DSLAB – MAP&REDUCE

517021910799 朱文杰

实验设计

PART I : Sequential Map/Reduce

根据注释，Mapper 需要读一个文件进行 Map 然后将结果写入临时文件中，Reducer 则需要对临时文件进行 Reduce 并将结果写入输出文件中。因此两者都需要读写文件，并且临时文件需要读后销毁。因此我选择将 read(file,deleteMode)/write(file,contents)等文件相关的操作封装成方法处理没被 Handle 的异常，统一放入 Mapper 类下，理论上应该独立一个 Util 类，但是因为不允许增加文件所以只能行此下策。

DoMap 首先需要构造 mapF 的参数，也就是 File/Content，然后将 map 结果的 list<key,value>使用 keyHash%nR 进行分组分配给 Reducer，获得临时文件名，并写入临时文件中。流程基本固定。

DoReduce 的关键就是，如何把从多个文件中获取的 list<key，value>整合成<key,list<value>>传递给 ReduceF 并得到最终的 map<key,string>,这里我参考了 flatMap 的用法，使用 flatMap 把**嵌套的二维结构降低成一维**的 map<key,list<value>>, 然后遍历这个 map 并建立最终的 map。难点在于如何进行重构上。

PART II: Word Count

Map 部分，针对每一个匹配到的结果为结果 list 增加一个键值对，内容为 key:"",value 本身不具备意义，仅仅供计数。

Reduce 部分，获得输入的 list 的长度即可。

PART III: Scheduling

为了实现并发，必须要使用 Java 中的 Thread 机制，因此这里要对 Thread 进行派生，。这里为了保证 schdule 在线程结束时才 return，创建一个倒计时为 nTasks 的锁交给线程共享，当线程结束时就减少倒计时。在 schdule 中循环 nTasks 次创建所有线程并且 start，然后 await。

线程执行的函数中调用 doTask 和 countDown，首尾使用 read/write 进行 worker 的分配。

PART IV: Fault Tolerance

这里的问题在于处理 RPC 异常，当 RPC 异常时，我们可以认为之前的 worker thread 已经与我们无关了，所以我们只需要在 try/catch 处理过程中按照原本的参数**重新创建新线程**即可。换而言之就是类比在计算机网络术语中，为了保证 At least once，我们通常采用 retry

来防止丢包，这里也是同理。

能够使用 retry 的原因是在于，我们的 reduce/map 具有**幂等性**，不管使用哪个 worker 执行执行几次，他们都能得到相同的输出。这个思想和 Google 分布式为了防止长尾的解决方案很类似，谷歌也是采用重复的方法，当某个节点卡住时，创建几个同任务节点谁先完成都可以。如果从幂等性进一步向下追究，那就在于 map/reduce 具有 FP 的特征，不具备副作用，因此只要输入参数相同，必然有相同的输出。

PART V: Inverted Index

Map 部分，针对每一个匹配到的结果为结果 list 增加一个键值对，内容为 key:filename

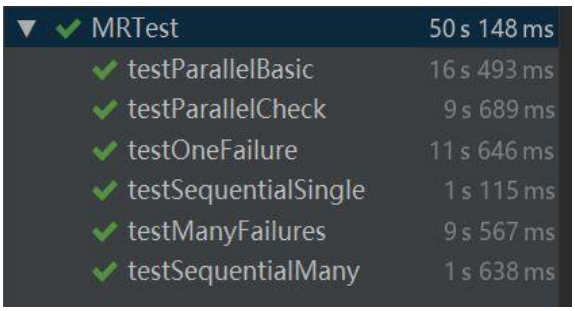
Reduce 部分，首先对于获得的 List<filename>转换成 stream 进行排序去重，使用 count 获得其数目，使用 collectors.joining 转换为,分割的字符串，这样就得到了格式要求的 num fileA,fileB,fileC 的结果字符串了。

实验感想

在做 PART4 时个人其实很不可思议，解决方案似乎太简单了，后来想了想觉得 FP 果然有其高明之处。分布式系统下，同步服务器的状态是很困难的，需要 PAXOS 那种复杂的协议，但是计算则完全独立，想要备份、集群无状态无副作用的服务器几乎是没有额外开销的，这保证了 scalability。难怪 SICP 也要用 Lisp/python 来讲 FP，后端也越来越强调服务器 stateless 了。

另外一点就是在 map/reduce 本身，那么多代码其实就是单纯通过映射在变换结构而已。这让我想起很早以前看过的函数式编程，通过各种高阶函数的引入把各种表面上完全风马牛不相及的函数同质化成拓扑学上的同胚。感觉在 Map/Reduce 中很重要的点就是一种数学的思想，通过各种变换，从一个数据映射到另一个数据，虽然 Mapper 和 Reducer 只认他们规定的输入格式，但是各种各样的复杂变换都能殊途同归到这种格式的组合上。

附录 实验结果



```
朱文杰@DESKTOP-VKQK9NG MINGW64 ~/Desktop/lab4/mapreduce
$ 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

朱文杰@DESKTOP-VKQK9NG MINGW64 ~/Desktop/lab4/mapreduce
$ LC_ALL=C sort -k1,1 mrtmp.iiseq | sort -snk2,2 | grep -v '16' |
> tail -10
www: 8 pg-being_ernest.txt,pg-dorian_gray.txt,pg-frankenstein.txt,pg-grimm.txt,p
g-huckleberry_finn.txt,pg-metamorphosis.txt,pg-sherlock_holmes.txt,pg-tom_sawyer
.txt
year: 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_sawye
r.txt
years: 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_sawy
er.txt
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,p
g-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,p
g-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_sawy
er.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_sawye
r.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_s
awyer.txt
zip: 8 pg-being_ernest.txt,pg-dorian_gray.txt,pg-frankenstein.txt,pg-grimm.txt,p
g-huckleberry_finn.txt,pg-metamorphosis.txt,pg-sherlock_holmes.txt,pg-tom_sawyer
.txt
```