Current Status:

wordcount code ready, native R(single thread), RHadoop, Rspark,

Running fine with smaller file but fails for the large file.

TODO: make sure it runs on the cluster and uses multiple nodes successfully.

start some initial timings for the different runs.

dataset: wikipedia data set.

wikipedia dataset: 100MB, 200MB, 1GB, 2GB, 12GB. 20GB, 30GB, 50GB.

checkout Hibench <https://github.com/intel-hadoop/HiBench>, for generating testing data.

--to consider: nativeR using Parallel package over multiple nodes,

R scripts + Hadoop streaming directly. vs. using RHadoop,

Todo: prepare Kmeans code

-----------

Parallelized R script for Wordcount

The script creates multiple “workers” and assigns each a file to process. After processing these workers return a list, which then are combined into a list of lists. Unlist and table does the word count by combining all the words appropriately.

<http://www.r-bloggers.com/how-about-a-snowdoop-package/>

###############################################

#

# give each node in the cluster cls an ID number

assignids <- function(cls) {

clusterApply(cls,1:length(cls),

function(i) myid <<- i)

}

# each node executes this function

getwords <- function(basename) {

fname <- paste(basename,".",myid,sep="")

words <- scan(fname,what="")

words

}

# manager

wordcount <- function(cls,basename) {

assignids(cls)

clusterExport(cls,"getwords")

lists <- clusterCall(cls,getwords,basename)

freq <- table(unlist(lists));

freq

}

# call example:

library(parallel)

c2 <- makeCluster(2)

wordcount(c2,"words")

#

#########################################