实验报告——HDFS创建大批量小文件

朱政烨 1700017760
薛飞跃 1700017831

2020 年 3 月 22 日

1 实验步骤

1.1 启动Hadoop（Master上执行）

start-dfs.sh
start-yarn.sh
mr-jobhistory-daemon.sh start historyserver

1.2 编程实现在HDFS中创建大批量小文件（100个）

HDFSWriteRead.java

```java
import org.apache.hadoop.conf.Configuration;

public class HDFSWriteRead {
    public static void main(String[] args){
        try{
            Configuration conf = new Configuration();
            conf.set("fs.defaultFS", "hdfs://Master:9000");
            conf.set("fs.hdfs.impl", "org.apache.hadoop.hdfs.DistributedFileSystem");
            FileSystem fs = FileSystem.get(conf);
            for(int i = 0; i < 100; i++){
                String fileName = "test/"+i;
                Path file = new Path(fileName);
                if(fs.exists(file)){
                    System.out.println("File exists");
                }else{
                    createFile(fs,file);
                }
            }
            fs.close();
        }catch (Exception e){
            e.printStackTrace();
        }
    }
    public static void createFile(FileSystem fs, Path file) throws IOException{
        byte[] buff = "Hello World\n".getBytes();
        FSDataOutputStream os =fs.create(file);
        os.write(buff,0,buff.length);
        System.out.println("Create:"+file.getName());
        os.close();
    }
}
```

图 1: java代码

1.3 使用stat命令查看文件备份数和blocksize

```
xuefeiyue@xuefeiyue-VirtualBox:/usr/local/hadoop/hadoop-2.7.7/myapp$ hdfs dfs -stat %r test/1
1
```

```
xuefeiyue@xuefeiyue-VirtualBox:/usr/local/hadoop/hadoop-2.7.7/myapp$ hdfs dfs -stat %o test/1
134217728
```

1.4 使用fsck命令查看文件在block中真实占据的大小

```
zzy@Master:~$ hdfs fsck test/0
Connecting to namenode via http://Master:50070/fsck?ugi=zzy&path=%2Fuser%2Fzzy%2Ftest%2F0
FSCK started by zzy (auth:SIMPLE) from /192.168.8.100 for path /user/zzy/test/0 at Sat Mar 21 12
:00:31 CST 2020
.
/user/zzy/test/0:  Under replicated BP-1262915069-192.168.8.100-1584086899106:blk_1073741958_113
4. Target Replicas is 3 but found 2 replica(s).
Status: HEALTHY
 Total size:    12 B
 Total dirs:    0
 Total files:   1
 Total symlinks:                0
 Total blocks (validated):      1 (avg. block size 12 B)
 Minimally replicated blocks:   1 (100.0 %)
 Over-replicated blocks:        0 (0.0 %)
 Under-replicated blocks:       1 (100.0 %)
 Mis-replicated blocks:         0 (0.0 %)
 Default replication factor:    2
 Average block replication:     2.0
 Corrupt blocks:                0
 Missing replicas:              1 (33.333332 %)
 Number of data-nodes:          2
 Number of racks:               1
FSCK ended at Sat Mar 21 12:00:31 CST 2020 in 81 milliseconds


The filesystem under path '/user/zzy/test/0' is HEALTHY
```

```
Status: HEALTHY
 Total size:    1200 B
 Total dirs:    1
 Total files:   100
 Total symlinks:                0
 Total blocks (validated):      100 (avg. block size 12 B)
 Minimally replicated blocks:   100 (100.0 %)
 Over-replicated blocks:        0 (0.0 %)
 Under-replicated blocks:       100 (100.0 %)
 Mis-replicated blocks:         0 (0.0 %)
 Default replication factor:    2
 Average block replication:     2.0
 Corrupt blocks:                0
 Missing replicas:              100 (33.333332 %)
 Number of data-nodes:          2
 Number of racks:               1
FSCK ended at Sat Mar 21 13:21:15 CST 2020 in 121 milliseconds


The filesystem under path '/user/zzy/test' is HEALTHY
```

1.5 查看文件实际占据的磁盘的大小

1.5.1 文件不存在时：

Configured Capacity:	40.05 GB
DFS Used:	260 KB (0%)
Non DFS Used:	7.79 GB
DFS Remaining:	30.2 GB (75.41%)
Block Pool Used:	260 KB (0%)
DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 0.00% / 0.00%

1.5.2 100个文件时：

Configured Capacity:	40.05 GB
DFS Used:	1.04 MB (0%)
Non DFS Used:	7.79 GB
DFS Remaining:	30.2 GB (75.4%)
Block Pool Used:	1.04 MB (0%)

1.5.3 100个文件时：

1058 files and directories, 1023 blocks = 2081 total filesystem object(s).

Heap Memory used 40.37 MB of 63.34 MB Heap Memory. Max Heap Memory is 966.69 MB.

Non Heap Memory used 59.16 MB of 60.31 MB Commited Non Heap Memory. Max Non Heap Memory is -1 B.

Configured Capacity:	40.05 GB
DFS Used:	8.15 MB (0.02%)
Non DFS Used:	7.8 GB
DFS Remaining:	30.18 GB (75.36%)
Block Pool Used:	8.15 MB (0.02%)
DataNodes usages% (Min/Median/Max/stdDev):	0.02% / 0.02% / 0.02% / 0.00%

2 实验结果分析

1. 每个小文件只有12B大小，block的大小为128MB，远远大于小文件的大小，但每个文件都分配一个各自的block。100个小文件就有100个block。
2. 通过DFS Used计算，每个小文件实际占用近8KB的空间，远远高于文件本身的大小12B。
3. 缺点：
   ①在HDFS中，数据和元数据是分离的。数据文件被拆分为块文件，这些块文件在群集中的DataNode上存储和复制。文件系统命名空间树和关联的元数据存储在NameNode上。小文件过多，会过多占用NameNode的内存，并浪费block文件。在此次实践中，12B大小的文件占用了8KB的空间。
   ②文件过小，寻道时间大于数据读写时间，这不符合HDFS的设计。HDFS是为了数据的高吞吐量设计的，而寻道时间长，降低了数据的传输速度。
4. 改进措施：
   传输数据时，每隔一段时间对小文件进行合并。
   Hadoop的文件方案：①HAR文件系统，在HDFS上构建一个分层文件系统，相当于将小文件合并成一个HAR文件。②Sequence Files方案，将小文件以数据流的形式合并到SequenceFile中

3 实验心得

1. 通过本次实习，学会了配置虚拟机以及Hadoop环境
2. HDFS处理大量小文件比较低效，更适合处理大文件