

# Hadoop Parallel Programming

Kai Shen

3/20/2014

CSC 258/458 - Spring 2014

1

## MapReduce

- A programming interface (two-stage Map and Reduce) and system support such that:
  - the interface is easy to program, and suitable for many big data processing applications;
  - the underlying system can automatically support parallelism, data movement, load balancing, and fault-tolerance behind this interface.
- Strength (compared with threads, MPI, etc.)
  - Ease of programming
  - Integration with data management to better support big data apps
- Weakness
  - Not applicable to all applications
  - More difficult to achieve good performance

3/20/2014

CSC 258/458 - Spring 2014

2

## Hadoop

- Original MapReduce/GFS implementation used at Google, but not shared with the public
- Hadoop: Java-based implementation of MapReduce
  - Originated at Yahoo but made open-source
  - Used by Facebook (more than 100 PetaBytes of data), Amazon (EC2 used by NYTimes), and many other places ...
- Hadoop implements MapReduce
  - Multiple TaskTrackers run map()/reduce() tasks
  - Single JobTracker assigns tasks to TaskTrackers and manage the entire workflow
  - Each TaskTracker sends periodic heartbeats to the JobTracker for fault detection and management

3/20/2014

CSC 258/458 - Spring 2014

3

## Hadoop Distributed File System

- Hadoop works with Hadoop Distributed File System (HDFS)
  - Inferior to GFS in many ways (poor support for mutable data replication, incompatible with POSIX standard); but open-source

3/20/2014

CSC 258/458 - Spring 2014

4



## Hadoop System Structure

[http://upload.wikimedia.org/wikipedia/en/2/2b/Hadoop\\_1.png](http://upload.wikimedia.org/wikipedia/en/2/2b/Hadoop_1.png)

- JobTracker and NameNode work together to support intelligent task placement
  - Dynamic load balancing
  - Data affinity

3/20/2014

CSC 258/458 - Spring 2014

5



## Hadoop Programming

- Java-based, implement several interfaces
- Mapper()
  - Maps input key/value pairs to a set of intermediate key/value pairs
- Reducer()
  - Reduces a set of intermediate values which share a key to a smaller set of values.
- Tool()
  - Handles your custom command inputs or any other custom preparation work

3/20/2014

CSC 258/458 - Spring 2014

6



## Hadoop Task Mapping

- Hadoop tries to assign map tasks to where the data is
- Surprisingly where the data is also affects how many map tasks are launched
  - The real number is typically no smaller than the number you specify, but could be larger if there are more distributed splits of your data
  - Always check the completion status of your run to be sure about the number of map tasks

3/20/2014

CSC 258/458 - Spring 2014

7



## Hadoop Performance

- What important issues affect Hadoop performance?
- Parallelism: number of map tasks; load at reduce (programmer control)
- Data locality (largely system control)
- Load balancing: uniform/heterogeneous nodes; uniform/heterogeneous tasks (system/programmer control)
- Overhead of task/data management – granularity of tasks and data partitions (programmer control)

3/20/2014

CSC 258/458 - Spring 2014

8

## Assignment #5

- You will program three applications on the Hadoop parallel data processing platform.
- Goal: gain practical experience on Hadoop programming and learn the performance implication of parallel data processing.
- This assignment alone won't be enough to make you a great Hadoop programmer.

3/20/2014

CSC 258/458 - Spring 2014

9

## Word Counting

- The input is one or more text files.
- You should count the number of files each word appears in. Your word identification should be case-insensitive and ignore anything that isn't a letter.
- We provide two datasets: one for correctness testing and another for performance tests.
- The first example of almost any MapReduce/Hadoop tutorial; you can pretty much have the solution to carefully go through the tutorial.

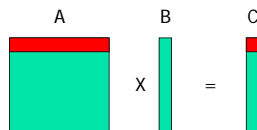
3/20/2014

CSC 258/458 - Spring 2014

10

## Matrix-Vector Multiplication

- Matrix-vector multiplication takes a file describing a matrix as an input and has to also load a file with a vector. The matrix will be split across map tasks, but every task will need to load the entire vector.



3/20/2014

CSC 258/458 - Spring 2014

11

## Iterative Linear System Solver

- Widely used for scientific computing and web data processing (PageRank).
- Iterative matrix-vector multiplication
  - MapReduce for each iteration
  - Data management between iterations

3/20/2014

CSC 258/458 - Spring 2014

12

## What performance to expect?

- You may be disappointed by the speedup you see
  - Depending on what you use as a base, you may not see any speedup
  - Larger problems/datasets has higher chance to allow speedup
- TA may provide more guidance on this

3/20/2014

CSC 258/458 - Spring 2014

13

## Our Hadoop Cluster

- One master node and 9 slave nodes
  - Small scale
- Status/usage check
  - Hadoop and HDFS
- Courtesy of using the shared Hadoop cluster
  - Particular for performance testing
  - Don't leave zombie tasks hanging in the system!

3/20/2014

CSC 258/458 - Spring 2014

14

## Grading

- Equal grading weights for the three applications (but nowhere near equal difficulty!)
- Grade on correctness and performance
- On performance, we care about the raw speed at multiple (up to 8) maps, not so much on the speedup over your own solution at 1 map, which can be poor

3/20/2014

CSC 258/458 - Spring 2014

15

## Miscellaneous Things

- Will post the assignment later today
- E-turnin
- Start working early
- Strongly encourage early turn-ins (bonus). Note that the last turn-in counts.
- Possibly run into issues, bear with us, potential award credit if discover and help solve problems with us
- TA session on April 3

3/20/2014

CSC 258/458 - Spring 2014

16