# Programming Assignment 3 Due Friday December 19th, 2014 11:59pm Hadoop and MapReduce (20 points)

The goal of this project is to get hands-on experience with Hadoop and MapReduce. You will run MapReduce jobs on the Amazon Hadoop EC2 clusters and report on the results.

What you should do:

1. Follow the Elastic MapReduce tutorial given in the Sakai Programming Assignment Resources.
2. Implement a MapReduce task that **returns the average rating of movies** in the Netflix dataset.

**Your results should be in a s3 bucket. You will need to give permission to the AWS user** [**cyramel@gmail.com**](mailto:cyramel@gmail.com) **to see the content of the bucket, otherwise I will not be able to see your results.**

You should submit:

1. The path to your s3 bucket which contains your output. The TA and instructor should have permission to read the bucket. If we do not, you will not get credit for your work.
2. The **output** of running the wordcount example from the tutorial on the EC2 cluster on the input directory s3n://worddatasets/wordcounttestinput Note that this is a different input directory from the one used in the tutorial. You should also include the name of the EC2 cluster you used for this computation. The results should be in your S3 output path **(3 points)**
3. The **output** of the Netflix movie average MapReduce job on the small dataset that is available on Amazon S3:  
   s3n://netflixdatasets/Netflix\_small.txt

You should also submit (in Sakai) the **java file(s)** you created for this task. You will need to tokenize the input file to get the correct values for the movie\_id and rating; look at the way the file is formatted. **(6 points)**.

1. The **output** of the Netflix movie average MapReduce job on the larger dataset (which share the same format as the small one) that is available on Amazon S3: s3n:// netflixdatasets/Netflix\_large.txt.

You should also return the **java file(s)** you created for this task (most likely the same as in 2 above). For this part you should try different numbers of clusters and study on the impact of the number of clusters on the processing time of the MapReduce job on the large Netflix dataset **(6 points)**.

1. Return the same **output** as in question 3, but sorted by movie average (highest-rated movies first). **(3 points)**.
2. A documentation of your code and description of what you did. **(2 points).**