Anthony Rezzonico
CS 455
19 April 2019

HW3-WC

1) Assume that you have complete control to reorganize and distribute the dataset. What is the fewest number of MapReduce jobs that you would need to implement the complete set of tasks in HW3-PC? Note that your program should emit outputs for each task into a separate file.[300-400 words]

The fewest number of jobs that it would take to implement question one through ten(depending on what the question was for ten) could be completed in one job. I was also able to write out the questions to seperate files using the multiple output class that allowed me to define named outputs for the different questions. While given the opportunity to reorganize the data, I still would have left it as is. I know a good portion of the class wanted the data set merged together, where I thought that because the data was broken up more it made it easier to refine on only the necessary fields that I need. For this assignment, I looked at some common MapReduce design patterns from lecture slides provided in the Big Data course and used the technique of binning my data. Since I have two mappers for the different data sets, I would have them both emit a similar key like the question number. For the value I would prefix the text string with either an A or a M to represent Analysis or Metadata so my reducer could appropriately handle what data it was receiving and operating on. I would then iterate over all of Text values that the reducer received and would put the data from the respective files into HashMaps that would operate on a shared key, whether this was an artists ID or a song ID. By saving the HashMaps at the reducer's class level, I was able to leverage the cleanup function and was able to answer questions four, five, and eight by using the results of previous questions without having to re-transmit any new data. The only thing that I might of changed would of been to create a small data set that contains the artist id, name, song Id and name that I could save in the distributed cache that would of helped with partial joins on the data without having to transmit minor things from both reducers to figure out something like the songs name.

Q2. You are designing a new streaming service that can scale to millions of users and stream songs that are personalized to a user. How would you extend assignments HW1-PC, HW2-PC, and HW3-PC to accomplish this goal?[400-500 words]

My approach for this design would be to create a central server that allows users to register and subscribe to the service. Upon registration that user will be given a regular socket on a server that resides based upon the users location. This node will be what extends HW2 in the sense of how it processes micro batches using a thread pool. When a user interacts with the service, their request will be sent to the server that they were assigned to by region and will be handled by an available worker thread that will: Retrieve the song that was requested and update a file that is unique to each of the users. For the worker thread to retrieve the song that the user requested, this would be where I would expand and implement the peer to peer network from HW1. The user will clearly want to stream the song at an expected rate, so using HW1 to provide the shortest path to retrieve the song makes the most sense to do. Not only would this peer to peer network be used for the retrieval of the song, but in addition this could be used for when the regional servers need to share trends with other servers that are not connected to each other. When the worker thread has finished retrieving the song, the thread will extract the metadata about the song and log this to the user specific file. This is important for the thread to log because later down the road, the user might want to see what music the service might suggest the user will like. With this design, the implementation will be able to direct the user to songs that they might like. At the end of each day, the central server could issue out a command to all of the regional servers to pool there records for the users together and submit a job to the Hadoop cluster to then analyze the patterns of the users and determine songs based upon the individual entries. By doing this massive job at one set time every so often, it prevents multiple jobs being submitted to the cluster and causing a considerable slow down in the system. Since these jobs are done incrementally, this still allows the service to suggest songs to the user and based upon the responses from the suggested list, the personal logs can be revised to further improve the experience.

Q3. What if a family has a shared account on your streaming service? How would your streaming service recommend songs so that each family member is satisfied? Please design your own measure for satisfaction.[300-400 words]

If I were to have a family use a shared account, the service would have to perform analysis at a finer grain level. The application would need to not only record metadata about the song itself, but also record information about what time was the song played at in the day and the geographical location of the request submitted by the family member. By getting the time and location, the system could guess who is using the service at that time. An example of this would be one of the family members goes to the gym at 6 a.m. Monday through Friday and listens to a considerable amount of electronic music at that time. Another family member listens to classical music around noon at the house to help soothe a baby to sleep. The family member at the gym would not want to listen to classical music during their workout, nor would the family member want to play the electronic music in fear of of disturbing the baby. By adding this level of classification to the requests, more songs that are similar can be suggested based upon the average types of songs that are played at that location and time. By grouping the request by location and time in conjunction with the metadata about the song provide classifications that can help differentiate who is listening to the music based upon the trends.
In order to measure satisfaction with the results, the service would consistently need to be analyzing how often song requests are being submitted at that time and if it the suggested songs are being played all the way through or are consistently being skipped through. If the service detects that it is receiving less requests for songs and the user is constantly skipping the suggested songs, then it can be said that the service is not correctly classifying the family member in which the application thought it was targeting.

Q4. You are starting a music production company and you are working with local artists in town. How would you perform micro-adjustments to a song so that it is a commercial success in different countries? How would you avoid concept drifts - what is in vogue today may not be in the near future? [300-400 words]

In order to make a local artist commercially successful in different regions would require different levels of analysis on the target region and micro adjustments on the artist songs. In order to project commercial success in different regions, the company would need to determine the interests of the region that resonate with the community the most. In order to avoid concept drifts with artist and their reach on the different regions, the artist will need to modify the songs in micro adjustments so that the interests of the region are within a tolerance window without changing the actual content of the song. So an example of this would be that an artist is trying to become popular in Europe and the artist has just produced an album that is successful in the

United States. The company could see that within Europe, the general population responds more positively to songs that have a higher values for beats per minute. So the company could modify the produced album so that the original music has modified values for songs that did not meet the beats per minute measure. Another possible way of promoting the artist to different regions would be to again find the average interests of the region and then cater songs from the artist that match those criteria to that region. In this case it is hard to find a model where one style that fits all, so by showcasing the songs that do pertain to the region's style, the artist can start to gain familiarity within the group. Once the artist has enough tracks that the audience likes, then the region could possibly adapt their preferences that accommodate the style of the artist.

Q5. The kind of music a person likes may change/evolve over time. Describe a potential scheme to design "interest trajectories" that allows you to recommend songs that a person may like in the future.[300-400 words]

To design this "interest trajectories" the scheme would require having information about the user's gender, location, and age to project songs that a user might like in the future. Much like what I was suggesting with question 3, location plays a large role what the user is interested in and what might be relevant to the group. Certain trends like music genres can pass down to other regions, just like with societal trends that originate in the west coast can move its influence to the east coast. The age and gender of the user can also play a large factor into the relevant music that they might be interested in. So by using these fields to group and categorize the users into relevant groups, the application could then project what the interest of that group might like to listen to. Since interests will vary among the different generations, this model will need to constantly adapt as the relevant user groups and their interests change. In order to predict future interests, the application will need to analyze the trends with older generations to see how they shaped over time. This could be that an older group used to be really interested in grunge music, but as they grew older they shifted their interests towards rock and roll. With this shift, the design would pick out the elements that remained consistent with the interest shift and would apply this to the genre that the current generation is involved in. Some of the elements that would be considered would include the subject matter of the song, the instruments used, timbre and pitch, beats per minute, how loud the song is, and overall energy levels of the song. Using these metrics can help predict future songs even as the user evolves their interest over time.