Programming for Big Data I

CMPT 732, Fall 2020

Greg Baker & Steven Bergner

https://coursys.sfu.ca/2020fa-cmpt-732-g1/pages/

Us

Instructors:

- Greg Baker, University Lecturer
- Steven Bergner, University Research Associate

TA:

- Nishit Angrish, previous-cohort Big Data Student

The Plan

Welcome to online-only instruction.

We will be making some changes to the course to adapt to online delivery, but not changing the course material in any significant way.

This Course

It's "Programming for Big Data I". So it's about:

1. programming,
2. big data.

What is Big Data?

A quote you have probably seen before:

> "Big data is like teenage sex: everyone talks about it, nobody really knows how to do it, everyone thinks everyone else is doing it, so everyone claims they are doing it…"
> [Dan Ariely]

It's a buzzword. But a useful one.

How big is "Big Data"?

Answer 1: Big enough that traditional techniques can't handle it.

The "traditional techniques" strawman is usually (1) highly structured data (2) in a relational database (3) on a single database server.

PostgreSQL can handle single tables up to 32 TB [*] (but maybe not as quickly as you'd like). Big data must be bigger than that?

Answer 2: Big enough that one computer can't store/process it.

"One computer" can mean dozens of cores, TBs of RAM and many TB of storage. So bigger than that?

“Big data” isn't always big.

Many describe with “The Four V's” (or 5 Vs' or 7 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.

Honestly, the term *big data* is often used to mean “modern data processing, 2013–2020 edition”.

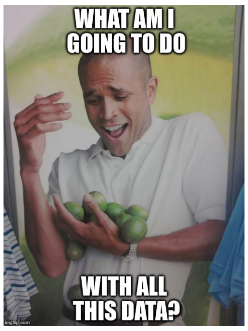
Even if most people don't work with truly-big data most of the time, it's nice to have the tools to do it when necessary.

Sometimes it's nice to know your computation can scale if a megabyte of data becomes many gigabytes.

Or maybe “can't be processed on one computer” should be “can't be processed in a time I'm willing to wait on one computer”.

An “overnight” calculation that doesn't complete until noon isn't very useful.

Greg's functional definition: people say they're doing “Big Data” when they have so much data, it's annoying.



Clusters

If our data is going to be too big for one computer, we presumably need many. Each one can store some of the data and do some of the processing and they can work together to generate “final” results.

This is a *cluster*.

Computer cluster

Actually managing work on a cluster sucks. You have all of the problems from an OS course (concurrency, interprocess communication, scheduling, ...) except *magnified* by being a distributed system (some computers fail, network latency, ...).

The MPI tools are often used to help with this, but are still very manual.

Do you want to worry about all that? Me neither. Obvious solution: let somebody else do it.

Hadoop

We will be using (mostly) the Apache Hadoop ecosystem for storing and processing data on a cluster. This includes:

- YARN: managing jobs in the cluster.
- HDFS: Hadoop Distributed File System, for storing data on the cluster's nodes.
- MapReduce: system to do parallel computation on YARN.
- Spark: another system to do computation on YARN (or elsewhere).
  ⋮

Our Environment

We have a cluster for use in this course: 6 nodes, each 16 cores and 110 GB memory. We will explore in the assignments.

Not a big cluster, but big enough we can exercise the Hadoop tools, and do calculations that aren't realistic for one computer.

In many ways, a two or three node cluster is enough to learn with: you have many of the same issues as 1000 nodes.

Things you will do

- Lectures (2 hours/week).
- Labs (4 hours/week).
- Assignments: complete on your own, with help in the labs/consultation times. 10 × 7%.
- Quizzes: online, during the lecture time. 3 × 3%.
- Project: a more open-ended big data analysis. 21%.

Lecture and Labs

Lecture: Mondays 13:30–15:20.

“Lab” hours: Wednesday + Friday, 9:30–10:30 + 14:30–15:30 (Vancouver time) in Zoom.

You can use your own computer for most of the work in the course. You can also access the workstations in the physical lab.

The compute cluster can be accessed remotely any time.

Assignments

The assignments will be most of the time in the course.

Work on the assignments as you like, and ~~come into the lab during your lab time to work and talk to us~~ talk to us during the lab/consultation times about the assignments/course.

Quizzes

[New for this offering.]

There will be three quizzes, tentatively:

- Oct 19 (lecture time, week 6)
- Nov 9 (lecture time, week 9)
- Nov 30 (lecture time, week 12)

They aren't worth a lot of marks, and their length will reflect that. (Tentatively, 30 minutes at the end of the lecture time.)

Course Topics

The current plan for the assignments is a good outline:

1. YARN, HDFS, MapReduce
2. MapReduce vs Spark
3. Spark RDDs
4. RDDs vs DataFrames
5. Spark DataFrames
6. DataFrames and ML
7. NoSQL and Cassandra
8. Spark + Cassandra
9. Spark Streaming
10. Hadoop reliability

The “express computation on a cluster” tools we'll see: MapReduce, Spark RDDs, Spark DataFrames.

There is a progression from “lower-level and more explicit” to “higher level and more declarative”.

Programming languages:

MapReduce will use **Java**: we will use it relatively little. Suggestion: don't bother setting up an IDE and go command-line-only for the few times you need it.

We'll use Spark with **Python**. Most of your programming in this course will be in Python.

Expectations

- The assignments and project are your chance to learn things, not a thing you have to do for marks.
- They are individual work: copying and understanding your friend's solutions isn't “your” work. [Academic Honesty]
- You are expected to be in the labs during your lab times, and working on the assignments.
- Everybody be nice to each other. [Code Of Conduct]