

Problem 1) Towards the end of the last lecture we described how to create one MapReduce job previously implemented in two jobs represented by classes `InverterCounter.java` and `CitationHistogram.java` and produce a “chain” of MapReduce jobs in a single class `ChainedHistogram.java`. Please follow the description and demonstrate that you can truly “chain” two previous jobs and execute them under one java class `ChainedHistogram.java`. The main objective is to avoid manually running two jobs. At the end of processing remove HDFS directory where you kept the intermediate results.

You can delete the intermediate data generated at each step of the chain or at the time when they are not needed any more, at the end. You can perform the file deletion with a command like this

```
FileSystem.delete(Path f, boolean recursive);
```

You can control input and output file locations (paths) programmatically using Path class.

```
Path in = new Path("HDFSDirectory/filename");
Path first_out = new Path("hdfs_directory");
```

Demonstrate that your chained MapReduce job for calculating Citation Histogram starting with patents citation file `cite75_99.txt` will produce the same result as the sequence of two jobs used in class. Implement class `ChainedHistogram.java` in “old” API.

Problem 2) Implement class `ChainedHistogram.java` in new API. Demonstrate that it works and produces identical results as the original. Provide working code and snippets of your results and console outputs.

Problem 3) Imagine that you are a Linux person and you have not a single machine with an Excel. Your boss is adamant and wants a graphical histogram of the number of patents cited (`no_patents_cited`) given number of times as determined by the attached `CitationHistogram.java`. Write a new Hadoop program that will print a true (graphical) histogram of $4\log_{10}(\text{no_patents_cited} + 1)$ values using a string of asterixes (*) indicating the value. Variable `no_patents_cited` is showing how many patents were cited once, twice, three-times and so on. Your histogram will look approximately like this:

```
1-20      *****
21-40     *****
41-60     *****
. . . .
760-780 *
```

Use buckets of size 20. $\log_{10}(1) = 0$, so if we try to present only $\log_{10}(\text{no_patents_cited})$, the very bottom of the tail would get lost in this particular histogram, since we cannot paint zero asterixes. To preserve the tail we added number 1 to the `no_patents_cited` in the expression $4\log_{10}(\text{no_patents_cited} + 1)$. By doing this, we introduce a tiny error for most of the histogram, but preserve the visibility of the tail. Number 4 is added for a similar reason. Namely, $\log_{10}(2) = 0.301$. Since you cannot paint 1/3 of an asterix, we are multiplying the logarithm by 4 to scale it out. That would produce a full point on the graph for the counts on the very bottom of the tail. You should try presenting the histogram without those embellishments, as well. Give your boss the one that looks better. Appearance matters.

Problem 4) Consider the following definite integral $\int_1^{10} 1/x \, dx$. This integral can be calculated exactly, and its value is $\ln(10) - \ln(1)$. Many other integrals cannot be calculated exactly and we resort to approximate techniques such as the Riemann sums. Riemann sum for an integral with lower and upper boundaries a and b of a function $f(x)$ is defined as:

$$\sum_{x_1=a}^{x_N=b} f(x_i) \Delta x$$

Here, the interval (a, b) is divided into $N - 1$ subintervals of width Δx , where Δx is the difference between two adjacent values of x , i. e. $\Delta x = x_{i+1} - x_i$. Also $a = x_1$ and $b = x_N$. Use MapReduce to calculate the Riemann sum with $N = 10,000$ for the above definite integral from 1 to 10 of the function $f(x) = 1/x$.

Problem 5) Extend the technique of previous problem to calculate the Riemann sum for the integral

$$\iint_{0,0}^{3,3} e^{-(x^2+y^2)} \, dx \, dy$$

Run the calculation on a 100x100 grid and then 1,000 X 1,000 grid.

Submission Note: Please capture all the steps of your implementation in an MS Word document. Please add comments indicating what is accomplished with every step. Please submit a copy of working code.

Please place all files you want to submit in a folder named: HW07. Compress that folder into an archive named E63_LastNameFirstNameHW07. ZIP. Upload the archive to the course drop box on the class web site. Please send comments and questions to cscie63@fas.harvard.edu