比MR至少快5倍的神器，竟然是它

原创 2017-03-30 巩传捷 中兴大数据

文 | 巩传捷@中兴大数据

我们都知道，有人的地方，就有江湖。其实在计算机的世界里，也存在着江湖，今天小哥就带领大家来初探Hive执行引擎Apache TEZ的崛起，看看某种神器是不是真如传说中那样的神奇。

Hive简介

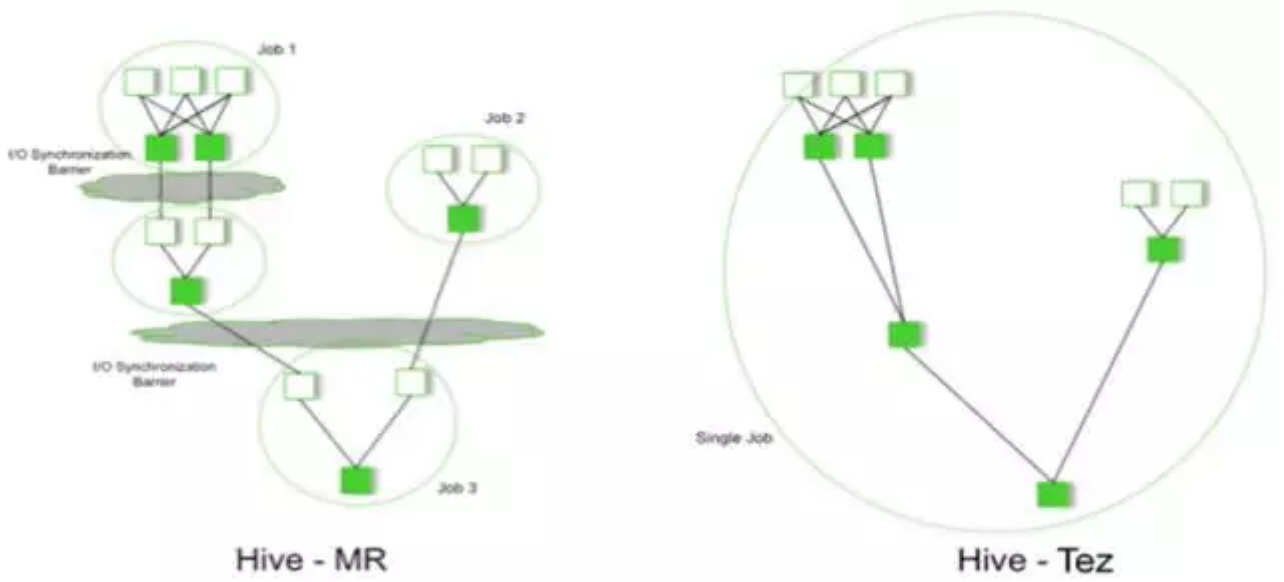
Hive是一个基于 Hadoop 的开源数据仓库工具，用于存储和处理海量结构化数据。它最初是应Facebook对每天产生的海量新兴社会网络数据进行管理和机器学习的需求而产生和发展的，Hive把海量数据存储于Hadoop文件系统，而不是数据库，但提供了一套类数据库的数据存储和处理机制，并采用HQL（类SQL）语言对这些数据进行自动化管理和处理。我们可以把Hive中海量结构化数据看成一个个的表，而实际上这些数据是分布式存储在 HDFS 中的。Hive经过对语句进行解析和转换，最终生成一系列基于 Hadoop 的 Map/Reduce 任务，通过执行这些任务完成数据处理。

Hive的MR和TEZ执行引擎

MapReduce：是一种离线计算框架，将一个算法抽象成Map和Reduce两个阶段进行处理，每个阶段都是用键值对（key/value）作为输入和输出，非常适合数据密集型计算。Map/Reduce通过把对数据集的

大规模操作分发给网络上的每个节点实现可靠性；每个节点会周期性地返回它所完成的工作和最新的状态。如果一个节点在设定的时间内没有进行心跳上报，主节点（可以理解为主服务器）就会认为这个节点down掉了，此时就会把分配给这个节点的数据发到别的节点上运算，这样可以保证系统的高可用性和稳定性。因此它是一个很好的计算框架。

TEZ：是基于Hadoop YARN之上的DAG（有向无环图，Directed Acyclic Graph）计算框架。核心思想是将Map和Reduce两个操作进一步拆分，即Map被拆分成Input、Processor、Sort、Merge和Output， Reduce被拆分成Input、Shuffle、Sort、Merge、Processor和Output等。这样，这些分解后的元操作可以任意灵活组合，产生新的操作，这些操作经过一些控制程序组装后，可形成一个大的DAG作业，从而可以减少Map/Reduce之间的文件存储，同时合理组合其子过程，也可以减少任务的运行时间，具体运行过程如下所示：



Apache TEZ的优化技术

- **ApplicationMaster缓冲池**

在YARN中，用户是将作业直接提交到ResouceManager上，而Apache TEZ则是将作业提交到一个叫AMPoolServer的服务上。当AMPoolServer服务启动后，会预启动若干个ApplicationMaster，形成一个ApplicationMaster缓冲池，所以当用户提交作业时，可通过AMPoolServer直接将作业提交到这些ApplicationMaster上。这样做的好处是，避免了每个作业启动一个独立的ApplicationMaster。

- **预先启动Container**

ApplicationMaster缓冲池中的每个ApplicationMaster启动时可以预先启动若干个Container，以提高作业运行效率。对于用户来说，可以通过yarn.app.mapreduce.am.lazy.prealloc-container-count这个参数来设置ApplicationMaster预启动container的数目，这一点使用起来比较方便。

- **Container重用**

当每个任务执行结束后，ApplicationMaster并不会立即释放其占用的Container，而是将其分配给其他未运行的作业任务，从而可以使得资源(Container)得以重用，我们可以通过yarn.app.mapreduce.am.scheduler.reuse.enable这个参数设置是否启用Container重用功能，并且能够通过参数yarn.app.mapreduce.am.scheduler.reuse.max-attempts-per-container设置每个container重用次数。

两种引擎比较

为了使大家更好的理解TEZ和MR区别，下面给出两个例子加以说明。

- **MRR应用**

比如有一张关于学生的信息Student表，按照年级查询出每个年级的学生人数：

```
SELECT Grade, COUNT(*) as num
FROM Student GROUP BY Grade ORDER BY num;
```

如果采用MapReduce计算框架，Hive SQL会翻译成两个MR作业，第一个MR作业执行完会存储到HDFS系统中，当下一次执行MR作业时，再从HDFS中读取数据来执行，如果采用TEZ计算框架，则生成一个简单的DAG作业，这样可大大的降低读取磁盘所消耗的时间。

我们可以在环境上简单的操作，对比下两种执行引擎操作的结果：

数据量（条数）	执行 MR 引擎使用时间（S）	执行 TEZ 引擎使用时间(S)
50000	23.3	3.9
100000	39.8	6.1
700000	71.6	10.3
2000000	99.3	14.8
10000000	132.6	18.6
15000000	176.6	25.3

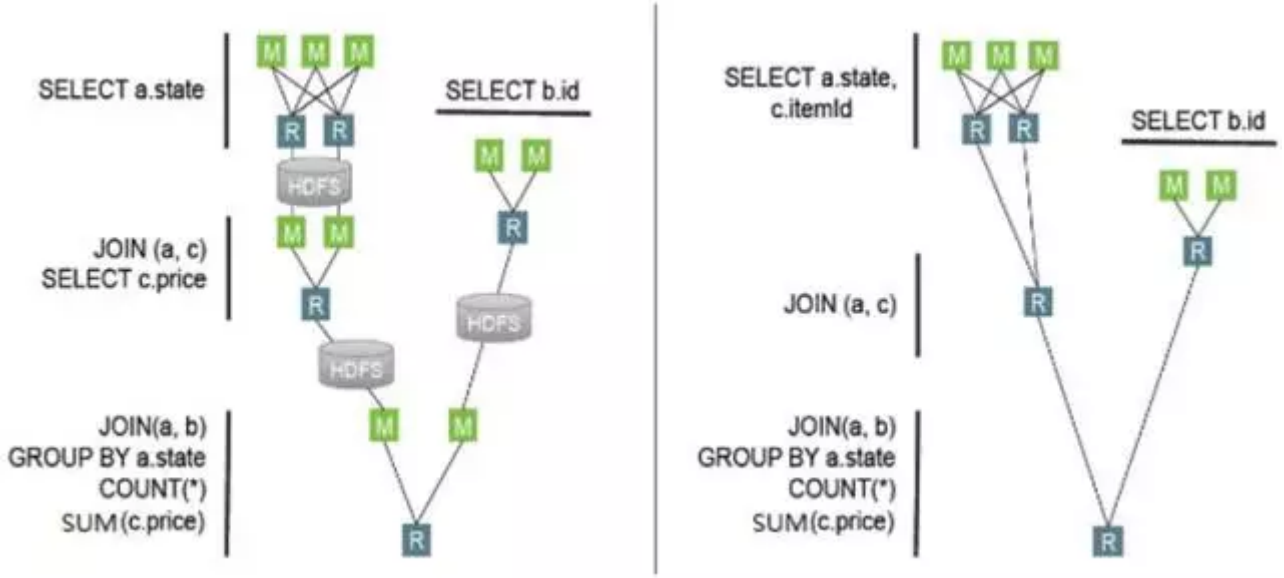
从上面对比表格可以看出，TEZ的执行效率明显高于MR，在大数据领域内，TEZ比MR至少快5倍，随着数据量进一步增大，TEZ的优势会变得越来越明显。

- **join应用**

下面来看一个稍微复杂的例子，比如有a，b，c三张关于商品信息的表，我们想执行以下语句得出每个商品产地的商品数量和总价钱。

```
SELECT a.state, COUNT(*), SUM(c.price)
FROM a
JOIN b ON(a.id = b.id)
JOIN c ON(a.itemId = c.itemId)
GROUP BY a.state
```

如果采用MapReduce计算框架，Hive SQL会翻译成四个MR作业，这样就会对磁盘进行多次的读写操作，这样启动多轮job的代价略有些大，不仅占用资源，更耗费大量的时间，而采用TEZ计算框架，就会生成一个简洁的DAG作业，算子跑完不退出，下轮继续使用上一轮的算子，这样大大减少磁盘IO操作，从而计算速度更快。下图是两种计算框架的流程图：



Hive-MR和 Hive-TEZ的计算流程比较

我们在环境上操作对比下两种执行引擎操作的结果：

数据量（条数）	执行MR引擎使用时间(S)	执行TEZ引擎使用时间(S)
50000	27.3	4.2
100000	50.8	6.9
700000	86.6	11.2
2000000	132.3	15.9
10000000	202.6	20.3
15000000	350.6	28.3

从上述表格可以看出，TEZ的执行效率比MR要高效很多，当数据量不断增大时，TEZ的优势会显得更明显。当然，如果对HQL语句再做修改，执行较复杂的语句，比如增加排序、求平均等操作，这样的计算量会增大，更能很好的看出TEZ的执行效率。

TEZ发展趋势

根据Apache TEZ官网的介绍，在以后的版本中将增加以下几个新特性：

1. 任务抢占，即可通过资源抢占的方式，让优先级更高的任务优先运行。
2. 任务执行断点检查。通过对任务执行过程记录断点，可在任务失败时从断点恢复运行，以避免任务重算。这个功能实现的难度不小，就从当前YARN的设计架构而言，只能做到已经完成的任务不重新计算，对于正在运行的任务需要重新开始计算。

3. Container重用。这个主要涉及两个方面：一方面，同一个应用程序的多个任务可重用一个Container，该功能是一个非常重要的Feature，社区里讨论也很热闹，感兴趣的可以进入社区看看。另一方面，不同应用程序的Container重用，即不同应用程序的多个任务可重用一个Container，内行人一看，就知道这个难度不小，也很让人期待。

总结

TEZ执行引擎的问世，可以帮助我们解决现有MR框架的一些不足，比如迭代计算和交互计算，除了Hive组件，Pig组件也将TEZ用到了自己的优化中。另外，TEZ是基于YARN的，所以可以与原有的MR共存，不会相互冲突，在实际的应用中，我们只需在hadoop-env.sh文件中配置TEZ的环境变量，并在mapred-site.xml设置执行作业的架构为yarn-tez，这样在YARN上运行的作业就会跑TEZ计算模式，所以原有的系统接入TEZ很便捷。当然，如果我们只想Hive使用TEZ，并不想对整个系统做修改，那我们也可以单独在Hive中做修改，也很简单，这样Hive可以在MR和TEZ之间自由切换而对原有的Hadoop MR任务没有影响，所以TEZ这款计算框架的耦合很低，让我们使用很容易和方便。

天下武功，唯快不攻，在互联网迅速发展的浪潮下，算法不断地被优化和改进，目的只有一个：不被现实淘汰。