

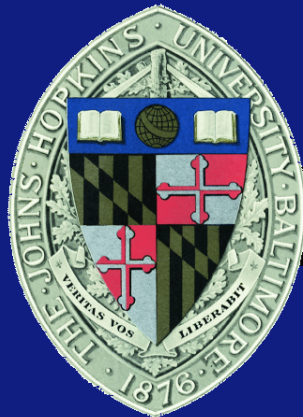
Lecture 15.3

Hadoop! Toolchain

EN 600.320/420

Instructor: Randal Burns

29 March 2017



Department of Computer Science, *Johns Hopkins University*

The Hadoop Tool Chain

- The command line tool chain
 - Build files into directory
 - Construct java archive (jar)
 - Point Hadoop! at the jar
- Many prefer to use Eclipse instead



Hadoop! Configurations

- Hadoop! is a heterogeneous, distributed system
 - Many components: namenode, hdfs, reporting
 - Parallelization (mappers, reducers, shuffle, loading)
 - Typically involves managing a cluster
- But can run in several simpler ways
 - Pseudo-distributed (full runtime on one machine)
 - Fully distributed (on a cluster)
- Running on pre-configured clusters
 - Specify size and types of nodes
 - Launch a compiled JAVA jar file or streaming scripts
 - AWS, Azure, Joyent, IBM, RackSpace
 - Metaservices: Cloudera



Hadoop! Streaming

- Given arbitrary string processing functions to the Hadoop! Environment
 - A map script and a reduce script
- Almost equivalent to:
 - `cat inputdir/* | mapper.py | sort | reducer.py`



Streaming and Sorting

- Streaming mode in Hadoop! Gives a different sorting guarantee
 - Recall: `cat inputdir/* | mapper.py | sort | reducer.py`
- *Why?*
- *Same or different semantics?*
- *Any performance implications?*



Streaming and Sorting

- Streaming mode in Hadoop! Gives a different sorting guarantee
 - Recall: `cat inputdir/* | mapper.py | sort | reducer.py`
- Why?
 - There is no schema
 - So, it sorts the whole output of `mapper.py` as a key
 - This is more restrictive than the default sort
 - And, thus, less efficient



Map/Reduce Recast (7 y.o. #s)

- Scanning engine
 - Use massive parallelism to look at large data sets
- Performance on 100 TB data sets
 - 1 node @ 50 MB/s (STR of disk) = 23 days
 - 1000 nodes = 33 minutes
- Batch Processing
 - Not real-time/user facing
- Large production environments
 - Not useful on small scales
 - Too much overhead on small jobs

