High Performance Computing for Data Science

Theses opportunities, useful tips, fInal remarks & project activity
Lecture24 - 01/12/2023

Prof. Sandro Fiore, Ph.D.
Department of Information Engineering and Computer Science (DISI)
University of Trento, 2023-2024

Theses opportunities (I)

Distributed caching systems for federated climate repositories

- The thesis's work will support
  - Automated "data space" instantiation in the cloud
  - Fast access to most-used datasets
  - Real testbed with ESGF nodes in EU, US and Australia
- Integration of the caching system with parallel data transfer systems & novel file catalogs

Theses opportunities (II)

NFT & Blockchain convergence for large-scale climate repositories

- The thesis’s work will support
  - NFT exploitation for scientific data
  - Integration of NFT and blockchain for climate data management purposes

Theses opportunities (III)

Hierarchical scheduling for exascale applications

- The thesis's work will support
  - Design of hierarchical scheduling approaches to tackle exascale
  - Application of hierarchical scheduling to existing systems & frameworks
    - In particular the analysis will target «Dask» scheduler extensions

Theses opportunities (IV)

On the integration of workflow systems and provenance services

- The thesis’s work will support
  - Extension of workflow management systems with provenance capabilities
    - Streamflow workflow management system
    - yProv service (for provenance tracking)

Theses opportunities (V)

On the extension of the yProv service with blockchain-based capabilities

- The thesis's work will support
  - Extension of provenance service with blockchain-based capabilities
    - Trustworthy provenance management
    - yProv service (for provenance tracking) and blockchain middleware integration

Theses opportunities (VI)

Expanding the sound speed dataset analysis to more advanced and novel directions (collaboration with OGS)

- The thesis's work can focus on different topics:
  - Sound speed dataset parallel analysis (more metrics)

Theses opportunities (VII)

AI at scale for extreme events detection and prediction

- The thesis's work could focus on different areas:
  - AI-based approaches (see Dr. Valentine Anantharaj's seminar for more details subjects in this area)
  - Target applications «environmental digital twins»
  - Big data engineering pipelines by leveraging frameworks like Dask

Theses opportunities (VIII)

Large-scale applications via containers over HPC

- Collaboration with CINECA in the context of the National Center on Big Data, HPC and QC

High Performance Computing for Data Science

Final remarks
Lecture 24 - 01/12/2023

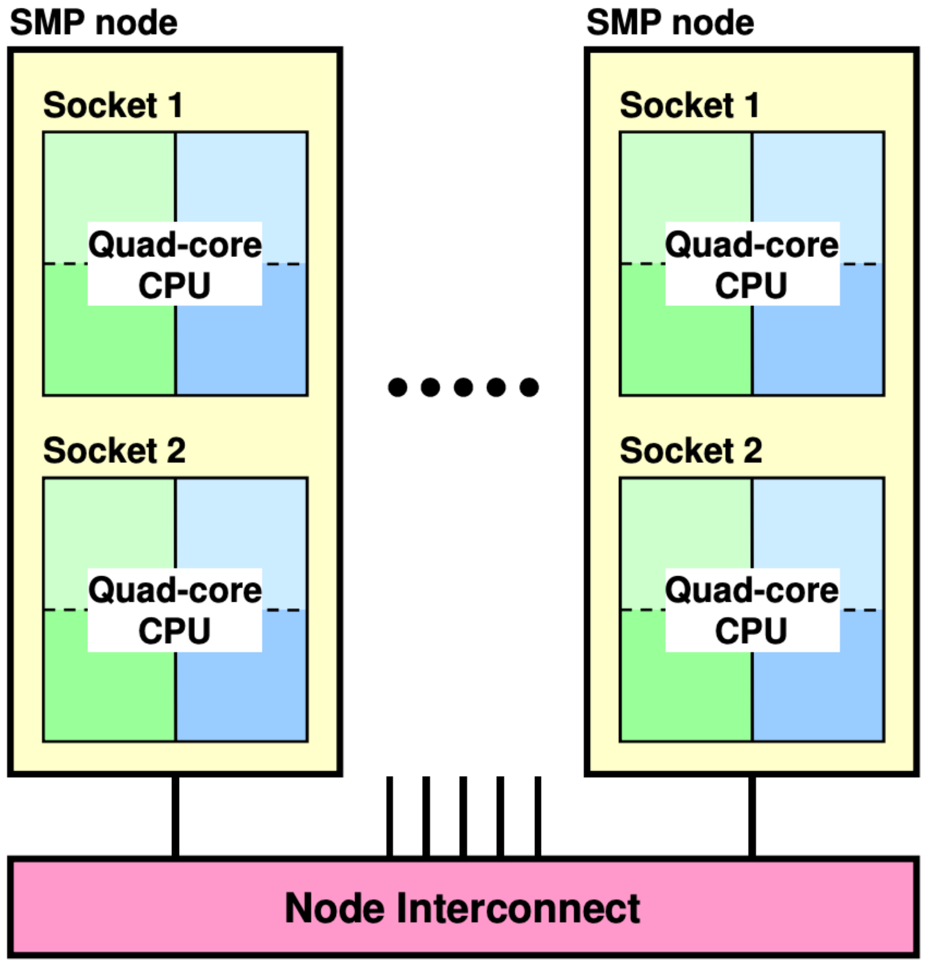
Some notes on hybrid and nested parallelization

Prof. Sandro Fiore
Department of Information Engineering and Computer Science (DISI)
University of Trento, 2023-2024

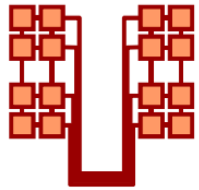
Nested parallelism

- In OpenMP it is possible to have nested parallelism
- Basically, having two nested parallel directives
- A thread in the first team, once it encounters a new parallel directive, creates a brand-new team of threads of which it will become the master until the end of such nested region
- To enabled nested parallelism:
  ```
  $ export OMP_NESTED=TRUE
  ```

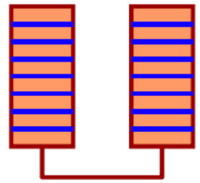
Hybrid parallelization: MPI+OpenMP Motivation



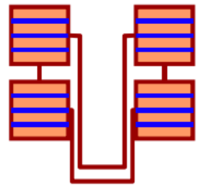
- Which programming model is fastest?
- MPI everywhere?



- Fully hybrid MPI & OpenMP?



- Something between? (Mixed model)



Source: Hybrid MPI and OpenMP Parallel Programming, Tutorial tut123 at SC13, November 17, 2013, Denver (CO) USA

Hybrid parallelization: MPI+OpenMP Master-only style

- Master-only MPI only outside of parallel regions
- Advantages
  - No message passing inside of the SMP nodes
- Major Problems
  - All other threads are sleeping while master thread communicates!
- Within your PBS script:

```
export OMP_PLACES=threads
mpiexec --report-bindings --map-by <x>
   --bind-to core
```

```
for (iteration ....)
{
 #pragma omp parallel
   numerical code
 /*end omp parallel */

 /* on master thread only */
   MPI_Send (original data
     to halo areas
     in other SMP nodes)
   MPI_Recv (halo data
     from the neighbors)
} /*end for loop
```

Binding

```c
#define _GNU_SOURCE // sched_getcpu(3) is glibc-specific (see the man page)

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <time.h>
#include <sched.h>
#include <mpi.h>
#ifdef _OPENMP
# include <omp.h>
#endif

int main(int argc, char **argv){
     // initialize MPI
    int provided;
    MPI_Init_thread(&argc, &argv, MPI_THREAD_FUNNELED, &provided);
    int rank, size;
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    MPI_Comm_size(MPI_COMM_WORLD, &size);
    int threads = omp_get_max_threads();
    if(rank==0){
        printf("Processes for MPI %d\n",size);
        printf("Threading for OMP %d\n",threads);
    }
    #pragma omp barrier
    #pragma omp parallel
    {
        int thread_id = omp_get_thread_num();
        int cpu_num = sched_getcpu();
        printf("Thread %d of process %d run on CPU %d \n",thread_id,rank,cpu_num);
    }
    #pragma omp barrier
    printf("Process rank %d is done \n",rank);
    MPI_Barrier(MPI_COMM_WORLD);
    MPI_Finalize();
    return 0;
}
```

Binding

PBS Script with binding:

```
#!/bin/bash
#PBS -l select=1:ncpus=2:mem=4gb
export OMP_PLACES=threads
mpiexec --report-bindings -np 1 --map-by node:pe=2 --bind-to core exec.out
```

Output with binding:

```
Processes for MPI 1
Threading for OMP 2
Thread 0 of process 0 run on CPU 0
Thread 1 of process 0 run on CPU 1
Process rank 0 is done
```

Without binding, you could replace the last line with the following one:

```
mpiexec  -np 1 exec.out
```

Feedback form

- Throughout the entire HPC4DS course there will be a form available for your feedback
  - Please provide any comment about:
    - Pros
    - Cons
    - Aspects that were not clear enough during the class
    - Any other feedback you think can be relevant for the course
    - …

- Information are gathered in an anonymous way

https://docs.google.com/forms/d/e/1FAIpQLScSVpNPfMP8poNNz-nLl9xlKlHkeBM2x_JgWyA0qQu6mOBl5A/viewform