

联合主办：msup® | 协办：  青云 QING CLOUD

卷皮大数据 平台演进

张知严 卷皮网大数据首席架构师

目录

CATALOG



初代架构



第二代架构



第三代架构

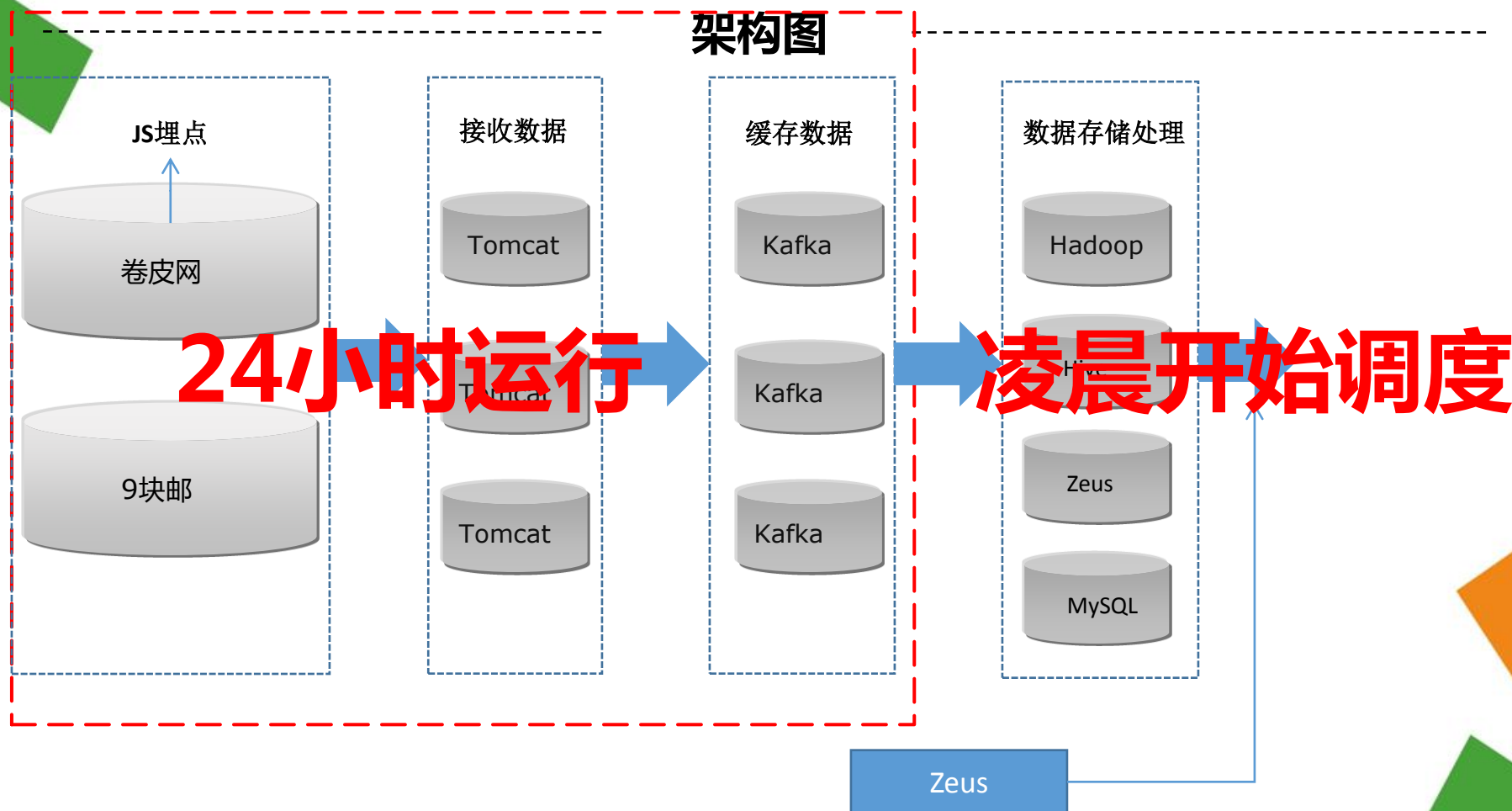


架构总结



第一部分 初代大数据架构

—小而紧



遇到的问题



环境搭建

如何从无到有搭建大数据分析平台



Kafka使用

如何对生产数据到Kafka以并进行消费



作业调度

如何实现蜘蛛网式的作业调度



数据质量

如何找出错误数据并持续监控

Cloudera

Cloudera和Hortonworks、MapR一起被称为大数据基础架构工具的三架马车，可以帮助企业快速安装，配置，运行hadoop和其周边生态系统，降低公司技术成本。

环境搭建

组件安装、配置管理

Cloudera的组件安装，横向扩展，配置变更等都通过web界面完成，无需进行命令行操作



日志查看、搜索

Cloudera 不仅对日志进行了自动切分，且在web界面可以对日志进行查看和搜索



运维监控

Cloudera 通过修改Hadoop等开源组件源码进行监控埋点，对于各项服务、角色都实现了详细的监控



版本升级

Cloudera 版本升级提供了多种方式，包括一键升级这种傻瓜升级方式，不过生产环境须慎用



Kafka使用

性能问题

初期对Kafka不熟悉，性能优化不够前端雪崩效应导致Kafka崩掉，后来通过合理的硬件配置，服务端配置以及客户端参数优化等方式进行了优化。

01
STEP

02
STEP



设计问题

初期Topic和Partition设置随意，导致热点现象比较严重，后续结合业务，对topic和partition进行合理设计，达到了多节点的负载均衡

03
STEP



消费问题

MR程序由于是离线消费，并且每次都是全量的，对网卡带宽压力很大。

作业调度



可视化管
理界面



JAVA
MySQL
技术栈



调度设计
简单



作业日志
查看方便

关于作业调度，我们调研了例如Oozie, Azkaban等多种调度框架，但是我们最终选择了Zeus，这我们的架构哲学有关，我们选择架构只要简单，够用就行了，太复杂的不适合我们，太笨重的也不行，从这些方面来说，Zeus比其他两个都要做得好。

数据质量

01

数据源质量把控

02

重要作业调度监控

03

核心数据质量检查

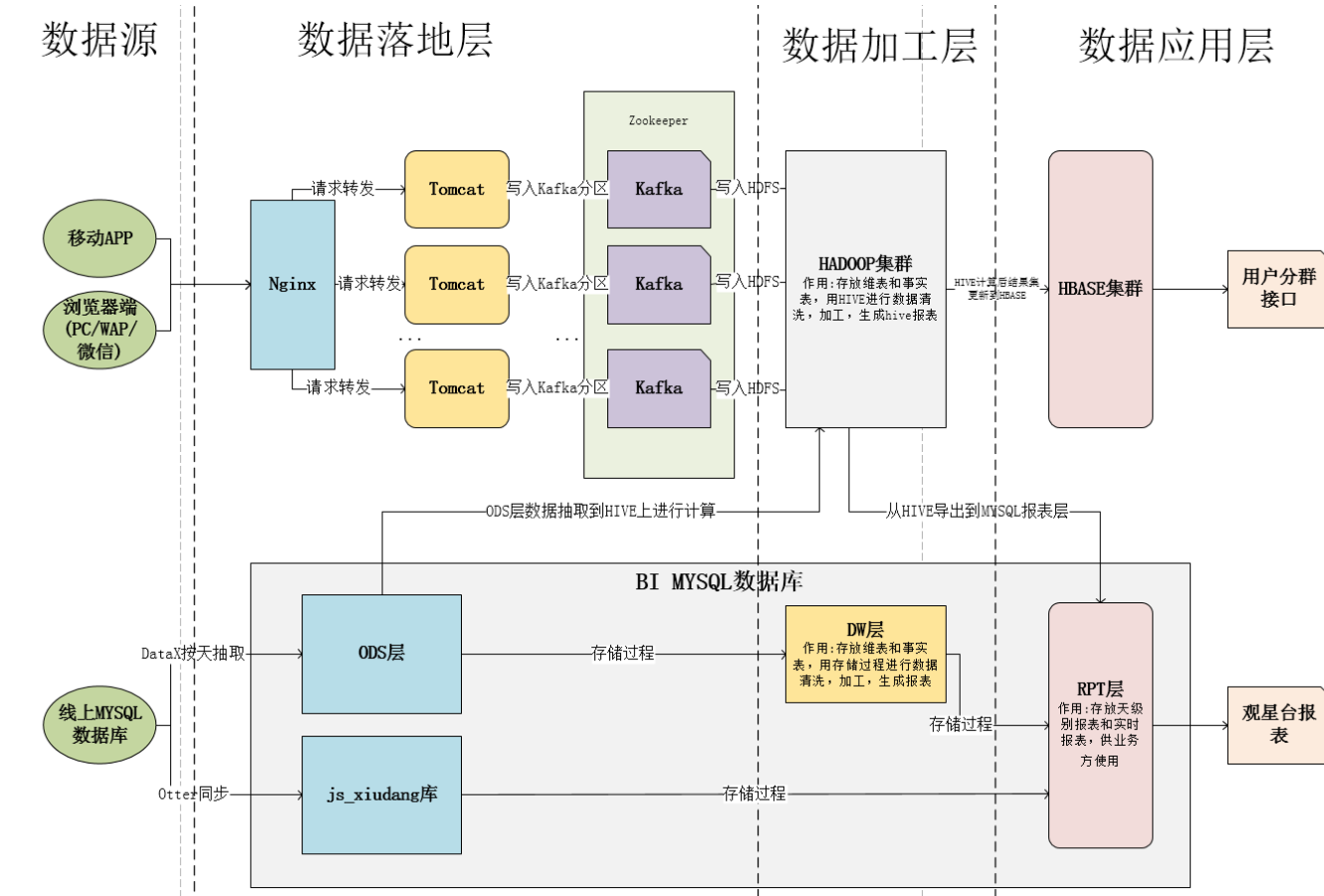


第二部分

第二代大数据架构

—分层&解耦

架构图



分层&解耦



数据分层

将数据划分为数据源、落地层、加工层、应用层等

01

具体来讲加工层又可以细分为ODS、DW、RPT层，基于这些数据分层，卷皮搭建起了企业级的数据仓库



架构分层

架构划分为Web服务、ETL工具、Hive、MySQL等

03

为了对数据进行接收、清洗、加工、展现，需要多个架构组合使用。



02



ETL工具

为了支撑日益复杂业务，BI对开源ETL进行了扩展，支持多种抽取场景



MySQL选型

为了适应前端的PHP展现以及对MySQL的抽取，我们选择了MySQL数据库做为DW，去达到更好的结合



04

ETL工具介绍

源	目的	开源软件&工具
Kafka	Hive	MapReduce
MySQL	Hive	Sqoop、DataX插件
Hive	MySQL	Hive JDBC
Hive	HBase	hive-hbase集成表
线上MySQL	BI MySQL	DataX
线上MySQL	BI MySQL	Otter(实时)

Sqoop扩展

- 1、全量同步，无分区；（适用于小表，参数表，全删全插）
- 2、增量或者全量抽取，并插入当天分区；（按分区全删全插，抽取效率较高，用户访问日志表）
- 3、增量同步合并全量，无分区（合并历史数据并插入,用户表）
- 4、增量同步合并全量，有分区（合并历史数据并插入多个分区,商品操作日志表）
- 5、增量同步合并全量，有分区，更新插入（合并历史数据并更新插入多个分区,订单表）



第三部分

第三代大数据架构

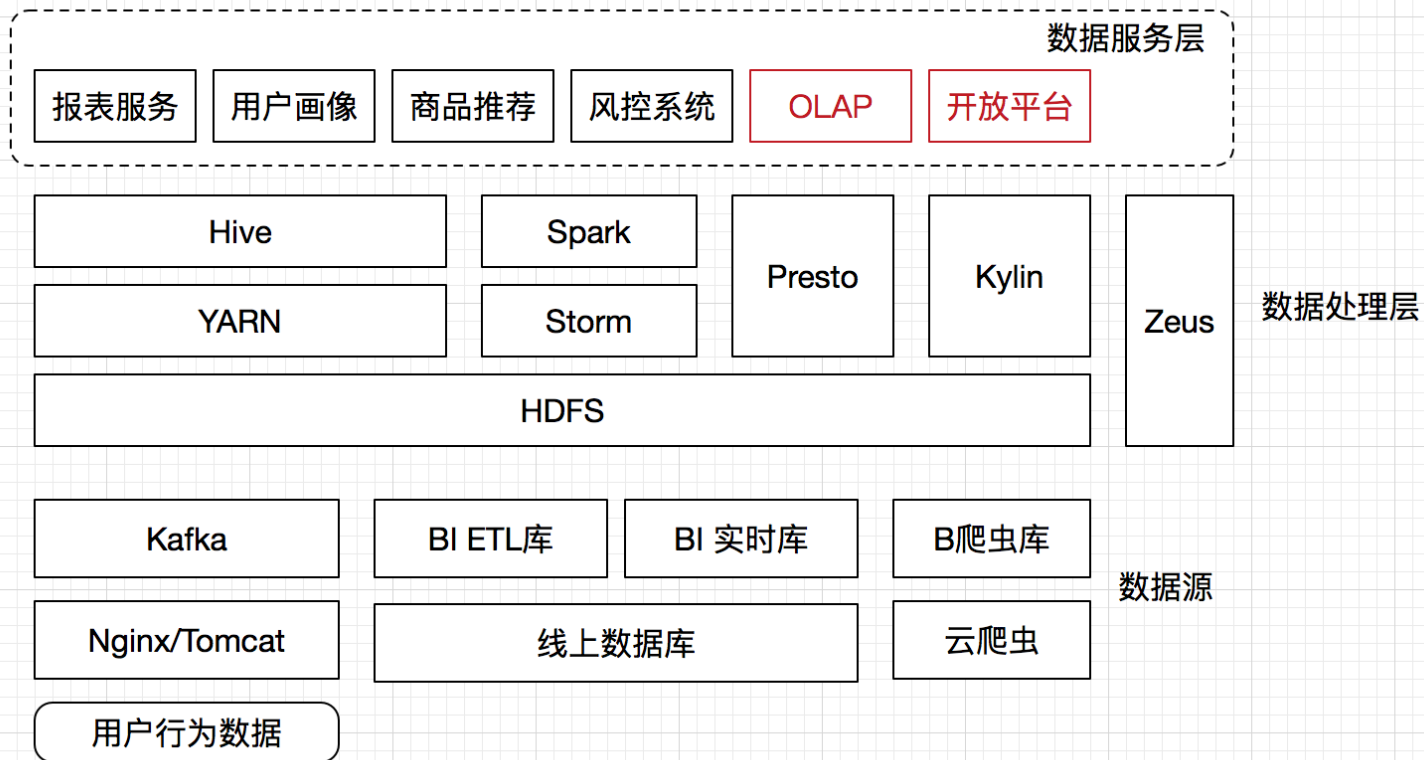
—面向服务

面向服务布局



架构图

面向服务的大数据架构



开放平台组成

用户系统：以web界面方式提供给用户使用，提供SQL查询，数据传输，作业上线等功能，并提供管理员权限管理功能。

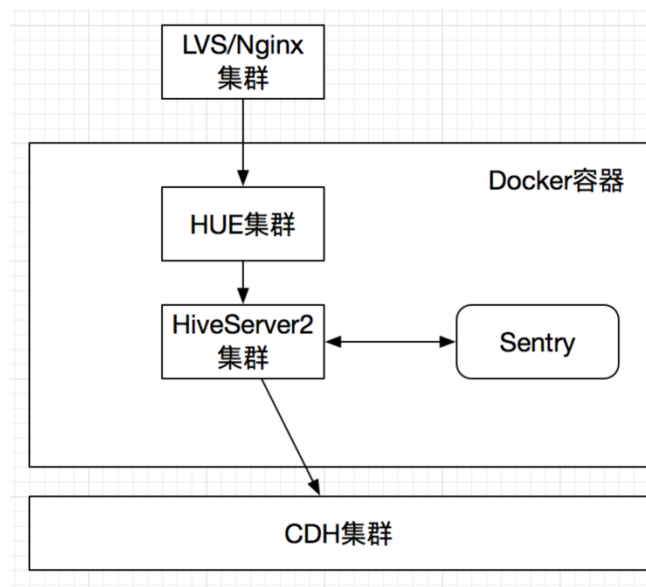
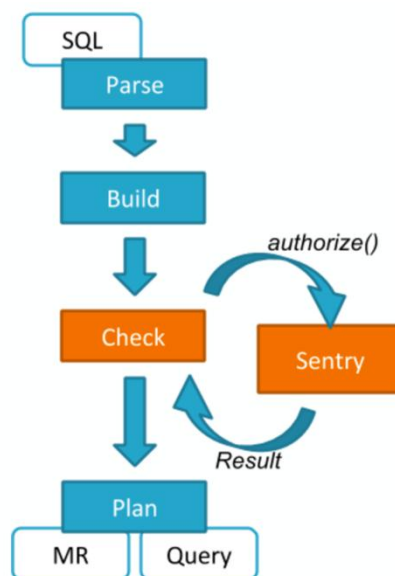
权限管理系统：基于RBAC模型设计，依据业务将分析师的数据访问权限进行精细控制，并实现权限操作审计功能。

资源调度器：实现合理的资源分配和调度，例如各部门资源单独分配，空闲资源互相共享，作业调度优先级设置，资源分配动态调整等。

业务调度系统：其他业务部门在界面上调试完SQL后，可以在界面上进行业务调度系统的作业上线，实现SQL的定期执行，用以固化结果数据。该系统与BI调度系统将隔离开独立部署，但是可以接收BI调度系统的作业状态广播。

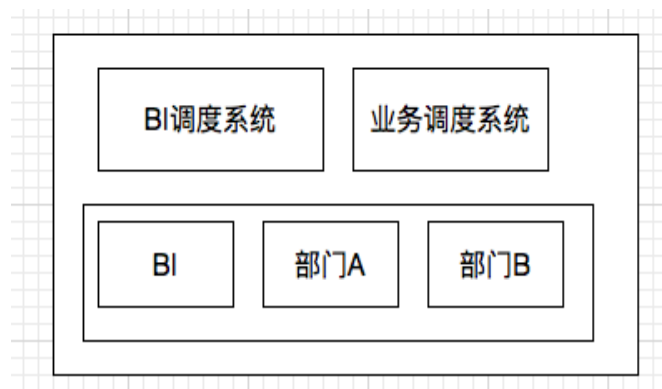
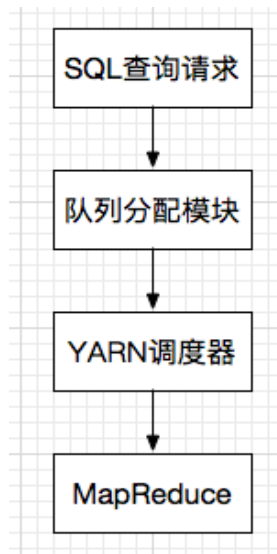
数据传输通道：设计为BI数据平台和外部的唯一数据通道，特点是高吞吐、稳定、提供审计功能。

权限模块



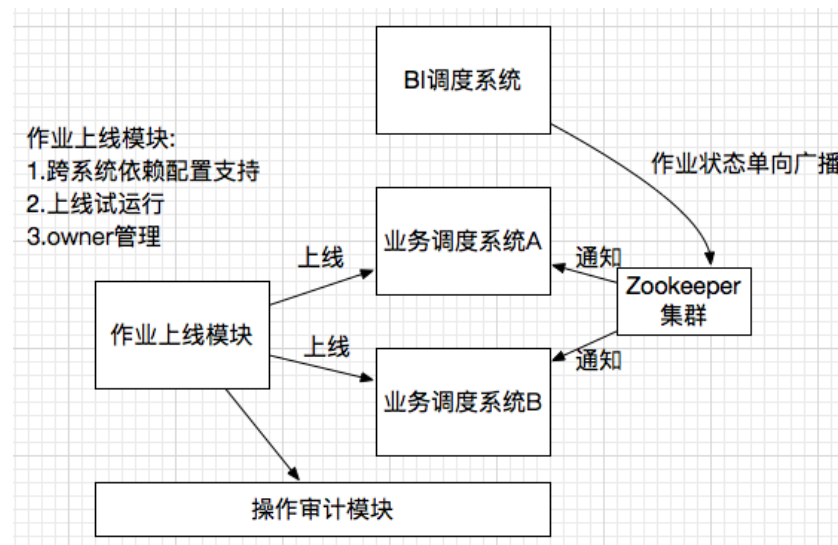
用户提交sql后进行权限检查

资源调度模块



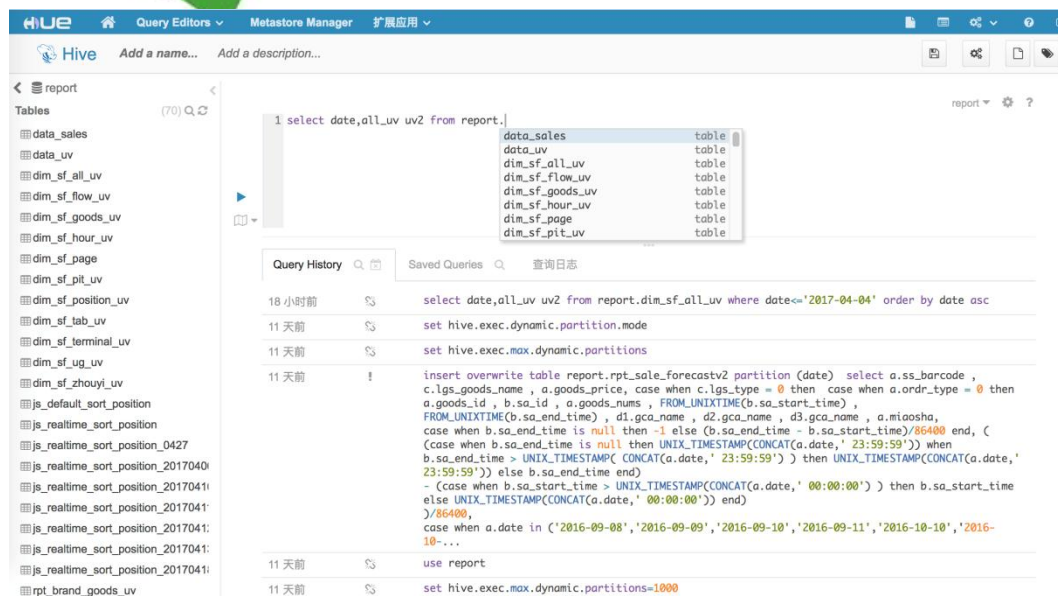
提交sql后通过队列进行资源协调、控制

调度模块



跨系统的依赖作业调度

用户界面



任务SQL

选择保存的查询

解析SQL

任务名称

任务依赖

任务说明

任务部门

作业类型

保存

取消

用户界面两大功能：查询、上线

技术栈



HUE

给用户提供了web界面进行自助查询，我们对其作了扩展以整合Zeus，LDAP，YARN等



LDAP

通过LDAP管理用户帐户信息



Sentry

用Sentry实现Hive最细到字段粒度级别的权限控制



Zeus

打通BI自身zeus和开放平台zeus系统，实现跨系统的依赖调度



YARN

对开放平台用户进行计算资源管理，充分利用大数据资源池



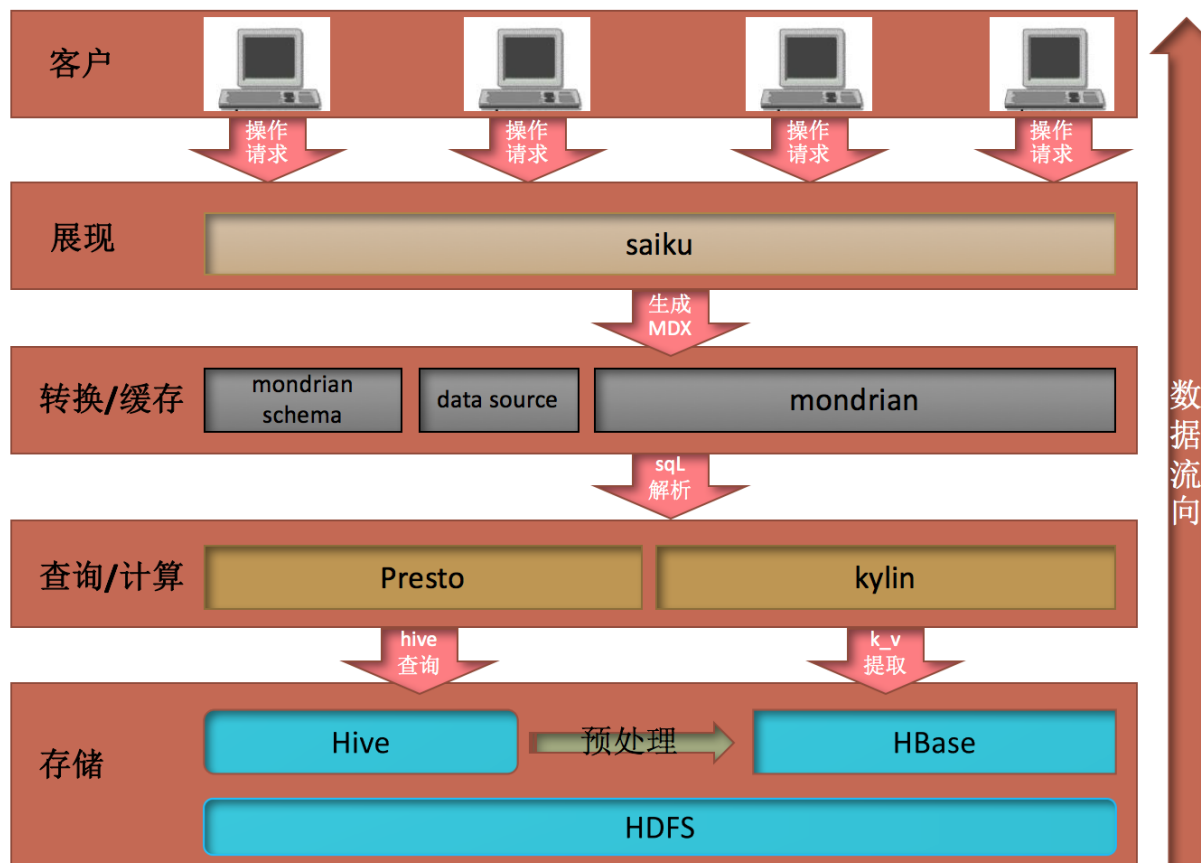
Hive

提供Sql查询功能，并将原有的UDF整合进开放平台

开放平台使用

- 1、上百注册用户
- 2、日均查询次数600+
- 3、资源使用情况30% - 40%
- 4、调度作业数70+
- 5、报表数量50+

OLAP架构图



演进历程

ROLAP

- saiku
- mondrain
- mysql

- 快速上线
- 数据量少

ROLAP分布式

- saiku
- mondrain
- presto
- hive

- 数据增长
- 响应慢

HLOAP/ROLAP分布式

- saiku
- mondrain
- presto/kylin
- hbase/hive

- 海量数据
- 响应慢
- 内存溢出



第四部分 架构总结

演进过程

初代大数据架构

—小而紧，初创期

第二代大数据架构

—分层&解耦，发展期

第三代大数据架构

—面向服务，成熟期

设计哲学

简单

开源

够用

分层

业务驱动

求变

面向服务

张胖 卷皮BI架构师

THANKS