



Small.txt:

As larger amounts of processes are added, the speedup decreases because there is extra communication and extra partitioning between processes. For the small amount of data, the extra processes (and communication overhead) will result in a decrease in the speedup.

Medium.txt:

The medium.txt file also contains a small amount of data. This is similar to small.txt's results. The extra overhead involved with adding more processes does not benefit the program as a whole because a single process could have calculated all the data by itself faster (as opposed to communicating with several other processes).

Large.txt:

The large.txt file contains a relatively small amount of data as well. The extra overhead involved with adding more processes does not benefit the program as a whole because a single process could have calculated all the data by itself faster.

Huge.txt:

The huge.txt file is fairly large, but is still relatively small. As more processes are added, speedup will continue to decrease because extra overhead will be introduced to the program involving partitioning work and communicating.

Overall, it can be seen that adding more processes will eventually result in the speedup decreasing. This is because for data sets that are not large (i.e. less than several thousand usually), speedup will not be seen because partitioning the data, communicating, and synchronizing all the processes takes time. This time outweighs the benefits seen from any speedup gained in decreased *calculation* time by a process.

My program avoids large decreases in speedup up to 16 processes by limiting the actual processes calculating data to the number of elements in the input file. This was done by creating a new communicator and decreasing the `comm_sz` when the number of processes allocated to a task was larger than the number of rows in the file's matrix. This is because each process in my program does the calculations for at least 1 row of data. This cannot be parallelized any further than each process doing 1 row.