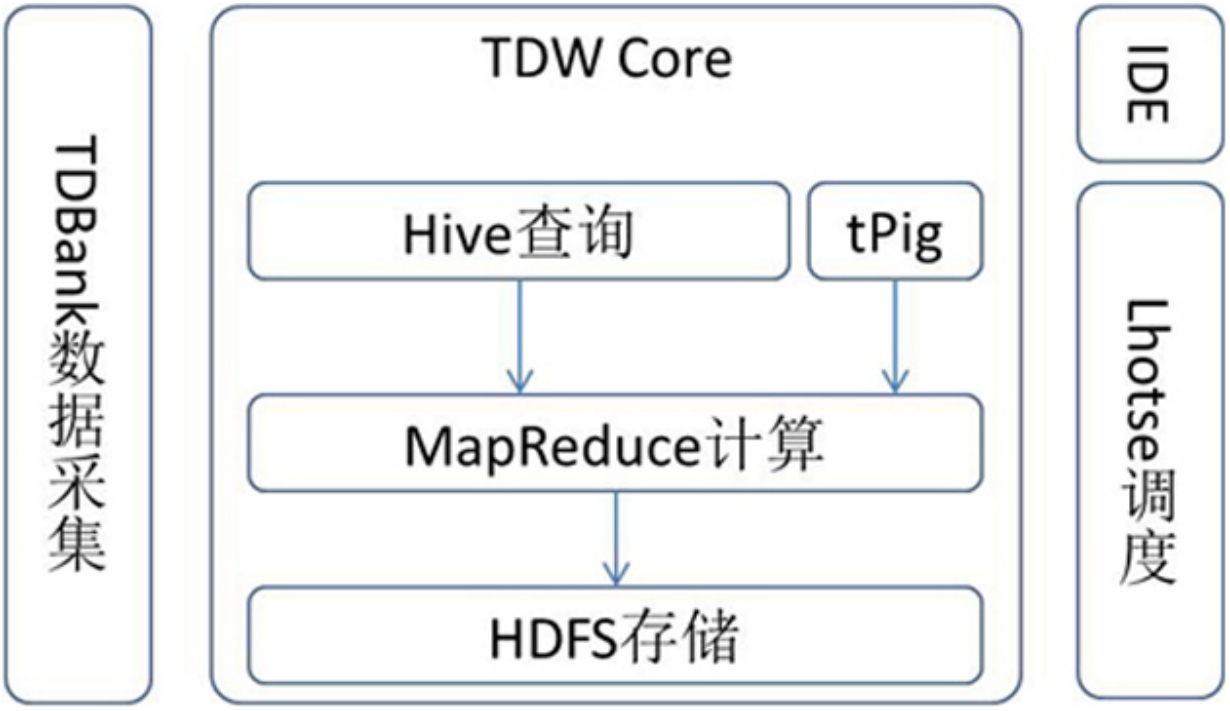
讨论课 #2

分布式存储 | 腾讯 TDW

腾讯分布式数据仓库（Tencent distributed Data Warehouse），其基于 Hadoop 、Hive，打破了传统数据仓库不能线性扩展、可控性差的局限，并且根据腾讯数据量大、计算复杂等特定情况进行了大量优化和改造。
整体架构如下图所示：



TDW Core 主要包括存储引擎 HDFS、计算引擎 MapReduce、查询引擎 Hive，分别提供底层的存储、计算和查询服务，并且根据公司业务产品的应用情况进行了很多的深度订制。
TDBank负责数据采集，旨在统一数据接入入口，提供多样的数据接入方式。Lhotse任务调度系统是整个数据仓库的总管，提供 一站式任务调度与管理。

架构瓶颈

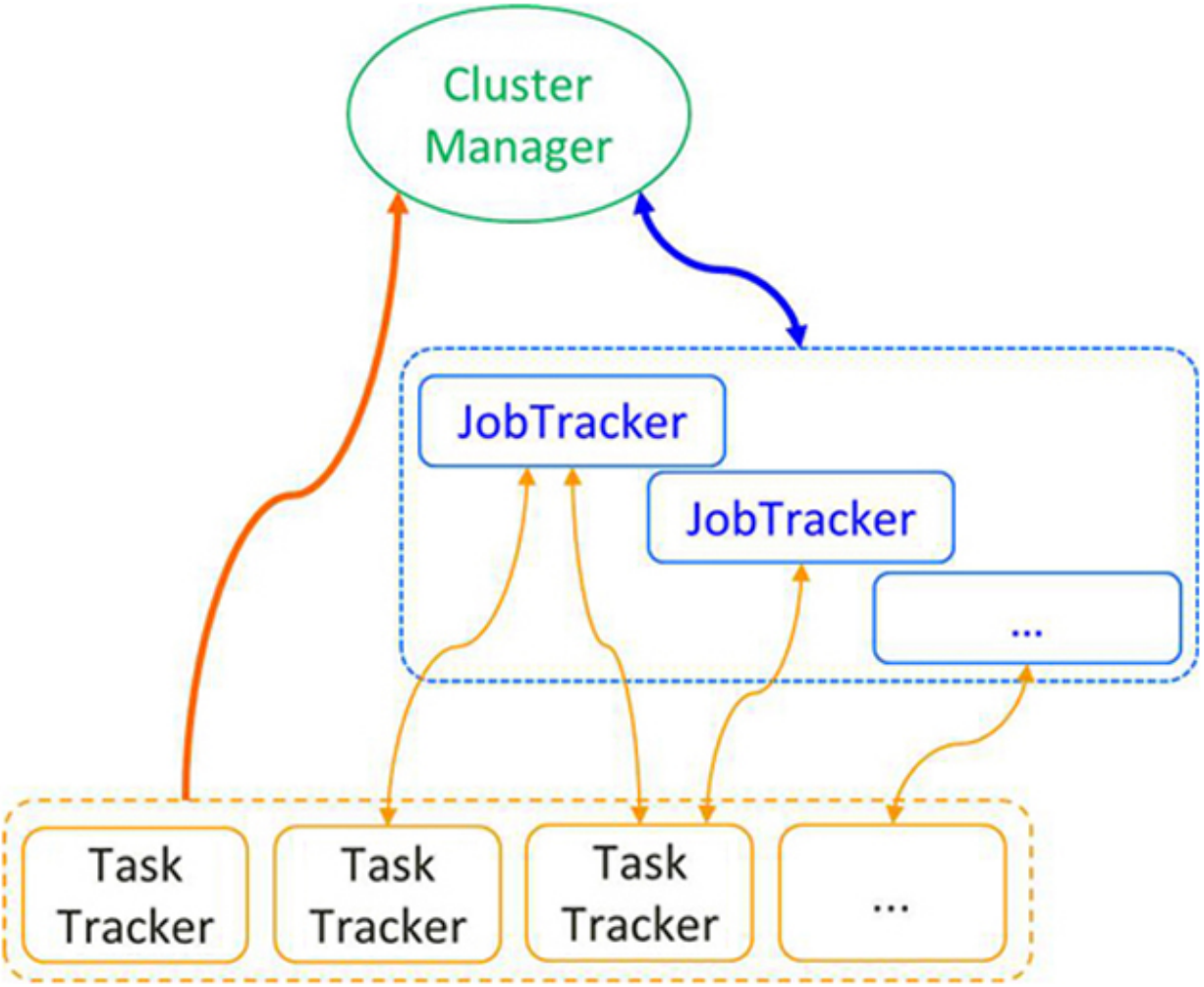
随着业务的快速增长，TDW的节点数也在增加，对单个大规模 Hadoop 集群的需求也越来越强烈。TDW需要做单个大规模集群，主要是从数据共享、计算资源共享、减轻运营负担和成本等三个方面考虑。

1. 数据共享。TDW之前部署了数十个集群，主要是根据业务分别部署，这样当一个业务需要其他业务的数据，或者需要公共数据时，就需要跨集群访问数据，这样会占用集群之间的网络带宽。为了减少跨集群的数据传输，有时会将公共数据冗余分布到多个集群，这样又会带来存储空间浪 费。

2. 计算资源共享。当一个集群的计算资源由于某些原因变得紧张时，例如需要数据补录时，这个集群的计算资源就捉襟见肘，而同时，另一个集群的计算资源可能空闲，但这两者之间没有做到互通有无。
3. 减轻运营负担和成本。十几个集群同时需要稳定运营，而且当一个集群的问题解决时，也需要解决其他集群已经出现的或者潜在的问题。一个 Hadoop版本 要在 十几个集群逐一变更，监控系统也要在十几个集群上部署。这些都给运营带来了很大负担。此外，分散的多个小集群，资源利用率不高，机器成本较大。

解决方案

- JobTracker 分散化
  TDW 的原始架构是传统的两层架构， 单点 JobTracker 负责整个集群的资源整理、任务调度和任务管理，TaskTracker 负责任务执行。JobTracker 的三个功能模块的耦合度非常高，并且由于是都有单一的 Master 结点负责运行，只有在并发任务数较少时，这种架构才能正常运行，而一旦任务数达到2000是，就会出现瓶颈，导致处理迟缓。之后，TDW 借鉴YARN和Facebook版corona设计方案，进行了计算引擎的三层架构优化（如图2所示）：将资源管理、任务调度和任务管理三个 功能模块解耦；JobTracker只负责任务管理功能，而且一个JobTracker只管理一个Job；将比较轻量的资源管理功能模块剥离出来交给新的 称为ClusterManager的Master负责执行；任务调度也剥离出来，交给具有资源信息的ClusterManager负责执行；对性能要求较 高的任务调度模块采用更加精细的调度方式。



新架构下三个角色分别是：ClusterManager负责整个集群的资源管理和任务调度，JobTracker负责单个Job的管理，TaskTracker负责任务的执行。

- NameNode高可用
  在初期，TDW 的存储引擎是单点的 NameNode，在一个业务对应一个集群的情况下，NameNode 压力较小，出故障的几率也较小，而且 NameNode 单点故障带来的影响不会涉及到全线业务。而在把各个小集群统一到大集群之后，NameNode 的压力就会变大，将会频繁地出现故障，故障的影响也会变得非常的严重。
  为此，TDW 设计了一种一主两热备的 NameNode 高可用方案。新架构下 NameNode 角色有三个：一主（ActiveNameNode）两热备 （BackupNameNode）。
  ActiveNameNode 保存 namespace 和 block 信息，对 DataNode 下发命令，并且对客户端提 供服务。BackupNameNode 包括standby 和 newbie 两种状态：standby 提供对 ActiveNameNode 元数据的热备，在 ActiveNameNode 失效后接替其对外提供服务，newbie 状态是正处于学习阶段，学习完毕之后成为 standby。

总结

TDW 从实际情况出发，采取了一系列的优化措施，成功实施了单个大规模集群的建设。为了满足用户日益增长的计算需求，TDW 正在进行更大规模集群的 建设，并向实时化、集约化方向发展。TDW 准备引入 YARN 作为统一的资源管理平台，在此基础上构建离线计算模型和 Storm、Spark、Impala 等各种实时计算模型，为用户提供更加丰富的服务。