Apache Hadoop/ Spark

- two things on class website that need to be read for homework

- hadoop is aframework that allows distributed processing of large data sets across clusters of computers..

- Sing simple programming models

- designed to scale well

- library is designed to detect and handle failures at the application layer, so delivering a highly-available service on to p of a cluster of computers, each of which may be prone to failures

- Original system written by Google

- everything is redundant in the system

Key components

built on top of hadoop distributed file system (HDFS) –

hadoop yarn(Yet Another Resource Navigator?) is used for scheduling and cluster resource management

processing – hadoop mapreduce mr2 – a yarn-based system for parallel processing of large data sets

HDFS

inspired by google file system

stores large files across multiple machines, replicating across multiple hosts

file system intentionally not fully posix-compliant

(write-once-read-many access)

hadoop must know which servers are closest to the data (HDFS does this

- sends computation to data instead of otherway around

MapReduce is a programming model for processing and generating large datasets with a parallel distributed algorithm

define

map(key,value)->list(key2,value2) and

reduce(k2, list(v2)->list(v3) and it will do all the datastuffs

MapReduce Word Count example

map(String inputkey, String inputValue):

// input-key is document name

// input value is document contents

for each word w in inputvalue:

emitintermediate(w, “1”) //prints word and number 1

reduce (String output\_key, Iterator intermediate\_values):

// outptkey a word

// outputvalues:a list of counts

int result = 0

for v in intermediate\_values:

result += parseint(v);

Emit(asString(result);// should output key and result

MapReduce Combiner (pre-reduces)

MapReduce problems & potential solution

Apache Spark- how to do more in memory

everything else is handled for you :)

graph algorithms in MapReduce

G = (V, E)

typically traversing and computation

How does MapReduce on Graphs?

(represent/ traverse?)

two common ways: adjacency matrix(nice for math, but there is lot of wasted space), Adjacency List (take matrix and toss out the zeros) (more compact, but difficult to compute inward links)

Single machine – Dijkstra’s works fine

parallel process – Breadth first search

Shortest path with maprecuce

Map: every element of adjacency list: emit(m, d+1)// node and a distance

sort/shuffle

Reducer:

select minimum distance path for each reachable node // also out put adjacency list

map and reduce until all levels of map complete/ certain degree of certain-ness is acquired