Handle Large Data
Meeting Notes: Nov 22 2025

Algorithm (Divide-N-Conquer)

To speculate or inspect large data size date file.

Step 1: Python to read the file cyclic. (do not write to mongo, but cut the file into each 10M, I have 10K files).

Step 2: Python and extract the file, and feed into Mongo, and you get a clear lines of errors, and 10K files, I feed them into mongo, then I know which files will fail.

1K failed files, I pick 1 of them, just peek at it.

ETL - Extract() - Transform() - Load()

1/ Identify the lines, problems, linear (line 100 we have an issue, line 100010 another shows only fixes)
2/ Data Cleaning

* Spark framework (divide the dataset into small piece)

1/ Data structure: Computer internal storage related proven algorithms to solve data storage and indexing.

2/ Algorithm: To express your idea to fix the problem, step-by-step.

Note:

* Storm framework.
* Based on zoo-keeper opensource distributed computing framework and Hadoop DFS.

Mongo vs Relational Database

* Has foreign keys and constraints, it was designed perfectly for large system and monolithic architecture.

* The relational db downside is that, with cloud, we prefer sometimes the microservice, so we literally have smaller components that only requires single table.

* Mongo came out as single smaller tables without relations + JSON.
Single tables, without any keys, their server instance is bulky/big and license is expensive.

* Today, I think MySQL does have a different engine to handle JSON and compete with Mongo.

Poll system, Recommendation Engine or a voting system.

RSS feeders.

If I have two relational database systems, let us say,

1/ I have an IBM DB2 dataset

2/ I have another database called Oracle

3/ I have another database called MySQL,

Q & A

Q: How do I join the data from all of these using the relational database concept?

A: No way, no key to join. You join it on the application level. Join the data using your Java or Python Array, List, Queue or Stack. Eventually we curated the data or combine data to form a new set of data.

Then what?

Save the new data into one of these relational database, maybe in a new data schemas or tables.

This is similar to be called psuedo union.

Q: When do I need the above thing? And Why?

A: One example, we have a lot of user authentication db or tables in various of database systems, let us Oracle and MySQL, so we use another new authentication way, and I would like to combine these silos of data into one good piece of user data.

1/ Authenatication: username + password
2/ Authentication: google gmail, Oauth

This is very normal. In DB1 you might have user's email1, and second db you have user's phone and address. Now we want to combine them all together, then question is how do identity they are the same user, and combine the info together.

Sometimes, the real problem is not just authentication, but customer data.

In healthcare, hospital/clinics, a lot of times, they register patients with all sorts of funny information, and it is the same person. Sometimes, they would have a urgent need to tell how many patients I have or have visits during some time, and they would like consolidate their data.
* Data Consolidation is a one of the missions/goals in large processing.

Q: HW3 extension. (Note)

Attempts unlimited.