

CSC4005

Tutorial 2

Sept 23, 2021

This tutorial will cover...

- How to connect to the server
- Secure your code
- Toolchains for your use
- How to build MPI projects
- How to submit jobs
- A glance at MPI/OpenMP/pthread/CUDA code

How to connect to the server

`ssh STUDENT_ID@10.26.1.30`

```
$ ssh 120999999@10.26.1.30
The authenticity of host '10.26.1.30 (10.26.1.30)' can't be esta
ED25519 key fingerprint is SHA256:d5RBtX0KuPvm4AzCxztUp3B09HwuHh
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])
Warning: Permanently added '10.26.1.30' (ED25519) to the list of
120999999@10.26.1.30's password:
Last login: Wed Sep 22 20:56:18 2021
-bash-4.2$
```

Demo

* Please contact us immediately if you cannot log into the server.

Secure your code

Only put your code in the following directories:

- /home/STUDENT_ID
 - For use in the main node only
- /pvfsmnt/STUDENT_ID
 - For use in all nodes

Or other users might be able to read your stuffs!

Toolchains for your use

The system is pretty outdated with gcc/g++ (CentOS 7), so we've built toolchains manually for your use:

(On the server & VM)

- cmake (CMake 3.21.2)
- clang++ etc. (clang 12.0.1)

How to build MPI projects

Instead of gcc/clang/g++/clang++, Use:

- mpicc
- mpic++
- mpicxx

Demo

```
mpic++ main.cpp
```


How to run my program on multiple nodes?

Submit a **job** to **Slurm**.

Useful commands

- `salloc` - Enter an interactive shell for MP
- `sbatch` - Submit a job
- `scancel` - Cancel a job
- `squeue` - View current queue
- `sacct` - View submission history
- `sinfo -a` - View node information
- `scontrol show job [JOB_ID]` - View information for the job

and so on...

Get help on these commands from these pages:

- [PKU HPC \(Chinese\)](#)
- [Harvard FASRC](#)
- [HPC2N](#)

salloc

```
1 salloc -N2 -n5 -t1
2 cd /pvfsmnt/12099999
3 srun hostname
4 mpirun a.out
```

mpirun was unable to launch the specified application as it could not
change to the specified working directory:

Working directory: /home/1209999999

Node: gpu02

while attempting to start process rank 3.

5 total processes failed to start

MPI, OpenMP, pthread, and CUDA

MPI Sample

mpic++ main.cpp

```
1 #include <mpi.h>
2 #include <iostream>
3 int main(int argc, char *argv[]) {
4     int rank, size;
5     MPI_Init(&argc, &argv);
6
7     MPI_Comm_rank(MPI_COMM_WORLD, &rank);
8     MPI_Comm_size(MPI_COMM_WORLD, &size);
9
10    std::cout << "Hello, World. I am " <<
11        rank << " of " << size << std::endl;
12
13    MPI_Finalize();
14    return 0;
15 }
```

MPI Sample

mpic++ main.cpp

```
1 #include <mpi.h>
2 #include <iostream>
3 int main(int argc, char *argv[]) {
4     int rank, size;
5     MPI_Init(&argc, &argv);
6
7     MPI_Comm_rank(MPI_COMM_WORLD, &rank);
8     MPI_Comm_size(MPI_COMM_WORLD, &size);
9
10    std::cout << "Hello, World. I am " <<
11        rank << " of " << size << std::endl;
12
13    MPI_Finalize();
14    return 0;
15 }
```


MPI Sample 2

```
1 #include <mpi.h>
2 #include <iostream>
3 int main(int argc, char *argv[]) {
4     MPI_Init(&argc, &argv);
5     int world_rank, world_size;
6     MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);
7     MPI_Comm_size(MPI_COMM_WORLD, &world_size);
8
9     int number;
10    if (world_rank == 0) {
11        std::cin >> number;
12        std::cout << "sending " << number << std::endl;
13        MPI_Send(&number, 1, MPI_INT, 1, 0, MPI_COMM_WORLD);
14    } else if (world_rank == 1) {
15        MPI_Recv(&number, 1, MPI_INT, 0, 0, MPI_COMM_WORLD,
16                MPI_STATUS_IGNORE);
```

OpenMP Sample

clang++ -fopenmp main.cpp

```
1 #include <stdio>
2 #include <omp.h>
3
4 int main(int argc, char *argv[]) {
5     using namespace std;
6     omp_set_num_threads(4);
7
8     #pragma omp parallel for
9     for (int i=1; i<20; i++)
10         printf("hello world from #%d\n", i);
11     return 0;
12 }
```

OpenMP Sample

clang++ -fopenmp main.cpp

```
1 #include <stdio>
2 #include <omp.h>
3
4 int main(int argc, char *argv[]) {
5     using namespace std;
6     omp_set_num_threads(4);
7
8     #pragma omp parallel for
9     for (int i=1; i<20; i++)
10         printf("hello world from #%d\n", i);
11     return 0;
12 }
```

OpenMP Sample

clang++ -fopenmp main.cpp

```
1 #include <stdio>
2 #include <omp.h>
3
4 int main(int argc, char *argv[]) {
5     using namespace std;
6     omp_set_num_threads(4);
7
8     #pragma omp parallel for
9     for (int i=1; i<20; i++)
10         printf("hello world from #%d\n", i);
11     return 0;
12 }
```

pthread Sample

clang++ -pthread main.cpp

```
1 #include <cstdio>
2 #include <vector>
3 #include <thread>
4
5 int main() {
6     using namespace std;
7     vector<thread> pool{};
8     for(int i=0; i<5; i++) {
9         pool.emplace_back([i] {
10             printf("hello world from #%d\n", i);
11         });
12     }
13     for(int i=0; i<5; i++) {
14         if (pool[i].joinable()) {
15             pool[i].join();
16         }
17     }
```

pthread Sample

clang++ -pthread main.cpp

```
2 #include <vector>
3 #include <thread>
4
5 int main() {
6     using namespace std;
7     vector<thread> pool{};
8     for(int i=0; i<5; i++) {
9         pool.emplace_back([i] {
10             printf("hello world from #%d\n", i);
11         });
12     }
13     for(int i=0; i<5; i++) {
14         if (pool[i].joinable()) {
15             pool[i].join();
16         }
17     }
18     return 0;
```

pthread Sample

clang++ -pthread main.cpp

```
4
5 int main() {
6     using namespace std;
7     vector<thread> pool{};
8     for(int i=0; i<5; i++) {
9         pool.emplace_back([i] {
10             printf("hello world from #%d\n", i);
11         });
12     }
13     for(int i=0; i<5; i++) {
14         if (pool[i].joinable()) {
15             pool[i].join();
16         }
17     }
18     return 0;
19 }
```

CUDA Sample (Just a glance...)

`nvcc` `main.cpp`

```
1 #include <stdio>
2 #include <cuda.h>
3 #include <cuda_runtime.h>
4
5 int main() {
6     int deviceCount = 0;
7     cudaError_t error_id = cudaGetDeviceCount(&deviceCount);
8     printf("%d CUDA devices detected\n");
9     return 0;
10 }
```


CUDA Sample (Just a glance...)

`nvcc` `main.cpp`

```
1 #include <stdio>
2 #include <cuda.h>
3 #include <cuda_runtime.h>
4
5 int main() {
6     int deviceCount = 0;
7     cudaError_t error_id = cudaGetDeviceCount(&deviceCount);
8     printf("%d CUDA devices detected\n");
9     return 0;
10 }
```

More on to talk about

- Workload distribution
- Communication

and so on!

Done for now!

Check out the slides on:

- Blackboard, or
- <https://csc4005-tut-slides.netlify.app/02/>, or
- <https://csc4005-tut-slides.pages.dev/02/>