Week 2

- Throughput: the amount of task completed given the time period
- Response time: the amount of time to complete taks
- Speed up: The time decrease by adding more resourses
- Scale up: The transaction size increase by adding more resourses
  - Transaction scale up: the increase in the rate at which the transactions are processed.
  - Data scale up:the increase in size of database.

Parallel obstacles

- Star up: Start up cost is associated with initiating multiple processes.
- consolidation costs: Consolidation cost refers to the cost associated with collecting results obtained from each processor by a host processor.
- Interference and Communication: Since processes executing in a parallel system often access shared resources, or one process may have to communicate with other processes. It will reduce overlap waiting time.
- Skew: Skew in parallel database processing refers to the unevenness of workload partitioning.

Parallelism form

- Interquery parallelism
  Interquery parallelism is parallelism among the queries.
- Intraquery parallelism
  Intraquery is parallelism within a query.
- Intraoperation parallelism
  speeding up the processing of query by parallelzing the excution of each individual operation.
- Interoperation parallelism
  (*i*) *pipelined parallelism* and (*ii*) *independent parallelism*.
  speeding up the processing of query by executing in parallel different operation.

Parallel database structure

- Shared-Memory and Shared-Disk Architectures
- Shared-Nothing Architecture
- Shared-Something Architecture

Analytical Models

Operations in parallel database systems normally follow these steps:

- Disk cost
  $* R_i / P \times IO$
- Main memory cost (select, result generation, data computation, data distribution)
  $| * R_i | \times t_x$

Parallel Search

Search queries

- Exact-match search
- Range search
- Multiattribute search

Data partitioning

Basic data partitioning

- Round-robin data partitioning
  each record in turn is allocated to a processing element in a clockwise manner
- Hash data partitoning
  Each record is allocated based on hash function
- Range data partioning
  spread the record based on given range of the partionning attribute
- Random-unequal data partitioning
  partionning function is unkonwn

Complex data partitioning

basic data partitioning is based on a single attribute, while complex data partitioning is based on multiple attributes.

- Hybrid-range partitioning strategy (HRPS)
  1) Range partitioning data into fragments;
  2) Round-robin data in fragments.
- Multiattribute grid declusterning (MAGIC)
- Bubba's Extended Range Declustering (BERB)
  1) Range partitioning on primary partioning attribute;
  2) Scan each fragment , create 'aux' table;
  3) Range partioning 'aux' table;
  4) combine result from 1) and 3)

Search algorithms

Serial search algorithms:

- linear search
  scanning cost: $\frac{1}{2} \times R/P \times IO$
  select cost: $\frac{1}{2} \times |R| \times (t_r + t_w)$
  Comparing cost: $\frac{1}{2} \times |R| \times t_c$
  result generation cost: $\sigma \times |R| \times t_w$

Disk write cost: $\sigma \times R/P \times IO$

- Binary search
  scanning cost: $log_2(R)/P \times IO$
  select cost: $log_2(|R|) \times (t_r + t_w)$
  Comparing cost: $log_2(|R|) \times t_c$
  result generation cost: $\sigma \times |R| \times t_w$
  Disk write cost: $\sigma \times R/P \times IO$

Parallel search algorithms

- Processor activation or inovation
  Hash for Discrete Range Selection: Selected processor involved
  Range for all Range Selection: Selected processor involved
- local searching method
  Ordered: Binary Search
  Unordered: Linear Search
- key comparision
  Only Exact Match for unique values can stop

Week 3

Parallel join

serial join algorithms

The complexity of the three join algorithms as discussed above is as follows:

- Nested-loop join algorithm O(NM)
- Sort-merge join algorithm O(NlogN+MlogM+N+M)
- Hash-based join algorithm O(N+M)

Parallel join algorithm

Divide and broadcast

"Divide and broadcast"-based parallel join algorithms are composed of two stages: data partitioning using the divide and broadcast method and a local join.

No load imbalance problems;

No broacast in share-memory structure.

cost model

- Sacan data from disk to memory as disk block(page)
  scanning cost: $S_i/P \times IO$
- get record out of page
  select cost: $|S_i| \times (t_r + t_w)$
- Data broadcasting
  Data transfer cost: $(S_i/P) \times (N-1) \times (m_p + m_l) = (S/P - S_i/P) \times (m_p + m_l)$
  Receiving cost: $(S/P - S_i/P) \times m_p$
- Disk cost for sorting table
  Disk write cost: $(S/P - S_i/P) \times IO$

Disjoint data partitioning

Disjoint partitioning-based parallel join algorithms also consist of two stages: a data partitioning stage using a disjoint partitioning and a local join.

cost model

- Sacan data from disk to memory as disk block(page)
  scanning cost: $(R_i/P + S_i/P) \times IO$
- get record out of page
  select cost: $(|R_i| + |S_i|) \times (t_r + t_w)$
- Data partioning
  Finding destination cost: $(|R_i| + |S_i|) \times t_d$
  Data transfer cost: $(R_i/P + S_i/P) \times (m_p + m_l)$
  Receiving cost: $(R_i/P + S_i/P) \times m_p$
- Disk cost for sorting table
  Disk write cost: $(R_i/P + S_i/P) \times IO$

Cost for local join

Scan cost: $(R_i/P + S_i/P) \times IO$

Select cost: $(|R_i| + |S_i|) \times (t_r + t_w)$

Join cost: $(|R_i| \times (t_r + t_h) + |S_i| \times (t_r + t_h + t_j))$

if main memory not enough for entire hash table, over flow buckets cost is incurred:

$$\left(1 - min\left(\frac{H}{|S_i|}, 1\right)\right) \times \left(\frac{S_i}{P} \times 2 \times IO\right)$$

generating result cost: $|R_i| \times \sigma_j \times |S_i| \times t_w$

Disk cost for sorting result: $(\pi_R \times R_i \times \sigma_j \times \pi_S \times S_i/P) \times IO$

Parallel outer join

- ROJA (redistribution outer join algorithms)
  1)redistribute;
  2)local outer join

  based disjoint partitoning.
- DOJA(duplication outer join algorithms)
  1)replicate;
  2)inner join;
  3)hash and redistribute;
  4)outer join.

  based on divide and bradcast
- DER(duplication & efficient redistribution)
  1)replicate;

2)inner join;
3)hash the ROW id and redistribute;
4)outer join.

when it comes join with more than two tables, say R,S,T
first redistribute first two R,S
then outer join R,S as J
redistribute J,T based on join attributes
outer join J,T

- OJSO(outer join skew optimization)
  1)redistribute R,S;
  2)outer join R,S, store the result into Jredis and Jocal;
  3)redistribute Jredis and T;
  4)union the final.

Week 4

Internal sorting

- Bubble
- Quick

External Sorting

Phase 0: divide the

Parallel External Sort

- Parallel Merge-All Sort
  1)local sort;
  2)final merge.
- Parallel Binary-Merge Sort
  1)local sort;
  2)binary merge.

  Binary merge vs k-way merge
- Parallel Redistribution Binary-Merge Sort
  1)local sort;
  2)redistribution;
  3)binary merge;
  4)redistribution;
  5)final merge.
- Parallel Redistribution Merge-All Sort
  1)local sort;
  2)redistribution;
  3)final merge.
- Parallel Partitioned Sort
  1)redistribution;
  2)local sort.

Serial GroupBy Processing

Read record from disk into memory;

if memory is not enough, hash data partioning base on attribute

Hash record into hash table in memory;

Store hash table as query result in disk

Parallel GroupBy

- Traditional methods (Merge-All and Hierarchical Merging)
  1)local aggregate ;
  2)global aggregation.
- Two-phase method
  1)local aggregate ;
  2)redistribution;
  2)global aggregation.
- Redistribution method
  1)redistribute(task stealing);
  2)global aggregation.

Week5

key concepts

- model: a specification of mathematical relationship btw different variables
- machine learning: modifying and implementing models that learnt from data
- bias: Difference between predict value and actual value
- variance: Difference between predictive value with others.
- how to prevent overfitting
  - train more data
  - remove features
  - early stopping
  - cross validation
- precision: TP/(TP+FP)
  focus on positive prediction, among the postive prediction, how many of them are true postive
- recall: TP/(TP+FN)
  focus on real postive, how many postive is predicted right among all the real positive.
- F1: harmonic mean of precsion and recall.

Types of machine learning:

- Supervised
  The input data has associate label
  - Classification: Binary and multinomial logistic regression, **decision tree, gradient boosted tree, random forest**, naive Bayes, support vector machine.
  - regression
    Linear regression, **decision tree, gradient boosted tree, random forest**.
- Unsupervised
  - Clustering
    K-means

- Association

Featurization

feature extractors

- count vectorizer
  word count problmes
- term frequency-inverse document frequency (TF-IDF)

  log based on 2

- Word2Vec
  using vector to calculate the similarity
- tokenization
  stopwords, appears requently but no meaning

feature transformers

- string index
  categorical -> numerical
  issues: assume nature ordering
- one hot enconding
  categorical -> binary array

Week6

Classification techniques

- Decison tree
- K-nearest neighbours
- Random forest
- Naive bayes
- support vecotr machine

Decison tree

Concepts

root node, leaf/terminal node(do not split), decison node(split into further sub-nodes)

Splitting: divide nodes into sub-nodes

Pruning: remove sub-nodes of descion nodes

ID3(interative dichotomiser 3)

1. Compute the entropy for data set
2. For every attribute/feature:
   2.1. Calculate entropy for all categorical values
   2.2 Take average information entropy for the current attribute
   2.3 Calculate gain for the current attribute
3. Pick the highest gain attribute
4. Repeat until the tree is complete

Advantage vs disadvantage

advantage

- Easy to understand/generate rules
- Less hyper-parameter
- Visualisation

disadvantage

- overfitting
- poor for non-numerical
- Low prediction accuracy
- Complex when much label

Random forest

1. consists of many decision trees.
2. vote final predictions

Optimisations

- Bagging: Bootstrap aggregating is a method that result in low variance.
  Rather than training each tree on all the inputs in the training set (producing multiple identical trees),
  each tree is trained on different set of sample data
- Gradient boosting: selecting best classifiers to improve prediction accuracy with each new tree.
  It works by combining several weak learners (typically high bias, low variance models) to produce an overall strong

Advantages and Disadvantages

Advantages

- Robust to correlated predictors
- Both regression and classification
- Unsupervised ML problems
- No variable selection
- As feature selection tool
- Take care missing data

Disadvantages

- Difficult to interpret
- Erratic predictions for observations out of range
- Longer time

Parallel classification

Data parallel: vertial data partioning

result parallel: horizontal data partitioning

Week 7

K-means

- K-means is a partitional clustering algorithm
- The k-means algorithm partitions the given data into k clusters.
- Each cluster has a cluster center, called centroid.
- k is specified by the user

Algorithm k-Means:

- Specifies k number of clusters, and guesses the k seed cluster centroid
- Iteratively looks at each data point and assigns it to the closest centroid,
  current clusters may receive or loose their members.
- Each cluster must re-calculate the mean (centroid)
- The process is repeated until the clusters are stable (no change of members)

The K-means Notice

- The number of clusters k is predefined.
- The final composition of clusters is very sensitive to the choice of initial centroid values.
- Pros
  - Simple and fast for low dimensional data
  - Scales to large data sets
  - Easily adapts to new data points
- Cons
  - It will not identify outliers
  - Restricted to data which has the notion of a centre (centroid)

Data Parallelism of k-means

Data parallel: each processor classify all clusters then combine in final stage

Result parallel: each processor classify only one cluster

week8

recommendation system

- content based
  requries sufficient amount of items (feature)
- collaborative filtering
  use previous user input/behaviours to make futhure recommendation

collaborative filtering

why?

- it benefits from large user bases
- flexible with diff domain
- produce level of recommendation
- capture nuance

Week 10

Overview of streaming join

- Nested-loop stream join
- Sorted-merge join
- hash join & symmetric hash join

  Problem for hash join: miss match (if hash Si, if r come later than its pair s, then r and s will not pair)

unbounded stream join

- Tuple-based window stream join
- Time-based window stream join
- handshake join
  soultion to fix missing match:
  - alternating tuples must be left empty in the stream.
  - handshake twice with adjcent then move

bounded stream join

- nested-loop join

  m-way join: first join R,S into RS, then join RS,T
- symetric hash join
- Mjoin: Probe hash table then hash
- Amjoin: probe Bit-vector hash table then update Bit-vector, last hash

Week 11

Granularity reduction in data stream

granularity is the level of details at which data are store in the database.

garanularity is not only of retirval efficency but also about managing complexity.

Reduce granularity from hourly to daily, then to weekly, it is easier to see the trend. Identify trend fron streaming data is very important.

Fixed-size windows

Overlapped windows

Slide time is less than window size

- no granularity reduction
  when the time slide is one unit of time
- with granularity reduction
  when the time slide is more than one unit of time

Non-overlapped windows

- with gramularity reduction
  consecitive windows are not overlapped (no gap between the windows)

Mixed-level of granularity

temporal-based

spatial-based

Sensor Array

a sensor array is a group of sensors, usually developed ina certain geometry pattern.

- multiple sensors measuring the same things (MSMST)
  - Why:
    - specialize on sensing a very specific samll region
    - Get more accuracy of the results
  - How:
    - reduce and the merge
    - merge and then reduce
- multiple sensors measuring the different things (MSMDT)
  - How:
    - reduce, normalize, and then merge
    - Normalised, merge and then reduce