METODA DE LUCRU

Nr. nuclee:6
1 master
5 workers

Se citeste directorul de date si se trimite pe rand la fiecare worker calea catre 1 fisier de date. Daca nu sunt workeri disponibili se asteapta pana se elibereaza unul.

Workerul imparte fisierul de date in cuvinte si creeaza fisiere temporare cu denumirea cuvantului si sursa acestuia (ex: andromeda_3 pt cuvantul andromeda gasit in fisierul 3.txt).

Se face reduce pe fisierele temporare apoi folderul cu fisiere temporare este sters.

ALGORITM

Master:

```
function MapReduceMaster ()
{
        createTempDirectory(tempDir)
        foreach data in dataDir
                while(freeWorkers<=0)
                        waitForWorkers()
                        freeWorkers++

                sendDataToWorker(data)
                freeWorkers--

        waitForAllWorkers()
        reduce(tempDir)
        deleteTempDirectory(tempDir)

}
```

Worker:

```
function MapReduceWorker ()
{
        while(getData(data))
                foreach word in data
                        createTempFile(word)
                sendDone()
        sendAllDone()

}
```

COD MASTER

```
if((dp = opendir(dir)) == NULL) {
    fprintf(stderr,"cannot open directory: %s\n", dir);
}
else
{
    chdir(dir);
    while((entry = readdir(dp)) != NULL) {
        lstat(entry->d_name,&statbuf);
            if(strcmp(".",entry->d_name) == 0 ||
                strcmp("..",entry->d_name) == 0)
                continue;
            while(sCores<=0)
            {
                MPI_Recv(Message,255,MPI_CHAR,MPI_ANY_SOURCE,MPI_ANY_TAG,MPI_COMM_WORLD,&status);
                sCores++;
                coreArray[status.MPI_SOURCE]=status.MPI_TAG;
            }
            for(int i=1;i<nrCores;++i)
            {
                if(coreArray[i]==0)
                {
                    sprintf(Message,"test-files/%s",entry->d_name);
                    MPI_Send ( Message , 255 , MPI_CHAR, i , tag , MPI_COMM_WORLD );
                    sCores--;
                    coreArray[i]=1;
                    break;
                }

            }

    }
    for(int i=1;i<nrCores;++i)
    {
        sprintf(Message," ");
        tag=2;
        MPI_Send ( Message , 255 , MPI_CHAR, i , tag , MPI_COMM_WORLD );
```

```
            }
            for(int i=1;i<nrCores;++i)
            {
                sprintf(Message," ");
                tag=2;
                MPI_Send ( Message , 255 , MPI_CHAR, i , tag , MPI_COMM_WORLD );

            }
            chdir("..");
            closedir(dp);

        }
        for(int i=1;i<nrCores;++i)
        {
            int statre=0;
            MPI_Recv(&statre,1,MPI_INT,i,2,MPI_COMM_WORLD,&status);
            sCores++;
        }
        printf("%d\n", sCores);

        //REDUCE<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
        char *tempdir="temp";
        if((dp = opendir(tempdir)) == NULL) {
            fprintf(stderr,"cannot open directory: %s\n", dir);
        }
        else
        {
            chdir(tempdir);
            while((entry = readdir(dp)) != NULL) {
                lstat(entry->d_name,&statbuf);
                    if(strcmp(".",entry->d_name) == 0 ||
                        strcmp("..",entry->d_name) == 0)
                        continue;
                        printf("%s\n",entry->d_name);
```

COD WORKER

```
while(tag!=2)
{

MPI_Recv(Message,255,MPI_CHAR,MPI_ANY_SOURCE,MPI_ANY_TAG,MPI_COMM_WORLD,&status);
if(status.MPI_TAG==2)
{
    tag=status.MPI_TAG;
    break;
}
FILE *fp;
FILE *fpc;


   char buff[255];

   fp = fopen(Message, "r");
   char* fileName;
   fileName=strtok (Message,"/");
   fileName=strtok(NULL, "/");
   while(fgets(buff,255,fp)!=NULL)
   {

       char pathcreate[255];
       char *token;

       token = strtok (buff," \t,.-\n()[]{}*\";?!1234567890/@#$%%^&_+=:<>|~`\r");
       while (token != NULL)
       {

           char *lower=token;
           while(*lower) {
               *lower = tolower(*lower);
               lower++;
           }
           sprintf(pathcreate,"temp/%s_%s",token,fileName);
           fpc = fopen(pathcreate,"w");
```

TPUT DEBUG CONSOLE TERMINAL

```
        token = strtok (buff," \t,.-\n()[]{}*\";?!1234567890/@#$%%^&_+=:<>|~`\r");
        while (token != NULL)
        {

            char *lower=token;
            while(*lower) {
                *lower = tolower(*lower);
                lower++;
            }
            sprintf(pathcreate,"temp/%s_%s",token,fileName);
            fpc = fopen(pathcreate,"w");
            fprintf(fpc,"%d\n",rank);
            fclose(fpc);
            token = strtok (NULL, " \t,.-\n()[]{}*\";?!1234567890/@#$%%^&_+=:<>|~`\r");
        }
    }

    fclose(fp);


    tag=0;
    sprintf(Message," ");
    MPI_Send ( Message , 255 , MPI_CHAR, 0 , tag , MPI_COMM_WORLD );


}
int stat=0;
MPI_Send ( &stat , 1 , MPI_INT, 0 , tag , MPI_COMM_WORLD );
// while(tag==2)
// {

// }
```

BIBLIOGRAFIE

https://sites.google.com/site/mriap2008/intro_to_mapreduce.pdf
https://sites.google.com/site/mriap2008/what_is_mapreduce.pdf
https://en.wikipedia.org/wiki/MapReduce
https://www.cplusplus.com/reference/cstring/strtok/
https://stackoverflow.com/questions/8149569/scan-a-directory-to-find-files-in-c