Data Compression through Duplicate Elimination and Tagging

Jeffrey Thomas

1. This project is to test the theoretical viability of a data compression algorithm that I invented. This project will test the algorithm in all of its potential variables and extract solid data on how much time is saved though this algorithm and how much data is compressed vs. the iterations of the algorithm and the variables involved.

2. The purpose of this project is to test a data compression algorithm that could, theoretically, be used on any kind of sent data. With some basic math, I have computed that this algorithm could, at a minimum, compress the data by twelve and a half percent. What I need to find out is how this relates to time saved, and whether or not the compression algorithm actually saves any time at all. The results can be applied very easily; if the algorithm is a success, then any computer can use it to improve the speed of sending data by sending less.

3. The algorithm works off of two principles; the "random" distribution in infinitely large data sets and that operations will take less time than sending data. The average file contains thousands to millions of bits. The algorithm works by analyzing data in sets of sixteen, 32-bit groups. The first sixteen bits of each group are then compared to the last sixteen. If they are different, then nothing happens and the algorithm moves on. If they are the same, the algorithm will discard the second half, add a four-bit number to the start of the data noting where the bits were removed, and add one to a running tally of duplicate groups. This final number will be added at the end of the algorithm. The data is sent and decompressed using the numerical markers.

For any group of 2^n bits, there are exactly (2^n)/4 "duplicate" patterns in bits. Therefore, with an infinitely large data set, roughly 1/4 of the groups will be duplicates. Also, the numbers 16 and 32 are arbitrary. They can be any value so long as they are powers of 2.

The testing portion of this project will test how changing those two numbers will affect the time that is saved and the amount of data that is compressed. From basic estimations beforehand, I have estimated that by using this method, I would send a minimum of 12.5% less data without losing any accuracy.

4. I have done some research into other methods of data compression, but so far none have taken the approach I have. The fundamental difference between my method and any other is that mine relies on basic probability instead of more complex statistical analysis. Also, my testing of the algorithm is specifically designed to save time when sending data between computers, instead of just saving space. There is also the fact that my algorithm can be run multiple times and still compress more data.

Today, there is no dominant data compression method. Each method has its own disadvantages and advantages.

5. To test my program, I will generate a very large text file with only the strings of "1" and "0". It does not matter what kind of data I use, as I will be measuring the percentage of data compressed and time saved. I will send the data to another computer, and have that computer send back a confirmation message for timing purposes. I will then compress the data, send it, decompress it, and then send the

confirmation message. The times of both of these actions will be taken and compared, as well as the amount of data compressed. I will follow this procedure for different numbers of sets, groups, and possibly iterations of the algorithm.

This testing procedure will have some user input, namely, the variables for number of sets, the size of groups, and the number of iterations. Before I can begin testing, I need to finish programming my compression and decompression algorithms in Java. I have already completed the file generator code.

I will have the compression and decompression algorithms completed by interims of second quarter, and will have started collecting data. The data collection should be complete by the end of second quarter.

6. I expect this algorithm to be a success. The only factor in question is how the variables effect each other.

The data will be presented in two three-dimensional graphs of number of groups and size of sets vs. time saved and data compressed, respectively.