Taskloop (7A)

- Task
-

Copyright (c) 2020 Young W. Lim.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using OpenOffice and Octave.

Taskloop (1)

The **taskloop** pragma is used to specify
that the iterations of one or more associated loops
are executed in parallel using OpenMP **tasks**.

The iterations are distributed across tasks
that are created by the construct and
scheduled to be executed.

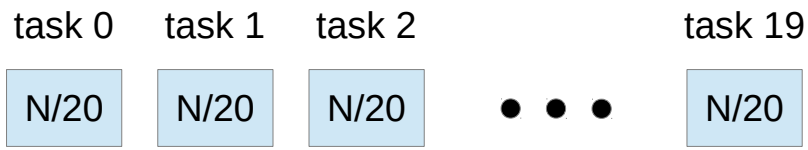
https://www.ibm.com/support/knowledgecenter/SSXVZZ_16.1.1/com.ibm.xlcpp1611.lelinux.doc/compiler_ref/prag_omp_taskloop.html

Taskloop (2)

The **taskloop** construct generates as many as 20 tasks.
num_tasks(20)

The iterations of the for loop are distributed
among the **tasks** generated for the **taskloop** construct.

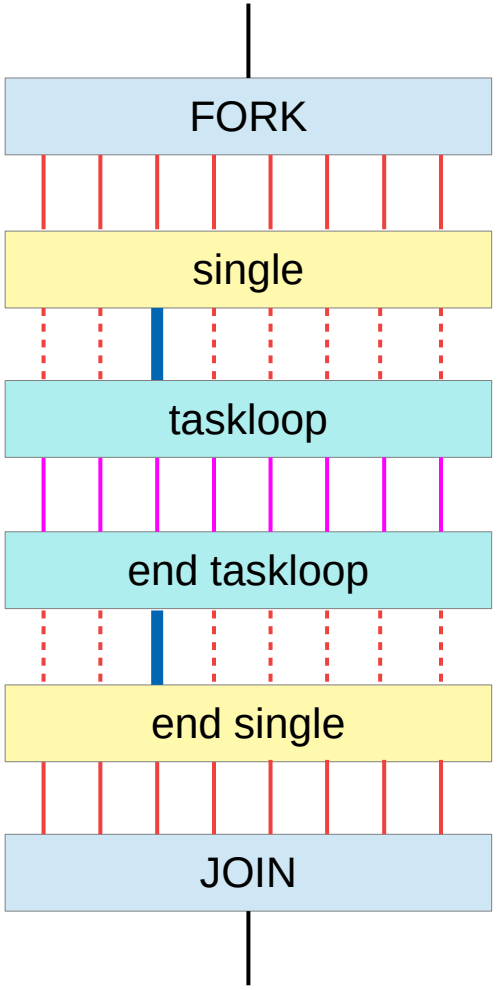
```
#pragma omp parallel
#pragma omp single                          // only one process performs taskloop
#pragma omp taskloop num_tasks(20)
 for (i=0; i<N; i++) {
   arr[i] = i*i;
 }
```



https://www.ibm.com/support/knowledgecenter/SSXVZZ_16.1.1/com.ibm.xlcpp1611.lelinux.doc/compiler_ref/prag_omp_taskloop.html

Taskloop (3)

```
#pragma omp parallel
#pragma omp single
#pragma omp taskloop num_tasks(20)
  for (i=0; i<N; i++) {
    arr[i] = i*i;
}
```



https://www.ibm.com/support/knowledgecenter/SSXVZZ_16.1.1/com.ibm.xlcpp1611.lelinux.doc/compiler_ref/prag_omp_taskloop.html

taskloop

```
Int main (int argc, char* argv[])
{
      ***
   #pragma omp parallel
   {
      #pragrma omp single
      {
         fib(input);
      }
   }
      ***
}
```

```
Int fib(int n)
{
   if (n < 2) return n;
   int x, y;

   #pragma omp task shared(x)
   {
      x = fib(n-1);
   }
   #pragma omp task shared(y)
   {
      y = fib(n-2);
   }
   #pragma omp taskwait;
   {
      return x+y;
   }
```

https://pop-coe.eu/sites/default/files/pop_files/pop-webinar-openmptasking.pdf

References

[1] en.wikipedia.org
[2] M Harris, http://beowulf.lcs.mit.edu/18.337-2008/lectslides/scan.pdf