

Parallel Programming using OpenMP

Malik Lechekhab

Università della Svizzera italiana

October 5, 2022

What is OpenMP?

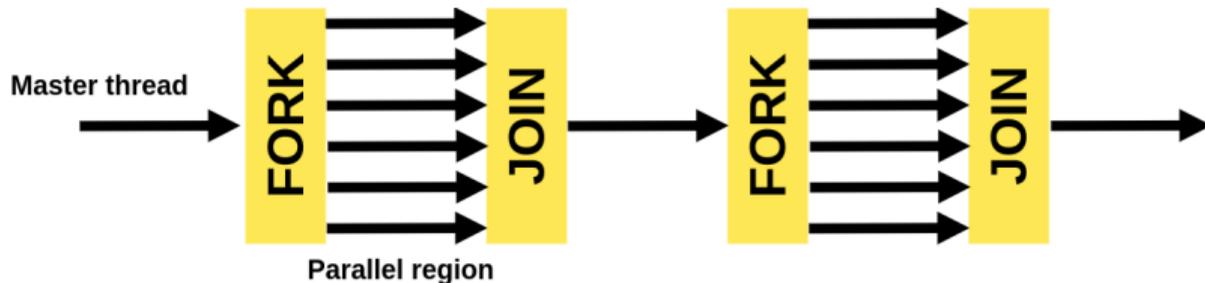
- API for writing multi-threaded applications
- Set of compiler directives and library routines for parallel application developers
- Simplifies writing multi-threaded programs in Fortran, C and C++
- Standardizes last decades of symmetric multi processing

Why use OpenMP?

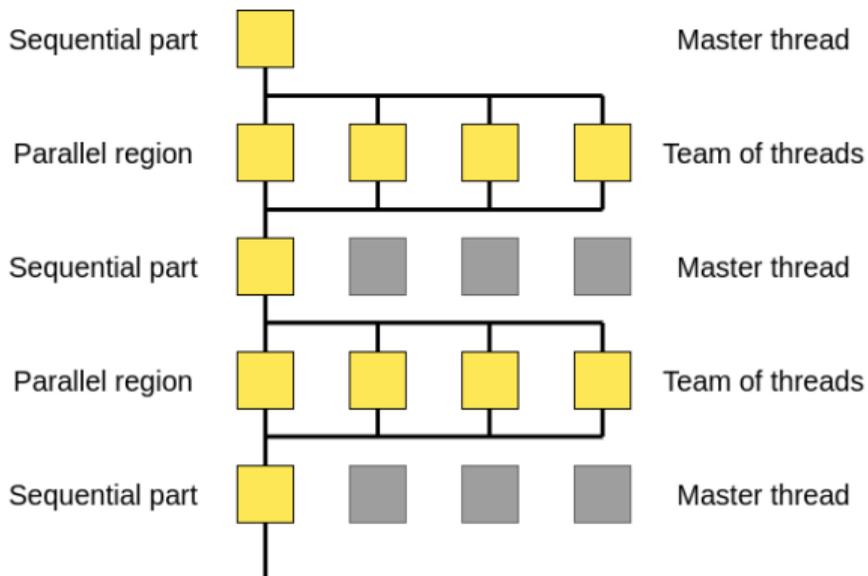
- Reducing MPI communication
- Improving scaling by exploiting
- Targeting new architecture

Fork-and-join model

- OpenMP programs begin as a single process called the master thread
- A team of threads is spawn when the parallel region is reached



OpenMP execution model



Creating parallelism

- Most constructs in OpenMP are compiler directives or pragmas
- Possibility to create portable code

```
#pragma omp directive
```

- Thread groups are created with the parallel directive

```
#pragma omp parallel
    /* structured block */
/* omp end parallel */
```

- The OpenMP ARB Fortran and C APIs: www.openmp.org
- NCSA online course on OpenMP:
www.hpc-training.org/xsede/moodle/enrol/index.php?id=43
- OpenMP tutorial from Lawrence Livermore National Laboratory:
hpc.llnl.gov/tuts/openMP/
- OpenMP Intel tutorial: youtu.be/cMWGeJyrc9w
- RohitChandra, et.al., "Parallel Programming in OpenMP" Morgan Kaufmann, ISBN 1-55860-671-8

- Take an array of integers and do some operation on each of them
- Single thread way:

```
int i;  
for(i=0; i<ARRAY_SIZE; i++){  
    /*do some operation*/  
}
```

- Let's add multi-threading. The most obvious and wrong way is:

```
#pragma omp parallel
int i;
for(i=0; i<ARRAY_SIZE; i++){
    /*do some operation*/
}
```

- That doesn't work. Why?

- Let's add multithreading. The most obvious and wrong way is:

```
#pragma omp parallel
int i;
for(i=0; i<ARRAY_SIZE; i++){
    /*do some operation*/
}
```

- We do loop operations in all available threads for the same value i

- All we need to fix that is to surround the loop with another omp block:

```
#pragma omp parallel
{
    #pragma omp for
    {
        int i;
        for(i=0; i<ARRAY_SIZE; i++){
            /*do some operation*/
        }
    }
}
```

- Alternatively, we can also combine different blocks:

```
#pragma omp parallel for
{
    int i;
    for(i=0; i<ARRAY_SIZE; i++){
        /*do some operation*/
    }
}
```

- Available on iCorsi and Github
- Due date: 26 October 2022, 11:59 p.m.
- Remember to work on compute nodes:

```
$ module load gcc
$ make
$ salloc --exclusive
$ export OMP_NUMBER_THREADS = t
$ ./yourProgram
```