Name: _____ USC ID: _____

INF 551 – Fall 2016 (Morning)

Quiz 4: File systems & HDFS (10 points)

10 minutes

1. [7 points] Consider writing a new file "bar" under the "/foo/more" directory. Fill in the following table by placing "read" or "write" operation in the proper cells. Each row has only one operation. The order of rows indicates the sequence of the operations. (Assume when the file is **opened** for write, the file system does not allocate any data blocks for the file.)

	inode bitmap	data bitmap	root inode	foo inode	more inode	bar inode	root data	foo data	more data	bar data[0]
open()			read							
							read			
				read						
								read		
					read					
									read	
	read									
	write									
						write				
									write	
					write					
write()						read				
		read								
		write								
										write
						write				

2. [3 points] Describe the process (i.e., the steps) of **writing** a file in HDFS

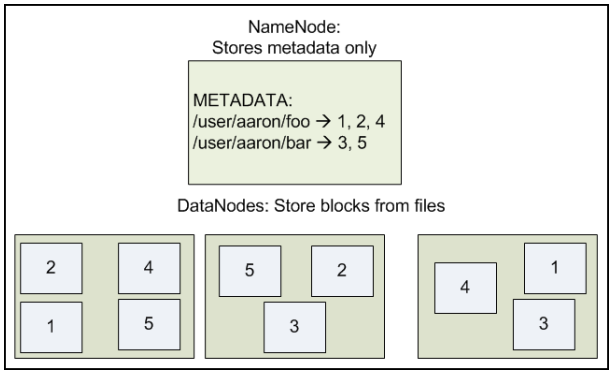
For every block (64MB typically) of the file, the client asks the NameNode to nominate a number of DataNodes to hold replica of the block.

It then divides the blocks into a number of packets (e.g., 64KB) and sends the packets to the DataNodes in a pipelined fashion. To speed up the process, the client does not wait for the acknowledge of previous packet before sending the next one.

Name: _____ USC ID: _____

Quiz 4: HDFS & File Formats (10 points. 15 minutes)

1. [5 points] Refer to the following diagram on an example HDFS. Answer the following questions.



a. [1 point] What is the replication factor in this HDFS?

Each block has two replicas distributed across three DataNodes.

Thus the replication factor in this HDFS is **2**.

b. [1 point] Which node does the client first contact when reading/writing a file?

Client first contacts **NameNode**, which informs the client of the closest DataNodes storing blocks of the file when reading, and selects DataNodes for holding its replicas when writing.

c. [1 point] What is the typical size of a block in HDFS?

64MB, which is much large than disk block size.

d. [2 points] When writing a file in HDFS, how many packets is each block divided into? What is the size of each packet?

One point for each

One block, which is 64MB, is divided into **1024** packets, each of which is **64KB**.

2. [3 points] Unicode code point for the Chinese character 中(means middle) is U+4E2D. Give its **UTF-8** encoding in both **binary** and **hexadecimal** formats.

U+4E2D is within the range from U+0800 to U+FFFF, denoting that the code sequence length being 3. U+4E2D in binary is **0100 1110 0010 1101**. Encode in the following steps:

1. Take 6 bits at a time backwards from end and add leading **10** to form the last two code units;
2. Add leading **111**, which indicates this code point consists of 3 code units, to the rest 4 bits and 0's to any unfitted spaces (one 0 in this case) to form the first code unit;
3. The binary code will be: **11100100 10111000 10101101**.
4. The hexadecimal code will be: **E4 B8 AD**

0.5 point for each transformation error between binary and hexadecimal formats, and each division and completion errors when forming code units. 2 points for wrong number of code units.

3. [2 points] What is the output of `json.dumps(['foo', {25: ('bar', None, 1.0, 2, False)}])`?

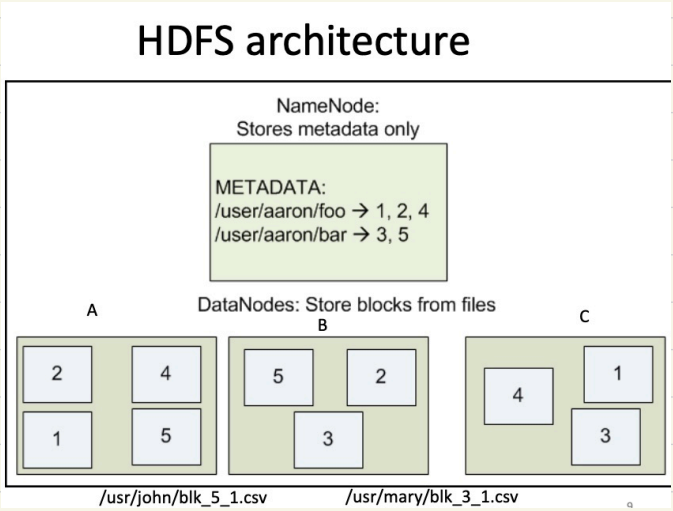
```
["foo", {"25": ["bar", null, 1.0, 2, false]}]
```

0.5 point deducted for each minor error, such as quotation marks and wrong capitalization. 1 point deducted for each wrong data structure.

HDFS

- a large-scale distributed & parallel batch-processing infrastructure

  process a series of jobs without human intervention



HDFS has:

1 single NameNode, storing metadata:

- namespace
- attribs of dir & files
  - permission
  - access time
  - modification time
- mapping file → DataNodes.

several DataNodes

1 secondary NameNode
- Maintaining checkpoints / images of NameNode
- For recovery
- not a fail over node

- In a single-machine setup.

Block Size

128MB (version 2 +)

Why: ① metadata ↓

② Faster Streaming read of data.

Reading & writing : 2 phases

Phase 1. client asks NameNode for block locations

reading: getBlockLocations()

I: ① File name
② offset (to start reading)
③ length (to be read)

O: Located blocks
data nodes
&
offsets

writing: create() / append() P23

addBlock()

Phase 2. client talks to DataNode for data transfer

reading readBlock()

writing writeBlock()

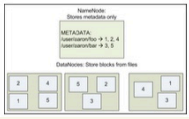
其它 copyBlock() > For load balancing

replaceBlock()

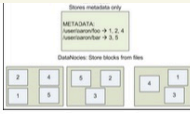
↳ move a block from one DataNode to another

Reading a file

1. Client first contacts NameNode which informs the client of the closest DataNodes storing blocks of the file
   - This is done by making which RPC call? getBlockLocations?
2. Client contacts the DataNodes directly for reading the blocks
   - Calling readBlock()

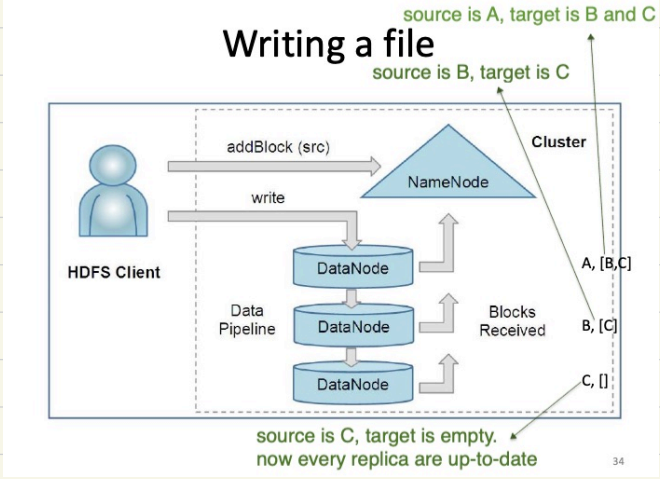


Writing a file



- Blocks are written one at a time
  - In a pipelined fashion through the data nodes
- For each block:
  - Client asks NameNode to select DataNodes for holding its replica (using which rpc call?) addBlock()
    - e.g., DataNodes 1 and 3 for the first block of /user/aaron/foo
  - It then forms the pipeline to send the block

高亮这句会考到，背诵



Data Pipelining

Consider a block X to be written to DataNode A, B & C.

1. X is broken down to packets (64KB)
   - 128 MB / 64 KB = 2048
2. Client sends the packet to DataNode A.
3. A sends it further to B
   B further to C

* Acknowledgement

- Client maintains an ack queue.
- Packets are removed from ack queue once received by all DataNodes
- When all packets are written, client notifies NameNode.
- ☆ Client does not wait for the ack of previous packet before sending next one.