final review

wbg231

January 2023

1 RDMS

review

- relational model (table and data frames) standardizes data organization
- schema constrains simplify making sure data is formatted correctly
- tables can speed up common access patterns
- sql is good
- constraining data organization makes high level operations easier

question

- none :(

map reduce

- map takes input outputs key value pairs
- reducer takes key values pairs and aggregates them on key value
- combiners are optional but work work like reducers within map nodes means less data shuffling
- constraining the form of computation makes parallelism easier

question

- Combiners in map-reduce improve efficiency by decreasing the number of intermediate keys prior to the reduce phase.

- false combiners preserve the key structure but produce fewer values they fix key skew
- map function in map reduce must produce at least one intermediate output
- false

HDFS

- data nodes store partitions of large data block
- name nodes store a map of file names/and the mapping of blocks to data nodes (that is what data node has what block of data)
- data is read and append only. this makes concurrent access easier
- Replication factor more copies of each block up front means less computation and shuffling later on improves locality
- when the name node fails spark fails it is not acid

spark

- RDD are the main abstraction
- transformations take RDD to RDD and actions take RDD to results
- lineage graphs represent overall computation
- provides a higher level interface for distributed compting than map reduce

question

- compare a RDD vs a data frame
- a RDD is closer to a list it is really fast with filters and maps
- both have lazy execution
- a dataframe is made up of RDDs good for more complex analysis
- data frames in spark are read only
- true the values of a spark data frame can not be modfied in place
- each step in an RDD lineage graph computation must complete before starting the next. false that is one of the reasons we have lineage graphs
- spark uses piellines to conect multiple map reduce programs
- false spark does not use map reduce

column oriented storage

- organizing data by columns makes things fater
- we are constraiing data types so faster memory acess
- data compresion makes faster comuncaiton
- dremel is used for taking nested structed documents to tabular representations
- tabels = coloumns = compression
- parquet is the defualt for park
- parquet can be faster than spark

questions

- explain parquet and dremel
- The Dremel system was designed to efficiently process all attributes for subsets of records in a dataset
- false that is the opesite of what it does
- When written to HDFS, Parquet files locate different columns in different HDFS blocks.
- false parquet files devide blocks by col

dask

- like spark but different
- works with the scipy stack
- integrated into python directly so more flexible with difrent types of data questions
- spakr is great when your data looks like dataframes
- when your data is not data frams dask is good
- but more flexibility means less automatic optimziation
- spark has bags that will hold it all

questions

- dask is entirely python, spark is slowed down a lot by going in and out of python
- dask is less polished
- dsak is suited for working with python packages that already exist (also can wrok better on a single machine)
- spark is better for badnas like things but dask uses pandas directly
- HOW TO OPTIMIZE: in SQL, Spark and Dask, i.e indices (types and uses), partitions, when to use RDDs/Data frames. Also, when to use each of the frameworks.
- use filters as quickly as posisble to cut down your data
- understand your lineage graph
- optimize partiiton strcuture to avoid wide dependinceis

similarity search

- minhas, represent each set in a colection by the smallest ha outputs of its ellements
- probailtiy that sets colide is the jaccard similairt
- use multiple hashes to estimate jacard similarity
- instead of paritioning us a hash function that is imperfect if there are collsiosn thsoe are candidte pairs do more similiary search on those
- combine multiple min hash outputs togther as a block
- block size + number of blocks can be used to boost likelyhood of collison
- idea generlies to other distance metrics

questions

- what is mutlip probe lsh
- multi probe lsh is insead of looking at ecact mathces for each block look at ner by blocks as well
- spatial trees just devide hte space uing a tree name explains it well
- why is lsh more efficient
- we are not partiioning and lowering the number of candite pair

quiz questions

- min hash fails when a single element bellongs to all sets in a collection
- true if any hash picks this item as a minimzer than all sets coldies

reproducibility

- reproducability rellyimportant in big tdata
- teher are lall kinds of best pactices
- rpiecjt folder standaried
- keep contetual information in reamdem
- neseative data kept in secured reposity

recomender systems

- idea predict which items a user will ineract with
- method popularity mdoel + dampening factor
- latnet factor model mdoel interactions as iner product of two vecors
- implicit vs explicit feedback

search , ranking evaluation

- can use min hash adn lsh to identfy similar documnetns
- page rank orders documetns by tehre porability of capture a random walk
- core compation is learing eigenvaect, and transiton matrix
- cna use power iteration

difenetial privacy

- being able to release data is criticla for reproducability but needs to be balancedd
- k annonomity is not enough
- de anomized atatack highly acrute

- difenetial pricay works throgu han api
- add lapplacian nosie to give plase dinaiablilty
- sensativity maxmial difrnet in output given a single row difrent extmeral agreates high slensiv each
- multiple queries reduce dp

gpus

- computaiton parllelism dedicated hard ware
- more restricite prpgram control flow
- but less resitricte in terms of data acess
- limited sharing of inomration betwen computaiton threads
-