**HOME ACTIVITY#01**

**Name:** Saman Khan

**ID:** 19K-0354

**Section:** G

**Question #2:**

* In coarse-grained parallelism, a program is decomposed into a small number of large tasks Due to this, a large amount of computation takes place in processors. This might result in load imbalance, which means certain tasks process the bulk of the data while others might be idle/not as bulky in terms of computation. Load-balancing is improper in coarse-grained parallelism. Moreover, coarse-grained parallelism does not exploit the parallelism in the program as most of the computation is performed sequentially on a processor. The benefit of this type of parallelism is low communication overhead and synchronization overhead. Message-passing architecture takes a long time to communicate data among processes which makes it suitable for coarse-grained parallelism.
* We know that in fine-grained parallelism, a program is decomposed into a large number of small tasks. These tasks are then allocated individually to many processors. Since there are many small tasks each of them has same amount of work meaning the work is evenly divided among the processors also the amount of work associated with each of these tasks is low. Since no task is too bulky or too idle, fine-grained parallelism facilitates the concept of load-balancing. As each task processes less data, the number of processors required to perform the complete processing is high which results in an increase in communication and synchronization overhead. Fine-grained parallelism is best suited in architectures which support fast communication. Shared memory architecture which has a low communication overhead is most suitable for fine-grained parallelism.
* As already mentioned above that fine-grained parallelism is best suited in architectures which support fast communication. Since Message passing architecture takes long long time to communicate data among processes which makes it unsuitable for fine-grained parallelism. On the other hand, coarse-grained parallelism has low level of parallelism as most of the computation is performed sequentially on a processor which means shared memory architecture not be as good of an option because it supports maximum parallelism.