Geetish Nayak Purvi Bafna

Sherry Jyoti Yadav

**Assignment 3**

**Objective**

The purpose of this assignment is to create a social network diagram that shows the number of emails sent from each email user to another email user in the dataset.

**Input Data**

The input is the Enron email dataset, which contains data from about 150 users, mostly senior management of Enron. The corpus contains a total of about 0.5 millions messages. This dataset does not contain attachments.

**How this is achieved**

Map – Reduce program

1. **Purpose of the mapper program**

This program reads data from the input file, checks for each email through a message id, and then looks for the corresponding “from” and stores the sender of the email. It then looks for the “to” in the email to retrieve the receiver of the email.

**Please read below for detailed implementation explanation.**

1. **Purpose of reducer program**

The reducer receives the data from the mapper and sums up the count of value for each email key.

**Please read below for detailed implementation explanation.**

**Detailed Explanation**

1. Mapper program

**EmailMapper.java** – This is a mapper class, which on reading each line from the input file, searches for the message id and flags a class variable as true. When the “From” and “To” are found for this email, the emailkey is formed as Sender$Receiver. The mappers send these key value pairs to the reducers.

1. Reducer Program

**EmailReducer.java –** This is the reducer class where it receives data from the mapper class. The purpose of the reducer in this case is to count the number of emails for a particular sender-receiver pair, by summing up the values for the emailkey passed by the mapper.

The output is formatted in the following manner

From : <Name> To : <Name> <count of emails between them>

**Actual Working**

1. In this program the global variables which are written for a specific mapper actually make it work. We have a flag which is set when we first encounter the <Message Id> and then search for the corresponding “From” and “To” email addresses.

**Is there a better methodology?**

1. We could have written our own custom classes which would extend the FileInputFormat and RecordReader.

**Output**

