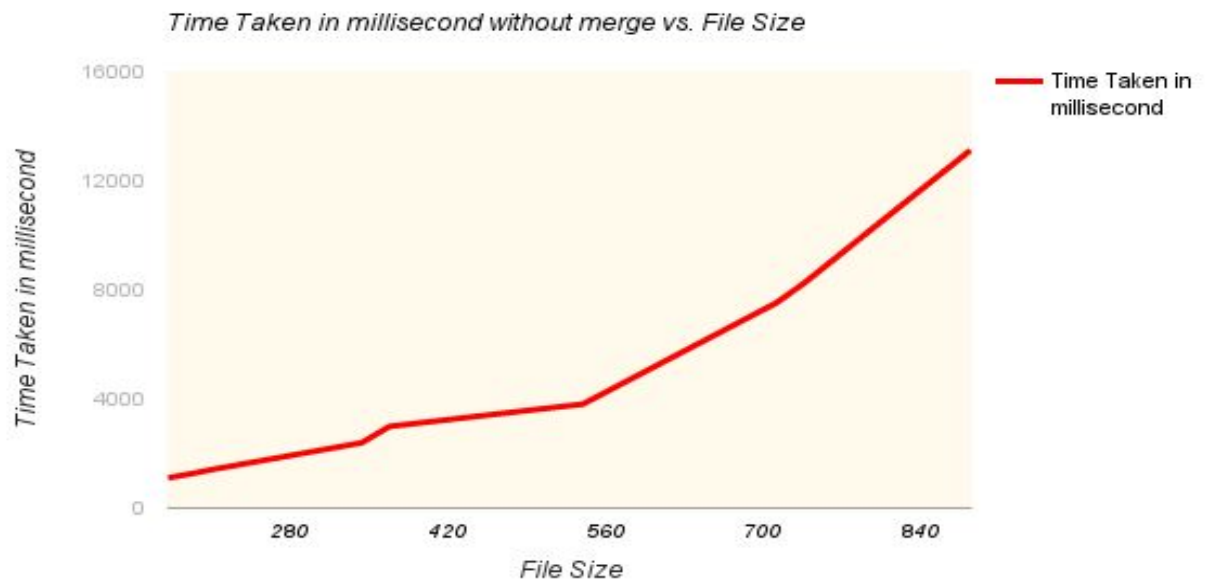


Ashish Rajthala  
CS260-01  
Dec 8, 2016

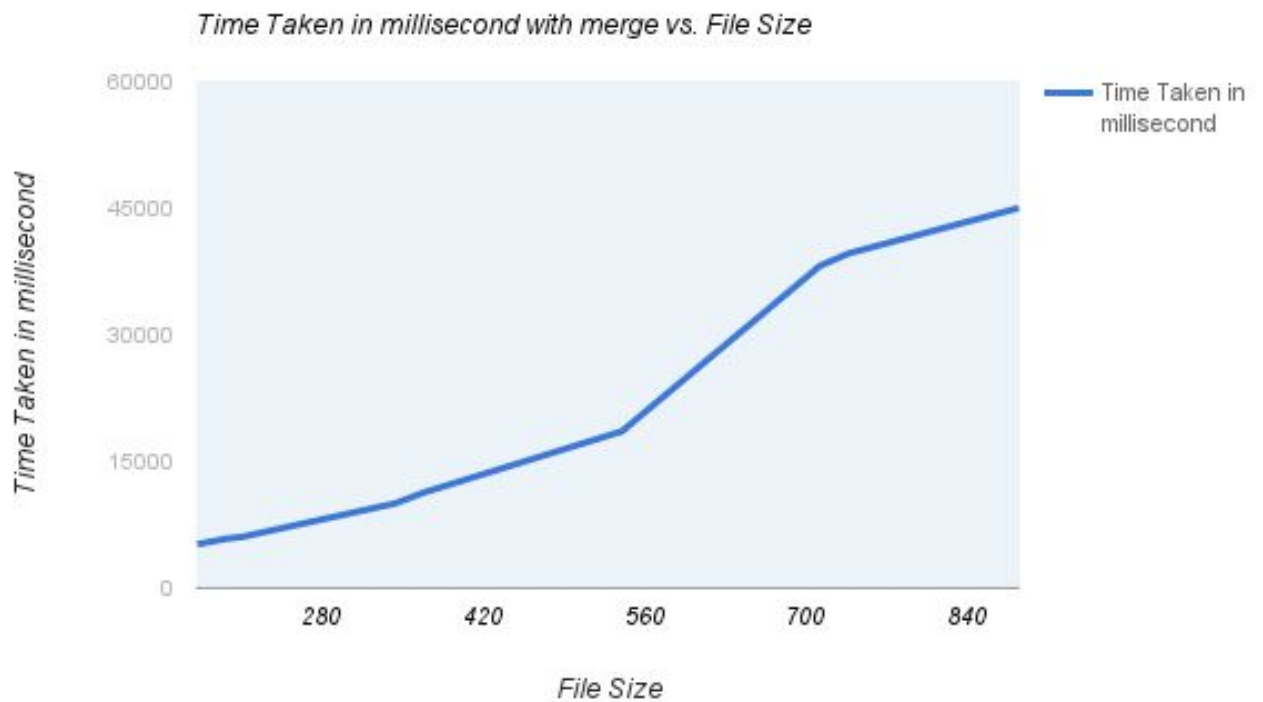
**Table contains data of the download accelerator program with and  
And without merge:**

File Size	Without merge (in millisecond)	With Merge (In millisecond)	Single Thread (in millisecond)
172 mb	1106	5202	1829
196 mb	1287	5831	2133
211 mb	1413	6079	4331
344 mb	2391	10046	7465
369 mb	2994	11341	8917
541 mb	3811	18579	12030
713 mb	7520	38176	17105
738 mb	8249	39658	17950
886 mb	13124	45042	20294

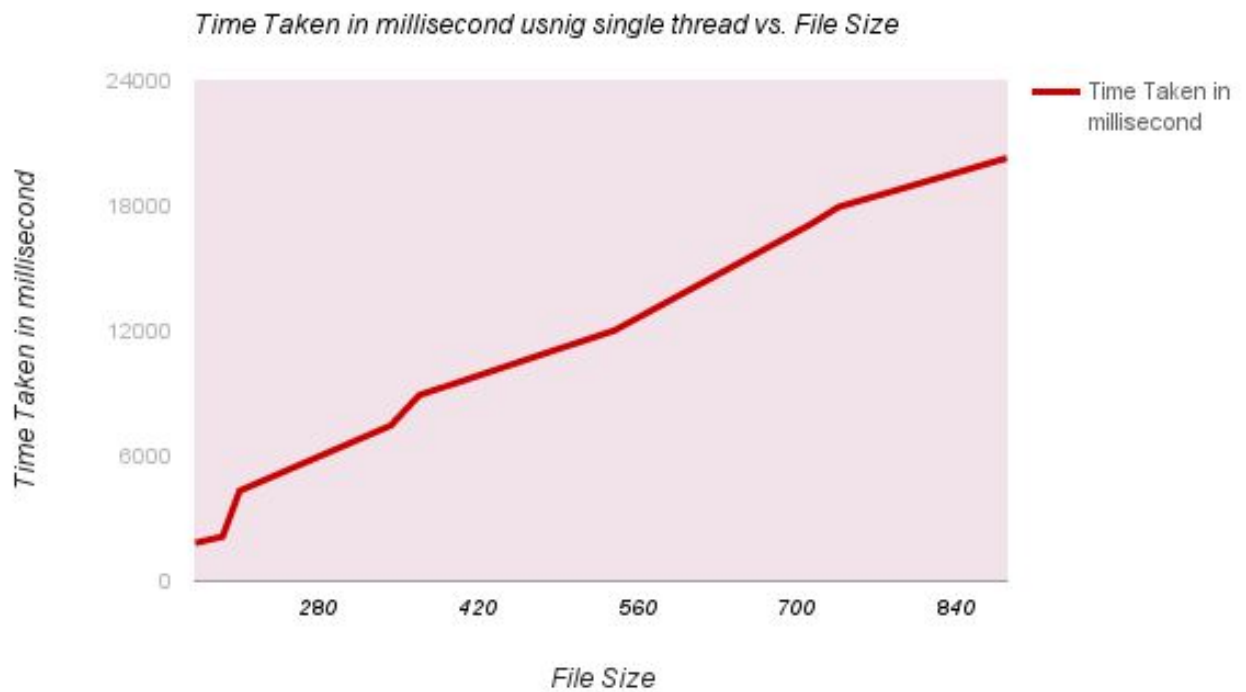
## 1. Multi-Threaded without merge delays:



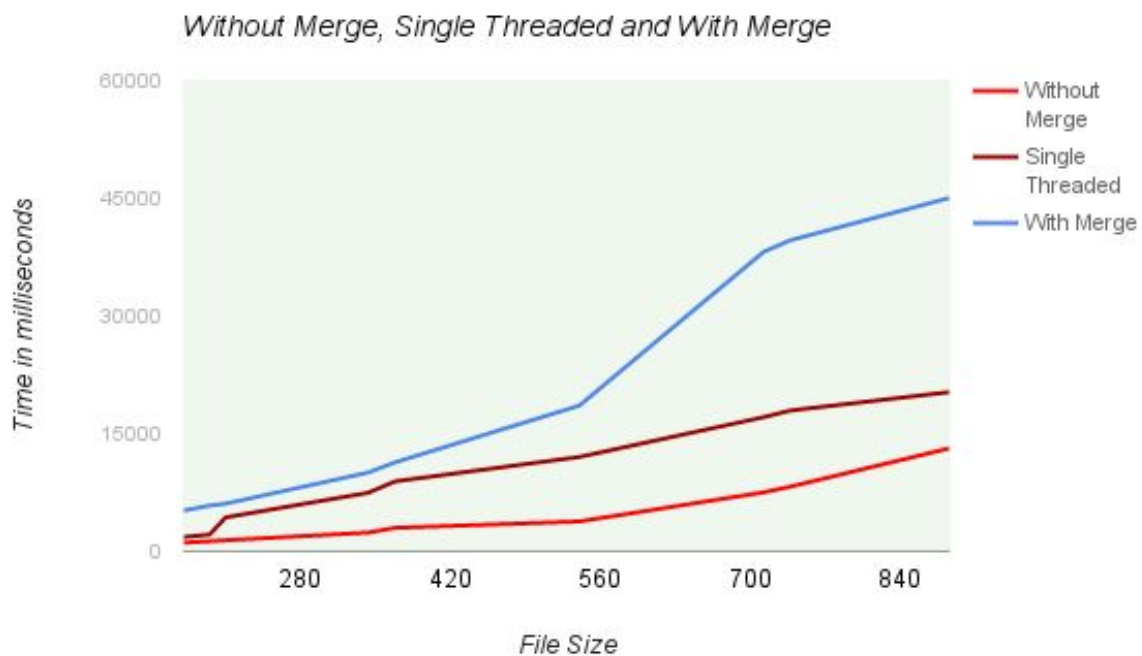
## 2. Multi-Threaded with merge delays:



### 3. Single Threaded without delays:



### 4. Graph containing all of the above:



## **Graph Analysis:**

- **Without Merge:**

The first graph shows the relationship between the download time (in milliseconds) for the *Download accelerator program* to download files before merge. Here, the red line represents points of the relationship between file size and time taken. The line increases at a slow pace at the beginning but as the file size increases the graph increases drastically. When the file size is small, the time taken to download is fast, and as the file size increases the download time slows down.

- **With Merge:**

The second graph shows the relationship between file size and the time taken to download a file added with the file merge time. The blue line in the graph represents the different points and the relation of the file size and time taken. Like the graph of without merge there is a direct relationship between time and size. The only difference here is that the starting time of merge is greater than that of without merge.

- **With Single Thread:**

The third graph represents the relation between the time taken for file transfer and the size of the file. The dark brown line in the graph shows that there is a gradual increase in the line as the file size increases. Here, when the file is small the download speed is high and as the file size rises the download speed slows down. The graph is similar to the graph of with merge and without merge. The only difference is the starting point of the graph.

- **With all of the above:**

The fourth and the last graph above is the one that contains all the graphs of before namely without merge, with merge and single threaded. Here, in this graph we can see that the one without merge is the fastest, and the one with the merge is the slowest. Whereas, the graph of the single threaded download is in the middle. From the graph we can see that, as the file size increases the time taken is increasing more in line with the merge, whereas the line without merge and single thread is increasing gradually.

In conclusion, we can see that before the line without merge and single thread is faster than that of the line that represents merge.

### **Lessons learned in the Project:**

Working on this project was an interesting experience for me. Creating a download accelerator program sounded like a really difficult program at the beginning. But as the classes passed I learned many new codes in java that I didn't know in CS181.

Among the many lessons I learned, the ones that I thought were important was multithreading, file handling and networking. I learned that using multithreading helps in making programs do multiple tasks at the same time. Likewise I learned that there are two types of networking models i.e UDP ( which is not so reliable for file transfer) and TCP ( which is not so reliable for gaming). Similarly, I learned how to partition the files into many smaller files and then merge them back together.

### **Problems and Adopted Solutions:**

I had quite a lot of problems doing this program. In Spite of having the professor's code and friends help I had many problems.

For phase one, I had a problem partitioning the files into equal parts and the extradata from the files that couldn't be divisible by 5. So, I researched a little and solved that problem by finding out that I could use append to add the remaining in one of the files that had already been made.

Similarly, I had few problems with networking and threads, I was thinking of making separate threads and separate sockets for every single thread which would be really time consuming and dirty code. So, after some googling and help from friends I completed the networking between the sockets and thread using the parameter passing arguments. Likewise, I had trouble merging the files in order as I used the join method in the wrong way. With some help I fixed that problem.

In a nutshell, this project helped me a lot , to get a wider knowledge on java programming. From this project I learned how to use multithreading, networking and file handling.