CSCI 6360: Parallel Computing Lecture Summary - 14

Anirban Das (dasa2@rpi.edu)

March 22, 2018

Summary on pthreads and Origin 2000 paper in Lecture 14:

The lecture explores multithreading concept using pthreads, mostly. Threads are like isolated streams of control in the flow of a main program, having access to a shared memory space and can be independently scheduled. Pthreads is specific implementation of multithrading architecture in POSIX standard.

Many a times, program or process can be implemented as such, that its execution can be broken into several parts all executing concurrently, ideally without interfering with eachother. If this is the case , a program can be made to be multithreaded. Threads can be thought of light weight processes. All threads spawned from a call executes the same set of instructions. The threads in a logical machine model, has global access to all memory, and hence can access the data concurrently.

The goal is to take advantage of this concurrency to achieve low latency in computing and waste less CPU cycles idling. For example in a single program , one thread can handle the I/O while the other thread can do background computations without wasting CPU. Moreover, sometimes having multiple threads results in fault tolerance of the main program compared to single threaded implementations. For e.g. if multiple threads from same program is connected to some sockets, if one socket dies, then only that thread dies, but rest of the program can be functional.

However, synchronization quickly becomes an issue. Multiple threads trying to access same memory space may cause deadlocks. For example some parts of the code needs to be 'critical', i.e. only one process should execute that at a time. This kind of mutual exclusion can be achieved using Mutexes and read-write locks at the penalty of some overhead. Also, barrier can be used for the same process and all these are available in the pthreads API.

The Origin 2000 paper, describes the architecture of Silicon Graphics Origin 2000 multiprocessor and its performance on NAS Parallel Benchmark, STREAM and SPLASH2. Origin is a distributed shared memory system also known as shared memoru multiprocessing architecture in the paper. All memory and IO subsystem is globally addressable from all nodes, with a non-blocking cache coherence protocol. STREAM benchmark reveals that memory bandwidth of a single thread per node offers almost linear scaling because each processor effectively utilised more than half memory bandwidth per node. But running two threads pernode achieves almost similar scaling performance and this is due to the shared memory architecture of Origin.