## 纠删码测试（目录测试）

因纠删码全局使用至少需要5 rack（机架）的规模，在现有5台机器的情况下，舍弃机架感知构建每组一台机器的机架测试意义不大，而且会对现有集群造成很大变动，因此只针对目录设置纠删码策略进行了测试。

公司目前已经有了一定量的数据，目前集群默认的副本因子为1，问过前辈之后不希望在现有集群上有较大变动避免对数据造成影响，因此对副本因子为1的情况下进行了较多测试，副本因子为3的情况下只测试了将某目录的副本因子改为3，单针对集群的读性能进行了测试。以下的测试结果可能涉及到网络传输的原因，可能存在少许偏差，但连续的测量网络情况应该变化不大，数据也具有一定的说明性。

### 副本因子：1 本地文件读取写入

写：时间较短的记录了三次的平均值，耗时较多的记录两次的平均值

读：只记录了第一次读数据所需的时间，因为连续多次传输同一文件后续传输时间会缩少，以dfs写单1.94G文件为例：

24.791 21.142 18.113 12.974 10.215 9.462。但每次传输进行至少三次测试，以防止意外情况发生。

注：第一组数据损失节点后文件读操作可能由于多次测试 传输速度已不具备重要参考性价值 可视为连续两次

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 文件大小 | Dfs写 | Dfs读 | dfs损失2节点写 | dfs损失2节点读 | 纠删码写 | 纠删码读 | 纠删码损失2节点写 | 纠删码损失2节点读 |
| 单文件1.94G | real 0m21.425s  user 0m17.597s  sys 0m6.079s | real 0m24.791s  user 0m9.323s  sys 0m5.265s | real 0m21.271s  user 0m16.531s  sys 0m5.631s | real 0m9.432s  user 0m8.048s  sys 0m4.174s | real 0m18.017s  user 0m31.864s  sys 0m11.234s | real 0m13.794s  user 0m11.304s  sys 0m5.849s | real 0m18.271s  user 0m31.723s  sys 0m10.867s | real 0m10.077s  user 0m11.365s  sys 0m5.556s |
| 346M  13586个文件 | real 5m48.630s  user 1m27.757s  sys 0m38.097s | real 0m53.416s  user 0m53.017s  sys 0m15.181s | real 5m31.911s  user 1m25.608s  sys 0m38.280s | real 0m55.079s  user 0m49.616s  sys 0m15.847s | real 5m40.063s  user 2m53.729s  sys 1m49.387s | real 0m53.950s  user 1m2.576s  sys 0m15.331s | real 5m49.941s  user 2m52.588s  sys 1m49.438s | real 0m55.793s  user 0m57.474s  sys 0m15.661s |
| 742M  6587个文件 | real 3m19.506s  user 1m2.190s  sys 0m22.533s | real 0m29.511s  user 0m39.363s  sys 0m8.602s | real 3m20.236s  user 1m2.033s  sys 0m22.638s | real 0m29.282s  user 0m40.716s  sys 0m8.670s | real 3m3.504s  user 2m4.118s  sys 0m59.383s | real 0m32.622s  user 0m43.356s  sys 0m9.457s | real 3m2.489s  user 1m59.801s  sys 0m58.777s | real 0m33.720s  user 0m43.445s  sys 0m8.866s |

上述数据可看出：

1. 损失节点对hdfs读影响很小，几乎可以忽略。
2. 纠删码写实际时间稍微快于hdfs副本写，但小文件较多且损失节点的情况下，较hdfs写稍慢。纠删码读效率实际时间稍慢于hdfs读。
3. user时间为单线程处理需要花费的时间，纠删码处理时间较直接读写处理时间多的多，sys为系统耗时，纠删码处理时间也要比直接读写处理时间要多。但多线程情况下，从real显示的实际耗时时间结果来看，两者处理时间几乎无差，在小文件较少的情况下纠删码策略处理效率还要比直接读写策略耗时要短，还是可以考虑使用的。
4. 期间也测试过不同datanode上的数据传输，虽然存在少许差异，大致结论无较大差别。

官方文档：

目录级EC策略仅影响目录中创建的新文件。创建文件后，可以查询其擦除编码策略，但不能更改。如果将擦除编码文件重命名为具有其他EC策略的目录，则该文件将保留其现有的EC策略。将文件转换为其他EC策略需要重写其数据。通过复制文件（例如通过distcp）而不是重命名来做到这一点。

因此在原来已有的数据规模下，集群恢复时间几乎不受影响：

集群恢复时间1m13s（三次平均 是指重启节点之后到cdh上面显示运行情况良好 100s左右集群检测到有节点挂掉）

### 副本因子：3 本地文件读取写入

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 文件大小 | Dfs读 | dfs损失2节点读 | 纠删码读 | 纠删码损失2节点读 |
| 单文件1.94G | real 0m24.054s  user 0m9.274s  sys 0m4.922s | real 0m24.119s  user 0m9.586s  sys 0m5.059s | real 0m11.096s  user 0m11.237s  sys 0m5.943s | real 0m10.683s  user 0m11.308s  sys 0m5.815s |
| 346M  13586个文件 | real 1m8.111s  user 0m54.674s  sys 0m18.562s | real 0m55.678s  user 0m50.049s  sys 0m16.984s | real 1m0.505s  user 1m4.915s  sys 0m18.605s | real 0m58.247s  user 1m3.307s  sys 0m18.420s |
| 742M  6587个文件 | real 0m41.089s  user 0m40.718s  sys 0m10.341s | real 0m33.624s  user 0m41.405s  sys 0m10.178s | real 0m39.393s  user 0m46.917s  sys 0m11.179s | real 0m36.497s  user 0m45.314s  sys 0m10.761s |

从数据可以看出，单文件的情况下，纠删码读的方案有着明显的速度优势，但随着文件数目的增加，速度的优势逐渐减弱，在损失的节点时速率要略低于直接的读写。

### 副本因子：1 kudu数据进imapla存储(hdfs)

将impala数据目录设置纠删码，数据会变得无法查询：

ImpalaRuntimeException: Scanning of HDFS erasure-coded file is not supported.

测试用数据文件row=207878 files=169 size=31.85GB

Ec目录写入数据（三次测试，扫描+writer时间之和即为总时间）

avg max sum

F00:HDFS WRITER 3 6m37s 6m43s 6m58s

00:SCAN HDFS 3 13s201ms 15s008ms

F00:HDFS WRITER 3 7m37s 7m58s 8m27s

00:SCAN HDFS 3 19s542ms 29s373ms

F00:HDFS WRITER 3 7m56s 8m7s 8m27s

00:SCAN HDFS 3 16s883ms 20s537ms

未指定ec目录写入数据（同上）

avg max sum

F00:HDFS WRITER 3 4m38s 4m42s 5m52s

00:SCAN HDFS 3 22s390ms 32s713ms

F00:HDFS WRITER 3 3m46s 4m37s 5m50s

00:SCAN HDFS 3 42s069ms 1m13s

F00:HDFS WRITER 3 3m38s 4m28s 5m55s

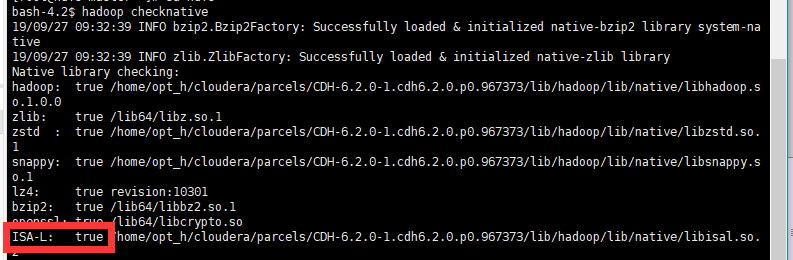
00:SCAN HDFS 3 54s327ms 1m27s

从上述数据可以看出

30G的数据量情况下,ec目录写入数据的效率远远不如未指定ec目录的文件写入时间，且对hdfs目录扫描时间也是未指定的两倍有余。

附：

对于Intel isa-l智能存储加速库，hadoop提供了源生的支持。



# 小结:

考虑到impala ec目录下的文件无法读取且从kudu方向过来的数据写入效率较低，不建议将impala数据文件置于纠删码文件目录下，而且数据查询之前还需要对文件的目录进行策略更改。但是就一定数据量的数据存储来说，考虑到纠删码的储存优势及少许的速度优势，还是可以采用的。

不过现有的纠删码资料不尽完全，出现的问题可能难以解决。例如<https://www.cnblogs.com/PigeonNoir/p/9226759.html> 这里可能是社区和论坛没有的bug。

测试结果如上，还希望可以综合考虑。