CSE Hands-on-3 20161109

Name: ChenJiayang ID: 5140379036

Question1：What do the first two parameters to WordCount's...?

- The first parameter "self" means the class instance itself , and the other parameter "maptask" refers to the num of the workers

Question2：Briefly explain how calling run triggers ...

- Because the class WordCount is a subclass of MapReduce, so when it executes, it first call run() in class MapReduce
- Then we can see in the run(), it first creates a pool in order to contain those processes whose capacity equals to the larger workers quantity.
- Next, it calls doMap() and doReduce(), and then in function doMap(), it calls Map() in subclass WordCount as well as doReduce() calls Reduce(), because in subclass it has rewritten this two functions in parent.

Question3：What do the parameters keyvalue and value of...

- In the function Map(), keyvalue represents the byte offset in the input file, which can be used as an index to mark those split files.
- In the function Map(), value represents the content in the split file.

Question4：What do the parameters key and keyvalues of...

- In the function Reduce(), key represents the word it wants to calculate.
- In the function Reduce(), keyvalues represents the (key, value) pairs it received from Map().

Question5：How many invocations are there to doMap...

- After adding print "" before "return" instruction in this two functions, there are four invocations to doMap() and two invocations to doReduce().
- The reason is that when we run WordCount instance, there is "wc = WordCount(4, 2, sys.argv[1])", the first two parameters represent how many invocations it will cause.

Question6：Which invocations run in parallel?...

- When we run doMap(), its invocations run in parallel and when doMap() finished, then doReduce() begis to run, its invocations also run in parallel.

Question7：How much input (in number of bytes) does a...

- What a signal doMap() can processes input number of bytes is 1237228. Because I firstly print the total size of the input file and then in the function split(), the size is divided by maptask, that is to say four under this circumstance.

Question8：How much input (in number of keys) does a ...

- What a signal doReduce() can processes input number of bytes is 4472.

Question9：For which parameters of maptask and reducet ...

- Due to my processor in my pc is Core i5 and it has four cores. So when the number of maptask and reducetask are both smaller than four, I can see speedup obviously.
- However, if we call WordCount(5,5,sys.argv[1]) and compare the cost time to WordCount(4,4,sys.argv[1]) then you will find a lower speed, and I think the reason is that the performance of mapreduce is limited by physical hardware, in this case is the numbers of cores. When each worker has reached the full capability it can provide, then just adding the number of maptask or reducetask will not work effectively.

Question10：Include the code for your ReverseIndex class ...

```
    # Produce a (key, value) pair for each word in value
    # TODO: your code here
    def Map(self, keyvalue, value):
        results = []
        i = 0
                      )
        while i < n:
            # skip non-ascii letters in C/C++ style a la MapReduce paper:
            while i < n and value[i] not in string.ascii_letters:
                i += 1
            start = i
            while i < n and value[i] in string.ascii_letters:
                i += 1
            w = value[start:i]
            if start < i and w.istitle():
                results.append ((w.lower(), start))
        return results

    # Reduce [(key,value), ...])
    # TODO: your code here
    def Reduce(self, key, keyvalues):
        valuelist = []
        for pair in keyvalues :
            if key == pair[0]:
                valuelist.append(pair[1])
        return (key, valuelist)
```

- Firstly, in the function Map(), the variable "start" and "i" represent the starting position and the ending position of a single word in the input file. So, when we find one valid word, we add its starting position "start" to the result[]
- Then, in the function Reduce(), in all pairs(key,value) input, we first judge whether the word is what we are looking for, if yes, we add its pair[1] to valuelist which is a tuple of its position we once created in the Map()