

Data Management

CMPT 732, Fall 2018

The V's

Remember the “[The Four V's](#)”?

- Volume: the amount of data.
- Velocity: the data arrives quickly and constantly.
- Variety: many differently-structured (or less-structured) input data sets.
- Veracity: some data might be incorrect, or of unknown correctness.

We have spent most of our time on *volume*. What about the others?

Velocity will be addressed by Spark Streaming.

Veracity (a.k.a. *correctness*) is generally solved with *data cleaning*... more coverage in CMPT 733.

What about *variety*? Let's get there...

OLAP vs OLTP

Most traditional databases are focussed on *OLTP* (OnLine Transaction Processing) work: processing queries quickly to back a web site or similar.

In contrast, *OLAP* (OnLine Analytical Processing) focusses on less-real-time analytics: reporting, business intelligence, data mining, etc.

The categories aren't mutually exclusive: the same database server *could* do both.

But complex reporting queries can slow down transactions, so it might also be reasonable to separate them. Easy solution: master database for OLTP and read-only replica for OLAP.

In this course, we are (I am?) generally thinking of OLAP-like workloads. “Big data” can be the result of aggregating smaller OLTP systems and doing analysis.

Extract-Transform-Load

Sometimes, the data you get isn't in the format you want.

e.g. schema for OLTP might not be what you need for OLAP: maybe it makes sense to denormalize or aggregate before doing any OLAP work.

e.g. data from multiple sources need to be combined for analysis.

e.g. data arrived in a slow-to-process format (like monolithic `.json.bz2`) but you want something faster (like partitioned `.parquet.lz4`).

e.g. as above, but it makes sense to load into Cassandra before further processing.

Generally, the solution is to take the format you have, transform to the format you need, and save.

Or *Extract-Transform-Load*, ETL.

Can also include extracting, validating, cleaning, joining, aggregating,

ETL can be done with tools you know: Spark, Spark Streaming, just programming.

There are also dedicated tools to define data processing steps: [Apache Sqoop](#), [Apache Flume](#), [Amazon Data Pipeline](#),

Data Warehousing

It's common for any organization to have many places data is stored: SQL database for web site, SQL database for HR system, spreadsheet with budgets,

For analysis, it probably needs to all be in one system: a *data warehouse*.

This addresses at least one kind of data *variety*.

The idea: take (all of the) OLTP system(s) and other data sources and ETL to get them all into one OLAP system.

Call that a data warehouse. Use it for analytics, reporting, etc.

e.g. SFU is currently working on a data warehouse. Possible inputs: databases from goSFU, the financial system, the HR system, computer accounts, ID photos. If that existed, it would be much easier to answer questions about what's happening around here.

The data store for a warehouse could be anything that makes sense: SQL database, NoSQL database, HDFS+Spark, dedicated warehouse tool.

Some data warehousing tools: [Amazon RedShift](#), [Google BigQuery](#), [Teradata](#).

It might not actually be necessary to copy the data into one system: maybe it can be queried in-place.

There are several tools to query data from different sources: [Spark SQL](#), [Hive](#), [Apache Impala](#), [Apache Drill](#).

When importing data into a warehouse, it may be useful to reshape it:

- denormalize or re-normalize for easier querying
- add indices that aren't necessary in OLTP
- unify identifiers ([entity resolution/record linkage](#))
- keep history of previous values
-

Make sure any data import task is *idempotent*.

It's going to fail one day: make sure it can be re-started safely and heal itself. i.e. never blindly insert data. Begin transaction; if not present, insert; commit.

... then use the warehouse to answer some questions.