

Statistics is FUN

Algorithm



April 22, 2015

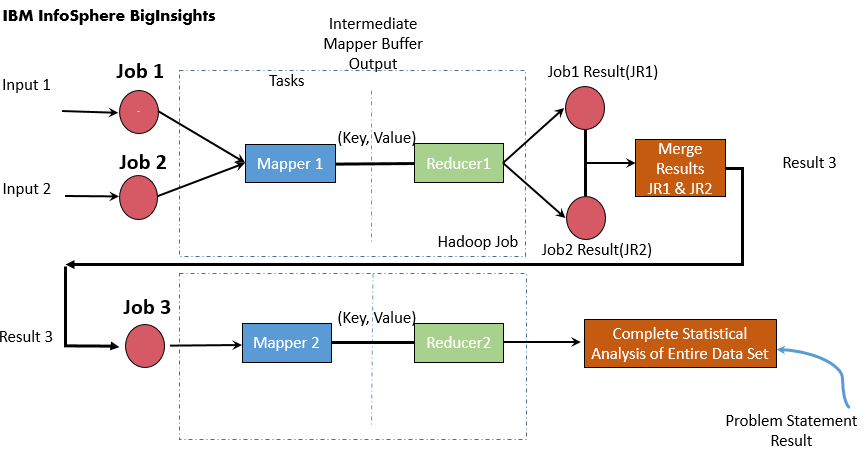
University of Missouri-Kansas city

UMKC 2015 Spring Hackathon

# Problem Statement

John has a file that contains lots of numbers (all integer). They are unordered. He wants to generate some statistics around these numbers, namely COUNT, MEAN, Standard Deviation, MIN, MAX, and 25th, 50th, and 75th percentiles. The numbers are so large that it cannot be run on a single machine. Write a map-reduce program to generate above-mentioned statistics (no APACHE PIG, no HIVE).

# Design Approach



# Algorithm

* In order to accommodate large data set from multiple systems we can create multiple jobs in the driver class to create intermediate values for each of the integer data set.
* To generate intermediate key values pair we make use of Set 1 map reduce program discussed below.
* For now, we have implemented two map-reduce jobs and each job takes individual input file containing integer dataset from different source path.
* Finally, a third map-reduce job is invoked to merge the output from above two jobs and compute the complete statistical analysis discussed in problem statement on the entire dataset.

Map-Reduce Jobs

**Job 1:** This job uses set 1 map-reduce tasks to generate intermediate data for one source file.

**Job 2:** This job also uses set 1 ma-reduce tasks to generate intermediate data based on another source file.

**Job 3:** It first merges the reducer output values of job 1 & job 2 and later, it computes the required statistical analysis on entire dataset.

There are 2 set of Map Reduce Programs developed for performing statistical analysis.

**Set 1:** This set of map-reduce program is used to compute intermediate statistical results.

**Mapper Class:** ReadInputMapper

*Input (Key, Values):* Source file path as Key,

Set of integer data set separated by new line as Values

*Output (Key, Values):* A common identifier assigned to each of the integer as key,

Individual integer data as Values

**Reducer Class:** IntermediateReducer

*Input (Key, Values):* A common identifier assigned to each of the integer from map as key, Individual iterable integer data as Values

*Output (Key, Values):* An identifier assigned by mapper to each of the integer as key,

Computation result i.e. count, sum, min, max and sum of individual squares of data as Values

Set 2: This set of map-reduce program takes the intermediate results generated from set 1 and performs aggregation of intermediate reducer outputs to compute complete statistical analysis on entire data set.

**Mapper Class:** ComputationMapper

*Input (Key, Values):* Merged file path containing results from Jobs executed from set 1 map-reduce programs as Key,

Merged result set as Values

*Output (Key, Values):* An identifier assigned to represent Set 2 Mapper as key,

Temporary results of required statistics - calculated count, sum, mean, min, max, standard deviation (except percentiles) as Values

**Reducer Class:** ComputationReducer

*Input (Key, Values):* An identifier assigned by Set 2 Mapper as key,

Temporary results of required statistics from mapper- calculated count, sum, mean, min, max, standard deviation (except percentiles) as Values

*Output (Key, Values):* An identifier assigned to represent final computation as key,

Complete statistical analysis results on Entire dataset from multiple sources as Values