Download Manager

Le Nhu Chu Hiep

Purpose

This is simple tool to download data from multiple node

Philosophy

- Simple in idea (RMI oriented)
- No complex structure
- Easy to add new data node to system
- Client heavy process tools

Design

Server

- NameNode: control DataNode + clean dead DataNode (time heartbeat: heartbeat - 2s and after 3 times ~ 6s –> node is dead)
- DataNode: receive request (upload/download chunk/full file) and update heartbeat to NameNode every 2s

Client

- Get list of DataNode from NameNode
- Upload case: open multiple thread do same thing - upload file to all DataNode
- Download case: open multiple thread for downloading each chunk file to temp dir, then merge together into single file

Protocol:

From DataNode point of view:

```
##### upload protocol #####
r: action \n
r: filename \n
r: size \n

-- repeat --
s: 200 \n
r: {data - chunk: 5000}
------------

s: 201 \n

##### download protocol #####
r: action \n
r: filename \n
s: 200 \n
```

```
s: filesize \n
r: [100 \n -- continues process | 200 \n -- ok now stop process]
IF(200) s: 200 \n
r: offset \n
r: size \n
s: 200 \n

-- repeat --
r: 200 \n
s: {data - chunk: 5000}
------------
```

Note:

For Client side:

- Upload: client open multi-thread to upload at same time to all nodes
- Download: since each DataNode could have different data for single filename –> get first getted DataNode as truth ground to compare another DataNode

QuickStart

1. Clone project and go to root base of project:

```
git clone https://github.com/larycoder/DownloadManagerTool.git
cd DownloadManager
```

2. Build project by command:

```
make -j
```

3. Go to build dir:

```
cd build
```

4. Run NameNode by command:

```
rmiregitry
java -Djava.rmi.server.hostname=<NameNode Host> NNServer <NameNode Host> <NameNode Naming>
```

5. Run DataNode by command:

```
java DNServer <DataNode Host> <DataNode Port> <NameNode Host> <NameNode Naming>
```

6. You can run client by command:

```
java Client <NameNode Host> <NameNode Naming> [store | download] <PathToFile>
```

Performance Evaluation

Local Test

The measurement count on single machine (my machine)

1. Machine info:

- CPU: Intel i5-8265U (4 cores - 8 threads)
- Mem: 8GB Speed: ~940 Mb/s (test command: dd)
- Disk: 512 bytes logical, 4096 bytes physical, 5400 rpm, SATA 3.1, 6.0 Gb/s (current: 6.0 Gb/s)

2. Time evaluation

- Single DataNode: 127493s
- Quater DataNode: 126283s

3. Discussion

Since the test happen in single machine, the network error as well as bandwidth limitation is not exists. therefore the different between parrallel download and single download is not actually large. (But it still heat up my machine a lot). And I also recognize that sometimes, there will be a delay in chunk transfer (it could be the time that HDD writes data from its buffer to disk)

Lan Test

The measurement count on 2 machines (my laptop and my PC)

1. Machine info:

- laptop: thinkpad E590 - above information
- PC:
  - CPU: i5 gen 3
  - Mem: 8gb DDR3
  - Disk: SATA 3.1, 6.0 Gb/s (current: 6.0 Gb/s)

2. Time evaluation

- Single DataNode: N/A
- Quater DataNode: N/A

3. Discussion

The test happen with 2 machines connected through and FPT router (a chip router in normal familly). The laptop connected through wireless while the PC is connected by ethernet. There are some error happen in lan network and cause the transfer process stucked after approximate 1.7M data transfered. For now, after 3 days checking, the most proper explanation that the FPT router, in someway, cause loss package in chunk transfer. And in accidentally, the importante package (confirm package which include **200** string) is loss despice of the fact that the data is transfered in TCP layer and the download process go to deadlock.

For limitation of time and the deadline is broken too long, the further effort will be suspend infinity until the developer have back time.