

CS 378 – Big Data Programming

Fall 2017

Lecture 1

Introduction

Class Logistics

- Class meets TTh, 9:30 AM – 11:00 AM, PAR 203
- Office Hours – GDC 6.402
 - T 11:00 – 12:00 AM, Th 12:30 – 1:30 PM
 - By appointment
 - Email: dfranke@cs.utexas.edu
 - Web page: cs.utexas.edu/~dfranke/courses/2017fall/cs378-BDP.htm
- TA: Vivek Pradhan
 - Office hours: TBD

Course Content

- Programming in Hadoop (map-reduce) and Spark
- Use ElasticMapReduce (EMR) on Amazon Web Services (AWS)
 - You can use a different Hadoop cloud service (like Azure)
 - You can use a local install of Hadoop if you want
- Local install of Spark (easiest)
 - Cloud based service from DataBricks (if you want)
 - Requires an AWS account

Textbooks

- MapReduce Design Patterns
 - Main content for Hadoop assignments
- Hadoop The Definitive Guide – 4th Edition
 - Recommended for your understanding, not required
- Learning Spark
 - Main content for Spark assignments
- All textbooks are available as ebooks from O'Reilly

Lectures

- PDF of (some) lecture notes accessible via syllabus
 - For your note taking, review, or whatever
 - cs.utexas.edu/~dfranke/courses/2017fall/cs378-BDP.htm

Assignments

- Assignments will be programming assignments
 - All work can be done using Java
 - Scala, Python are options for Spark
- 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

Assignments

- We'll be using Hadoop for the first assignments
- You'll need to do one of:
 - Create a personal account for cloud-based Hadoop
 - AWS (Amazon Web Services), MicroSoft Azure, ...
 - Install Hadoop locally on your personal machine
 - I recommend you start on this now
- Instructions for cloud-based Hadoop account options or local Hadoop install are available in Assignment 0

- Questions?

Managing Big 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 (3 years old now):
 - Facebook has ~800 petabyte data cluster (Hadoop)
 - 1 petabyte = 10^{15} bytes
- Big data is spread across a network of machines

Managing 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 becoming established– 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 is 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, 4th Edition*, by Tom White
 - O'Reilly Media
 - Print ISBN: 978-1-4919-0163-2 | ISBN 10: 1-4919-0163-2
 - Ebook ISBN: 978-1-4919-0162-5 | ISBN 10: 1-4919-0162-4
- *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
 - Cloudera, HortonWorks, MapR, ...
- Cloud-based Hadoop examples:
 - Amazon Web Services, MicroSoft Azure, ...

Resources for Spark

- *Learning Spark*, 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 are joining the party