CSE 4/587
Data Intensive Computing

Dr. Eric Mikida
epmikida@buffalo.edu
208 Capen Hall

Dr. Shamshad Parvin
shamsadp@buffalo.edu
313 Davis Hall

Hadoop Software
Announcements

- Midterms being returned today
- Regrade requests due by Friday (see Piazza for instructions)
- Key/Rubric posted to website
Additional References

- Lauran Serhal: “introduction to the hadoop ecosystem” Introduction to the hadoop ecosystem – Oracle
Key Aspects

- The key aspects of Hadoop we will be discussed are:
  - Architecture
  - Robustness
  - Data Organization
  - Communications Protocol
  - API to access services
  - Software (Mapreduce)
  - Evaluation of Hadoop
  - Hadoop Ecosystem
Software
Software System

- MapReduce requires a distributed file system, and an engine to distribute, coordinate, monitor, and gather results.
- MapReduce performs the processing of large data sets in a distributed and parallel scheme.
- MapReduce consists of two tasks: **Map** and **Reduce**.
- Hadoop processes the file system (HDFS), through its **JobTracker** and **TaskTracker** system.
Software (Map reduce)

- Job Tracker
- Task tracker
- Name node
- Data Node

MapReduce

HDFS
Job Tracker

- Scheduler service in the Hadoop system
- Manages Resources
- Client application is sent to the JobTracker
  - It talks to the NameNode
  - Locates TaskTrackers near the data
- Moves scheduled work to the TaskTracker
  - JobTracker is updated via heartbeat
  - Failure of a task is detected through a missing heartbeat
TaskTracker

- Accepts tasks (Map, Reduce, Shuffle, etc) from JobTracker
- Each TaskTracker has a number of slots for tasks
  - These are execution slots available on the machine or rack
- Indicates the number of available slots through the heartbeat message with the JobTracker
- Informed the JobTracker with the task status
From Brad Hedlund: a very nice picture
Software (Map reduce)

Large scale data

Splitting

Mapping (Key, Value)

Shuffling

Reducing

Key, Value:
- 8, 7
- 9, 9

- Pairing and grouping

- Process of reducing and combining

- Final output
A very good example by Lauran
Word Count Example (Map Reduce)

- How can we count the number of word occurrence in the Input file?
Word Count Example (MapReduce)
Evolution of Hadoop

- Hadoop has undergone an evolution from Hadoop 1.0 to Hadoop 2.0 (and today Hadoop 3.0)
- While the underlying principles related to the distributed file system (HDFS) have remained largely the same, resource management and software support has evolved
Evolution of Hadoop

**HADOOP 1.0**
- MapReduce
  (cluster resource management & data processing)
- HDFS
  (redundant, reliable storage)

**HADOOP 2.0**
- MapReduce
  (data processing)
- Others
  (data processing)
- YARN
  (cluster resource management)
- HDFS
  (redundant, reliable storage)
HDFS (discussed last lecture) provides the reliable distributed file system as the backbone of Hadoop.
Evolution of Hadoop

Originally, MapReduce was the only supported software system, and also had to handle resource management (via JobTracker and TaskTracker).

HDFS (discussed last lecture) provides the reliable distributed file system as the backbone of Hadoop.
Evaluation of Hadoop

- Hadoop 3 version was released on 2017 and comes with some new features.
- Hadoop-2 used replication concept and Hadoop-3 used HDFS Erasure coding(EC).
- Example: a 3X replicated file with 6 blocks will consume 6*3=18 blocks of disk space. With EC 9 blocks of disk will consume.
- Hadoop-3 introduced multiple standby Namenodes support.
A Brief Look at YARN

- The fundamental idea of YARN is to split up the functionalities of resource management and job scheduling/monitoring into separate daemons.
- The idea is to have a global ResourceManager (RM) and per-application ApplicationMaster (AM).
- RM is the ultimate authority of resource distribution among all the applications.
A Brief Look at YARN

- The ResourceManager has two main components: Scheduler and ApplicationsManager.
- The Scheduler allocates resources to the various running applications.
- The Scheduler is pure scheduler, it performs no monitoring or tracking of status for the application.
- No restarting ability for failed tasks either due to application failure or hardware failures.
A Brief Look at YARN

- The ApplicationsManager is responsible for accepting job-submissions, negotiating the first container for executing the application specific ApplicationMaster.
- It provides the service for restarting the ApplicationMaster container on failure.
A Brief Look at YARN

- The NodeManagers along with the ResourceManager form the data-computation network
- The NodeManagers monitor their local jobs and report back to the RM
A Brief Look at YARN

- Each application has an ApplicationMaster which negotiates resource requests with the RM
- Working with the NodeManager(s) to execute and monitor the tasks.
Hadoop Ecosystem

- Hadoop Ecosystem is a platform or framework which solves big data problems. We can consider it as a suite which encompasses a number of services (ingesting, storing, analyzing and maintaining) inside it.
Apache Ecosystem

- **HDFS** (Data storage)
- **YARNs** (Resource management)
- **MapReduce** (Data processing)
- **Hive** (SQL)
- **Pig** (Scripting)
- **Mahout** (machine learning)
- **Flume** (Structured/Unstructured data)
- **Scoop** (Structured data)
- **Oozie** (Workflow)
- **Spark** (Data flow engine)
- **ZooKeeper** (Co-ordination)
- **HBASE** (NoSQL)