala、Drill）	满足	**差** （偏离线，分钟级）	**优** （支持绝大部分标准 SQL语义）	优	**优** （可直接支持mapreduce框架）	**优** （parquet格式按列压缩和编码存储）
预聚合系 （Druid、Kylin、Pinot）	**不满足** （预先将原始数据进行聚合， 会丢失列值信息）	准实时	中 （不支持join）	优	中 （预先聚合，调整聚合规则， 需重新聚合，且不支持嵌套）	**中** （预先聚合后，缓存大量中间聚合结 果数据，导致存在数据膨胀，kylin尤 为明显）
Lucene系 （ElasticSearch、Solr）	满足	准实时	中 （支持Restful API）	优 （需开启正向索引）	中 （不能直接支持mapreduce计算 框架）	**中** （索引多，存在数据膨胀）

博睿Net产品在线数据分析技术路线演进

2007~2014 依赖关系型数据库（Oracle）的存储和计算

2014~2016年初 切换至基于storm + redis的自研OLAP计算架构（完全基于内存存储和计算，成本太高，被迫放弃）

2016~至今 切换至基于impala + parquet的自研OLAP计算架构



博睿Net产品在线数据分析技术路线演进

—— Impala+parquet架构所遇到的坑

使用impala + parquet技术方案，我们遇到的一些坑

- 数据延时太大，实时性差（10分钟），但如果强行将数据时延减小会导致产生大量小文件（分区），每次查询会扫描很多小文件，导致集群I/O压力骤增，性能下降严重，这个问题是Dremel系的通病。
- 某些数据查询由于用户设置查询时间跨度不合理，导致大量没有目标数据集的分区也被频繁SCAN，导致系统整体查询性能严重拖累。
- 某一个用户的随机超大查询请求会把集群I/O资源集中耗尽，导致其他用户并发查询排队等待，导致平台整体查询响应缓慢。

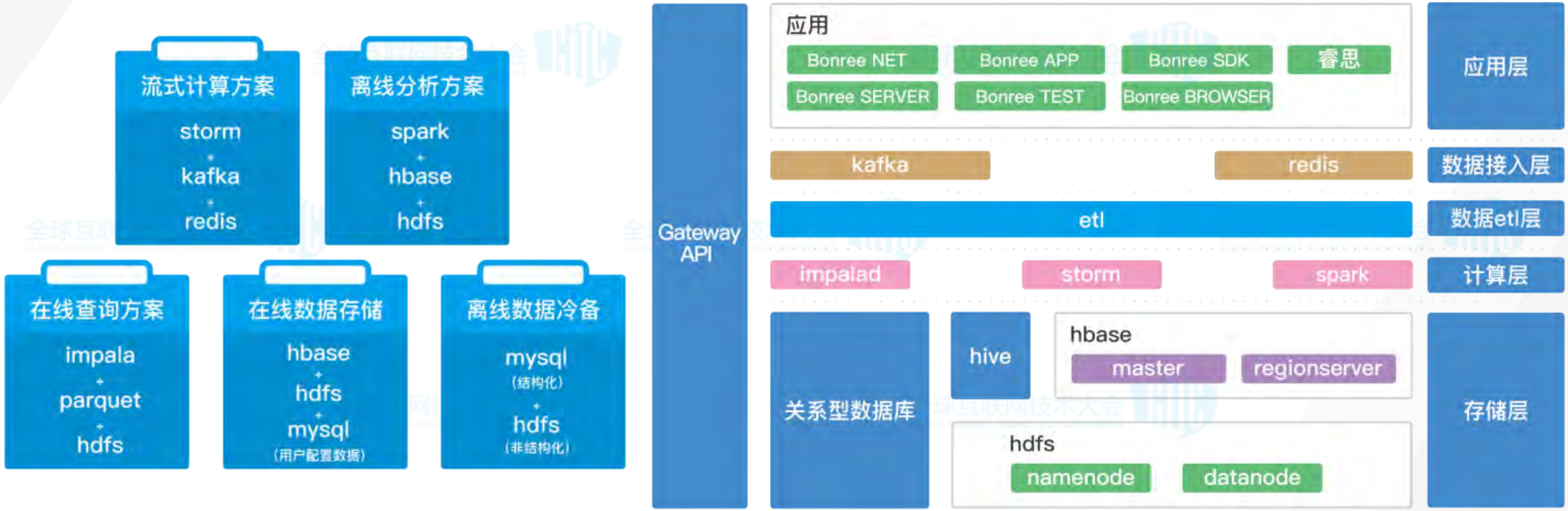
博睿Net产品在线数据分析技术路线演进

—— Impala + Parquet技术方案优化

- 为提高数据实时性和查询性能，我们进行了第一次集群拆分，降分区合并操作单独拆分一组集群，做“读写分离”，将数据延时由10分钟降低至1分钟左右，且由于小分区进行了提前合并，使得查询扫描分区文件数大为减少，降低了查询集群整体负载，极大提升系统查询并行度和性能。
- 为避免扫描无目标数据分区的问题，我们在hbase中设计并维护了查询条件与数据分区的对照索引，从源头上避免全表扫描，提高系统性能。
- 为避免随机超大查询对其它查询请求的干扰，我们再次进行了集群拆分，从物理上再将查询集群拆分为大查询集群和小查询，彻底分离，并由查询网关负责调度。

Bonree Net产品	Oracle架构	Impala架构
生产环境查询平均性能	4.6s	1.2s
极限查询平均性能（5TB/60GB/30+维度）	无结果	32s
使用机器数量	18台（一主一备）	12台（2*2副本）
落盘数据总量（12个月）	142TB	24TB

博睿数据目前的大数据技术栈



博睿数据在大数据方向继续探索

——大数据开放融合平台Bonree Platform

性能监控数据

- 终端用户性能数据（SDK、Browser）
- 内网应用性能数据（Server）

环境监控数据

- 外网链路性能数据（Net、APP）
- 主机环境性能数据（Server）
- 中间件运行状态数据（Server）

报警日志数据

- 外网链路报警数据（Net）
- 内网业务报警数据（Server）
- 外网业务报警数据（SDK、Browser、Net、APP）
- 主机环境报警数据（Server）
- 中间件报警数据（Server）

用户行为数据

- 终端用户行为数据（SDK、Browser）
- 行业细分标杆数据（SDK、Net）

第三方日志数据