University of Amsterdam
BSc Informatica

Distributed and Parallel Programming

DAS5 Tutorial

Becoming Familiar with the Supercomputing Infrastructure

dr. Adam Belloum

October 2023

Students

1. Boris Vukajlovic, **15225054**
2. Marouan Bellari, **14675218**

Questions

1. DAS5 login and data copying

   a) To access the DAS5 you have to use SSH, by executing the command
   `"$ ssh -X USERNAME@fs0.das5.cs.vu.nl"`
   where you substitute USERNAME with the given username. After it will ask you for your given password.
   To access DAS5 from home or outside of UvA, you need to use the UvA VPN.

   b) In the DAS5 terminal, you can change the password by executing the `passwd` command.

   c) To copy a file from our local machine to the DAS5 we use the `scp` command, for example:
   `"scp hello_cuda.cu Makefile das5:  "`
   To copy a folder back from the DAS5 to our local machine we use the `rsync` command, for example:
   `"rsync -avz -ssh das5:folder ./folder"`

2. Usage policy

   a) The defualt run-time for a job is 15 min.

   b) The maximum run-time for a job during the day is 15 min.

c) During the night 8pm to 8am, jobs can run up to 2 hours. Use `prun -t <time>` to set the runtime for each job, so that they don't exceed the limit.

d) On the headnode we edit and compile our code, and obviously run the commands to launch the executable on the cluster nodes. On the regular nodes we can now execute the code.

e) The consequences are losing accesss to the account.

3. Job Execution

a) The prun interface is used to schedule and manage jobs on the cluster. Firstly run `module load prun` to load the prun module type. After you can use the prun command with options -np 1 and -N 1 to run on 1 node with 1 process each:
```
$ module load prun
$ prun -np 1 -N 1 -script $PRUN_ETC/prun-openmpi ./cpi
```
The output will show only one proecss running on a single node, as a example:
```
Process 0 on node027
output
```

b) To run it on the headnode:
```
./assign1
```
To run it on a singular regular node (an example):
```
prun -np 1 -N 1 -script $PRUN_ETC/prun-openmpi ./assign1
```

4. GPU computing

a) Load prun as shown above, then add CUDA to your env:
```
$ module load cuda10.1/toolkit
```
Now we can check whether the module has been loaded using "nvcc –version". Then we can run CUDA jobs like this for example:
```
$ prun -v -np 1 -native '-C <gpu-name> --gres=gpu:1' <program-name>
```

b) The output is:
```
[dpp2417@fs0  ]$ make
nvcc -O2 -g --ptxas-options=-v -c hello_cuda.cu -o hello_cuda.o
ptxas info    :  0 bytes gmem
ptxas info    :  Compiling entry function '_Z11helloKerneliPi' for 'sm_30'
ptxas info    :  Function properties for _Z11helloKerneliPi
0 bytes stack frame, 0 bytes spill stores, 0 bytes spill loads
ptxas info    :  Used 5 registers, 336 bytes cmem[0]
nvcc hello_cuda.o -o hello_cuda
```