香港天文台
HONG KONG OBSERVATORY

Practical Guide for HKO WMO VCP Workshop on NWP (8 - 11 December 2020)

(Knowledge and experience of Linux Operating System assumed.)

1 LOGIN

(individual user name and key are given separately)

```
## in a Linux/Unix terminal

$ chmod go-r $keyfile (execute once)
$ ssh –i $keyfile $yourusername@202.88.99.86
$ whoami
r2_xxxx
(use this login name used when reporting problems to us)
```

## Windows
Putty or Kitty are recommended.
One has to convert the provided private key to PPK format.
Search the web for the detail how-to.

2 SOFTWARE AND LIBRARY INSTALLED

GCC compiler

1. Intel MPI
2. hdf5-1.8.16
3. netcdf-4.4.0
4. netcdf-fortran-4.4.3
5. parallel-netcdf-1.7.0

3 FOLDER STRUCTURE

$HOME [your home directory]
/r2 [20TB shared LUSTRE storage for work]

4 Running programmes, Submitting jobs and job control

```
$ srun --pty bash
        (get a console on one of the computer nodes for work)
$ sinfo
        (check the status of the computer partition assigned)
$ sbatch yourscript.sh
        (schedule a job to run on the partition)
$ squeue
        (list what jobs are currently running/scheduled)
$ scancel #id
        (cancel a running job)
```

5 Warm up exercise

5.1 Compile a test parallel program

Copy the samples to your own directory. Check if the files are properly copied.

```
$ cp -r /r2/sample_program $HOME
cd $HOME/sample_program
$ ls
cxxpi.cxx  slurm_job.sh
```

Compile the sample code with g++ compiler (with MPI lib loaded) and output as executable "parallel_pi.out"

```
$ mpicxx cxxpi.cxx -o parallel_pi.out
$ ls
cxxpi.cxx  parallel_pi.out  slurm_job.sh
```

Now you have a '*parallelised*' programme making use of a MPI library, which can run spanning multiple nodes and processor cores. However, it cannot be invoked normally as you would do, instead the correct MPI environment must be setup as illustrated below. Look into the sample 'slurm_jobs.sh' for hints.

5.2 RUN THE PROGRAM ON THE CLUSTER

Submit your job to the cluster using command sbatch:

```
$ sbatch -p sq slurm_job.sh
Submitted batch job 64837
```

After the job is complete, a log file (paratest_XXXXX.log) will be saved in your working directory as shown:

```
$ ls
cxxpi.cxx  parallel_pi.out  paratest_64837.log  slurm_job.sh
```

Check the contents of the log file to see if a correct result is obtained:

```
$ cat paratest_64837.log
change dir to:
/r2/sample_program
Calculating Pi parallely with 2 nodes each with 2 cores
Process Process Process 1 of 4 is on r630-235.super.clustertech.com
3 of 4 is on r630-235.super.clustertech.com
Process 0 of 4 is on r630-234.super.clustertech.com
2 of 4 is on r630-234.super.clustertech.com
pi is approximately 3.14159 the calculation delta is 8.33331e-10
wall clock time = 0.00155997
```

Content of the 'slurm_job.sh' is reproduced below for reference.

```
#!/bin/bash
#SBATCH --job-name=paratest
#SBATCH --nodes=2
#SBATCH --ntasks=2
#SBATCH --output=paratest_%j.log   # Standard output and error log

echo "change dir to:"
echo ${SLURM_SUBMIT_DIR}
cd ${SLURM_SUBMIT_DIR}

echo "Calculating Pi parallely with 2 nodes each with 2 cores"

source /opt/intel/compilers_and_libraries_2017/linux/mpi/bin64/mpivars.sh
```

```
mpirun -n 4 ./parallel_pi.out
```

6 Practical 1/2

Compiling and running of WRF basics.
More detail to be available on the VCP website.

7 Practical 2/2

NWP data processing and visualisation.
More detail to be available on the VCP website.

8 General Reminders

All users shall respect that hands-on sessions are running on a shared server cluster, hence submission of extended workload is discouraged, and such job(s) may be terminated by the system administrator without prior notice nor compensation.

While exploring and testing the platform is not strictly prohibited out of good will common to the scientific community, exploiting the system is not allowed. Users will be held accountable to damage they cause for executing unrelated or even malicious code.