Need for Parallel Computers

Dr.N.Sairam & Dr.R.Seethalakshmi
School of Computing,
SASTRA Univeristy,
Thanjavur-613401.

Contents

1. Need for Parallel Computers

The computer era has a very long history

- 1642 – Blaise Pascal’s Calculating machine
- 1812-1832 Charles Babbage’s difference engine
- 1853-1871 Analytical engine
- 1930-1940 Electro mechanical computers
- 1940- First electronic computer
- 1950s -compact systems
- 1986- till date - Parallel and distributed systems

1.1 Uniprocessor Systems

First Personal computers were introduced. These computers were compact,but only could execute one instruction at a time, since there is only one processor. It is a single user system and it is dedicated system. In a single processor system there will never be more than one running process as shown in figure 1 where the tasks task1,task2,task3 and task4 are executed in sequence. If there are more processes, the rest will have to wait until the CPU is free. Further if the currently executing process is waiting for the I/O then there can be a context switch to the waiting process. Multiprocessing can occur in some operating systems. If it is a personal computer then the processes have to run one after the other.

1.1.1 Disadvantages of Uniprocessor system

Sufficient time is wasted when the process waits for i/o. If the system crashes then there is none to replace it. The uniprocessor system provides slow performance. Further the hard drive is small and we can feed only small input data. The same hard drive is used for all the processes and there is a chance that the hard drive easily crashes and the data is lost. There is a limitation to the RAM that can be supported by a PC. One cannot work on imaging and gaming applications with PC. Even if it is executed there will be a lot of struggle. The motherboard of the PC can easily get crashed. This is yet another problem. Next issue is on the power supply. We have to build a reliable power supply. Most PCs are to be supported by the SMPS which again adds to the cost. With PC, database applications cannot be run efficiently. There are limitations to both virtual memory and hard disk. Hence we migrate to parallel systems.

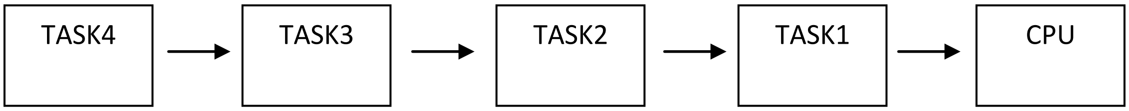

Figure 1 Uniprocessor Representation

1.2 Parallel systems

Parallel processing is much faster than sequential processing when it comes to doing repetitive calculations on vast amount of data. This is because a parallel processor is capable of multithreading on a large scale, and can therefore simultaneously process several streams of data. The one advantage of parallel processing is that it is much faster for simple, repetitive calculations on vast amounts of similar data. If a difficult computational problem needs to be attacked, where the execution time of program code must be reduced, parallel processing may be useful, which is as shown in figure 2.Some (not all) problems may be subdivided into pieces. If so, the pieces may be simultaneously processed by multiple processing units. Examples of such divisible problems include graphics, graphs, mathematical problems like sets and signal processing. The advantage is the processing time is reduced, possibly by up to 1/n (for n processors). Another advantage is the power consumption. As the electrical power used by a processor generally increases by the square of its switching speed, power-sensitive applications can reduce power use by compensating for reduced switching speed with additional computing units. For example, halving the switching speed and doubling the processing elements results in half the power use for about the same computing ability. Generally, parallel processing refers to the division of a single computation problem into pieces for simultaneous work, where using multiple computation units for multiple unrelated problems is just multiprocessing. Although parallel processing is often done in a single machine with multiple processors, it can also be done among multiple computers connected with a network. Although most current PCs with 2 or more cores are capable of running parallel processing programs, complex problems, like weather forecast or climate simulation, are often run on specialized computers with many CPUs (possibly thousands). In recent years cluster technology, such as linux "beowulf" clusters, has enabled connecting generic PCs for parallel processing, an inexpensive alternative to expensive special-purpose equipment. In software, the use of threads is a common way of enabling the user program to make use of the computer's parallel processing abilities. A process is a single running program on computer, consisting of a memory space in the computer inhabited by a thread. Modern computer operating systems allow the user to write programs to create multiple threads to achieve the desired goal. It is up to the programmer to create and organize these threads, assure they will not step on each other's toes, and will receive, process, and deliver their assigned work.

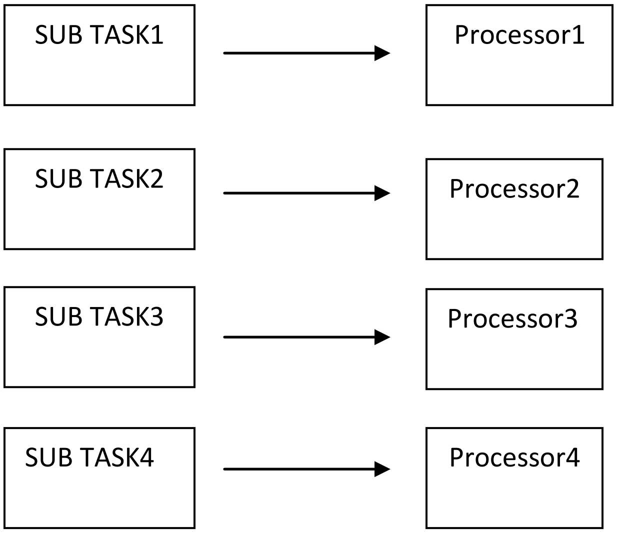

Figure 2 Parallel Processing Representation