Damian Franco

101789677

dfranco24@unm.edu

CS-542

**Homework 1**

1. The first question gives us a scenario in which a company is hiring workers to build fences and railroads. This company is trying to find an efficient way to assign the workers to various tasks to maximize efficiency of production. There are three questions that the company specifically asks us below.
   1. *False*

First, the company gives us a hypothetical scenario where one worker can dig a post hole in one hour. Knowing that, we are asked if it is possible for sixty workers to dig the post hole in one minute. This seems like a logical way of thinking, but this is not true. Sixty workers would be able to dig sixty holes in sixty minutes, but sixty workers working on a singular post hole would be impossible to achieve sixty times the speed. The workers will be more utilized if they were used on a singular task. To think that all the workers can work on a singular task is impractical. This is similar to thinking that all threads in a computer could work on a singular process without overhead and data processing failures occurring. Issues will arise if all workers work on digging on hole, which is why it would be improbable to believe that sixty workers can dig a single post hole in one minute.

* 1. *False*

Next, the company gives us another hypothetical situation where a single worker can drive a single railroad spike in two minutes. The question asks us if two workers can drive a single railroad spike in one minute. This is another instance of a false statement. Two workers will not be able to drive a single railroad spike in one minute due to the same reasons stated in part (a). In instances of parallel work, we would assume that two workers working on a singular task only increases the time it would take to complete the task. I believe a better way to think about this is that the first worker will hit the railroad spike halfway in thirty minutes and the second one will also take thirty minutes to hit the other half of the railroad spike. This would still take an hour to hammer down the railroad spike and will not reduce the time it will take to finish the task. Therefore, this question that the company asks is not true.

* 1. Last, a question is proposed to us asking us what accounts for the difference in answers (a) and (b). Well, I did not have any difference in answers. I assume that there would be some reasoning to believe part (a) or (b) could be true, but with the way parallel processing was presented to us, I would like to double down on both (a) and (b) being false with no differences in answers. This is due to the fact that each task has instructions that it must follow to complete the task. Those instructions indicate how long they return and although parallel computing can compute multiple tasks at once, a singular task will not sufficiently run faster if multiple tasks are attempting to compute the same task. What I gathered from the problems in part (a) and (b) is that multiple people were trying to do the same tasks at the same time, which would not allow for any speedup in how fast the task is finished, rather that multiple tasks can be finished simultaneously.

1. This question proposes a different hypothetical scenario in which an automobile assembly line has 15 stations. At each station a portion of a car is assembled and requires six minutes to perform the task.

First we are asked how long it takes to build a single car at this automobile assembly line. I believe it would take a full amount of time to only build a single automobile. Specifically, I believe that it would take 90 minutes to take due to 15 stations\*6minutes/station.

Next, we are asked how long it would take for 10 automobiles to be produced. I used pipelining to figure out exactly how long this would take. With pipelining, each station would start on a new automobile exactly 6 minutes after the first automobile was started. For example it would still take 90 minutes to complete the first car and only 96 minutes to complete two cars due to the fact of pipelining. This would be true because all the stations start exactly 6 minutes after the first car goes into production. Knowing that we can say that it will take about 60 more minutes with a total of ***150 minutes*** for 10 cars to be made due to pipelining.

For 100 cars, we will also use pipelining which would add a total of 100\*6 minutes to the overall time of 90 minutes. I derived 90 minutes from the time it will take for the first car to be produced. A total of 600 minutes will be added to make 100 cars which would result in an overall time of ***690 minutes*** for 100 cars to be made.

Lastly, 1000 cars will also use pipelining to achieve faster production time which would add a 1000\*6 minutes to the overall time of 90 minutes for the first car to be produced. A total of 6000 minutes will be added to make 1000 cars which would result in an overall time of ***6090 minutes*** for 1000 cars to be made.

1. For this question, we are given two statistical values for two ovens. The first (conventional) oven can bake one potato in one hour or six potatoes in one hour. The second (microwave) oven can bake one potato in ten minutes and six potatoes in one hour. Each oven has very different behavior and it is our job to describe these behaviors in terms of the physical processes involved and use analogies involving serial and parallel computers.

The conventional oven physical processes allow a single potato or six potatoes to cook fully at one hour due to the fact of how these ovens work. In a conventional oven, the heating occurs by heat released underneath the food and around the oven’s cavity space which cooks the food faster and more evenly. Heat will rise and cook the food from underneath, much similar to a stove top. It is also worth noting that this heat source is fixed and even across the entire bottom of the oven. This means that one potato will cook the same as multiple potatoes with the same amount of time to cook a single potato as multiple potatoes. This is effectively describing some parallelism in the cooking of potatoes. An analogy I can draw that is similar to a conventional oven relates to the number of processes a computer has and the amount of time it will take to run a task. Let's say you have a single process that does some simple math arithmetic and takes 0.5 seconds to compile and complete the task on core 1. Now you want to compute the same math arithmetic multiple times in parallel with the first one on core 2, core 3 and core 4. So you will assign each process on separate cores and compile all in parallel. Each process will have close to or exact same compile time and runtime (0.5 seconds). This is the same premise as the potatoes cooking in the conventional oven.

The microwave oven's heating process is much more complicated than the conventional oven. Inside a microwave oven, microwaves are produced by an electron tube and then reflected within the metal interior. When the microwaves get absorbed by the food, the water molecules in the food will begin to vibrate which will then cause the food to heat and cook. One potato can cook very fast in a microwave oven because it is the only object in the microwave and is able to absorb all the microwaves that are produced by the electron tube. More microwaves absorbed means the quicker the food will heat and fully cook. When you put six potatoes into a microwave oven, that lessens the amount of microwaves each potato absorbs which will result in a much longer cook time. This results in parallelism within the microwave oven, but not similar parallelism as the conventional oven. For example, let's say we have a single process that does some complex math arithmetic that runs in 0.5 seconds on core. Let’s now add more processes to run on the same core, but all of these programs' runtime is highly dependent on how many processes are being run at the same time. This leads to some sort of dependency between some processes which can then result in longer runtime for the overall goal. Resources within the computer cannot fully parallelize all processes because there are not enough available resources to be efficient for every process. This is similar to what the microwave oven is experiencing. Each potato needs a certain amount of microwaves to be absorbed to cook fast, but cannot get all the microwaves needed which will result in a longer time to cook until the potatoes do receive all the microwaves needed to fully cook.

1. There are four composers given to use but all four live in different cities. The composers collaborate on a string quartet that has four movements. The question asks us to discuss how these composers should arrange to get the best final outcome. Through all three hypothetical cases I believe each composer should put in a similar or same amount of work into composing their movement. All composers should communicate well with the other composers but not over or under communicate with the others. This would allow a great balance of collaboration, but not fully overindulging or under performing. I believe that this would allow for the best outcome in their string quartet piece. Below, I will go into more detail of the specific hypothetical cases that are proposed to us.
   1. In this case, all the composers are given each a movement to write. This arrangement would fit the outline I presented in the first paragraph. As long as each composer is doing their work, then there are no strong dependencies on the other composers. I will be assuming that each composer will put in the same amount of effort as the others. If one composer is lacking in workload, then they do not intensely ruin the amount of time it would take to finish a string quartet. As far as quality goes, this is a way of having great quality within each movement but that does not mean that all of the movements are going to be cohesive. Composers must share information with each other to keep the semi-cohesiveness of the entire piece together. Communication is key here for cohesiveness, but does not intensely slow down the overall finish time of the entire string quartet piece. Overall, this arrangement is very good for collaboration between each composer and seems to have little problems (if the composers provide the same amount of effort) in my opinion.
   2. Next, we are given a case where each composer writes only certain instrumental section parts within all four movements. Two are in charge of writing violin parts, one writes the viola part, and the last writes the cello part of each section. Once again, I will be assuming that all composers put in the same amount of effort in their parts as the others. Now, this can cause some problems with cohesiveness and runtime in my opinion. I say this because each composer will have multiple dependencies based upon the other composer they are working with. Each composer must have very strong communication with other composers to keep the cohesiveness of each movement or it would be a disorganized mess. This also leads me to believe that all composers will have a tough time if all worked on the movements concurrently. The amount of time it will take for all the movements to get finished may be quicker, but it will be at the cost of a bad overall product. Although I do believe this would be a disaster if all composers worked in the arrangement, I also do have a good suggestion. The arrangement could possibly benefit from pipelining where composer one starts first and then composer two starts some time after and so on. This would allow all the composers to work on their parts of the pieces with success and cohesiveness at the cost of a bit of more time spent till the finished product. Overall, this arrangement seemed horrible to me, but after some insight about some parallel processing techniques, this could possibly be a great approach.
   3. This last arrangement breaks down the work even further by assigning each composer the task of writing notes. In this hypothetical situation, there are only two composers. The first writes all the first notes played on the first string for four instruments and the second composer writes all the second notes played on the four instruments as well. This is a smaller task which would be very difficult to manage through parallel work. I believe there is no good outcome if both composers work concurrently with the other. The finished outcome will be a jumbled mess due to the fact of almost to no cohesiveness within the pieces. Even if the composers have very strong communication, it would still be terribly difficult to pull this arrangement off. Pipelining would somewhat help, but not as much as the scenario given in part (b). Regarding runtime, I believe that the movements will be able to be written relatively quickly compared to other arrangements. Overall, I believe that this arrangement has too small of a task that we assign the composers which will then result in an unfortunate outcome. Even if they all put in the same amount of time and effort then it will still result in something not well.