Project A3

Tek Acharya

900758663

GSU-Spring 2020

TASK3a. Parallel Programming Skill(97P)

1. **Define the flowing: Task, Pipelining, Shared Memory, Communications, Synchronization (in your own words) (5P)**

**Task**

A task is a set of programs/instructions that a processor can execute. In other words, any executable file to the microprocessor is a task to it.

**Pipelining**

Pipelining is a type of parallel computation in which a task is broken into several individual tasks by the processor by streaming the inputs, much like an assembly line.

**Shared Memory**

A shared memory, from a hardware point of view, is a computer architecture in which a common memory location is shared between several processors through a bus. And in a programming point of view shared memory is a common data reservoir accessing through instructions regardless of where the memory is located.

**Communication**

The exchange of data during parallel computation is commonly referred to as communication.

**Synchronization**

Closely associated with communication, the collaboration of parallel tasks in real-time, is called synchronization. This is implemented by generated by creating a synchronization point. Often next task many not proceed until one is done creating a wait increasing the wall clock execution.

1. **Classify parallel computers based on Flynn’s taxonomy. Briefly describe every one of them (8P)**

The classification of parallel computers based on Flynn’s taxonomy is briefly described below.

1. Single Instruction stream, Double Data stream (**SISD**):

A very classic non-parallel computer in which single instruction is acted on by CPU during a clock cycle that uses single data as its input. This type is basically the first-generation computer which is very deterministic in execution. Example of such computer type includes minicomputers, workstations, and single-processor/cores Pcs.

1. Single Instruction, Multiple Data (**SIMD**):

A basic type of parallel computer which uses single instruction but can execute multiple data element at any clock cycle. Most suited for graphics processors units (GPU), this type comes with two varieties: Processor Array and Vector Pipelines. An example includes Thinking Machines CM-2, IBM 9000, Hitachi S820, etc.

1. Multiple Instruction, Single Data (**MISD**):

This also is a type of parallel computer in which each processing unit operates on the data independently via separate instruction using single data. This type of computer system is rare, it is believed that the malware programmer uses this type trying to decrypt the information to heck.

1. Multiple Instruction, Multiple Data (**MIMD**):

This is also a type of parallel computer which is the most modern. This type of computer architecture, every individual processor will be running a different instruction using their own data stream. This type of system works synchronous or asynchronous, deterministic or non-deterministic. Examples include intel core i7, supercomputers.

1. **What are the parallel Programming Models? (7P)**

There are several parallel programming models in common use. Some of them are:

1. Shared Memory (without thread)
2. Threads
3. Distributed Memory / Message Passing
4. Data-Parallel
5. Hybrid
6. Single Program Multiple Data (**SPMD**)
7. Multiple Program Multiple Data (**MPMD**)
8. **List and briefly describe the types of Parallel Computer Memory Architectures. What type is used in OpenMP and why? (12P)**

The list along with a brief description of Parallel Computer Memory Architecture is done bellow.

1. Uniform Memory Access (**UMA**)

This type of parallel memory architecture represented mostly by Symmetric Multiprocessor (**SMP**)machines with identical processors. In this architecture, each execution uses equal access and access time to the common memory location. This type of architecture also uses a coherent cache memory system in which an update on any one of the processors leads to update to the rest of the processors. The term is often called as **CC-UMA** (Cache Coherent UMA). This way processors remain vigilant at each time an execution occurs.

1. Non-Uniform Memory Access (**NUMA**)

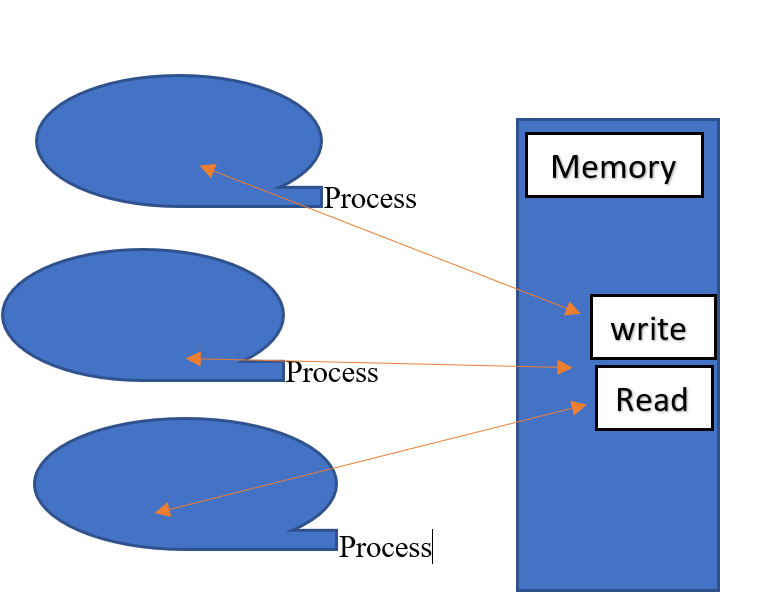
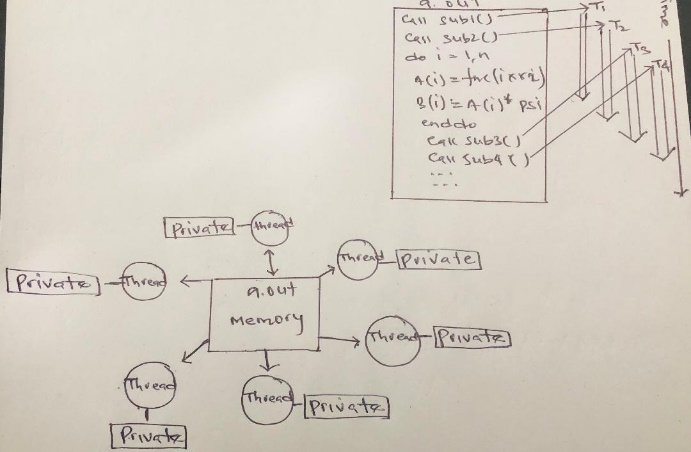
This type of architecture is often made by physically linking two or more SMPs where one SMP can access the other one directly, though it is a slower method. Not all processors have equal access time to all memory thus minimizing the error. If Cache coherency is maintained, then may also be called **CC-NUMA**; Cache Coherent NUMA

* Advantages:
* Global address space provides a user-friendly programming perspective to memory.
* Data sharing between tasks is both fast and uniform due to the proximity of memory to CPUs.
* Disadvantages:
* The lack of scalability between memory and CPUs. Adding more CPUs increases traffic to the signal transfer thereby creating performance reliability.
* Programmers are more responsible for correct access to global memory.

OpenMP is designed for multi-processor/core, shared memory machines. The underlying architecture can be shared memory UMA or NUMA. Because OpenMP is designed for shared memory parallel programming, it largely limited to single-node parallelism

1. **Compare Shared Memory Model with Thread Model (in your own word and show pictures) (10P)**

|  |  |
| --- | --- |
| Shared Memory Model (**SMM**) | Threaded Memory Model (**TMM**) |
| This programming model shares a common address space to read and write to. | This model is a special type of SMM |
| This is a basic type of memory model | In this type, a single thread process can have multiple execution paths. |
| Difficult to understand and manage data locality while implementing | The implementation using OpenMP is user-friendly and potentially fewer errors. |



Threaded Memory Model

Shared Memory Model

1. **What is parallel programming (in your own words) (5P)**

A programming technique that is used to perform several computations or execution simultaneously. In this programming, several tasks are performed at a time. This is possible due to the involvement of multiple processors/cores that are used concurrently. This method has met the demand of modern computational need. It is fast and more accurate while using this technique.

1. **What is a system on chip (SoC)? Does Raspberry PI use the system on SoC? (5P)**

System on chip is a modern approach of reducing the size of a computer by integrating CPU (Central Processing Unit), RAM (Random access memory), or GPU (Graphics Processing Unit) and many other components essentially an entire computer into a single chip. Yes, the Raspberry PI is a SoC

1. **Explain what the advantages are of having a system on chip rather than having separate CPU, GPU and RAM components(5P)**

The most advantage of having SoC is its highly reduced size. That means portability and less expensive. Due to the high level of integration and much shorter wiring, an SoC uses considerably less power. Also, mobile computing will hike as everything will be handheld. The rate at which data are processed is also significantly increased due to this integration.