University Göttingen
Department of Computer Science
Patrick Höhn, Jonathan Decker

Exercise 2 / October 16, 2023
HPC System Administration / WiSe 2023/24
30 Minutes Total

Slurm Exercise

Since a supercomputer has to provide compute resources to potentially many different customers and jobs, it is important to distribute the resources in a reliable and fair fashion. The program to execute this task is a HPC-Scheduler. A number of different solutions exist in the market, e.g. PBS based schedulers and Slurm. The GWDG utilizes Slurm as its HPC-scheduler. In order facilite the quic execution of the exercises a reservation was created for this course on the SCC cluster. In order to use the reserveration `hpcsa-course` either use the flag `--reservation=hpcsa-course` for each job or set the `SLURM_Reservation` environment variable when using the *medium* partion. For other partitions this variable must be unset.

Contents

Task 1: Your first job (5 min)

As a first example of running jobs on a cluster, you will start one simple command (`hostname`) with different options.

1. Connect to the HPC cluster using SSH (`ssh login-mdc.hpc.gwdg.de`)
2. Run the command directly on the front end node (`hostname`)
3. Run the command in the same fashion on a compute node (`srun hostname`)
4. Run the command with the flag "-f" on a compute node (`srun hostname -f`)

Observe the output. What do you see? Are the outputs similar compared to Slide 15 of the lecture notes?

Task 2: Interactive GUI job (5 min)

New you will start a programm with a graphical user interface (GUI) on a compute node. MATLAB is a fairly popular Computer Algebra System (CAS) which is used in many engineering applications, e.g. control engineering. It is used in this module as an example for an interactive job with a Graphical User Interface.

Of course, any other GUI program installed in the cluster can be started in an analogous way. Please use the following command and verify that the commands are executed successfully.

1. Connect to the HPC cluster using SSH (`ssh -Y login-mdc.hpc.gwdg.de`)
2. Load the "matlab" module (`module load matlab`)
3. Start matlab (`srun --x11 -p medium matlab -desktop`)

Task 3: Interactive Console job (5 min)

Python is a popular open source pgramming language for beginners and widely used in the fields of data science and machine learning. It is applied here as an example for an interactive console session. Please use the following command and verify that the commands are executed successfully.

1. Connect to the HPC cluster using SSH (`ssh login-mdc.hpc.gwdg.de`)
2. Load the "anaconada3" module (`module load anaconda3`)
3. Start an interactive python shell (`srun --pty -p medium python3`)
4. Import "os" and "socket" python modules (`import socket, os`)
5. Run the following command (`print(socket.gethostname())`)
6. Run the following command (`print(os.system("slurm_resources"))`)

Task 4: Starting simple Batch Jobs in different Configurations (5 min)

Batch jobs can use many different configurations as specified during the submission of the job. Try these job configurations:

1. 10 tasks
2. 10 tasks distributed over 3 nodes
3. 3 nodes with 3 tasks each
4. 1 task with 5 cores
5. 2 tasks per node on 2 nodes with 4 cores per task

use `slurm_resources` script to get see the resources of your job

Task 5: Hello World with Slurm and MPI (10 min)

The Message Passing Interface (MPI) is a standard for inter process communication on a single or distributed system. It provides efficient methods for running a single program over a number of systems. However, the MPI methods are relatively low level such that the programmer is responsible for implementing communication patterns and determining how and when processes will exchange data. An in-depth tutorial to MPI is not part of this course but it is still meaningful to understand the basics.
In this exercise you will compile a C program for usage with MPI and run it on multiple nodes using Slurm.

```c
#include <mpi.h>
#include <stdio.h>

int main(int argc, char** argv) {
```

```c
    MPI_Init(NULL, NULL);

    // Get number of processes
    int world_size;
    MPI_Comm_size(MPI_COMM_WORLD, &world_size);

    // Get rank of process
    int world_rank;
    MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);

    // Get name of processor
    char processor_name[MPI_MAX_PROCESSOR_NAME];
    int name_len;
    MPI_Get_processor_name(processor_name, &name_len);

    // Print hello world message
    printf("Hello world from processor %s, rank %d out of %d processors\n",
    processor_name, world_rank, world_size);

    // Finalize the MPI environment
    MPI_Finalize();
}
```

Complete the following tasks:

1. Connect to the HPC cluster using SSH (`ssh login-mdc.hpc.gwdg.de`)
2. Create a new file using `nano mpi-hello-world.c`
3. Insert the code from the listing into the file
4. Save and exit nano, CTRL + o to save and CTRL + x to exit
5. Load the *openmpi* module using `module load openmpi`
6. Compile your program `mpicc -o mpi-hello-world mpi-hello-world.c`
7. Test it `mpirun -np 2 mpi-hello-world`
8. Run it via Slurm `srun --nodes=2 --tasks-per-node=16 mpi-hello-world`
   It might take a moment for nodes to be allocated for you
9. Observe the output, what do you notice?

Task 6: Write your own Batch Script (Complementary) (0 min)

Batch scripts are essential for the submission of complexer non-interactive jobs. You have seen in the slides the basics to write your own scripts. In this voluntary exericse you can try to write your first own batch script for Slurm.

1. Use `echo`, `hostname`, and `sleep X` (sleep for X seconds) to generate output or have it running for a longer time.
2. Have the job send you an email. Advanced: Take a look at the different mail-type options. What do they do?
3. Write the output to a different file. Redirect output and error into different files. Advanced: Take a look

at the filename pattern options. Include node and job name in the output file.