

Introduction to OpenMP

Dr. Christian Terboven



Task Synchronization

Dr. Christian Terboven

Introduction to OpenMP



- OpenMP `barrier` (implicit or explicit)
 - All tasks created by any thread of the current *Team* are guaranteed to be completed at barrier exit

C/C++

```
#pragma omp barrier
```



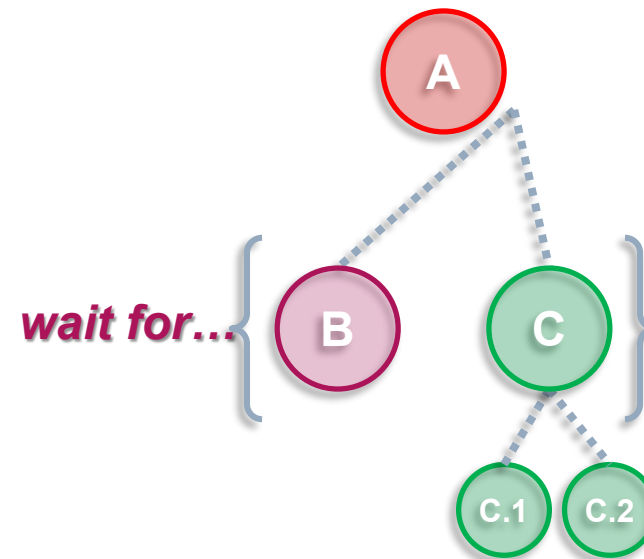
- The taskwait directive (shallow task synchronization)

- It is a stand-alone directive

```
#pragma omp taskwait
```

- wait on the completion of child tasks of the current task; just direct children, not all descendant tasks; includes an implicit task scheduling point (TSP)

```
#pragma omp parallel
#pragma omp single
{
  #pragma omp task : A
  {
    #pragma omp task : B
    { ... }
    #pragma omp task : C
    { ... #C.1; #C.2; ... }
    #pragma omp taskwait
  }
} // implicit barrier will wait for C.x
```



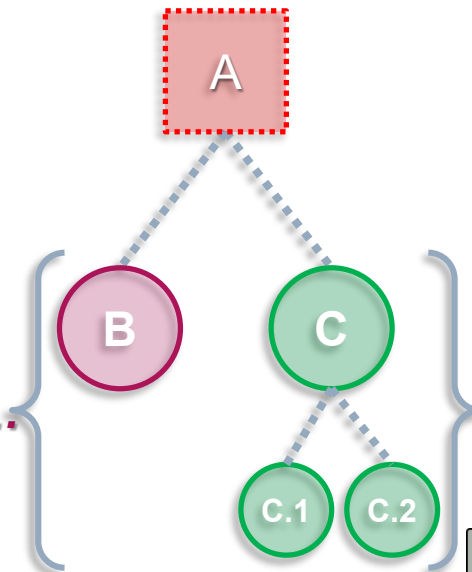
- The taskgroup construct (deep task synchronization)
 - attached to a structured block; completion of all descendants of the current task; TSP at the end

```
#pragma omp taskgroup [clause[[,] clause]...]  
{structured-block}
```

- where clause (could only be): reduction(reduction-identifier: list-items)

```
#pragma omp parallel  
#pragma omp single  
{  
  #pragma omp taskgroup :A  
  {  
    #pragma omp task :B  
    { ... }  
    #pragma omp task :C  
    { ... #C.1; #C.2; ... }  
  } // end of taskgroup  
}
```

wait for...



The Barrier and Taskwait Constructs

– Task Synchronization explained:

```
#pragma omp parallel num_threads(np)
{
#pragma omp task ←
    function_A();
#pragma omp barrier ←
#pragma omp single
{
#pragma omp task ←
    function_B();
} ←
}
```

np Tasks created here, one by each thread

All Tasks guaranteed to be completed here

1 Task created here

B-Task guaranteed to be completed here



Questions?

