Cloud Computing

Project 8

By:

Saloni Sharma(sharma36)

Arpitha N Kashyap(arpkashy)

We use the following algorithm for implementing mini batch k-Means in project 8.



We first need to get the output of all the Mappers and then use “allReduce” method (which was explained in the lab) on it. We cannot directly implement this algorithm without doing so. The allReduce operation is applied on the centroids as well as v[c]. After this, the centroids are updated. Hence we use the algorithm for Mini-batch k-means with allReduce which is as follows:



# Steps for MBKmeans using AllReduce method:

Step 1: Loading data from HDFS to the mapper.

Step 2: Randomly initialize Centroids C0 .

Step 3: Broadcast the centroids to all the mappers after initializing.

Step 4: Select a batch of data points (Mi) from the data on all the mappers at random.

Step 5: Locate the closest centroids for data in Mi. Update Ci and Vi for each of the mappers.

Step 6: Perform allReduce task for Ci and then for Vi.

Step 7: Calculate Ci and Vi based on Ci-1 and Vi-1.

Step 8: Repeat the above steps for each iteration. Output Ci

# Compilation Steps:

1. cd $HARP\_ROOT\_DIR
2. mvn clean package
3. cd $HARP\_ROOT\_DIR/harp-tutorial-app
4. cp target/harp-tutorial-app-1.0-SNAPSHOT.jar $HADOOP\_HOME
5. cd $HADOOP\_HOME

# Run:

cd $HADOOP\_HOME

# Usage:

hadoop jar harp-tutorial-app-1.0-SNAPSHOT.jar edu.iu.mbkmeans.common.KmeansMapCollective <number of data points> <number of Centroids> <size of vector> <batch size> <number of map tasks> <number of iteration> <workDir> <localDir> <communication operation>

<number of data points >: the number of data points you want to generate randomly

<number of centroids>: the number of centroids you want to clustering the data to

<size of vector>: the number of dimension of the data

<batch size>: size of the batch to be randomly generated from the data points

<number of map tasks>: number of map tasks

<number of iteration>: the number of iterations to run

<work dir>: the root directory for this running in HDFS

<local dir>: the harp mbkmeans will firstly generate files which contain data points to local directory. Set this argument to determine the local directory.

<communication operation> includes: allreduce

# Example Run:

hadoop jar harp-tutorial-app-1.0-SNAPSHOT.jar edu.iu.mbkmeans.common.MBKmeansMapCollective 1000 10 10 100 2 10 /mbkmeans /tmp/mbkmeans allreduce

# Fetching Results:

hdfs dfs -ls /

hdfs dfs -cat /mbkmeans/centroids/\*

# Output:

3.2154281021546804 9.414672896953672 5.3936958930835015 5.455695872479432 6.210653155320154 0.5647244860527789 3.405481207218868 5.392105798470621 6.042415633481573 6.792740149492927

0.8990528534952813 8.191576953332149 6.888623607006787 3.9059369554877943 5.354388192723261 1.1109915688223726 5.319159086911601 0.6604088084390625 8.691273489332861 0.275848540299084

4.389242828716659 9.456428085249703 7.126225226494257 1.2485037323960058 4.874000478863407 8.329383651559905 3.89219688661138 8.924422417751959 5.6562497766045094 6.2801107517973165

1.4473563289628044 5.797277257468723 7.086177224578433 8.68564725830401 8.012103712867209 4.987237378047226 8.250705532371535 2.0412048945170893 1.7988937325863685 4.592817668296682

1.5220015117023489 7.606416826629064 5.1373169773297604 2.9051609696776026 5.498707859086469 1.663685986119826 9.847280972993815 9.814654515245502 0.510126286895326 6.043724606044833

4.6760309751421625 8.343285866917766 9.222723176784148 6.333140256635366 5.390132274748955 1.6521689017911156 8.569959664546229 6.751189683813931 8.20143019676619 5.102178660695623

8.220601287316011 0.06759047776722205 7.606396811522945 9.986654898405577 5.107292882174149 8.130538616705344 0.43115987194810135 5.408645464184364 3.156330913610794 3.5838546615059466

8.8999024506271 7.441180075013341 2.2912021959820383 6.174440633587196 6.882212239060786 0.18757712473564503 1.0266027984301873 7.685997381943891 5.125349275329823 1.649805354561641

2.526286488384586 7.982218254014936 3.240291475797583 8.41643493387628 4.867238949318357 0.9024398739862227 8.485810251624073 7.097728748838232 2.434306121875738 2.8249964403982766

3.98271564508609 7.163942234691987 6.863555296370726 8.598289884499568 1.8908032978987677 9.383913402597555 2.759281756319937 9.236853587594632 0.29392522321407655 6.414540955675401