Distributed systems lab4

515015910005 丁丁

dingd2015@sjtu.edu.cn

Part1: Map/Reduce input and outqput

- doMap函数：
  - 读取输入文件内容
  - 调用用户定义的map函数，传入文件内容。用户定义的map函数将返回一个KeyValue的List。
  - 根据key的hash值将key分入不同的List中。
  - 使用JSON.toJSONString将List转为字符串。将上一步所分成的不同的List写入不同的中间文件中。
- doReduce函数：
  - 读取自己所需要reduce的中间文件。
  - 调用JSON.parseArray将中间文件（字符串形式）转换回List对象。
  - 使用Map<String, String[]>对象，将List里的KeyValue对根据Key分类好，逻辑上对应Key:Value1, Value2, ...的形式.
  - 对于每一个Key，调用用户定义的reduce函数，传入内容为Key和所有整理好的Value。reduce函数返回一个String。
  - 使用Map<String, String>对象来存储Key和reduce的返回结果。
  - 使用JSON.toJSONString将Map<String, String>转为字符串，写入到输出文件中。

Part2: Single-worker word count

- mapFunc函数：
  - 使用正则表达式“[a-zA-Z0-9]+”进行匹配，遍历所有的matcher，当发现一个matcher时，向结果List中增加一个(matcher对应单词,"1")的KeyValue对。
- reduceFunc函数：
  - 由于mapFunc函数中输出的结果每个KeyValue的value都是"1"，所以只需要count传入的String[]数组有多少个元素即可（多少个1相加）。将count的结果转为字符串返回。
- 测试结果：

```
jos@cosmic:~/test$ sort -n -k2 mrtmp.wcseq | tail -10
that: 7871
it: 7987
in: 8415
was: 8578
a: 13382
of: 13536
I: 14296
to: 16079
and: 23612
the: 29748
```

Part3: Distributing MapReduce

- schedule函数：
  - schedule函数需要每次给worker分配任务，而每个worker只能1次做1个任务。这个通过registerChan来实现。当schedule试图分配任务时，用registerChan.read()来获取一个worker，在worker结束这个任务后，用registerChan.write()写回这个worker。这样即可保证worker只会1次做1个任务。
  - schedule需要等所有任务完成后再return，这个通过CountDownLatch来实现。首先用task数量来初始化latch，随后如果某个任务做完调用latch.countDown。schedule等待latch归0即可。

Part4: Handling worker failures

- schedule函数：
  - 在分配任务过程中，catch Call超时的异常，倘若出现超时，分配给下一个worker即可。

Part5: Inverted index generation (optional, as bonus)

- mapFunc函数：
  - 使用正则表达式“[a-zA-Z0-9]+”进行匹配，遍历所有的matcher，当发现一个matcher时，向结果List中增加一个(matcher对应单词, file)的KeyValue对。
- reduceFunc函数：
  - 将传入的values数组中的各个元素根据要求拼接在一起即可，注意去除重复元素。
- 测试结果：

```
jos@cosmic:~/test$ sort -k1,1 mrtmp.iiseq | sort -snk2,2 | grep -v '16' | tail -10
yesterday: 8 pg-being_ernest.txt,pg-dorian_gray.txt,pg-frankenstein.txt,pg-grimm.txt,pg-huckleberry_finn.txt,pg-metamorphosis.txt,pg-sherlock_holmes.txt,pg-tom_sawyer.txt
yet: 8 pg-being_ernest.txt,pg-dorian_gray.txt,pg-frankenstein.txt,pg-grimm.txt,pg-huckleberry_finn.txt,pg-metamorphosis.txt,pg-sherlock_holmes.txt,pg-tom_sawyer.txt
you: 8 pg-being_ernest.txt,pg-dorian_gray.txt,pg-frankenstein.txt,pg-grimm.txt,pg-huckleberry_finn.txt,pg-metamorphosis.txt,pg-sherlock_holmes.txt,pg-tom_sawyer.txt
You: 8 pg-being_ernest.txt,pg-dorian_gray.txt,pg-frankenstein.txt,pg-grimm.txt,pg-huckleberry_finn.txt,pg-metamorphosis.txt,pg-sherlock_holmes.txt,pg-tom_sawyer.txt
YOU: 8 pg-being_ernest.txt,pg-dorian_gray.txt,pg-frankenstein.txt,pg-grimm.txt,pg-huckleberry_finn.txt,pg-metamorphosis.txt,pg-sherlock_holmes.txt,pg-tom_sawyer.txt
young: 8 pg-being_ernest.txt,pg-dorian_gray.txt,pg-frankenstein.txt,pg-grimm.txt,pg-huckleberry_finn.txt,pg-metamorphosis.txt,pg-sherlock_holmes.txt,pg-tom_sawyer.txt
your: 8 pg-being_ernest.txt,pg-dorian_gray.txt,pg-frankenstein.txt,pg-grimm.txt,pg-huckleberry_finn.txt,pg-metamorphosis.txt,pg-sherlock_holmes.txt,pg-tom_sawyer.txt
Your: 8 pg-being_ernest.txt,pg-dorian_gray.txt,pg-frankenstein.txt,pg-grimm.txt,pg-huckleberry_finn.txt,pg-metamorphosis.txt,pg-sherlock_holmes.txt,pg-tom_sawyer.txt
yourself: 8 pg-being_ernest.txt,pg-dorian_gray.txt,pg-frankenstein.txt,pg-grimm.txt,pg-huckleberry_finn.txt,pg-metamorphosis.txt,pg-sherlock_holmes.txt,pg-tom_sawyer.txt
zip: 8 pg-being_ernest.txt,pg-dorian_gray.txt,pg-frankenstein.txt,pg-grimm.txt,pg-huckleberry_finn.txt,pg-metamorphosis.txt,pg-sherlock_holmes.txt,pg-tom_sawyer.txt
```