Introduction to OpenMP

EAS 520
High Performance Scientific Computing

University of Massachusetts Dartmouth

Spring 2015
Thread Control

**Barrier**

Each thread wait at the barrier until all threads reach the barrier.

**Fortran Example**

```fortran
!$omp parallel private(myid, istart, iend)
call myrange(myid, nthreads, istart, iend)
do i = istart, iend
   a(i) = a(i) - b(i)
end do
!$omp barrier
call dowork(a)
!$omp end parallel
```

**C/C++ Example**

```c
#pragma omp parallel private(myid, istart, iend)
{
   myrange(myid, nthreads, &istart, &iend);
   for(i=istart; i<=iend; i++){
      a[i] = a[i] - b[i];
   }
   #pragma omp barrier
dowork(a);
}
```
Thread Control Continued

**Master**

A section of code that runs only on the master (thread with rank=0)

**Fortran Example**

```fortran
!$omp parallel private(myid, istart, iend)
call myrange(myid, nthreads, global_start, global_end, istart, iend)
do i = istart, iend
   a(i) = b(i)
end do
!$omp barrier
!$omp master
write(21) a
!$omp end master
call do_work(istart, iend)
!$omp end parallel
```
**Thread Control Continued**

**Master**

A section of code that runs only on the master (thread with rank=0)

**C/C++ Example**

```c
#pragma omp parallel private(myid, istart, iend)
{
    myrange(myid, nthreads, global_start, global_end, &istart, &iend);
    for(i=istart; i<=iend; i++){
        a[i] = b[i];
    }
    #pragma omp barrier
    #pragma omp master
    {
        n = global_end - global_start + 1;
        write_size = fwrite(a, 1, n, file_pointer);
    }
    do_work(istart, iend);
}
```

**Single**: Similar to Master except runs only on the first thread to reach it
Thread Control Continued

Critical

- Only one thread executes a specified section of the code at a time
- Threads can execute in any order
- Similar to ORDERED directive except ordered specifies that threads go in numerical order

Fortran Example

```fortran
the_max = 0.0
!$omp parallel private(myid, istart, iend)
   call myrange(myid, nthreads, global_start, global_end, istart, iend)
   call compute_a(a(istart:iend))
   !$omp critical
     the_max = max( maxval(a(istart:iend), the_max )
   !$omp end critical
   call more_work_on_a(a)
!$omp end parallel
```
Thread Control Continued

C/C++ Example

the_max = 0.0;
#pragma omp parallel private(myid, istart, iend)
{
    myrange(myid, nthreads, global_start, global_end, &istart, &iend);
    nvals = iend-istart+1;
    compute_a(a[istart],nvals);
    #pragma omp critical
    the_max = max( maxval(a[istart],nvals), the_max );
    #pragma omp end critical
    call more_work_on_a(a)
}
Thread Control Continued

**Sections/Section**

- A section of code that is run by only one thread
- Sections are performed in parallel

**Fortran Example**

```
!$omp parallel
!$omp sections
  !$omp section
  call init_field(field)
  !$omp section
  call check_grid(grid)
!$omp end sections
!$omp end parallel
```

**C/C++ Example**

```
#pragma omp parallel
{
  #pragma omp sections
  {
    #pragma omp section
    init_field(field);
  }
  #pragma omp section
  check_grid(grid);
}
```