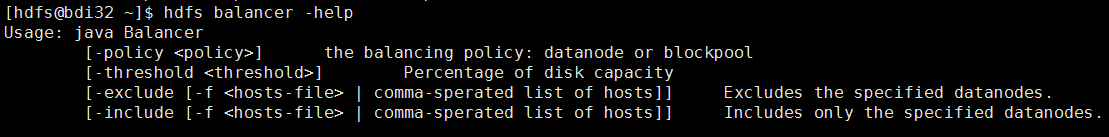
HDFS Balancer源码分析

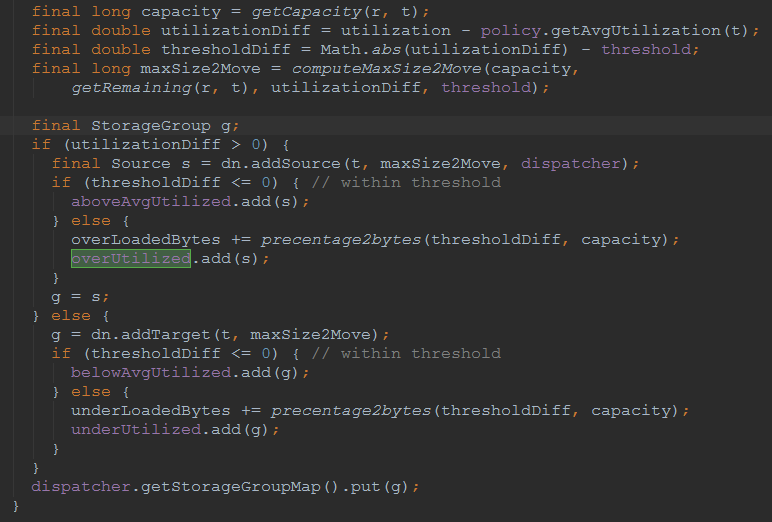
命令如下：



Balancer原理

计算磁盘使用率

avgUtilization 🡨 totalUsedSpaces/totalCapacity



Utilization:

[avgUtilization+threshold, ∞] => overUtilizedDatanodes

[avgUtilization,avgUtilization+threshold] => aboveAvgUtilizationDatanodes

[avgUtilization-threshold,avgUtilization] => blowAvgUtilizedDatanodes

[-∞, avgUtilization-threshold] => underUtilizedDatanodes

Threshold的默认值为10

步骤1：Datanode的utilization处于

[avgUtilization-threshold,avgUtilization+threshold] ：

blowAvgUtilizedDatanodes和aboveAvgUtilizationDatanodes

认为datanode已经是平衡状态

步骤2：

Sources [overUtilizedDatanodes,aboveAvgUtilizedDatanodes]

Target [blowAvgUtilizedDatanodes,underUtilizedDatanodes]

将Source对应的DataNode中的数据移动到Target对应的Datanode中

步骤3：

选择负载节点：优先级逐渐递减

同一机架上的过载节点

同一机架上高于平均使用率的接待你

其他机架上的过载节点

其他机架上的高于平均使用率的节点

选择目标节点：

同一机架上的负载节点

同一机架上的低于平均使用率的节点

其他机架上的负载节点

其他机架上的低于平均使用率的节点

步骤4：

1）匹配over和underUtilization

从over List及under List中选择节点构成[overUtilized -> underUtilized]

2）匹配over和blowUtilization

从over List及blowUtilization中选择节点构成[overUtilized -> blowUtilization]

3) under区有节点 //没有和above节点匹配

从avoveList及under List中选择节点，构成[aboveUtilized -> underUtilization]

4) 然后异构rack，重复1-3

5）计算source可移动的数据量

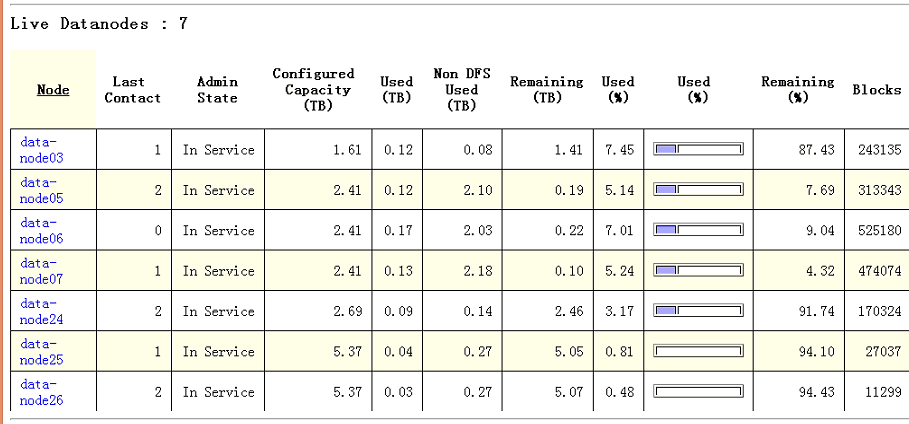
步骤5：

Dispatcher#dispatch //dispatchExecutor

将source -> target

moveExecutor#execute(PendingMove.dispatch)

Example:



Avg= 4.18，threshold=5

|  |  |  |
| --- | --- | --- |
| 节点号 | utilization | 所属区域 |
| Node03 | 7.45 |  |
| Node05 | 5.14 |  |
| Node06 | 7.01 |  |
| Node07 | 5.24 |  |
| Node24 | 3.17 |  |
| Node25 | 0.81 |  |
| Node26 | 0.48 |  |

Avg+threshold : 9.18

Avg : 4.18

Avg-threshold: -0.72

[-0.72,9.18] 所有节点的utilization都处于这个区域， 处于均衡状态

解决方案：

1）调小threshold的值，例如 -threshold 调整为1

2）设置参数 -include

本地新建文件:include.hosts //该文件用于设置block的移动目的主机，以","为分割符，例如:

host1,host2,host3,......

执行命令：

hdfs balancer -include -f include.hosts