随着分布式计算技术的推广，越来越多的大数据计算任务迁移到hadoop平台上进行，模型类的hadoop应用也越来越多。经过这一段时间在hadoop上的[**测试**](http://www.51testing.com/?uid-300987-action-viewspace-itemid-818957)项目，在此简单分享下hadoop上项目测试的经验。本文主要介绍项目测试过程中一些常见的现象以及问题的说明和一些常见的测试方法

**一、测试常见问题**

**1、reduce输出文件，上传文件，下载文件等操作的目的文件的删除。**

　　【现象】程序第一次运行还是成功的，数据和程序都没有修改，同样的命令，运行第二次的时候，怎么就失败了呢？

　　【问题说明】由于hdfs文件系统没有覆盖写的特性。对于reduce的输出，本地上传文件到hdfs上，下载hdfs文件到本地等操作，当目的文件已经存在，这些操作均会失败。

　　【测试方法】对于具有上述操作的程序，一定要在程序运行前把对应的目的文件删除，特别是具有多轮迭代程序的临时目录需要清楚。

**2、HADOOP\_HOME环境变量的设置**

　　【现象】在自己独自使用的测试机上，利用hadoop命令新建了一个目录，并利用hadoop dfs –ls path命令能够查看到该目录存在，换到一个公用的机器上就找不到该目录？

　　【问题说明】同一台测试机器可能会有多个hadoop客户端连接到多个不同的hadoop平台。而当在[**shell**](http://www.51testing.com/?uid-300987-action-viewspace-itemid-818957)命令行直接输入hadoop命令时，系统默认是使用HADOOP\_HOME下的hadoop客户端。当HADOOP\_HOME环境变量被别的用户修改后，就会连接到别的hadoop平台，当然就找不到所要的目录：）。

　　【测试方法】当在程序中使用hadoop命令的时候，一定要指定hadoop命令的路径，特别在rd提供的程序中，hadoop命令的路径一定要可配。

**3、Hadoop上程序输入目录的标准化**

　　【现象】程序的输入数据完全没问题：文件路径和格式均正确，为什么结果文件都为空呢？

　 　【问题说明】对于多路输入（即多种格式的输入文件），rd进行设计程序的时候，常常会根据路径名来进行文件类型的判断，进而进行不同的操作。此时，当外 界输入的路径名没有标准化（比如存在：./a/，/a//b，/a/./b），map阶段通过比较传递的路径参数和map环境变量获取的当前处理文件路径 来判断当前处理的文件块来自哪个目录，结果会判断当前处理的文件不来自任何输入目录，从而无法获取输入数据的类型。（当时针对这个问题排查很久，曾一度认 为是hdfs的问题，最后通过查看程序源代码才发现该问题）

　　【测试方法】出现该情况，首先查看该任务的监控页面： Map input records的输入是否为0，若是为0，则需检查输入数据地址正确性。Map output records是否为0. Map output records代表map的输出，若是为0，那么就是数据在map阶段就被过滤掉，需要检查输入数据的格式正确性。然后查看Reduce input records是否为0，若rduece的输入为0，那输出肯定就为0了。

**4、Hadoop副本任务对程序结果的影响**

　　【现象】在reduce中生成的本地文件需要上传到hdfs上。在上传之前，为了避免目的文件存在而导致上传失败，需要先进行删除操作，然后再上传。所有的reduce任务都正常结束，可是结果文件偶尔会有缺失。而且是不能稳定复现。

　 　【问题说明】hadoop运行map，red任务的时候，为了防止单个task运行缓慢，拖累整个任务的完成时间，会对一些task启用备奋task， 即多个task运行同一份数据，当一个task运行完成后，系统自动kill掉备份task。这样可能导致备份task被kill前，正确的文件上传后， 被备份任务删除，导致最后结果文件的缺失。而该现象还不是稳定复现。

　　【测试方法】对hdfs上的同一个文件，目录进行操作时，一定要注 意并行操作的干扰。特别当在reduce中进行hdfs操作的时候，一定要考虑到副本的影响（该问题比较隐蔽）。解决方案是：1，禁止平台生成副本任务 （可以配置启动参数达到目的）。2，在一个统一的单机进行此类操作。比如，现在单机处理好环境，然后启动mapred任务。

**5、Reduce数据分桶不均**

　　【现象】通过查看任务的监控页面发现有的reduce运行时间很短，而有的reduce运行时间很长。

　　【问题说明】既然利用hadoop的任务，处理的数据一定是大数据量的。简单的hash映射分桶可能导致分桶不均，从而多个reduce处理的数据量差别很大。

　　【测试方法】当前hadoop任务处理的数据很多都上T，若是在处理这么大规模的数据，分桶不均，可能导致单个节点处理数据过大，导致性能降低，甚至可能导致内存超过阈值被平台kill。因此在测试之前，一定要弄清楚，分桶的key和分桶函数是否会造成分桶不均。

**6、worker资源分配的指定**

　　【现象】每个task运行时间很短，集群资源很充足，可是任务运行时间却很长。

　　【问题说明】当处理的数据量很大时，任务会被分成很多的task，而当任务启动时，集群默认分配的worker会比较少，导致即使集群资源空闲，运行该任务的worker数仍然很少，运行结束时间延长。

　　【测试方法】若是处理的数据很大，在任务启动的时候，一定要指定资源参数，否则按照系统默认值，分配的work会很少（在hy集群为50）。对于大数据量，该限制会大大降低性能。任务启动的时候，可以通过监控页面，查看该任务运行的worker数。

**7、单个worker内存限制**

　　【现象】小数据量，测试通过，可是运行大数据，任务总是被平台kill。

　　【问题说明】现在hadoop平台对每个task运行时的内存进行了限制，默认是800M，当程序运行内存超过800M，平台自动会kill该任务

　　【测试方法】针对该点测试有2种方法：1，在集群运行大数据量，被平台kill后，查看[**日志**](http://www.51testing.com/?uid-300987-action-viewspace-itemid-818957)，确认是因为内存超出被平台kill。2，在本地运行mapred程序，查看程序内存占用。若是在800M左右，上线就会有很大风险。

**8、MPI程序对hadoop上文件目录的操作**

　　【现象】在mpi节点上对同一目录上的文件进行操作，偶尔会失败。

　　【问题说明】该问题同Hadoop副本任务对程序结果的影响 原因一样，都多个节点对hadoop上同一个文件的操作。只是一个是hadoop上，一个是mpi上。

　　【测试方法】多个地方对hdfs上的同一个文件或者目录进行操作。特别是同一个模块即在hadoop，又在mpi集群上运行。不要在每个mpi节点同时对同一个hadoop目录或者文件进行修改操作。

**9、对于map reduce的运行参数的设置**

　　【现象】程序在本地就能执行成功，可是在hadoop上却无法运行

　　【问题说明】有时候，map reduce的运行参数比较长，为了阅读方便，rd可能会对程序参数进行折行，添加tab键排版，这样反倒使hadoop上解析命令失败。

　　【测试方法】对于map reduce的运行参数比较长的情况，可以督促rd用shell变量进行设置。然后在hadoop程序启动参数用shell变量进行替换。即阅读方便，又不会出错。

**10、Hadoop程序的结果使用的bistreaming的二进制文件结果**

　　【现象】程序结果文件是二进制格式，可是下载到本地，按照详细设计中描述的格式，解析格式总是错误。

　　【问题说明】当前流模式可以用streaming和bistreaming，当使用bistreaming时，生成的结果文件是hadoop的 sequence file文件格式，文件格式中含有key length和value length信息。在hadoop上使用该数据时，这个格式对用户是透明的。下载到本地后，就不能直接使用了。

　　【测试方法】当任务的输出outputformat=SequenceFileAsBinaryOutputFormat 时，可以使用hadoop dfs -copySeqFileToLocal –ignoreLen命令，把二进制数据下载到本地，并去掉长度信息，就和文档中写的格式一致了。

**11、Hadoop对于输入文件的切分**

　　【现象】输入文件是基于session的query日志行，session之间空行分割。当设置一个map的时候，程序结果正确，当设置多个map时，运行结果错误。

　　【问题说明】hadoop对于输入文件会以行为最小单元切分，因此，当输入以空行为分割，有二次数据格式的时候，hadoop无法保证不把一个session切到两个map task中。这样就把一个session拆分为多个session了。

　　【测试方法】当程序实现逻辑依赖于比行粒度更大的单元时，需要设置map的切分大小比单个输入文件大，否则就回出现把输入文件切分成多个map输入，导致切断更大的输入单元。

**二、常用测试方法**

**1、跨集群拷贝或者大数据量集群内拷贝**

　　在测试过程中，可能需要从摸个集群或者某个目录拷贝大量测试数据，若是先把数据拷贝到本地，在上传到目的集群，将会非常的耗时间，这时，可以考虑用distcp命令。

　　DistCp（分布式拷贝）是用于大规模集群内部和集群之间拷贝的工具。它使用Map/Reduce实现文件分发，错误处理和恢复，以及报告生成。 它把文件和目录的列表作为map任务的输入，每个任务会完成源列表中部分文件的拷贝。

　　hadoop distcp hdfs：//nn1：8020/foo/bar hdfs：//nn2：8020/bar/foo这条命令会把nn1集群的/foo/bar目录下的所有文件或目录名展开并存储到一个临时文件中，这些文 件内容的拷贝[**工作**](http://www.51testing.com/?uid-300987-action-viewspace-itemid-818957)被分配给多个map任务， 然后每个TaskTracker分别执行从nn1到nn2的拷贝操作。注意DistCp使用绝对路径进行操作。

　　由于distcp无法指定两个用户名和密码，因此，拷贝的源和目的集群的用户名和密码必须一致，且该用户名密码在源集群具有读权限，在目的集群具有写权限。

**2、单机模拟分布式测试功能点**

　　在测试部分功能点时或者内存不超过800M的性能时，可能会考虑到先用单机模拟分布式上运行进行测试：

　　cat input | mapper | sort | reducer > output

　　在用单机模拟分布式测试时，有以下注意点：

　　1）Streaming的输入是按行划分的文本，可以使用cat input即可；但是BiStreaming是“<key-length><key><value- length><value>”的格式，所以在输入前要进行一定的处理；常用方法是：

　　cat input | ./ reader |./mapper |./ reducer >output

　　reader程序负责将文件转化为mapper程序能够识别的keyLength， key， valueLength， value的二进制格式。当输入已经是sequencefile格式时，也不需要reader。

　　2）当Mapper或Reducer中使用到hadoop的环境变量时，单机模拟时，需要先修改这些环境变量或者在运行时设置这些环境变量的值。

**3、分布式程序和单机程序结果的对比**

　　在校验分布式程序结果时，我们常常使用的方法是，实现一个单机版的程序，然后diff单机版和分布式版本的运行结果。

　　由于hadoop对于输入文件的切分，以及map后的reduce分桶。在和单机版做结果对比时，需要考虑到输入行乱序对于结果的影响。若输入行的乱序对结果的正确性没有影响。那在做分布式的结果和本地单机模拟的结果diff时，一定要先sort，然后在diff。

**4、总控脚本的测试**

　　分布式程序虽然以map-reduce程序为主体，但每一轮map-reduce任务都是在单机用脚本进行任务提交和启动，大多数项目都包含多轮map-reduce 任务。因此，各轮任务之间的调度协调和项目的系统运行就需要总控脚本来完成。

　　在测试总控脚本的时候，用-x参数运行，并把运行log重定向到输出文件。运行介绍后，即使结果正确，也需要看看运行脚本的log，很可能发现一些意想不到的问题。