EXPERIMENT -7

MAPREDUCEPROGRAM3

OBJECTIVE:

Implement matrixmultiplicationwithHadoopMap Reduce.

RESOURCES:

VMWare,Webbrowser,4GBRAM, Hard Disk80GB.

PROGRAMLOGIC:

Weassumethattheinputfiles for AandBarestreamsof(key,value)pairsinsparse matrix format, where each key is a pair of indices (i,j) and each value is the correspondingmatrix element value. The output files for matrix C=A*B are in the same format.

Wehave thefollowinginput parameters:

Thepathoftheinput fileordirectoryfor matrixA.
Thepathoftheinputfileor directoryformatrix B.

Thepathofthedirectoryfortheoutputfilesformatrix C. strategy = 1, 2, 3 or 4.

R=thenumberofreducers.

I=the numberofrowsinAandC.

K=thenumberofcolumnsin AandrowsinB. J = the number of columns in B and C.

IB=the numberofrowsperAblockandCblock.

KB=thenumberofcolumnsperAblockandrowsperBblock. JB = the number of columns per B block and C block.

Inthepseudo-codefortheindividualstrategiesbelow,wehaveintentionallyavoided factoringcommoncodeforthepurposesofclarity.Notethatinallthestrategiesthememory footprint of both the mappers and the reducers is flat at scale.

Notethatthestrategiesallworkreasonablywellwithbothdenseandsparse matrices. For sparse matrices we do not emit zero elements. That said, the simple pseudo-code for multiplying the individualblocksshownhere iscertainlynotoptimalforsparse matrices.Asa learningexercise, our focus here is on mastering the MapReduce complexities, not onoptimizing the sequential matrix multipliation algorithm for the individual blocks.

Steps

1. setup ()
2. varNIB=(I-1)/IB+1
3. varNKB=(K-1)/KB+1
4. var NJB=(J-1)/JB+1
5. map(key,value)

6. iffrommatrixAwithkey=(i,k) andvalue=a(i,k)
7. for 0<=jb<NJB
8. emit(i/IB,k/KB,jb,0),(imodIB,kmodKB,a(i,k))
9. iffrommatrixBwithkey=(k,j)andvalue=b(k,j)
10. for 0<=ib<NIB
emit(ib,k/KB,j/JB,1),(kmodKB,jmodJB,b(k,j))

Intermediatekeys(ib,kb, jb, m)sortinincreasingorderfirst byib,thenbykb,thenbyjb, then by m. Note that m = 0 for A data and m = 1 for B data.

Thepartitionermapsintermediatekey (ib,kb,jb,m)to areducerrasfollows:

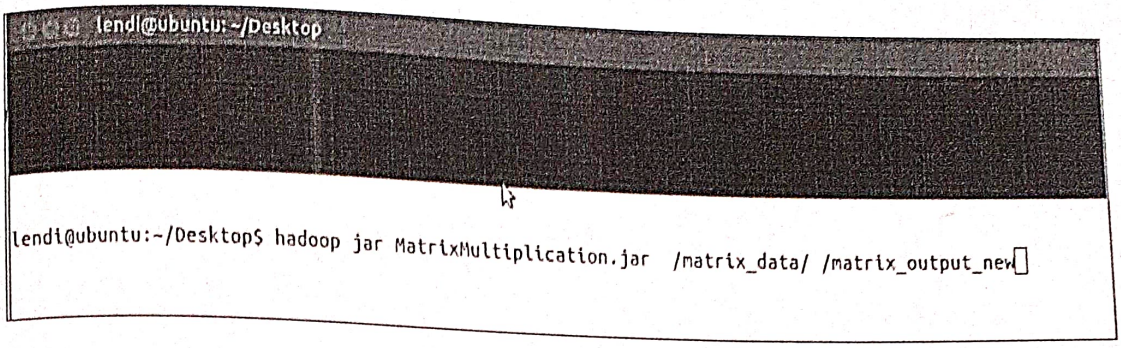
11. r=((ib*JB+jb)*KB+kb)modR
12. These definitions for the sorting order and partitioner guarantee that each reducer R[ib,kb,jb]receivesthedatait needsforblocksA[ib,kb]andB[kb,jb],withthedatafor the A block immediately preceding the data for the B block.
13. varA=newmatrixofdimensionIBxKB
14. var B=newmatrixofdimension KBxJB
15. varsib=-1
16. varskb=-1

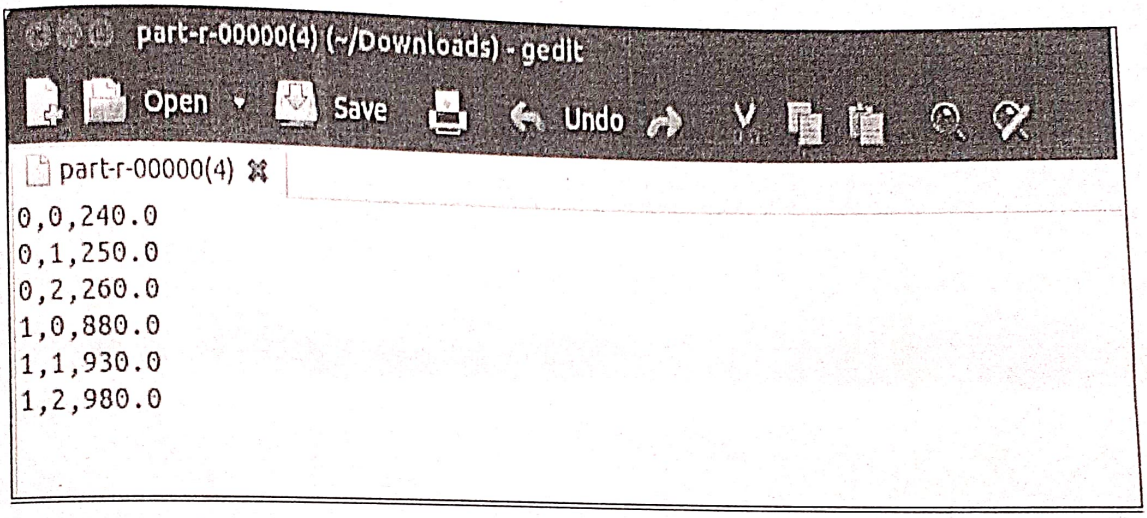
Reduce(key,valueList)

17. ifkeyis(ib, kb,jb, 0)
18. //Savethe Ablock.
19. sib=ib
20. skb= kb
21. ZeromatrixA
22. foreachvalue=(i,k, v)invalueListA(i,k)=v
23. ifkeyis(ib, kb,jb, 1)
24. ifib!=sibor kb!=skbreturn//A[ib,kb]mustbe zero!
25. //BuildtheB block.
26. ZeromatrixB
27. foreachvalue=(k,j,v)invalueListB(k,j)=v
28. //Multiplytheblocksandemitthe result.
29. ibase=ib*IB
30. jbase=jb*JB
31. for0<=i< rowdimensionofA
32. for0<=j< column dimensionofB
33. sum= 0
34. for 0<=k<columndimensionofA= rowdimensionofB
a.sum+=A(i,k)*B(k,j)
35. ifsum!=0emit(ibase+i,jbase+j),sum

SetofDatasetsoverdifferentClustersaretakenasRowsandColumns

INPUT/OUTPUT:





PRE-LABVIVAQUESTIONS:

1. Explainwhatis“map”andwhatis“reducer”inHadoop?
2. Mentionwhatdaemonsrunonamasternodeandslavenodes?
3. MentionwhatistheuseofContextObject?

LAB ASSIGNMENT:

1. ImplementmatrixadditionwithHadoopMap Reduce.

POST-LABVIVA QUESTIONS:

1. Whatispartitioner inHadoop?
2. ExplainofRecordReaderinHadoop?