Università della Svizzera italiana

Parallel Programming using OpenMP

Malik Lechekhab

Università della Svizzera italiana

October 5, 2021

Università della Svizzera italiana

What is OpenMP?

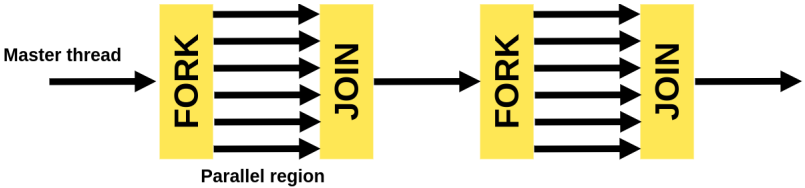
- API for writing multi-threaded applications
- Set of compiler directives and library routines for parallel application developers
- Simplifies writing multi-threaded programs in Fortran, C and C++
- Standardizes last decades of symmetric multi processing

Why use OpenMP?

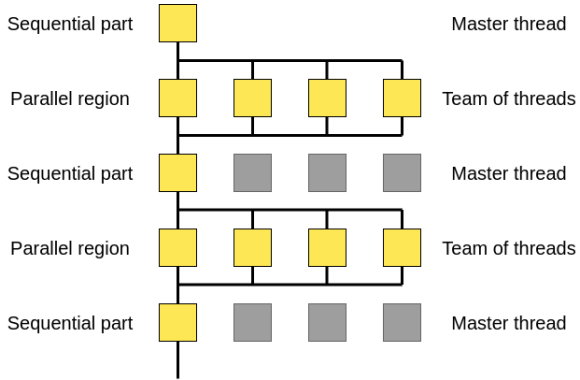
- Reducing MPI communication
- Improving scaling by exploiting
- Targeting new architecture

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

```
1          #pragma omp directive
```

- Thread groups are created with the parallel directive

```
1          #pragma omp parallel
2              /* structured block */
3          /* omp end parallel */
```

Università della Svizzera italiana

Information

- 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

Examples

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

```
1       int i = 0
2       for(i; i<ARRAY_SIZE; i++){
3           /*do some operation*/
4       }
```

Examples

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

```
1       #pragma omp parallel
2       int i = 0
3       for(i; i<ARRAY_SIZE; i++){
4           /*do some operation*/
5       }
```

- That doesn't work. Why?

Examples

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

```
1       #pragma omp parallel
2       int i = 0
3       for(i; i<ARRAY_SIZE; i++){
4           /*do some operation*/
5       }
```

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

Examples

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

```
1       #pragma omp parallel
2       {
3           #pragma omp for
4           {
5               int i = 0
6               for(i; i<ARRAY_SIZE; i++){
7                   /*do some operation*/
8               }
9           }
10      }
```

Examples

- Alternatively, we can also combine different blocks:

```
1       #pragma omp parallel for
2       {
3           int i = 0
4           for(i; i<ARRAY_SIZE; i++){
5               /*do some operation*/
6           }
7       }
```

Università della Svizzera italiana

Project 3

- Available on iCorsi and Github
- Due date: 24 October 2021, 11:59 p.m.
- Remember to work on compute nodes:

```
1  $ module load gcc
2  $ make
3  $ salloc --exclusive
4  $ export OMP_NUMBER_THREADS = t
5  $ ./yourProgram
```