# Class Average Hadoop Program Report

### By Santipab Tipparach

## Introduction

This program is designed to aggregate a data set by applying weights to 5 variables in each row and find the min, max, and average of each class/category via the hadoop zoidberg cluster.

## Design & Implementation

The program is implemented in Java in the Eclipse IDE. The design is fairly simple. Ot takes advantage of the serialization interface in the Hadpop API called the ‘Writable’ interface. This interface is implemented in the ‘Class Writable’ class. It contains the various writable objects like IntWritable for the count of rows in each class, some DoubleWritables for the min,average, and max statistics,  and the total value so far. The idea here is to allow the same class objects to merge, evaluating the statistics as we go. The key is the class name itself, this could be simplofied to just an IntWritable, but I used a text since it looks nicer on the output.

The code flow is simple, all values are mapped to their class ID as a ClassWritable object in the mapping step. The reduce step merges all the objects ofthe same class contained on the same node. The nodes are split in the main class, and its also where the benchmark portion is being done.

## Experimentation

The process of experimentation was straightforward. Using the file provided, I applied the weights by hardcoding them into the program.

The Hadoop commands were saved into a file, and executed via the SSH terminal “Putty”.

The execution command was run 10 times for each number of nodes specified

**hadoop jar ca.jar ClassAverage /user/tipparac/classaverage/input /user/tipparac/classaverage/output /user/tipparac/classaverage/logs 2**

The results were retrieved via this command

**hadoop fs -copyToLocal /user/tipparac/classaverage/output /home/tipparac/output**

or could be simply read using this:

**hadoop fs -cat /user/tipparac/classaverage/output/part-r-00000**

## Result

The results of this experimentation is in the graphs provided below, along with an excel sheet that will be attached, and a set of sample output files.

Here we can begin analyzing and making assertions as to what’s going on.

In this graph