Parallel Programming Models
CS 540- High Performance Computing

Spring 2017

Programming Models

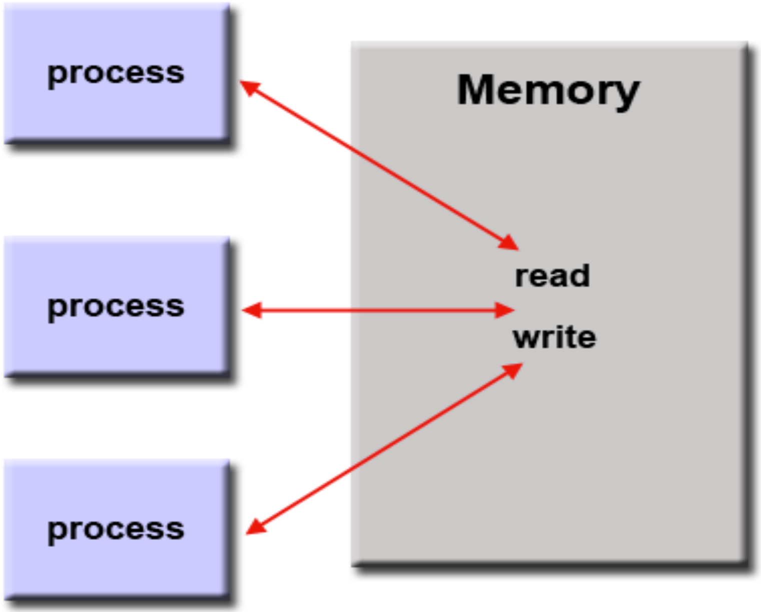
- Shared Memory (without threads)
- Threads
- Distributed Memory / Message Passing
- Data Parallel
- Hybrid
- Single Program Multiple Data (SPMD)
- Multiple Program Multiple Data (MPMD)
- Parallel programming models are an abstraction layer above hardware and memory architectures.

Programming Models

Shared Memory (without threads)

- processes/tasks share a common address space, which they read and write to asynchronously.
- We have mechanisms such as locks / semaphores, which are used to control access to the shared memory,
- prevent race conditions and deadlocks.
- On shared memory machines, native operating systems provide support for shared memory programming.

Programming Models



Source: Lawrence Livermore National Laboratory

Programming Models

Threads Model

- is a type of shared memory programming.
- In the threads model of parallel programming, a single process (heavy weight) can have multiple threads (light weight)
- POSIX Threads
- OpenMP

Programming Models

POSIX Threads

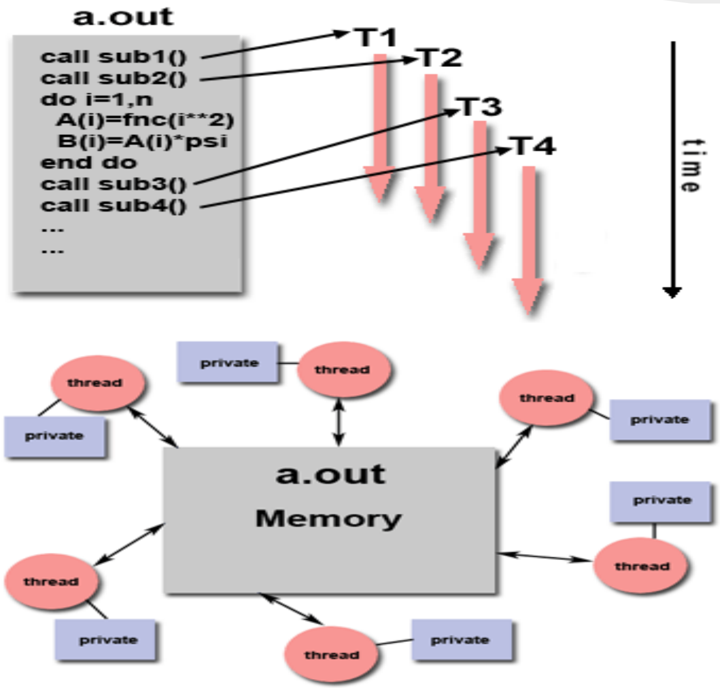
- Specified by the IEEE POSIX 1003.1c standard (1995). C Language only.
- Part of Unix/Linux operating systems
- Library based
- Commonly referred to as Pthreads.
- Very explicit parallelism; requires significant programmer attention to detail.

Programming Models

OpenMP

- Industry standard, jointly defined and endorsed by a group of major computer hardware and software vendors.
- Compiler directive based
- Portable / multi-platform, including Unix and Windows platforms
- Available in C/C++ and Fortran implementations
- Can be very easy and simple to use - provides for "incremental parallelism". Can begin with serial code.

Programming Models



```
call sub1()
call sub2()
do i=1,n
  A(i)=fnc(i**2)
  B(i)=A(i)*psi
end do
call sub3()
call sub4()
...
...
```

Source: Lawrence Livermore National Laboratory

Programming Models

OpenMP

- Industry standard, jointly defined and endorsed by a group of major computer hardware and software vendors.
- Compiler directive based
- Portable / multi-platform, including Unix and Windows platforms
- Available in C/C++ and Fortran implementations
- Can be very easy and simple to use - provides for "incremental parallelism". Can begin with serial code.

References

1. Blaise Barney, Lawrence Livermore National Laboratory, https://computing.llnl.gov/tutorials/parallel_comp/#Overview
2. https://en.wikipedia.org/wiki/Flynn's_taxonomy
3. https://www.citutor.org/index.php, Parallel Computing Explained