Introduction to OpenMP

EAS 520
High Performance Scientific Computing

University of Massachusetts Dartmouth

Spring 2015

References

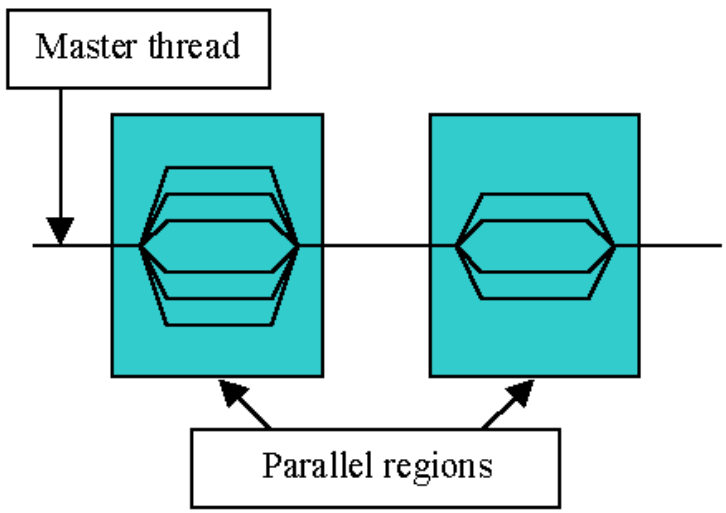
This presentation is almost an exact copy of Dartmouth College's openMP tutorial. The link can be found in:

`http://www.dartmouth.edu/~rc/classes/intro_openmp/`

Changes from the original document are related to compilers and job submissions for UMass Dartmouth clusters.

OpenMP Features

- Used for multi-threaded parallel processing
- Used on shared-memory multi-processor (core) computers
- Part of program is a single thread and part is multi-threaded
- Has 3 components
  - directives that can be put into C/C++ or Fortran programs
  - runtime library for setting and querying parallel parameters (ex. # of threads)
  - environment variables that define runtime parallel parameters (ex. # of threads)



An example of a **directive** in Fortran

```
call omp_set_num_threads(nthread) !requests "nthread" threads
!$omp parallel do
 do i=1,N
 do j=1,M
 .
 .
 end do
 end do
!$omp end parallel do
```

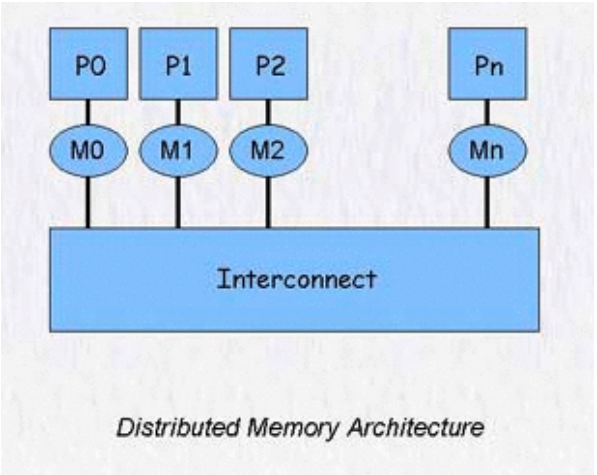
An example of a **directive** in C

```
omp_set_num_threads(nthread); /* requests nthread threads */
#pragma omp parallel for
{
 for (i=0; i<n; i++) {
 for (j=0; j<m; j++) {
 .
 .
 }
 }
}
```

Memory Architectures and Parallel Programming

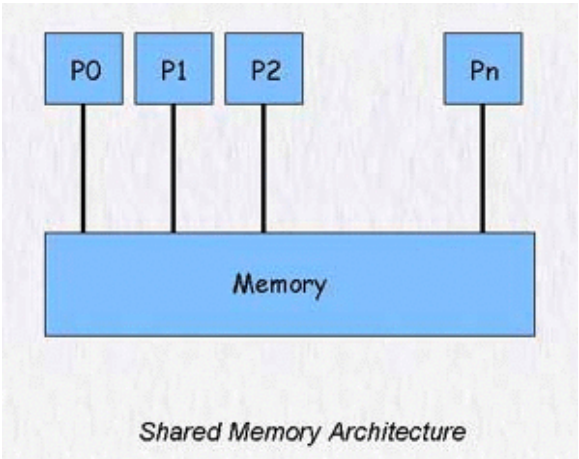
Distributed Memory

- each processor has its own memory
- parallel programming by message passing (MPI)



Shared Memory

- processors shared memory
- two parallel programming approaches
  - message passing (MPI)
  - directives-based interface - OpenMP



Pros and Cons of OpenMP

Pros

- Prevalence of multi-core computers
- Requires less code modification than using MPI
- OpenMP directives can be treated as comments if OpenMP is not available
- Directives can be added incrementally

Cons

- OpenMP codes cannot be run on distributed memory computers (exception is Intel's OpenMP)
- Requires a compiler that supports OpenMP (most do)
- limited by the number of processors available on a single computer
- often have lower parallel efficiency
    - rely more on parallelizable loops
    - tend to have a higher % of serial code
    - Amdahl's Law - if 50% of code is serial will only half wall clock time no matter how may processors

Example OpenMP Hello World Program (C)

```
#include <omp.h>
#include <stdio.h>
#include <stdlib.h>
int main (int argc, char *argv[]) {

int nthreads, tid;

/* Fork a team of threads giving them their own copies of variables */
#pragma omp parallel private(nthreads, tid)
 {

 /* Obtain thread number */
 tid = omp_get_thread_num();
 printf("Hello World from thread = %d\n", tid);

 /* Only master thread does this */
 if (tid == 0)
 {
  nthreads = omp_get_num_threads();
  printf("Number of threads = %d\n", nthreads);
 }

 } /* All threads join master thread and disband */
}
```

Fortran example:

```
program helloomp
use omp_lib
implicit none
integer nthreads, tid, ncores

! Set the number of cores as opposed to using export command
ncores = 8
call omp_set_num_threads(ncores)

! Fork a team of threads giving them their own copies of variables
!$omp parallel private(nthreads, tid)

! Obtain thread number
tid = omp_get_thread_num()
write(*,*) 'hello world from thread = ', tid

! Only master thread does this
if (tid .eq. 0) then
  nthreads = omp_get_num_threads()
  write(*,*) 'number of threads = ', nthreads
end if

! All threads join master thread and disband
!$omp end parallel
end program helloomp
```

Loop level Parallelization

Requirements for Loop Parallelization

- no dependencies between loop indicies
- an element of an array is assigned to by at most one iteration
- no loop iteration reads array elements modified by any other dependency
- due to overhead of parallelization - use only on loops where individual iterations take a long time

Example of Code with No Data Dependencies

Fortran example

```
!$omp parallel do
   do i = 1, n
      a(i) = b(i) + c(i)
   enddo
```

C/C++ Example

```
#pragma omp parallel for
  for(i=1; i<=n; i++)
     a[i] = b[i] + c[i];
```

continued...

Example of Code with Data Dependencies

Fortran example

```
do i = 2, 5
   a(i) = a(i) + a(i-1)
enddo
```

C/C++ Example

```
for(i=2; i<=5; i++)
   a[i] = a[i] + a[i-1];
```

Examples of Applications and Libraries That Use OpenMP

Applications:

- Matlab
- Mathematica

Libraries:

- Intel Math kernel Library (MKL)
- AMD Core Math Library (ACML)
- GNU Scientific Library (GSL)