ps6

Hangyu Huang

31 October 2017

NOTES

Partner: YUE HU As I forgot my password of savio account, I used yue_hu's account to process the dataset on Spark, and use R on a single Savio node.

Problem2

To find the users ask questions related R but no questions related to python. 1. create a view usersPython that the userid of users ask questions related to python. 2. crate a view usersR that the userid of users ask questions related to R 3. use outer join to find the userid from usersPython that is null. 4. use count() to find the total number of such users

```
library(RSQLite)
drv <- dbDriver("SQLite")
dir <- 'C:/Users/crypress/Desktop/243/ps6'
dbFilename <- 'stackoverflow-2016.db'
db <- dbConnect(drv,dbname = file.path(dir, dbFilename))
dbGetQuery(db, "select distinct tag from questions_tags")
dbGetQuery(db, " create view usersPython as
           select distinct userid from users U join
           questions Q on U.userid = Q.ownerid join
           questions_tags T on Q.questionid = T.questionid
           where tag ='python'")
dbGetQuery(db, "create view usersR as
           select distinct userid from users U join
           questions Q on U.userid = Q.ownerid join
           questions_tags T on Q.questionid = T.questionid
           where tag ='r'")
dbGetQuery(db, "select count(R.userid) from usersR R left join
           usersPython P on R.userid = P.userid
           where P.userid is NULL ")
drop("usersPython")
drop("usersR")
```

count(R.userid) is total 18611

problem 3

The subprime crisis started after Lehman's failure in mid September 2008, I would like to see how popular this topic is by counting the numbers of website hit related to the word "Lehman_brothers". And I am also interested that how people using english pay attention to the subprime crisis.

1. To run the following coade on PySpark in Savio to fileter the lines with key words "lehman_brothers"

```
##knitr read-filter
dir = '/global/scratch/paciorek/wikistats_full'
lines = sc.textFile(dir + '/' + 'dated')

### filter to sites of interest ###
def find(line, regex = "Lehman_Brothers", language = None):
    vals = line.split(' ')
    if len(vals) < 6:
        return(False)
    tmp = re.search(regex, vals[3])
    if tmp is None or (language != None and vals[2] != language):
        return(False)
    else:
        return(True)


Lehman = lines.filter(find).repartition(480)


### map-reduce step to sum hits across date-time-language triplets ###
def stratify(line):
    # create key-value pairs where:
    #   key = date-time-language
    #   value = number of website hits
    vals = line.split(' ')
    return(vals[0] + '-' + vals[1] + '-' + vals[2], int(vals[4]))

counts = Lehman.map(stratify).reduceByKey(add)

# sum number of hits for each date-time-language value
counts = lehman_brothers.map(stratify).reduceByKey(add)

### map step to prepare ouput ###
def transform(vals):
    # split key info back into separate fields
    key = vals[0].split('-')
    return(",".join((key[0], key[1], key[2],       str(vals[1]))))

### output to file ###
outputLeham = '/global/scratch/yue_hu/hhy'
counts.map(transform).repartition(1).saveAsTextFile(outputlehman)

### download the file into local dir ###
scp yue_hu@dtn.brc.berkeley.edu:/global/scratch/yue_hu/hhy/part-00000 counts
```

2. analysis the data by R

```
install.packages('readr')
install.packages('dplyr')
library(readr)
library(dplyr)
Lehman <- readr::read_delim("counts", ",", col_names = FALSE)
names(Lehman) <- c('date','time', 'lang','hits')
```

```
#count the number of website hits by date
dateLehman <- Lehman %>% group_by(date) %>% summarise(hits = sum(hits))
dateLehman$date <- as.Date.character(dateLehman$date, "%Y%m%d")
plot(dateLehman$date, dateLehman$hits, type = 'l', xlab = 'date', ylab = 'hits')

# count the number of website hits in english website

sub <- Lehman %>% filter(Lehman$lang == 'en')
enLehman <- sub %>% group_by(date) %>% summarise(hits = sum(hits))
as.Date.character(enLehman$date, "%Y%m%d")
plot(enLehman$date, enLehman$hits, type = 'l', xlab = 'date', ylab = 'hits')

# count the number of website hits by language
langLehman <- Lehman %>% group_by(lang) %>% summarise(hits = sum(hits))
langLehman$lang <- as.character(langLehman$lang)
langLehman <- langLehman %>% filter (langLehman$hits >10000)
pie(c(langLehman$hits),labels = c(langLehman$lang), main = "Lehman website hits by langague")
```

problem 4

1. use 24 cores
2. use read_delim to get the lines of each file and use grep() to get the lines with pattern "Barack Obama"
3. use foreach() to do the parallel work.
4. test 2 files to check the code is correct
5. run the code with 96 files (1/10 of totall)

```
library(readr)
require(parallel)
require(doParallel)
library(foreach)

path = '/global/scratch/paciorek/wikistats_full/dated_for_R'
files <- list.files(path, full.names = T)
nCores <- 24
cl <- makeCluster(nCores)
registerDoParallel(cl)

nsub <- files[1:2]

obama <- foreach (file=nsub, .combine = rbind) %dopar% {
  lines <- readr::read_delim(file,
  delim = " ", col_names = F, quote = "")
  obama <- lines[grepl("Barack_Obama", lines$X4),]

  return(obama)
}

outfile = '/global/scratch/yue_hu/hhy/obamaSample.csv'
write.csv(obama, file = outfile, row.names = FALSE)
```

download the sample into my local disk to check if it works

$ scp yue_hu@dtn.brc.berkeley.edu:/global/scratch/yue_hu/hhy/obamaSample.csv sample.csv

test the sample of 96 files to get the running time

```
library(readr)
require(parallel)
require(doParallel)
library(foreach)

path = '/global/scratch/paciorek/wikistats_full/dated_for_R'
files <- list.files(path, full.names = T)
nCores <- 24
cl <- makeCluster(nCores)
registerDoParallel(cl)

nsub <- files[1:96]

obama <- foreach (file=nsub, .combine = rbind) %dopar% {
  lines <- readr::read_delim(file,
                             delim = " ", col_names = F, quote = "")
  obama <- lines[grepl("Barack_Obama", lines$X4),]

  return(obama)
  gc()
}

outfile = '/global/scratch/yue_hu/hhy/obama96.csv'
write.csv(obama, file = outfile, row.names = FALSE)
```

problem 4(b)

I didnt get the result as it shows that "error: Exceeded step memory limit at some point.", i have tried to use gc() in the foreach() loop to remove the temporary memory.

problem 4(c)

Based on Unit7, if each task takes similar time, you want to preschedule the tasks to reduce communication. If you have few tasks or tasks with highly variable completion times, you don't want to preschedule, to improve load-balancing.

In this question, we have 960 files, and each file should take similar times, therefore, prescheduling may improve load-balancing.

problem 5

submited by hand writing