学着站

记录成长心路沥程，记录技术成长轨迹的资料，技术文章的回收站

查看文

Apache Pig入门1 –介绍/基本架构/与Hive对比

2011年03月16日 星期三 下午 03:53

一、介绍

Google的工程师为了方便自己对MapReduce的实现搞了一个叫做Sawzall的工具，Google就放了几篇论文放在网上，但这玩意在代码上不开源在设计思想是开源的，在**前面一篇文章**中我也提到过Hadoop也推出了类似Sawzall的Pig语言，就是根据Google放出来的论文山寨的。

Pig是对处理超大型数据集的抽象层，在MapReduce中的框架中有map和reduce两个函数，如果你亲手弄一个MapReduce实现从编写代码，编译，部署，放在Hadoop上执行这个MapReduce程序还是耗费你一定的时间的，有了Pig这个东东以后不仅仅可以简化你对MapReduce的开发，而且还可以对不同的数据之间进行转换，例如：包含在连接内的一些转化在MapReduce中不太容易去实现。

Apache Pig的运行可以纯本地的，解压，敲个“**bin/pig -x local**”命令直接运行，非常简单，这就是传说中的local模式，但是人们往往不是这样使用，都是将Pig与hdfs/hadoop集群环境进行对接，我看说白了Apache的Pig最大的作用就是对mapreduce算法(框架)实现了一套shell脚本，类似我们通常熟悉的SQL语句，在Pig中称之为**Pig Latin**，在这套脚本中我们可以对加载出来的数据进行排序、过滤、求和、分组(group by)、关联(Joining)，Pig也可以由用户自定义一些函数对数据集进行操作，也就是传说中的UDF(user-defined functions)。

经过Pig Latin的转换后变成了一道MapReduce的作业，通过MapReduce多个线程，进程或者独立系统并行执行处理的结果集进行分类和归纳。Map() 和 Reduce() 两个函数会并行运行，即使不是在同一的系统的同一时刻也在同时运行一套任务，当所有的处理都完成之后，结果将被排序，格式化，并且保存到一个文件。Pig利用MapReduce将计算分成两个阶段，第一个阶段分解成为小块并且分布到每一个存储数据的节点上进行执行，对计算的压力进行分散，第二个阶段聚合第一个阶段执行的这些结果，这样可以达到非常高的吞吐量，通过不多的代码和工作量就能够驱动上千台机器并行计算，充分的利用计算机的资源，打消运行中的瓶颈。

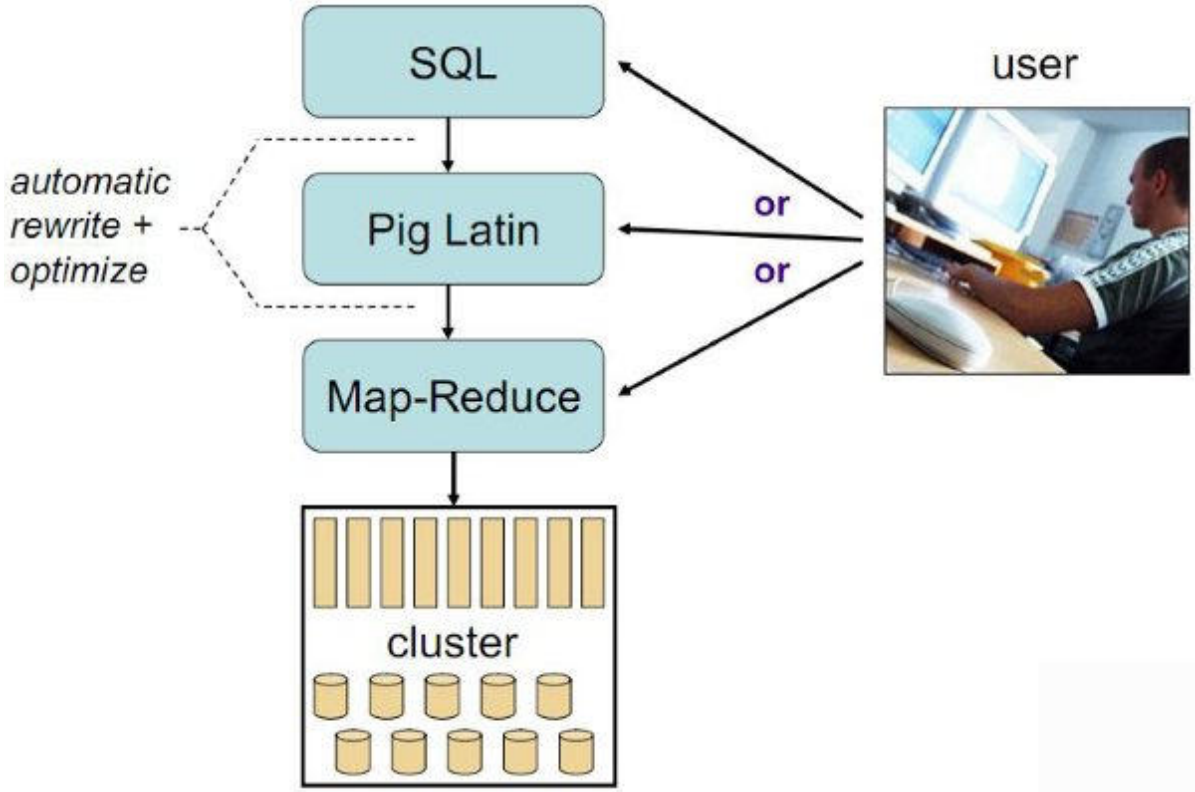
所以用Pig可以对TB级别海量的数据进行查询非常轻松，并且这些海量的数据都是非结构化的数据，**例如：**一堆文件可能是log4j输出日志存又放于跨越多个计算机的多个磁盘上，用来记录上千台在线服务器的健康状态日志，交易日至，IP访问记录，应用服务日志等等。我们通常需要统计或者抽取这些记录，或者查询异常记录，对这些记录形成一些报表，将数据转化为有价值的信息，这样的话查询会较为复杂，此时类似MySQL这样的产品就并非能满足我们的对速度、执行效率上的需求，而用Apache的Pig就可以帮助我们去实现这样的目标。

反之，你如果在做实验的时候，把MySQL中的100行数据转换成文本文件放在在pig中进行查询，会让你非常失望，为何这短短的100行数据查询的效率极低，呵呵，因为中间有一个生成MapReduce作业的过程，这是无法避免的开销，所以小量的数据查询是不适合pig做的，就好比用关二哥的大刀切青菜一样。另外，还可

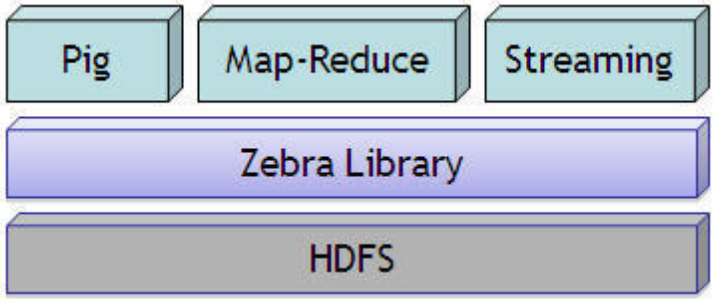
以利用Pig的API在Java环境中调用，对Apache的Pig以上内容请允许我这样片面的理解，谢谢。

二、基本架构

从整体上来看大量的数据聚集在HDFS系统上，通过输入类似SQL的脚本简化对MapReduce的操作，让这几行代码/脚本去驱动上千台机器进行并行计算。
如图所示：



Pig的实现有**5**个主要的部分构成：
如图所示：



1.Pig自己实现的一套框架对输入、输出的人机交互部分的实现，就是Pig Latin 。
2.Zebra是Pig与HDFS/Hadoop的中间层、Zebra是MapReduce作业编写的客户端，Zerbra用结构化的语言实现了对hadoop物理存储元数据的管理也是对Hadoop的数据抽象层，在Zebra中有2个核心的类 TableStore(写)/TableLoad(读)对Hadoop上的数据进行操作。
3.Pig中的Streaming主要分为4个组件: 1. Pig Latin 2. 逻辑层(Logical Layer) 3. 物理层(Physical Layer) 4. Streaming具体实现(Implementation)，Streaming会创建一个Map/Reduce作业，并把它发送给合适的集群，同时监视这个作业的在集群环境中的整个执行过程。
4.MapReduce在每台机器上进行分布式计算的框架(算法)。
5.HDFS最终存储数据的部分。

三、与Hive对比
请允许我很无聊的把飞机和火车拿来做比较，因为2者根本没有深入的可比性，虽然两者都是一种高速的交通工具，但是具体的作用范围是截然不同的，就像Hive和Pig都是Hadoop中的项目，并且Hive和pig有很多共同点，但Hive还似乎有点数据库的影子，而Pig基本就是一个对MapReduce实现的工具(脚本)。两者都拥有自己的表达语言，其目的是将MapReduce的实现进行简化，并且读写操作数据最终都是存储在HDFS分布式文件系统上。看起来Pig和Hive有些类似的地方，但也有些不同，来做一个简单的比较，先来看一张图：

Feature	Hive	Pig
Language	SQL-like	PigLatin
Schemas/Types	Yes (explicit)	Yes (implicit)
Partitions	Yes	No
Server	Optional (Thrift)	No
User Defined Functions (UDF)	Yes (Java)	Yes (Java)
Custom Serializer/Deserializer	Yes	Yes
DFS Direct Access	Yes (implicit)	Yes (explicit)
Join/Order/Sort	Yes	Yes
Shell	Yes	Yes
Streaming	Yes	Yes
Web Interface	Yes	No
JDBC/ODBC	Yes (limited)	No

查看大图请点击这里

再让我说几句废话：
Language
在Hive中可以执行 插入/删除 等操作，但是Pig中我没有发现有可以 插入 数据的方法，请允许我暂且认为这是最大的不同点吧。

Schemas
Hive中至少还有一个“表”的概念，但是Pig中我认为是基本没有表的概念，所谓的表建立在Pig Latin脚本中，对与Pig更不要提metadata了。

Partitions
Pig中没有表的概念，所以说到分区对于Pig来说基本免谈，如果跟Hive说“分区”(Partition)他还是能明白的。

Server
Hive可以依托于Thrift启动一个服务器，提供远程调用。 找了半天压根没有发现Pig有这样的功能，如果你有新发现可以告诉我，就好像有人开发了一个**Hive的REST**

Shell
在Pig 你可以执行一些个 ls 、cat 这样很经典、很cool的命令，但是在使用Hive的时候我压根就没有想过有这样的需求。

Web Interface
Hive有，Pig无

JDBC/ODBC
Pig无，Hive有

四、使用
1启动/运行
分为2台服务器,一台作为pig的服务器，一台作为hdfs的服务器。
首先需要在pig的服务器上进行配置，将pig的配置文件指向hdfs服务器，修改pig/conf目录下的
vim /work/pig/conf/pig.properties
添加以下内容：
fs.default.name=hdfs://192.168.1.201:9000/ #指向HDFS服务器
mapred.job.tracker=192.168.1.201:9001 #指向MR job服务器地址

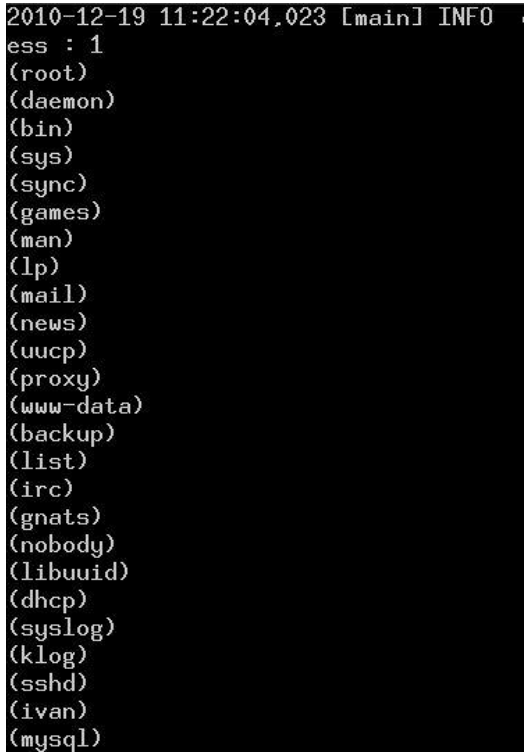
如果是第一次运行请在Hadoop的HDFS的服务器上创建root目录，并且将etc目录下的passwd文件放在HDFS的root目录下，请执行以下两条命令。
hadoop fs -mkdir /user/root
hadoop fs -put /etc/passwd /user/root/passwd

创建运行脚本，用vim命令在pig的服务器上创建javabloger_testscript.pig 文件，内容如下：
```
LoadFile = load 'passwd' using PigStorage(':');
Result = foreach LoadFile  generate $0 as id;
dump Result;
```

运行pig脚本，例如:pig javabloger_testscript.pig，执行状态如图所示：

```
root@serv2:~# pig javabloger_testscript.pig
10/12/19 08:28:56 INFO pig.Main: Logging error messages to: /root/pig_1292765336669.log
2010-12-19 08:28:56,940 [main] INFO  org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to hadoop fi
e system at: hdfs://192.168.1.201:9000
2010-12-19 08:28:57,820 [main] INFO  org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to map-reduc
 job tracker at: 192.168.1.201:9001
2010-12-19 08:28:58,701 [main] INFO  org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - (Name: Store(hdfs://serv
1:9000/tmp/temp190/629036/tmp/37292790:org.apache.pig.builtin.BinStorage) - 1-19 Operator Key: 1-19)
2010-12-19 08:28:58,745 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MultiQueryOptimizer - MR pl
n size before optimization: 1
2010-12-19 08:28:58,746 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MultiQueryOptimizer - MR pl
n size after optimization: 1
2010-12-19 08:28:58,864 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.JobControlCompiler - mapred
 job.reduce.markreset.buffer.percent is not set, set to default 0.3
2010-12-19 08:29:00,383 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.JobControlCompiler - Setting
 up single store job
2010-12-19 08:29:00,437 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - 1 map-r
duce job(s) waiting for submission.
2010-12-19 08:29:00,470 [Thread-11] WARN  org.apache.hadoop.mapred.JobClient - Use GenericOptionsParser for parsing the argume
nts. Applications should implement Tool for the same.
2010-12-19 08:29:00,961 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - 0% compl
ete
2010-12-19 08:29:01,029 [Thread-11] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to proces
 : 1
2010-12-19 08:29:01,030 [Thread-11] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths t
 process : 1
2010-12-19 08:29:02,248 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - HadoopJo
bId: job_201012190611_0006
2010-12-19 08:29:02,249 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - More inf
ormation at: http://192.168.1.201:50030/jobdetails.jsp?jobid=job_201012190611_0006
2010-12-19 08:29:13,971 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - 50% comp
```

执行结果：



2.java 代码 运行并且打印运行结果

```
import java.io.IOException;
import java.util.Iterator;

import org.apache.pig.PigServer;
import org.apache.pig.data.Tuple;

public class  LocalPig {
    public static void main(String[] args) {
        try {
            PigServer pigServer = new PigServer("local");
            runIdQuery(pigServer, "passwd");
        } catch (Exception e) {
        }
    }

    public static void runIdQuery(PigServer pigServer, String inputFile)  throws IOException {
        pigServer.registerQuery("LoadFile = load '" + inputFile+ "' using PigStorage(':');");
```

```
        pigServer.registerQuery("Result = foreach A generate $0 as id;");
        Iterator<Tuple> result = pigServer.openIterator("Result ");
        while (result.hasNext()) {
              Tuple t = result.next();
              System.out.println(t);
           }
//         pigServer.store("B", "output");

    }
}
```

类别：并行分布式存储与计算-云计算方向p2p | 分享 | 添加到搜藏 | 分享到i贴吧 | 浏览(624) | 评论 (0)
上一篇：epoll_create, epoll_ctl和epoll... 下一篇：ZeroMQ 的模式

已有2人分享了这篇文章：


baixuejiyi1111 Ta的分享 ChinaGuowei Ta的分享

最近读者：








lxw_cq uwon tt86_123456 joker_000 lovebirdsx icymary if1i 16668215

网友评论：

发表评论：