CS 378 – Big Data Programming

Lecture 1

Introduction

Class Logistics

- Class meets MW, 9:30 AM – 11:00 AM

- Office Hours – GDC 4.706
  - MW 11:00 – 12:00 AM
  - By appointment
  - Email: dfranke@cs.utexas.edu
  - Web page: cs.utexas.edu/~dfranke/courses/2015spring/cs378-BDP.htm

- TA: Swadhin Pradhan
  - Office hours:

Course Content

- Programming in Hadoop (map-reduce) and Spark
- Use ElasticMapReuce (EMR) on Amazon Web Services (AWS) initially
  - Hope to use the Hadoop cluster at TACC (if available)
  - Local install of Hadoop
- Looking into cloud based Spark cluster
  - From DataBricks
  - TACC is another possibility
  - Local install can also be used

Textbooks

- MapReduce Design Patterns
  - Main content for Hadoop assignments
- Hadoop The Definitive Guide – 3RD Edition
  - Recommended for your understanding, not required
- Learning Spark (early release)
  - Main content for Spark assignments

Lectures

- PDF of lecture notes accessible via syllabus
  - For your note taking, review, or whatever
- These notes are my outline for each class

Assignments

- Assignments will be programming assignments
  - All work can be done using Java
  - Scala might be an option

- IDE for developing code recommended
  - Eclipse, IntelliJ IDE (community edition) are free
  - Use maven to build “uber” JAR to upload to the cloud
  - I’ll provide the pom.xml file used by maven

Assignments

- I'll review a solution in class on the due date
  - Work submitted after the start of class considered late
  - 25% penalty for late submission
  - Can be submitted until the next assignment is due
  - After that deadline, no credit is given
    - Will consider these in determining final grade

- I encourage you to keep pace with the assignments
  - Most assignments will build on previous work

- Questions?

Learning from Data

- What can we do when the data gets big?
  - Too big for the CPU memory of any single machine
  - Larger than the disk storage of a single machine

- Recent data point:
  - Facebook has ~800 petabyte data cluster (Hadoop)
  - 1 petabyte = 10^{15} bytes

- Big data is spread across a network of machines

Learning from Big Data

- Need to bring distributed storage and distributed processing to bear to handle big data

- Issues:
  - Distributing computation across many machines
  - Maximizing performance
    - Minimize I/O to disk, minimize transfers across the network
  - Combining the results of distributed computation
  - Recovering from failures

Managing Big Data

- We'll look at two popular tools/systems

- One well established – Hadoop
- One up and coming – Spark

- Basic concepts of each
- How they address the aforementioned issues
- How to solve various problems with these systems

Managing Big Data

- When writing a program with these tools …
  - You don't know the size of the data
  - You don't know the extent of the parallelism

- Both try to collocate the computation with the data
  - Parallelize the I/O
  - Make the I/O local (versus across network)

- Data is often unstructured (vs. relational model)

Big Data vs. Relational

- RDBMS normalization
  - Goal is to remove redundancy and retain/insure integrity
- Big data apps want reads to be local
  - Send the code to the data, as it much smaller (Jim Gray)
  - Normalization makes read non-local
- Processing examines one input record at a time
  - Minimal state in programs – it's in the data

Big Data Tools

- This all sounds great. What are the issues?
  - Coordinating the distributed computation
  - Handling partial failures
  - Combining the results of distributed computation

- Tools offer a programming model that abstracts
  - Disk read and write
  - Parallelization (computation and I/O)
  - Combining data (keys and values)

MapReduce Design Patterns

- Summarization
- Filtering
- Data Organization
  - Partitioning/binning, sorting, shuffle
- Joins
  - Merging data sets
- Meta-patterns
  - Optimizing map-reduce chains (data pipelines)

Resources for Hadoop

- *Hadoop: The Definitive Guide, 3rd Edition*, by Tom White
  - O'Reilly Media
  - Print ISBN: 978-1-4493-1152-0 | ISBN 10: 1-4493-1152-0
  - Ebook ISBN: 978-1-4493-1151-3 | ISBN 10: 1-4493-1151-2
- *MapReduce Design Patterns*, by Donald Miner and Adam Shook
  - O'Reilly Media
  - Print ISBN: 978-1-4493-2717-0 | ISBN 10: 1-4493-2717-6
  - Ebook ISBN: 978-1-4493-4197-8 | ISBN 10: 1-4493-4197-7

- http://hadoop.apache.org/
- Several vendors provide Hadoop distributions
- Amazon Web Services – ElasticMapReduce

Resources for Spark

- *Learning Spark*, (early release) by Holden Karau, Andy Konwinsky, Patrick Wendell, Matei Zaharia
  - O'Reilly Media
  - Print ISBN: 978-1-4493-5862-4 | ISBN 10: 1-4493-5862-4
  - Ebook ISBN: 978-1-4493-5860-0 | ISBN 10: 1-4493-5860-8

- http://spark.apache.org/
  - Can download a version that runs on your local machine

- Cloud services
  - Spark on AWS
  - DataBricks offers a cloud service
  - Others will join the party