大数据平台建设方案

1. 项目概述

1.1 平台意义

从2016年起，银科集团数据分析工作主要基于SQL Server集群（内部称为分析集群）。在大数据、人工智能快速发展的今天，分析集群逐渐暴露一些问题：

1. 集群的存储是10T级的，而新采集的数据量会超过这个数据量10倍以上；
2. 数据处理基本都是单节点，运算能力无法扩展，当复杂查询来临时只能等若干小时；
3. 传统数据架构不支持非结构化数据的存储；
4. 传统数据架构只支持标准的SQL操作，无法支持复杂的批处理和流处理；

基于以上问题我们决定构建一套新的大数据平台。

1.2 平台建设目标

构建的大数据平台要求：

1. 可以保存百T级的数据；
2. 集分布式数据存储和分布式计算于一体；
3. 允许横向扩展存储和运算能力；
4. 支持复杂的编码或机器学习计算；
5. 支持复杂的数据任务调度；
6. 安全、可靠

除了在大数据平台本身的硬件要求上，我们希望支持更适合大数据的工作流程：

1. 基于推送而不是拉取的数据采集
2. 基于任务调度而不是一系列脚本的ETL
3. 易于修改和查询的数据元信息

1.3 平台性能需求

存储

- 可以保存100T的数据
- 可以保存大小为100G的单个文件

运算性能

- 平台最多可以使用不少于350个核进行运算
- 平台最多可以使用不少于2,000G内存进行运算
- 集群的运算性能应和数据量级呈线性降低

扩展性

- 集群的存储可以较容易的扩展
- 集群的运算资源（CPU核和内存）可以较容易的扩展

稳定性

- 2个节点永久失效不影响数据访问
- 1个机架上的节点永久失效不会导致数据丢失

2. 整体架构设计

2.1 整体设计原则

我们选择以Hortonworks公司的HDP开源平台为核心，搭配其他开源组件的整理设计方案。

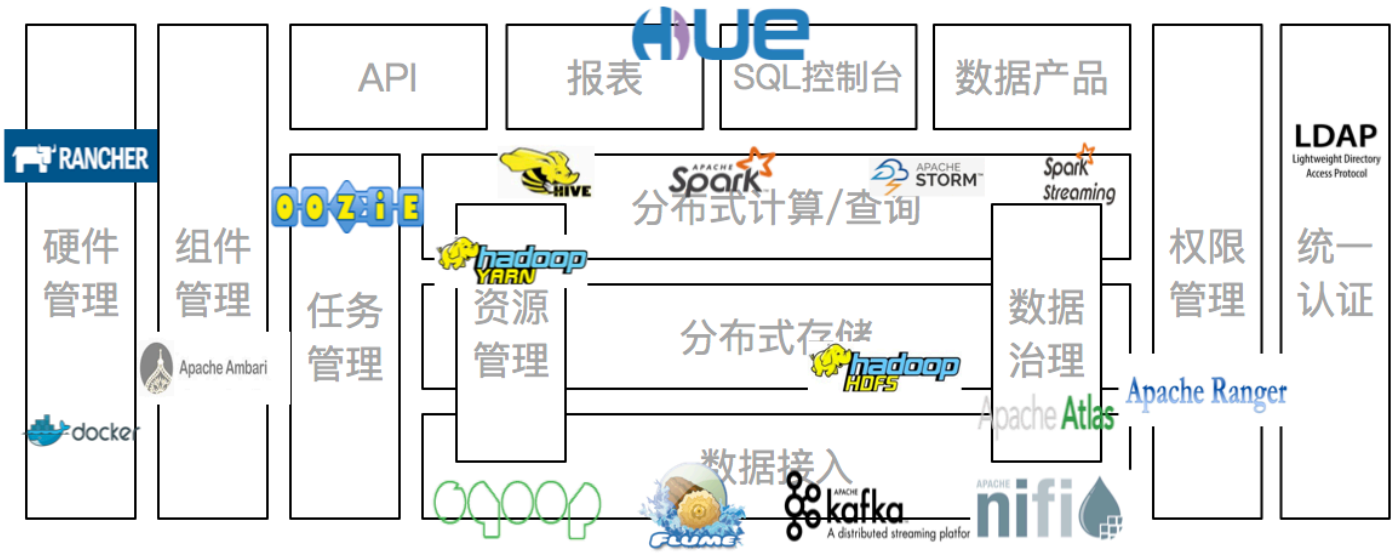
我们选择开源体系而非商业体系的主要原因是：

- 我们希望团队对底层平台有不错的理解，以便于使用时对平台性能有足够的掌控力
- 大数据技术依然处于剧烈变化的年代，我们希望未来有更换一些组件的自由度

架构上我们尽量遵循最简原则，只把最核心最常用的组件包括进“大数据平台”。维持小而稳定的核心组件意味着我们和用户在非核心组件上的选择更自由。

HDP，一个Hadoop为基础的体系，其本身就解决了扩展性和稳定性的许多问题：当我们读写数据时，HDFS本身就有很好的容错机制；当我们调度运算资源时，Yarn本身就解决了任务分布和资源扩展的问题。

2.2 逻辑架构

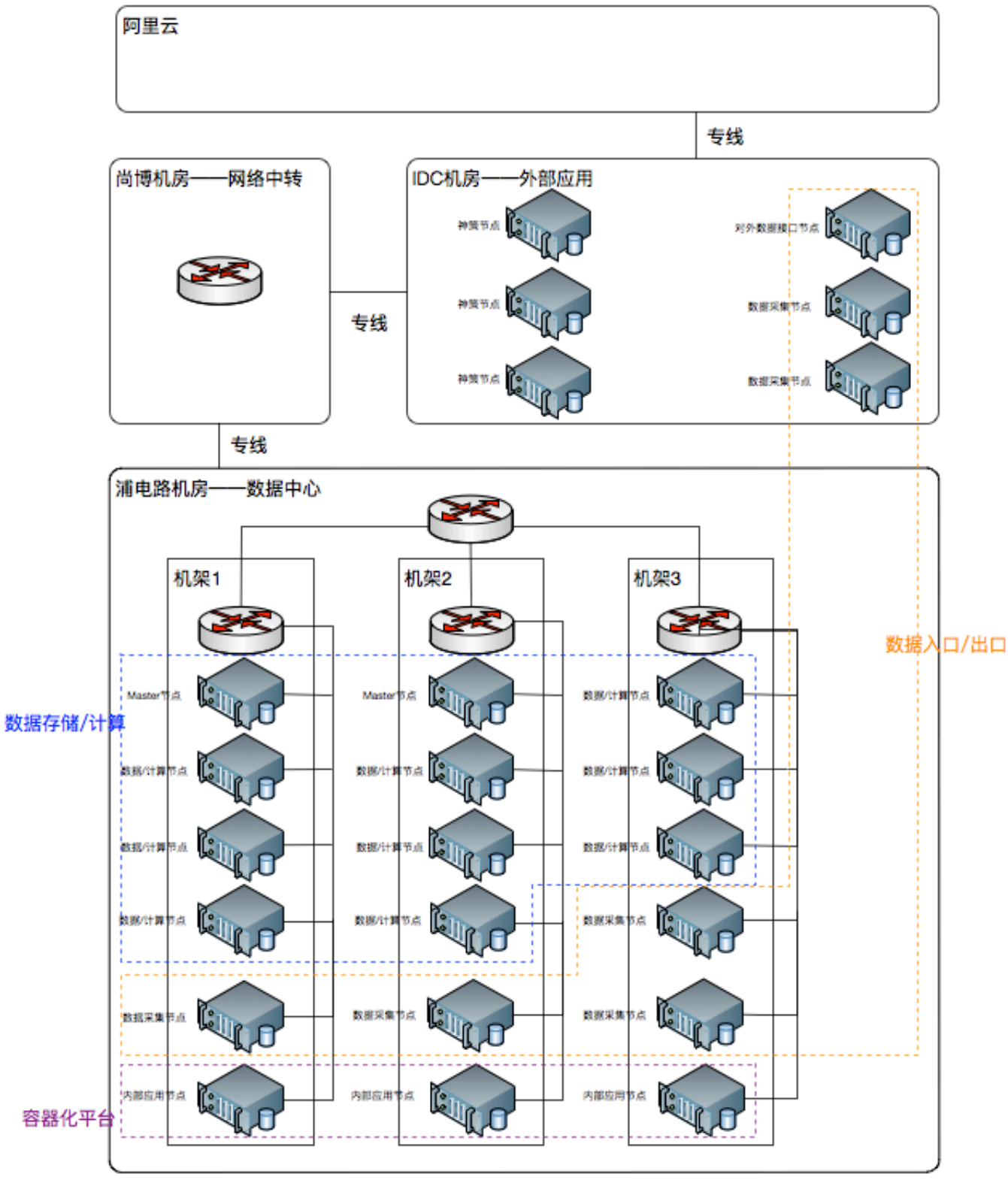


2.3 物理架构

服务器资源清单

机器资源	用途	物理位置	数量
R730 E5-2640 V4*2/16G*8	神策埋点数据采集	IDC机房	3
R730 E5-2650 V4*2/16G*16	SQLServer节点	浦电路机房	1
R730 E5-2650 V4*2/16G*16	数据采集	浦电路机房	3
R730 E5-2650 V4*2/16G*16	对外应用部署节点	IDC机房	2
R730 E5-2650 V4*2/16G*16	容器化应用部署环境	浦电路机房	3
R730 E5-2650 V4*2/16G*16	大数据平台测试环境	浦电路机房	6
R730 E5-2650 V4*2/32G*12	大数据平台生产环境	浦电路机房	11

物理拓扑图(生产环境相关)



2.4 迁移计划

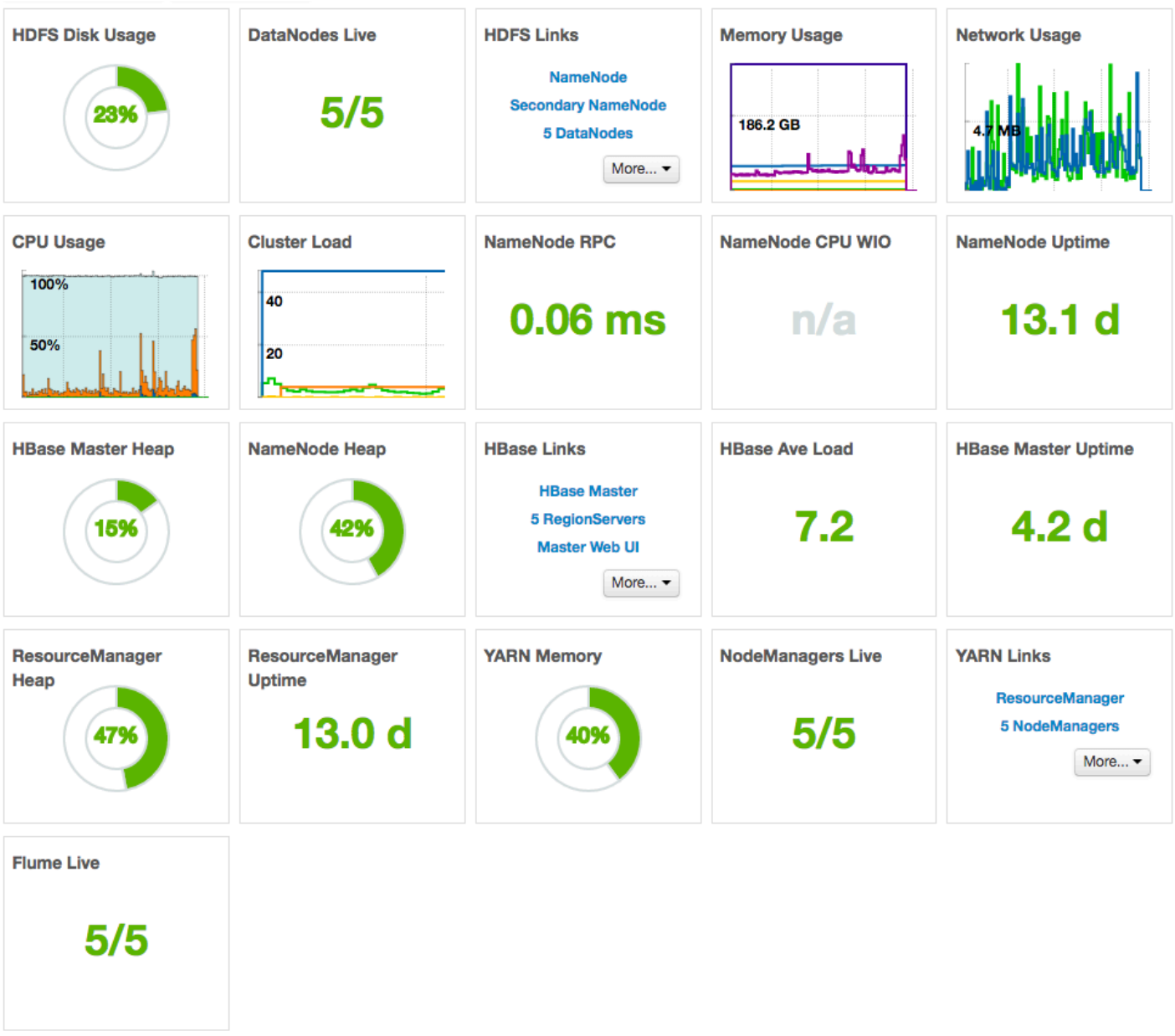
目前在测试环境的大数据平台上，我们已经接入了几个TB的数据，在生产环境完成建设滞后

3. 平台功能和关键技术

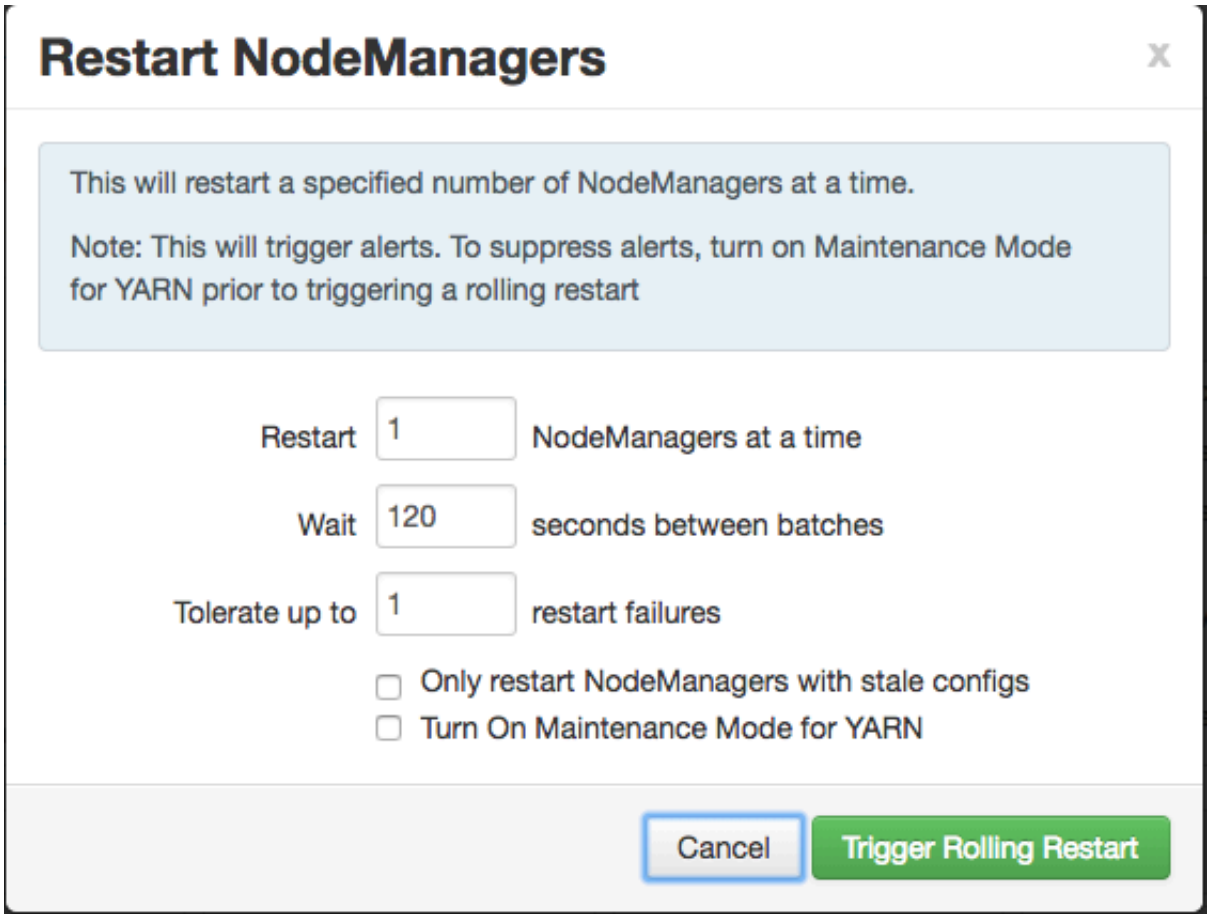
3.1 基于HDP+Ambari的管理平台

核心功能

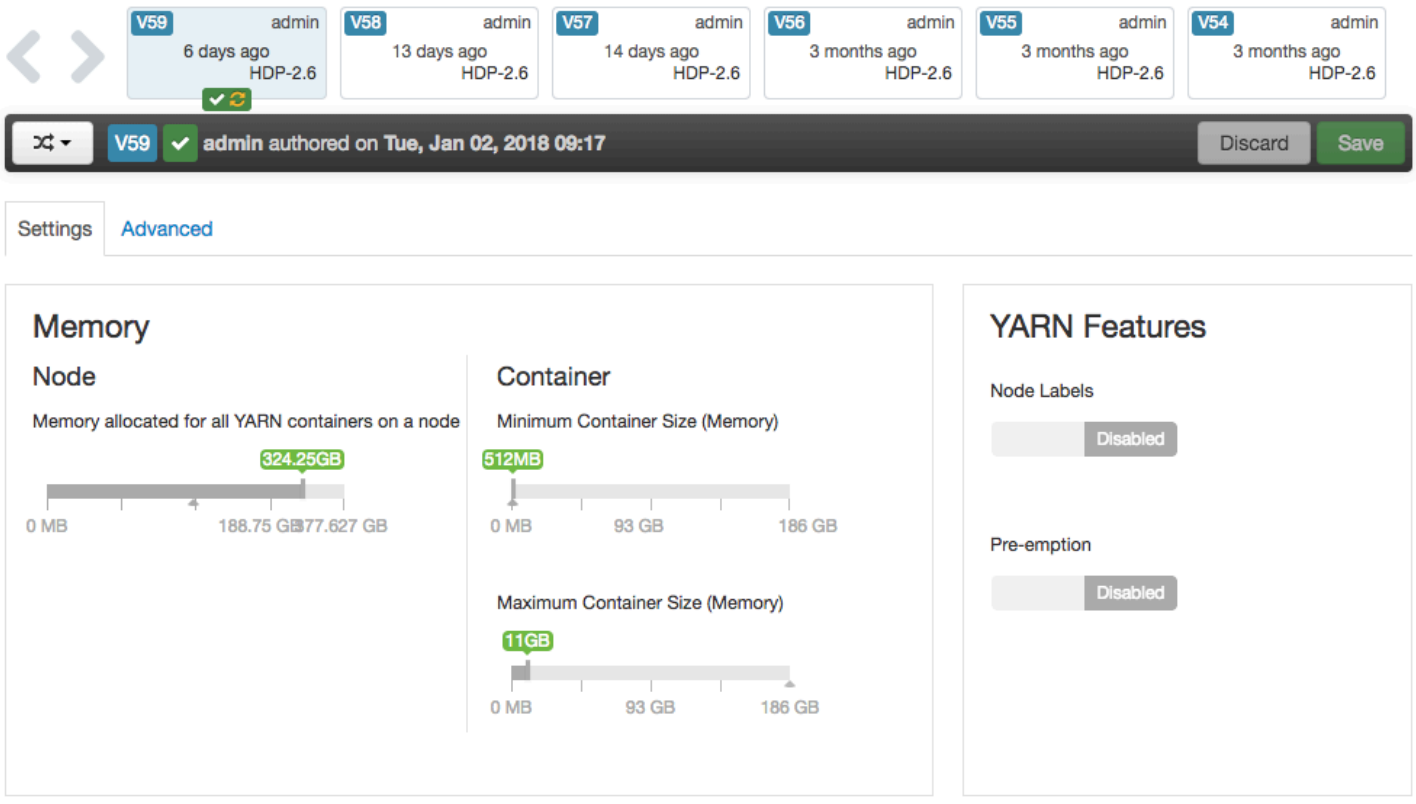
- 系统监控



- 滚动升级


Restart NodeManagers
This will restart a specified number of NodeManagers at a time.
Note: This will trigger alerts. To suppress alerts, turn on Maintenance Mode for YARN prior to triggering a rolling restart
Restart 1 NodeManagers at a time
Wait 120 seconds between batches
Tolerate up to 1 restart failures
Only restart NodeManagers with stale configs
Turn On Maintenance Mode for YARN
Cancel
Trigger Rolling Restart

- 配置管理


V59 admin 6 days ago HDP-2.6
V58 admin 13 days ago HDP-2.6
V57 admin 14 days ago HDP-2.6
V56 admin 3 months ago HDP-2.6
V55 admin 3 months ago HDP-2.6
V54 admin 3 months ago HDP-2.6
V59 admin authored on Tue, Jan 02, 2018 09:17
Discard
Save
Settings
Advanced
Memory
Node
Memory allocated for all YARN containers on a node
324.25GB
0 MB
188.75 GB
377.627 GB
Container
Minimum Container Size (Memory)
512MB
0 MB
93 GB
186 GB
Maximum Container Size (Memory)
11GB
0 MB
93 GB
186 GB
YARN Features
Node Labels
Disabled
Pre-emption
Disabled

当前版本

	HDP-2.6.3.0 Show Details
HDFS	2.7.3
YARN	2.7.3
MapReduce2	2.7.3
Tez	0.7.0
Hive	1.2.1000
HBase	1.1.2
Pig	0.16.0
Sqoop	1.4.6
Oozie	4.2.0
ZooKeeper	3.4.6
Ambari Metrics	0.1.0
Ranger	0.7.0
SmartSense	1.4.3.2.6.0.0-267
Spark	1.6.3
Spark2	2.2.0
Kerberos	1.10.3-10
Slider	0.92.0
Solr	5.5.2

技术选型

业内有多种成熟的大数据平台解决方案，其中最为流行和成熟的包括Cloudera的CDH以及Hortonworks的HDP，而CDH又分为社区版和商业版。比较几种方案，其中，CDH社区版中不包含企业级的数据安全、数据治理等必要功能，而CDH商业版价格昂贵(50W/年)，开源性较差，不利于后期基于平台扩展具有企业特色的

功能。相反，HDP虽然在商业支持上不及CDH，但是，所有组件均为开源，社区活跃。因此，我们选用HDP作为大数据平台的基础解决方案。

3.2 基于HDFS原始数据存储

核心功能

Hadoop分布式文件系统(HDFS)被设计成适合运行在通用硬件上的分布式文件系统。HDFS是一个高度容错性的系统，适合部署在廉价的机器上。HDFS能提供高吞吐量的数据访问，非常适合大规模数据集上的应用。HDFS放宽了一部分POSIX约束，来实现流式读取文件系统数据的目的。

Hive, HBase 大数据存储都基于 HDFS，我们从各种数据源同步至 Hive 中的库表后最终都以文件形式存放在 HDFS 中。

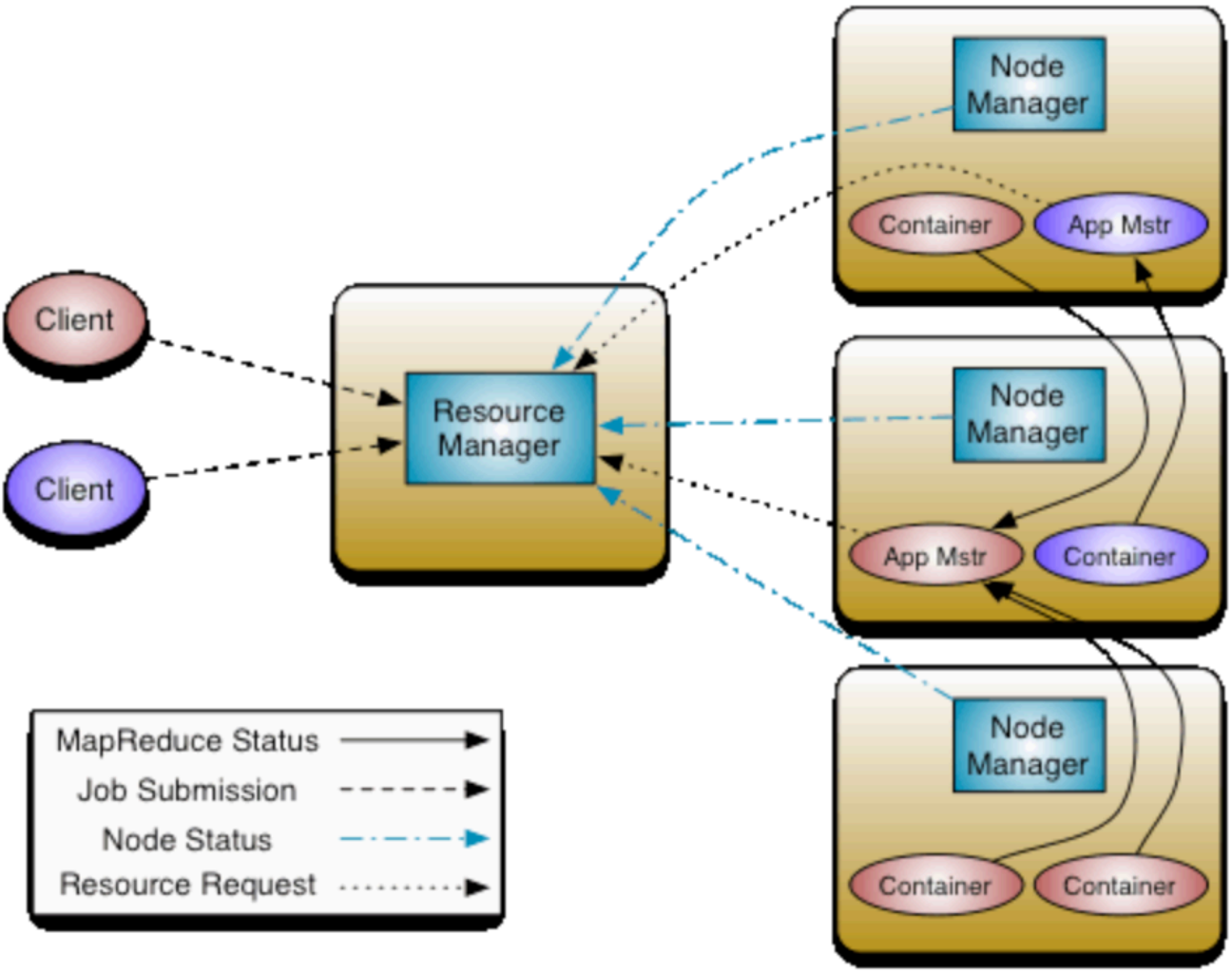
存储规划

集群规划有9个存储节点，其中每个节点出去操作系统盘，分别设置10块3.5T HDD作为数据存储盘，一块3.5T HDD作为日志盘。数据备份选择为3份，因此整个集群的实际数据容量约为100T。

机器名	ip	机架	硬盘	备注
namenode1.prod.bigdata.ytx.com	192.168.150.18	1	2*600G SSD Raid1	操作系统Centos 7
			2*3.5T HDD Raid1	NameNode
			2*3.5T HDD Raid1	Solr
			1*3.5T HDD Raid0	日志
namenode2.prod.bigdata.ytx.com	192.168.150.122	2	2*600G SSD Raid1	操作系统Centos 7
			2*3.5T HDD Raid1	NameNode
			2*3.5T HDD Raid1	Solr
			1*3.5T HDD Raid0	日志
datanode1.prod.bigdata.ytx.com	192.168.150.20	1	2*500G SSD Raid1	操作系统Centos 7
datanode2.prod.bigdata.ytx.com	192.168.150.22		10*3.5T HDD JBOD	DataNode
datanode3.prod.bigdata.ytx.com	192.168.150.24		1*3.5T HDD Raid0	日志
datanode4.prod.bigdata.ytx.com	192.168.150.123	2	2*500G SSD Raid1	操作系统Centos 7
datanode5.prod.bigdata.ytx.com	192.168.150.131		10*3.5T HDD JBOD	DataNode
datanode6.prod.bigdata.ytx.com	192.168.150.132		1*3.5T HDD Raid0	日志
datanode7.prod.bigdata.ytx.com	192.168.150.133	3	2*500G SSD Raid1	操作系统Centos 7
datanode8.prod.bigdata.ytx.com	192.168.150.26		10*3.5T HDD JBOD	DataNode
datanode9.prod.bigdata.ytx.com	192.168.150.174		1*3.5T HDD Raid0	日志

3.3 基于Yarn的资源调度

目前生产系统节点 Spark worker 可用节点为9个，每个节点有48个核，377 GB 内存，累计 432 个核，3300 GB 内存，通过 Yarn 进行调度。



- dataflow, 35%资源， 数据采集和汇入、ETL
- bi，45%资源，数据查询、...
- app, 数据应用
- admin, 特殊任务队列
- default, 暂未分类的任务队列

考虑到每日的数据同步任务，通过数据量调研，我们为数据采集分配了至少35%的队列资源。因平台的重要功能体现在事业部的数据查询上，使用次数较为频繁，查询的数据量多，任务需求紧急，故而为其分配至少45%的队列资源。

3.4 基于Hive的结构化数据存储

核心功能

Hive是建立在 Hadoop 上的数据仓库基础构架。它提供了一系列的工具，可以用来进行数据提取转化加载（ETL），这是一种可以存储、查询和分析存储在 Hadoop 中的大规模数据的机制。Hive 定义了简单的类 SQL 查询语言，称为 HQL，它允许熟悉 SQL 的用户查询数据。

简单的说，Hive是Hadoop体系中最接近传统数据库存储的组件。

核心参数

平台中的表默认采用ORC格式的存储方式，ORC全名为Optimized Row Columnar File，是一种面向大数据场景的列式存储引擎，其具有很高的压缩比例和查询效率。对比另外一种常用的存储格式Parquet，ORC在嵌套式数据支持不够完整，但是ORC支持事务，支持ACID操作，因此针对平台常见的使用场景，我们选用了ORC作为默认存储格式。

查询性能

TPC-DS

随着国内外各代表性的Hadoop发行版厂商以TPC-DS为标准测评产品，TPC-DS也就逐渐成为了业界公认的大数据系统测试基准。这个测试集包含对大数据集的统计、报表生成、联机查询、数据挖掘等复杂应用，测试用的数据和值是有倾斜的，与真实数据一致。可以说TPC-DS是与真实场景非常接近的一个测试集，也是难度较大的一个测试集。

TPC-DS 测试结果

> TODO: 测试结果

技术选型

Hive是大数据领域近SQL查询的行业标准，类似的标准还有pig，但Hive的使用更广泛，生态发展也更好。

Hive的执行性能因执行引擎而有所区别，我们考察了流行的查询引擎，包括Impala，Tez，MR，Hive2Llap，最终我们选择Hive2Llap作为新集群的默认查询引擎。

> TODO：上面一段写得不准确，用户其实是可以用spark sql之类的的。另需补充兼容性等其他对比内容。

以下为以实际业务查询为例我们做的不同查询引擎下的基线测试

> TODO：性能对比测试柱状图

[性能对比测试]

3.5 基于NiFi的数据采集体系

核心功能

Apache NiFi的设计目标是自动化系统间的数据流，它的主要功能包括：

- 友好的用户交互界面能更加直观的监控和管理数据的流向
- Apache NiFi支持的多种数据源如ftp、kafka、主流关系型数据库、Hive等，可以适用于大多数的数据是接入场景

- 数据自然缓存提高了数据容错性同时易于跟踪和异常定位
- NiFi采用分布式高并发框架，具有很大的数据吞吐量及处理能力
- 为数据自动化接入工作提供安全可靠的处理平台

NiFi可以方便的大数据平台集成。

物理规划

- 测试节点1个：大数据测试环境NiFi节点，为数据流向图的提供测试环境
- 生产节点5个：
  - 3个内部节点：为内网数据流提供数据接入平台
  - 2个外部节点：用于公网数据接入
  - 集群的主节点由Zookeeper选举

技术选型

NiFi作为一个面向数据流的数据管理平台，相比于kettle面向加载静态数据的缺点，NiFi能够处理实时的数据流，数据的自然缓存能有效的降低数据丢失的风险。NiFi基本涵盖了Sqoop的全部功能，并且提供给了额外的交互界面。我们综合考察了不同ETL工具对数据采集场景的适应性、实时性、并发性等特点，选定了NiFi作为数据采集平台。

3.6 基于Spark大数据运算体系

核心功能

Spark 是一个基于内存的通用分布式计算引擎，通过 RDD(弹性分布式数据集)对处于内存和磁盘的数据进行了高度抽象并对上层提供了极其便利的高阶调用。

选用 Spark 作为我们集群上的分布式计算引擎能够完美地和当前大数据集群进行深度整合并进行复杂而且高效的分析计算。使用Spark的典型场景如： - 用户行为数据的清洗及入库 - 智能推荐和文本反垃圾广告 - 客户画像的后端计算引擎

技术选型

- 运算速度快：和 Hadoop 的 MapReduce 相比，Spark 可以让程序在内存中运行时速度提升100倍，或者在磁盘上运行时速度提升10倍
- 开发易用性好：API 同时支持 Java, Scala, Python 和 R 四种编程语言，提供80多种高阶算子构建大规模集群并行计算程序
- 使用通用性强：同时支持 SQL 查询，流数据，机器学习和复杂数据处理，有 Spark Streaming, Spark SQL, Spark MLlib 等附加库支持
- 数据源支持广：可以很好地和 Hadoop 生态系统和数据源（HDFS、Hive、HBase等）进行集成，同时也支持通过 JDBC 读取 SQL Server, MySQL 等传统关系型数据库

比较其他分布式计算引擎，还没有发现能和 Spark 在开发运行效率和生态系统综合指标相媲美的产品。

3.7 基于Rancher的Docker容器集群

核心功能

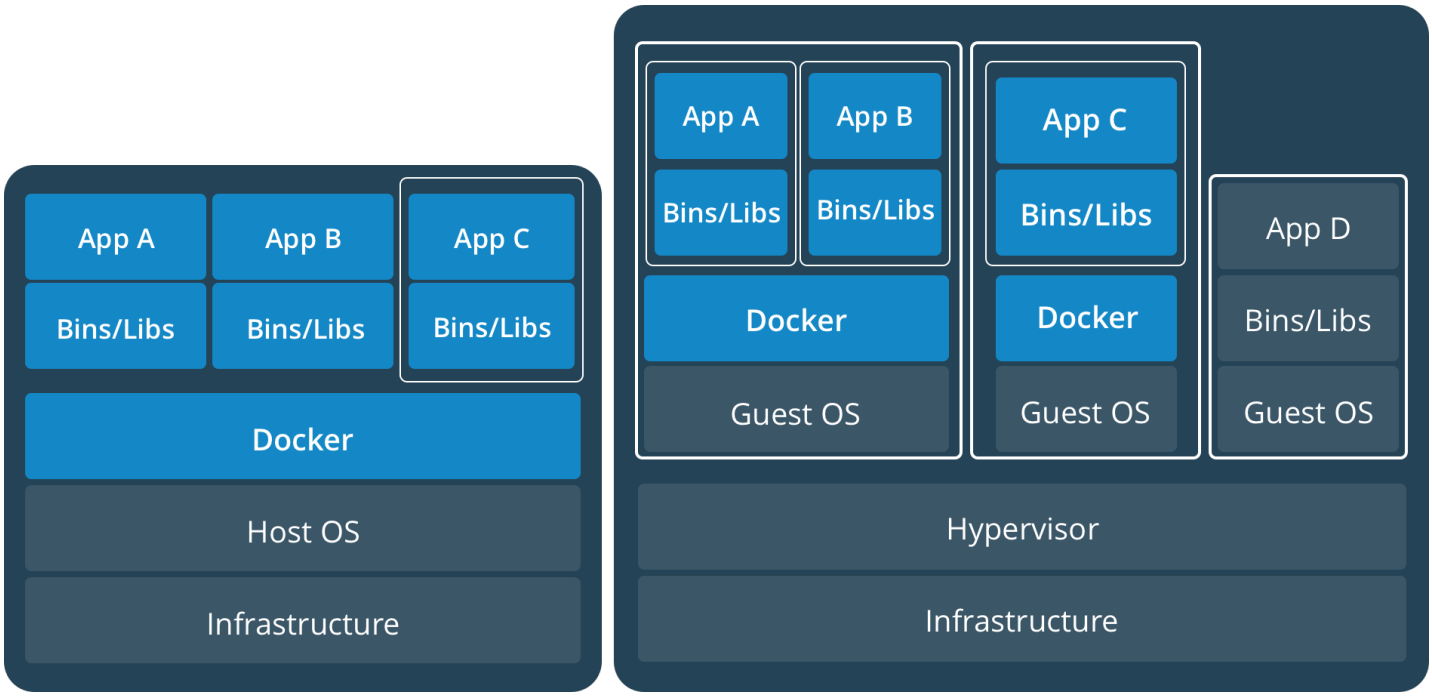
除了构建大数据的存储与计算平台，我们仍需要开发一些数据产品与应用，因此，我们就需要对这一部分服务器硬件资源做有效的管理。

Rancher是一个开源的企业级全栈化容器部署及管理平台。它提供了网络服务、存储服务、主机管理、负载均衡、防火墙等基础架构服务，并能够使上述服务跨越公有云、私有云、虚拟机、物理机环境运行，真正实现一键式应用部署和管理。此外，Rancher可以与各种CI/CD工具协同工作，可以实现开发、测试、预生产和生产环境的自动部署，提供整体可视化的主机、容器、网络及存储管理，大幅简化开发和运维人员故障排除和生产部署的工作量。

技术选型

docker vs 虚拟化

相比于虚拟化技术，容器化技术拥有更高效的资源利用率，更轻量的部署形式。



而从大数据团队内部考虑，我们没有独立的运维人员但又有各种应用的部署需求，并且需要持续集成的环境支持我们快速迭代，因此，我们选用Docker作为容器化方案。

Rancher vs Kubernetes

除了Rancher之外，Kubernetes也提供了较为完整的容器编排工具和UI，简单比较如下

	k8s	Rancher
UI	kubeUI交互性略差	交互更友好
容器伸缩	支持	支持
环境管理	需二次开发	支持
项目管理	支持	支持
资源管理	更加完整，细粒度	仅支持根据机器信息选择机器
监控	有完整的方案	自带的监控比较弱，需要自建
日志	不支持web方式查看	支持web方式查看

结合我们的场景，两者功能差距不大，而我们在易用性上更加注重，因此选择Rancher。

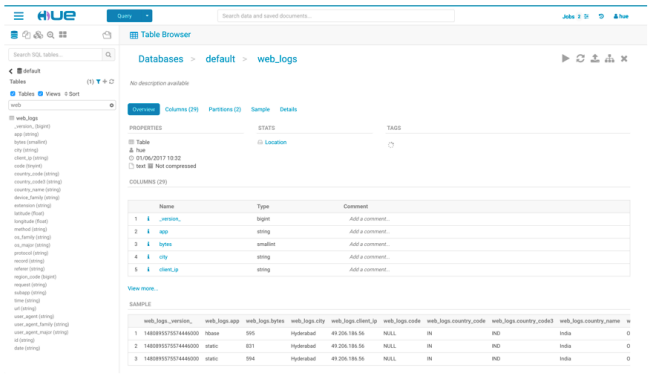
3.8 基于Hue的数据访问

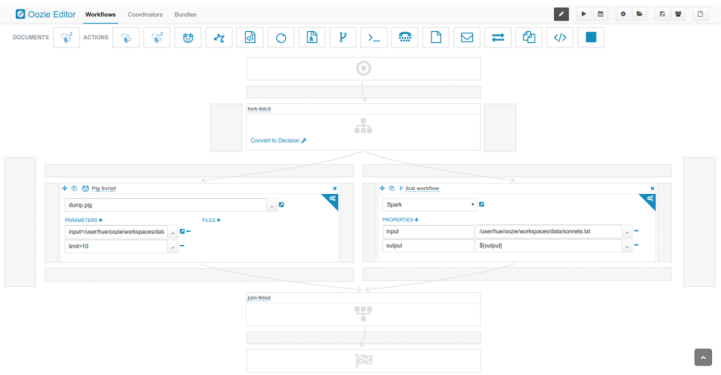
核心功能

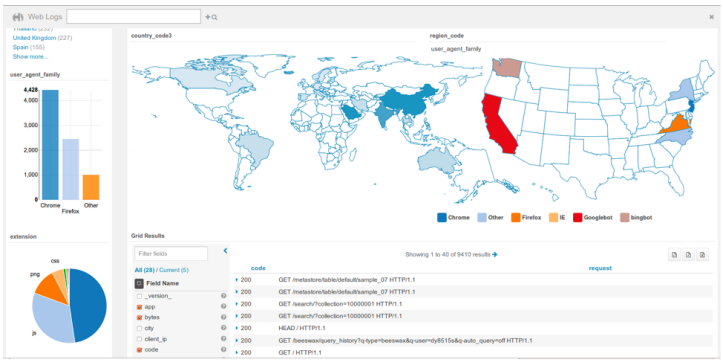
Hue的核心功能包括

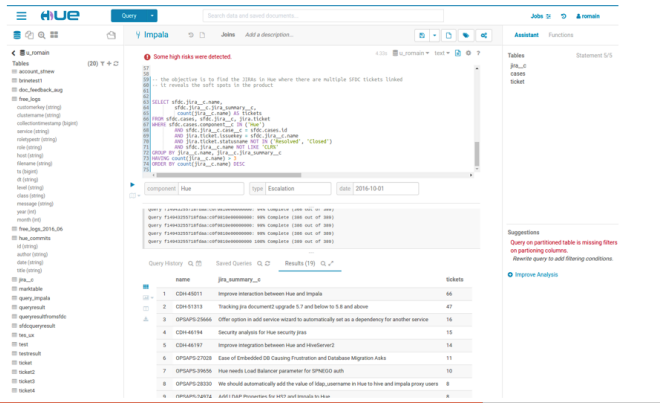
- 提供方便的SQL编辑和执行功能
- 提供简单易用的仪表盘来展示数据
- 提供创建和调度 Oozie 任务的界面

Hue 定位为 Hadoop 平台各组件的 Web UI，可通过浏览器访问大数据集群内部各组件及数据仓库，入口可统一用户认证及鉴权，更容易管理和升级。查询数据时类似于传统数据库的SQL客户端，同时Hue还承担类似于传统架构中Jenkins或Kettle的工作。









技术选型

尽管大数据集群下其他组件基本都有独立的 Web UI 可用，但使用时难以达到集中鉴权管理和使用上的便利，因此我们最终选用了 Hue 这一目前非常成熟的 Hadoop Web UI.

3.8 其他组件

工作流引擎 Oozie

Apache Oozie是用于调度Apache Hadoop作业的Java应用程序。 Oozie将多个作业按顺序组合成一个逻辑工作单元。 它与Hadoop技术栈集成，以YARN为架构中心，并支持Apache MapReduce，Apache Pig，Apache Hive和Apache Sqoop的Hadoop作业。 Oozie还可以调度特定于系统的作业，如Java程序或shell脚本。

Apache Oozie是一个Hadoop操作工具，它允许集群管理员从多个组件任务中构建复杂的数据转换。 这提供了对作业的更好的控制，也使得在预定的时间间隔重复这些作业变得更容易。

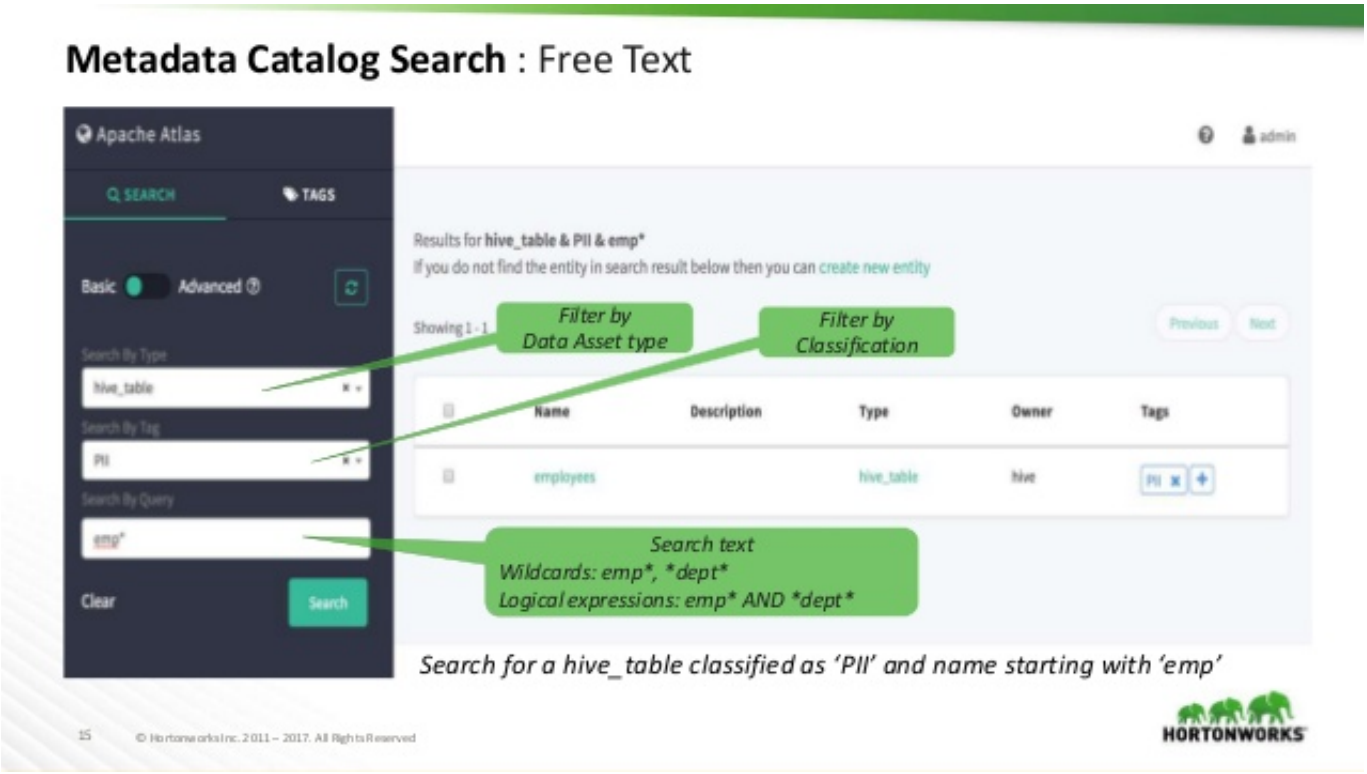
Apache Oozie的角色类似于BI工作中常用的一些任务管理工具如Kettle和Talend。典型的使用场景是执行复杂的一系列数据任务，或需要定时/重复执行一些任务。

元数据管理工具 Atlas

Atlas是一套数据治理服务，它帮助企业有效地满足Hadoop中的合规要求，并允许与整个企业数据生态系统进行整合。它的主要功能包括：

- 数据分类，导入或添加数据相关的标签
- 中心化的数据审计
- 搜索或关联数据

在过去，Atlas所承担的数据管理工作是由一系列的word或excel文档来完成，如数据字典文档。使用Altas，不仅我们可以自动获取数据转移过程信息，也将有统一的地方对元数据进行管理。



因Atlas项目仍处于初级阶段（当前版本0.9），团队有计划在Atlas之上开发我们急需的功能。

4. 典型平台使用场景

4.1 数据汇聚和采集

一个标准的从数据库采集数据流程如下：

1. 安全运维部门获知一个新的数据源的上线或部署
2. 数据团队提供一系列NiFi的处理器（同步、增量、数据结构变更检查等）
3. 启动NiFi处理器

有时数据会以文件方式提供，一个标准的从文件格式采集数据的方案是：

1. 数据提交者将文件放至HDFS指定位置
2. 数据团队提供一系列NiFi的处理器（同步、增量、数据结构变更检查等）
3. 启动NiFi处理器

NiFi的处理器有时会需要数据团队来编写，但是大多数情况下使用早已编好的处理器模板即可。 我们现在已

提供的处理器模板包括：

- 全量同步数据
- 增量同步数据
- 检查数据表结构变更
- 解析标准的csv

数据采集的过程在一定时间内仍需要人员参与，而不是完全自动的。NiFi需要源的连接信息，以及数据增长方式等信息来高效的同步数据。NiFi有友好的界面来管理数据采集任务，但它主要是面向数据运营人员使用的。我们有计划开发更傻瓜式的界面来启动和监控NiFi的工作。

4.2 分析师数据访问

数据分析师可以通过Hue、Zeppelin或者其他客户端来连接大数据集群。

4.3 ETL批量处理数据

数据团队和分析师都可以在集群上运行数据处理的程序，标准的做法是在hue中创建一个ozzie的任务并运行。

数据处理程序可以连接HDFS、Hbase或Hive。集群内的程序无法访问集群外的数据源，如果需要使用这种数据，需要先把它们同步到集群中来。

数据处理程序的结果可以写到Hive、HDFS或Hbase中，并不能直接写到集群外的数据地址，如外部的一个mysql。如果有这种需求，你需要在一个独立的程序把数据结果同步到集群外部。

4.3 流处理数据

大数据平台的建设目的不仅限于提供查询和分析，更希望在智能应用、实时计算等方面提供支持，因此大数据平台也设计为包含支持面向实时处理的流处理框架。

我们建议数据工程师创建一个Spark程序，打包后从Oozie提交集群运行。

目前比较使用比较广泛的流处理框架如Storm、Spark和Flink。相较而言，Storm完全针对于流式处理设计，对于延迟需求很高的纯粹的流处理工作负载，其可能是最适合的技术。而Spark和Flink均能同时支持批处理和流式处理，不同之处在于Spark基于短批模型来处理流式数据，Flink则是设计了纯粹的流式模型。

考虑平台目前的使用场景，以同时具有批处理和流处理的混合型工作场景居多，而Spark相较Flink更加成熟和稳定，因此，前期选择Spark作为平台的流式处理框架，但保留之后选用其他框架的扩展性。

5. 平台安全体系

5.1 访问控制

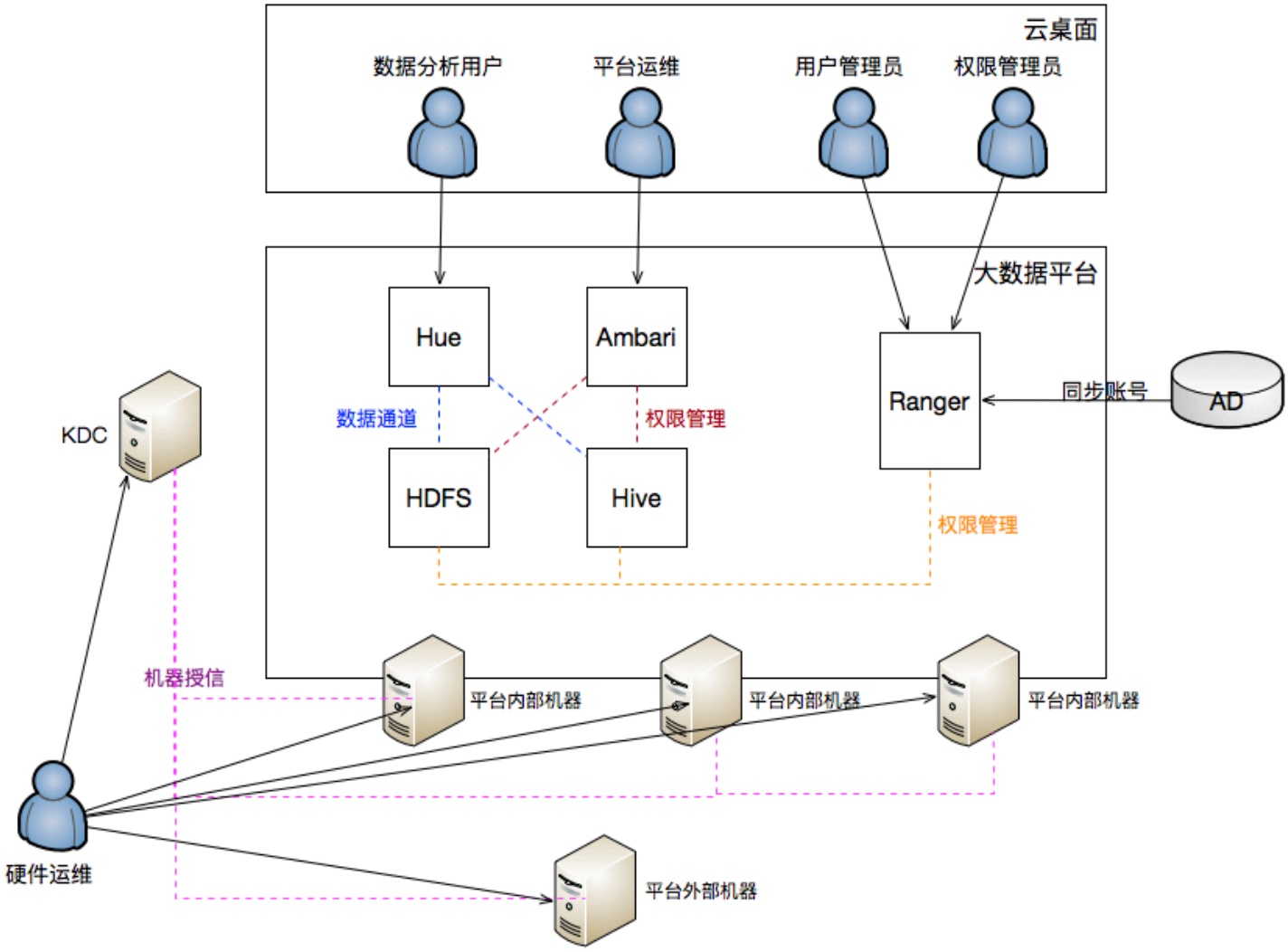
平台用户角色及访问控制要求

根据平台的使用和运维场景，我们本着职能单一、权限隔离的原则，将用户分为以下几种角色

角色	允许	禁止
硬件运维	管理服务器资源、网络、安全	接触到数据文件但无法还原数据
平台运维	大数据平台软件方面的常规维护、升级及故障处理	一般情况下无法接触服务器 无法修改平台组件的使用密码
用户管理员	管理平台用户与用户组的对应关系	无法增加、删除、修改用户
权限管理员	管理用户组对数据的使用权限	无法控制用户和用户组的关系
数据分析人员	以数据应用部提供的方式使用数据(如Hue、作战室、Excel等)	无法在云桌面外访问数据 无法访问没有被授权的数据 无法提升自己的权限
应用程序	以程序的形式使用大数据平台数据或计算资源	无法在指定服务器外访问数据 无法访问没有被授权的数据/资源 无法提升自己的权限

技术实现

在平台内部，Hadoop平台上的各个组件大多本身就自带访问控制体系，可以根据用户和组来控制访问，但是往往来讲，用户的一个行为如查询会经过多个平台组件，因此我们选用了Ranger来做集中式的权限管理，使用户管理和权限管理都能方便实现。 而在平台外部，我们希望避免恶意的使用或篡改Hadoop集群的问题，同时也不愿意将密码在网络上传输，我们选用基于Kerberos的认证体系，通过硬件运维人员管理可信赖的机器节点，保证物理环境的安全。



5.2 授权机制

授权原则

授权流程

用户获得数据权限的流程如下

1. 公司员工的统一账号都存在AD中，这些账号会定期同步至Ranger中
2. 权限管理员根据公司业务预设用户组，例如qzg_security表示期掌柜的投顾业务，并对每一个用户组分配对应的数据权限
3. 用户管理员根据用户所参与数据工作将用户分至对应的组里，统一用户可对应多个组

按照此流程，我们能够确保

- 离职员工账号能够及时清理
- 分离了用户管理和权限管理，避免用户给自己提升权限的可能
- 能够响应数据使用人员申请使用的数据与其本身所在的事业部不一致的情况

5.3 审计和日志

平台日志和审计要求

平台的访问、操作均需被记录，且支持在一定时间内进行审计。其中包括：

- 物理机器的访问
- 平台组件的新增、删除、停止、升级等操作
- 数据的增加、查询、修改、删除、下载等操作

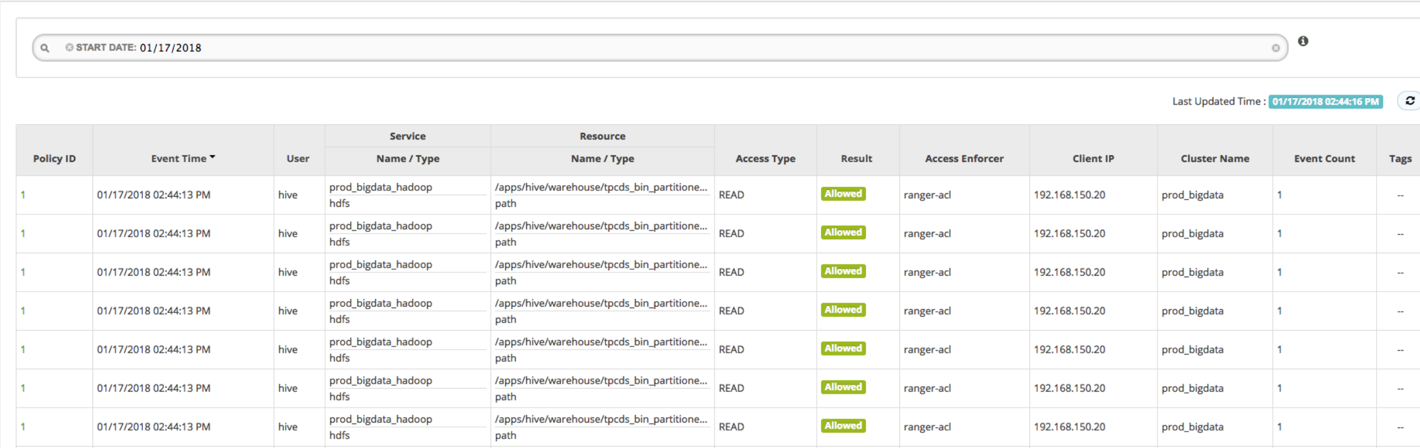
技术实现方式

对于不同的操作，我们以不同的方式来进行日志记录

- 物理机的访问仅能通过堡垒机进行，所有行为都会被记录，此部分日志被安全运维团队管理和审计
- 平台组件的操作会被Ambari记录

Service	Config Group	Created	Author	Notes
All	All	Any	Any	Any
V40 YARN	Default Current	Wed, Jan 17, 2018 14:30	admin	
V59 Hive	Default Current	Wed, Jan 17, 2018 14:19	admin	
V39 YARN	Default	Wed, Jan 17, 2018 14:18	admin	
V13 Oozie	Default Current	Wed, Jan 17, 2018 14:17	admin	更正正确密码
V58 Hive	Default	Wed, Jan 17, 2018 13:54	admin	
V57 Hive	Default	Wed, Jan 17, 2018 13:53	admin	
V56 Hive	Default	Wed, Jan 17, 2018 13:52	admin	
V12 Oozie	Default	Wed, Jan 17, 2018 13:49	admin	Created from service config version V10
V55 Hive	Default	Wed, Jan 17, 2018 13:45	admin	
V11 Oozie	Default	Wed, Jan 17, 2018 13:45	admin	调整smtp认证方式

- 数据的操作会被Ranger记录并保留最长一年



START DATE: 01/17/2018

Last Updated Time : 01/17/2018 02:44:16 PM

Policy ID	Event Time	User	Service Name / Type	Resource Name / Type	Access Type	Result	Access Enforcer	Client IP	Cluster Name	Event Count	Tags
1	01/17/2018 02:44:13 PM	hive	prod_bigdata_hadoop hdfs	/apps/hive/warehouse/tpcds_bin_partitione... path	READ	Allowed	ranger-acl	192.168.150.20	prod_bigdata	1	--
1	01/17/2018 02:44:13 PM	hive	prod_bigdata_hadoop hdfs	/apps/hive/warehouse/tpcds_bin_partitione... path	READ	Allowed	ranger-acl	192.168.150.20	prod_bigdata	1	--
1	01/17/2018 02:44:13 PM	hive	prod_bigdata_hadoop hdfs	/apps/hive/warehouse/tpcds_bin_partitione... path	READ	Allowed	ranger-acl	192.168.150.20	prod_bigdata	1	--
1	01/17/2018 02:44:13 PM	hive	prod_bigdata_hadoop hdfs	/apps/hive/warehouse/tpcds_bin_partitione... path	READ	Allowed	ranger-acl	192.168.150.20	prod_bigdata	1	--
1	01/17/2018 02:44:13 PM	hive	prod_bigdata_hadoop hdfs	/apps/hive/warehouse/tpcds_bin_partitione... path	READ	Allowed	ranger-acl	192.168.150.20	prod_bigdata	1	--
1	01/17/2018 02:44:13 PM	hive	prod_bigdata_hadoop hdfs	/apps/hive/warehouse/tpcds_bin_partitione... path	READ	Allowed	ranger-acl	192.168.150.20	prod_bigdata	1	--
1	01/17/2018 02:44:13 PM	hive	prod_bigdata_hadoop hdfs	/apps/hive/warehouse/tpcds_bin_partitione... path	READ	Allowed	ranger-acl	192.168.150.20	prod_bigdata	1	--

5.4 灾备和恢复

可靠性要求

平台的可靠性包括

1. 保证历史数据不丢失
2. 保证在线数据服务24*7不间断
3. 保证数据接入任务24*7不间断
4. 保证数据使用人员在工作时间不间断可用

技术实现

- 单个节点方面，对于重要的数据存储在RAID1方式的磁盘阵列上，保证单块硬盘出现故障时不导致系统奔溃。其中包括操作系统数据，NameNode数据，JournalNode数据。
- 软件层面，平台的基本存储技术HDFS，本身就是以支持分布式存储为目标，每份数据都会被复制在不同的节点上，当单个节点失效时，数据可以从另外的备份上恢复，因此对于DataNode，考虑到平台性能，我们采用JBOD作为基础存储，并设置备份数为3，可以保证集群任意两台机器节点故障时、或者一个机架三台机器节点故障时，数据不会丢失。
- 整个集群方面，设计不存在单节点的服务，如MySQL、NameNode、ResourceManager均按照HA方式部署，可以在服务发生故障的时候自动切换到StandBy节点
- 机制方面，每周设定常规维护窗口，对于HDFS中未完全备份(小于3份)的数据进行手动触发备份