Suzanne McIntosh
Cloudera and New York University
mcintosh@cs.nyu.edu

HPC and Distributed Computing for Students in Science and Non-Science Programs

ABSTRACT

The unprecedented growth in applications that leverage high performance computing (HPC) platforms ranging from supercomputers and grid computing systems, to Hadoop clusters, has created demand for graduates with skills in parallel programming, data science, and big data engineering.

Traditionally, HPC courses have been part of the Computer Science curriculum. As HPC applications find their way into finance, medicine, life sciences, advertising, and other industries, an HPC curriculum to simultaneously address the needs of Computer Science students and students studying business, life sciences, or mathematics is required.

Keywords*—distributed computing, education, curriculum, Hadoop, parallel programming*

INTRODUCTION

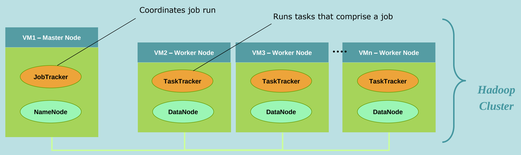
In developing a High Performance Computing (HPC) curriculum for students from various disciplines, one challenge was bridging the programming languages gap. In particular, Computer Science and Engineering students typically learn C, C++, Java, or Python, all of which are supported by Hadoop's compute component, MapReduce. However, students in other disciplines may program in higher level languages such as SQL or R, for example.

Fortunately, Hadoop provides several programming language options to bridge the programming gap and make HPC available to a wider user base.

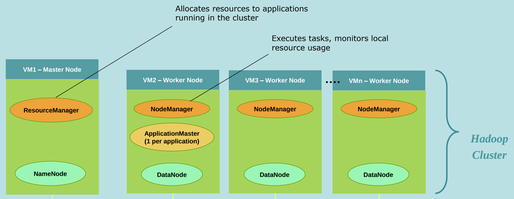
Another challenge in creating a Hadoop course is the continuously and rapidly changing Hadoop landscape. For example, the drawings below depict a major shift that occurred in the past year as enterprises started adopting YARN/MapReduce 2 in place of MapReduce 1. In order to keep pace with the technology, this course must be updated frequently.

The next major course upgrade includes the addition of Spark, a language that introduces the concept of resilient distributed datasets (RDDs), which can be persisted in memory to allow for highly performant and efficient distributed data processing [2].

MAPREDUCE 1 ARCHITECTURE

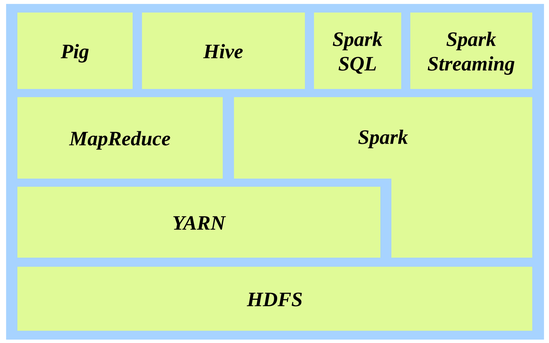


YARN / MAPREDUCE 2 ARCHITECTURE



INSTRUCTIONAL APPROACH

- Curriculum designed to teach parallel programming concepts using Hadoop tools
  - Distributed computing system that has become
  - Widely adopted by a variety of industries
  - Low cost SIMD HPC solution
- Early homework labs completed using
  - Hadoop cluster
  - Cost-free, fully configured Hadoop virtual machine (VM) [1]
- Lectures on 'Core Hadoop', which is comprised of a
  - Distributed storage component, HDFS (Hadoop Distributed File System) [4]
  - Distributed compute component, MapReduce [3]
- Lectures on the Hadoop ecosystem tools, including
  - Hive, a SQL-like language [7]
  - Pig, a data flow language for transforming datasets [6]
  - Both of these languages facilitate programming tasks, particularly for users who are less comfortable programming in languages supported at the MapReduce level (e.g. Java, C++, Python, etc.).



THE PROJECT

In the second half of the course, students form project teams to develop self-designed analytics projects using the tools learned in the first half of the course.

Project Requirements

- Form a team with maximum four team members
- Complete a project proposal describing the analytic to be developed
  - Who will benefit from the analytic?
  - What insight(s) are expected to be produced by the analytic?
  - How will the goodness of the analytic be assessed?
  - What are realtime use cases directly or indirectly related to the analytic?
- Identify and obtain at least two large data sources – this can be challenging due to a number of issues
  - Data owner is not willing to release data
  - Privacy concerns
  - Data requires owner's time to perform anonymization
  - Difficulty with transfer of data
  - Not enough storage to store the data source
  - Difficulty with data format
- Utilize at least two Hadoop compute technologies, e.g. MapReduce, Hive, Pig
- Identify a platform on which to run the analytic, for example
  - University Hadoop cluster
  - Amazon Elastic MapReduce (EMR)
  - Self-installed Hadoop on Elastic Compute Cloud (EC2) [5] [8]

STUDENT ANALYTICS PROJECTS

Business, Finance, Legal Projects

- News-Based Stock Performance Prediction
- Predictions for Success of Startup Investments
- U.S. Patent Office Analytic
- Predicting Local Real Estate Value
- Social Media-Based Stock Performance Prediction
- Yelp Business Data Analysis to Recommend Business Location

Health and Safety

- Breast Cancer Staging and Prediction
- Healthcare Relationship to Employment in the United States
- Public Safety and Mapping Analytic

Sports

- Prediction of NBA Game Pace
- Soccer: Dynamically Ranking Football Players

Entertainment

- Effect of Title, Posting Time, Comments and Community on Popularity of Social Media Posts
- Rating Prediction on NetFlix Dataset
- Song Recommendation
- Song Popularity Based on User Listening Patterns
- Book Recommendation

Travel and Transportation

- Detecting Movement Paths and Patterns Using Wireless Access Point Logs
- Urban Traffic Monitoring
- Impact of Weather on Road Travel
- Creating a Smarter Transportation Plan for New York City

Miscellaneous

- Drought Prediction
- Earthquake Data Analytic
- Resume Matching for Job Seekers
- Understanding Hierarchies in Technical Conferences
- Twitter Feed Sentiment Analysis

PROJECT DELIVERABLES

- Project Proposal – Students submit proposals describing their team analytic, the data sources, data sizes, and references to related work. The proposal is refined through several iterations. Sample Project Proposal:

Realtime and Big Data Analytics Project Proposal		
Part 1. General Information		
Team Name (optional): Team Members: Project Title: - Project Description: Refine what you have already written, add to it as needed. Data Sources - Use the table below to list and describe potential data sources.		
Part 2. General Data Source Information		
Data Sources - E.g. tweets	**Data Source Description**	**Data Size** Estimate size, e.g. MB? GB? TB?
Data Source 1		
…		
Data Source n		
Part 3. Detailed Data Source Information		
Data Sources - From Part 2. above	**Data Characteristics** - Is data source a realtime source? - Is it realtime and stored (e.g. a log)? - Is it statically loaded data (e.g. historic)?	**Data Frequency** - If realtime data, what is the frequency?
Data Source 1..n		
Part 4. Technologies		
Describe technologies. Will your project make use of MapReduce? Pig? Twitter? HDFS? Flume? HBase? Hive? Impala? Other?		
Part 5. References		
References – Please add references to all papers/articles read by the team (should be at least two references per team member).		

- Once the project proposal is approved, students create diagrams to describe data sources, data flows, data storage, and data processing.
- When the project concept is solidified, students create a task list which they use during weekly in-class scrum meetings.
- Students complete software development, perform results analysis, and adjust software or add additional data sources as necessary. Once satisfied with the results, students prepare for a class demonstration.
- Finally, students use a conference paper template to produce a paper describing their project. By the time of the paper writing, students will find that they have already produced material that can be leveraged for most of the paper sections, making the writing chore a little bit lighter. Paper format:

Abstract:	*A shortened version of the Introduction*
Introduction:	*Leverage the project proposal text already written*
Related Work:	*Use summaries of the relevant papers previously read and summarized*
Design:	*Add the design diagrams previously generated, write text to describe each diagram*
Results:	*The most time-consuming part of writing this paper*
Future Work:	*Based on results, what are next steps*
Conclusion:	*Wrap-up*
References:	*Add references to the papers that the team read*

REFERENCES

1. http://www.cloudera.com/content/cloudera-content/cloudera-docs/DemoVMs/Cloudera-QuickStart-VM/cloudera_quickstart_vm.html
2. M. Zaharia, M. Chowdhury, M. Franklin, S. Shenker, and I. Stoica. Spark: Cluster computing with working sets. 2nd Usenix Conference on Hot Topics in Cloud Computing, 2010.
3. J. Dean and S. Ghemawat. MapReduce: Simplified data processing on large clusters. In Proceedings of the 6th Symposium on Operating Systems Design and Implementation, 2004.
4. S. Ghemawat, H. Gobioff, and S. Leung. The Google File System. 19th ACM Symposium on Operating Systems Principles, 2003.
5. http://cloudera.com/content/cloudera/en/products-and-services/director.html
6. A. Gates, O. Natkovich, S. Chopra, P. Kamath, S. M. Narayanamurthy, C. Olston, B. Reed, S. Srinivasan, and U. Srivastava. Building a High-Level Dataflow System on top of Map-Reduce: The Pig Experience. Proc.of the VLDB Endowment, 2009.
7. A. Thusoo, J. S. Sarma, N. Jain, Z. Shao, P. Chakka, S. Antony, H. Liu, P. Wyckoff, and R. Murthy. Hive - A Warehousing Solution Over a Map-Reduce Framework. Proceedings of the VLDB Endowment, 2009.
8. Apache Whirr. https://whirr.apache.org/
9. T. White. Hadoop: The Definitive Guide, 3rd Ed.. Yahoo Press, 2010.
10. http://www.cloudera.com/content/dev-center/en/home/academic-partnership.html
11. Impala. http://www.impala.io/

ACKNOWLEDGEMENTS

Thanks to Amazon for supporting student projects through educational grants. Thanks also to Pennsylvania State University's CiteSeerx team for providing access to the CiteSeerx data used in student projects. Finally, thanks to the domain experts for supporting the development of student analytics projects.