M. HASSAN NAEINI 01.05.20

HPC
High Performance Computing - (Baobab)

A short briefing of HPC:

HPC (High Performance Computing) is the ability to process data and perform complex calculations at high speeds. To put it into perspective, a laptop or desktop with a 3 GHz processor can perform around 3 billion calculations per second. While that is much faster than any human can achieve, it pales in comparison to HPC solutions that can perform quadrillions of calculations per second.

One of the best-known types of HPC solutions is the supercomputer. A supercomputer contains thousands of compute nodes that work together to complete one or more tasks. This is called parallel processing. It's similar to having thousands of PCs networked together, combining compute power to complete tasks faster.

The cluster at Unige is mainly composed of two head nodes (or login nodes) and a bunch of computation nodes. Each node provides 32, 20, 16 cores, 12 cores or 8 cores. When you submit a job to the cluster, you allocate resources for your work. You can either request exclusives resources (allocation by node) or request shared resources (allocation by cores).

Why use HPC at Unige?

- For all researchers from all the different fields of studies.
- Holds two clusters: **Baobab** & Yggdrasil->*(available soon)*
- Rich offer of applications and high flexibility of usage
- High availability and limited waiting time for using the Baobab Cluster
- Suitable to parallel computing (more than 4000 cores)
- Have at your disposal a large storage space for temporary data(1PB= 1024TB)
- Really helpful services and assistance:
  - To contact the HPC: hpc@unige.ch
  - And if you have questions or trouble figuring out your problems you can also ask questions on their HPC Forum: hpc-community.unige.ch
  - There's also the possibility that you can ask them to install an application which the cluster doesn't already support and you could also buy additional compute nodes to increase the availability and performance of the infrastructure.

How to get access to the Baobab cluster?

To get access to the cluster you would first need to request for rights to connect to the cluster using your usual ISIs account. For that you need to send a mail containing the following information written below to hpc@unige.ch .

—The informations that you would need to provide inside your email:

- Your ISIs account (*only your username and not password)*
- The name of the department you belong to
- The name of your principal investigator (PI)
- A short description of the intended usage
- Add your PI in cc of the email (*so they can know the request is done with his/her agreement)*

** And normally it will take them 1-2 days of working days to reply back **

Afterwards they'll reply to your mail and let you know that with your usual ISIs username and password you have access to the cluster.

Step-By-Step to execute your code on the cluster

Now that you have access to cluster you need to follow the step-by-step instructions below to execute your code on the cluster.

(1) Add the Bash-script(.sh) next to your python file:

Attached to this there is a script (**Object_detection_image.sh**) which you should put next to your python script (Object_detection_image.py). It will load the Tensorflow module and other modules which you need to load for your code.

(2) Copy your files to baobab:

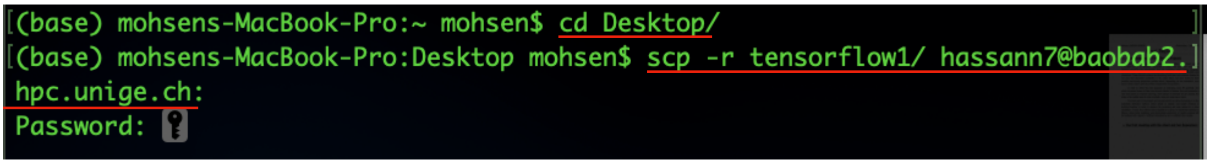
1. Locate your folder which contains your *Object_detection_image*.py code using your terminal. And for that the easiest way to do that is to copy the folder on to your Desktop and after words open a terminal. After you open your terminal, you need to use the **cd** command to get to your Desktop on the terminal: `cd Desktop/`

2. And now that you are in the correct directory where you have your folder containing your code, use the command below to copy your file to the server:

```
scp -r tensorflow1 hassann7@baobab2.hpc.unige.ch:
```

(note: use your own username instead of **hassann7**)

3. Afterwords you need to enter your ISIs password and then wait a couple of minutes to for the folder to be copied to the cluster.



```
[(base) mohsens-MacBook-Pro:~ mohsen$ cd Desktop/                                   ]
[(base) mohsens-MacBook-Pro:Desktop mohsen$ scp -r tensorflow1/ hassann7@baobab2.]
 hpc.unige.ch:
 Password:
```

(3) How connect to the Baobab cluster?

There are different ways to connect to the cluster with your account depending on your operating system (Windows/Mac/Linux/…):

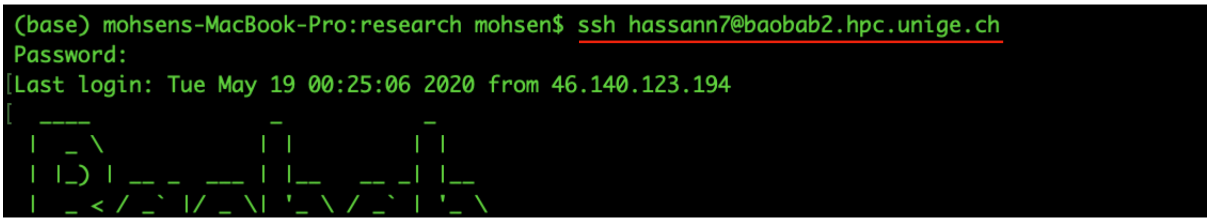
— If you are on a **Mac** or **Linux**:

All you need to do is to open your Terminal and type in the command below:

```
ssh youruser@baobab2.hpc.unige.ch
```

Afterwards you to enter the password which you received by mail from HPC and there you have it, a connection to the Baobab cluster.

The computations were performed at University of Geneva on the Baobab cluster.

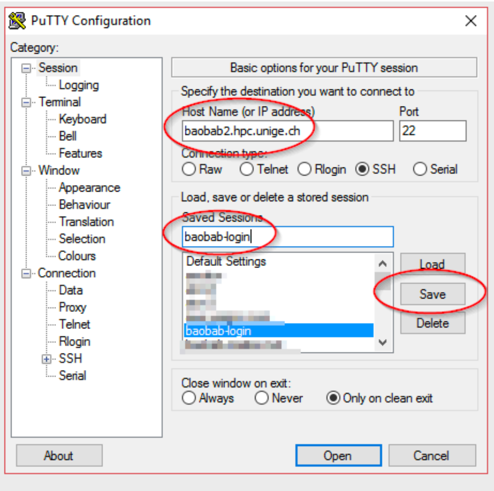


```
(base) mohsens-MacBook-Pro:research mohsen$ ssh hassann7@baobab2.hpc.unige.ch
Password:
[Last login: Tue May 19 00:25:06 2020 from 46.140.123.194
[  ____            _          _
  |  _ \          | |        | |
  | |_) | __ _  ___| |__   __ _| |__
  |  _ < / _` |/ _ \| '_ \ / _` | '_ \
```

— If you are on **Windows**:

It’s a little more difficult but the easiest method is to use a software called PuTTY You will find the needed informations on the screenshots below:

1)



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): baobab2.hpc.unige.ch Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

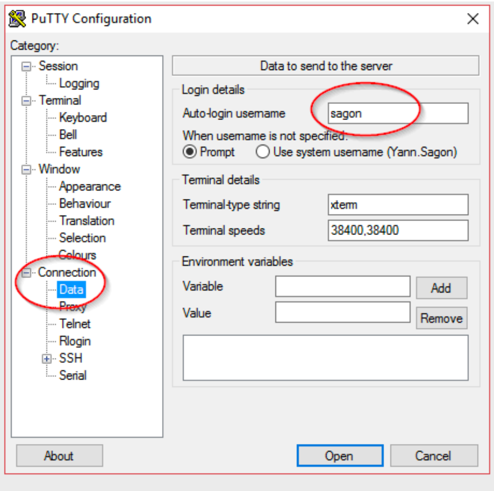
Load, save or delete a stored session

Saved Sessions: baobab-login

Close window on exit: ○ Always ○ Never ● Only on clean exit

PuTTY create session

2)



PuTTY Configuration

Data to send to the server

Login details

Auto-login username: sagon

When username is not specified: ● Prompt ○ Use system username (Yann.Sagon)

Terminal details

Terminal-type string: xterm

Terminal speeds: 38400,38400

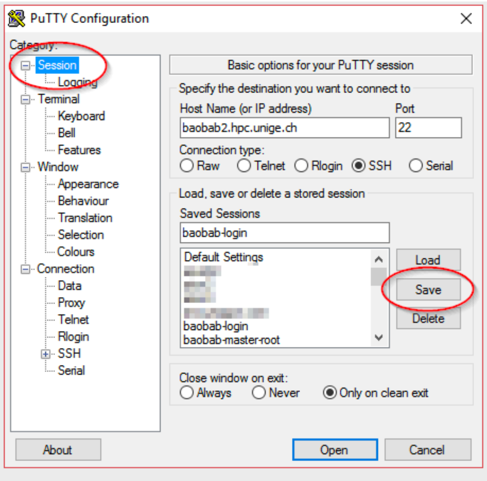
Environment variables

Variable:

Value:

PuTTY enter username

3)



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): baobab2.hpc.unige.ch Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

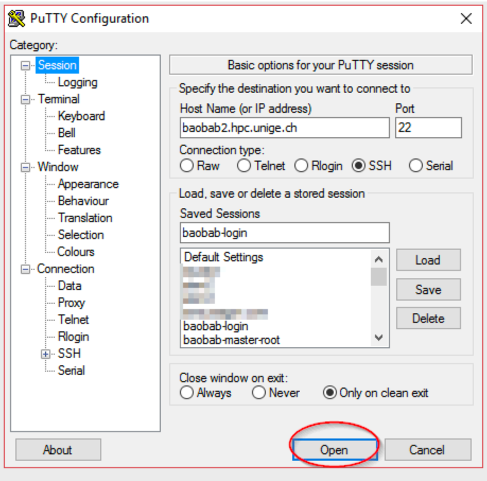
Saved Sessions: baobab-login

Default Settings
baobab-login
baobab-master-root

Close window on exit: ○ Always ○ Never ● Only on clean exit

PuTTY save session

4)



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): baobab2.hpc.unige.ch Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

Saved Sessions: baobab-login

Default Settings
baobab-login
baobab-master-root

Close window on exit: ○ Always ○ Never ● Only on clean exit

PuTTY open session

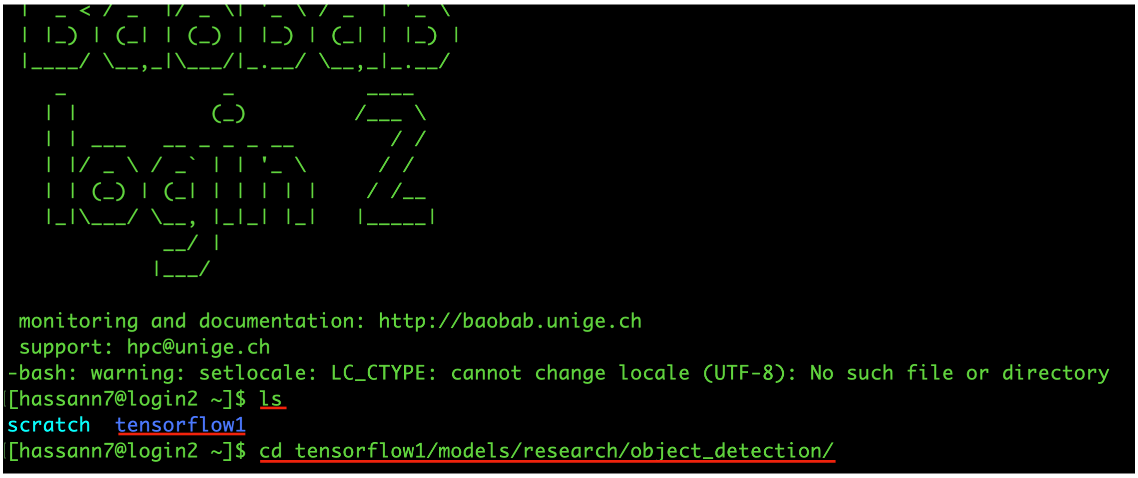
The computations were performed at University of Geneva on the Baobab cluster.

(4) How to execute your Tenserflow code on the cluster:

1. Now that you're logged onto the cluster, you can check if the your file has successfully been transported to the cluster. And for that you just need to type "`ls`" and press enter. (And this while show the file and folders on your account).

2. If you can see folder that you just copied on the server, now you need to enter in the directory where you have your *Object_detection_image*.py, with the “`cd`" command. For example if your *Object_detection_image*.py is located in *research* and *research* is in *models* and *models* is in *tensorflow1* you would have:

```
cd tensorflow1/models/research/object_detection/
```



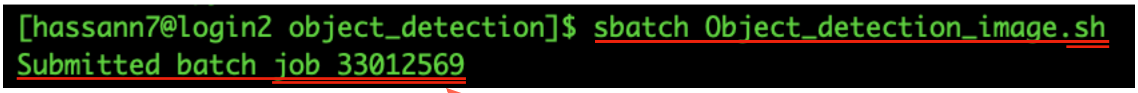
```
 |  _ < / _` |/ _ \| '_ \ / _` | '_ \
 | |_) | (_| | (_) | |_) | (_| | |_) |
 |____/ \__,_|\___/|_.__/ \__,_|_.__/

   _              _          ____
  | |            (_)        /___ \
  | | ___   __ _ _ _ __        / /
  | |/ _ \ / _` | | '_ \      / /
  | | (_) | (_| | | | | |    / /__
  |_|\___/ \__, |_|_| |_|   |_____|
            __/ |
           |___/

 monitoring and documentation: http://baobab.unige.ch
 support: hpc@unige.ch
-bash: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such file or directory
[hassann7@login2 ~]$ ls
scratch  tensorflow1
[hassann7@login2 ~]$ cd tensorflow1/models/research/object_detection/
```

3. Now it's time to run the bash-script “*Object_detection_image*.sh” to execute your “*Object_detection_image*.py” code by doing:

```
sbatch tensorflow.sh
```



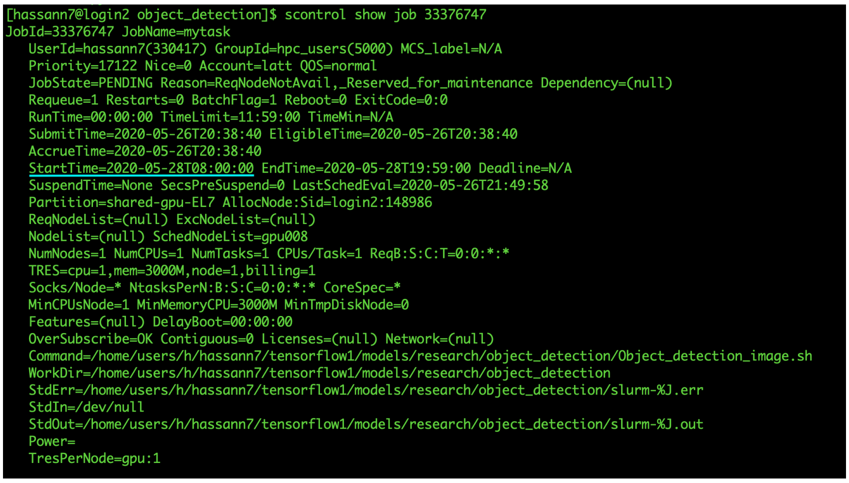
```
[hassann7@login2 object_detection]$ sbatch Object_detection_image.sh
Submitted batch job 33012569
```

JobID

The computations were performed at University of Geneva on the Baobab cluster.

4. After you execute the bash script, the cluster will start the execution of the *Object_detection_image*.py code and return a jobID; Which you can use to check on the pending of your job on the cluster by writing the following command:

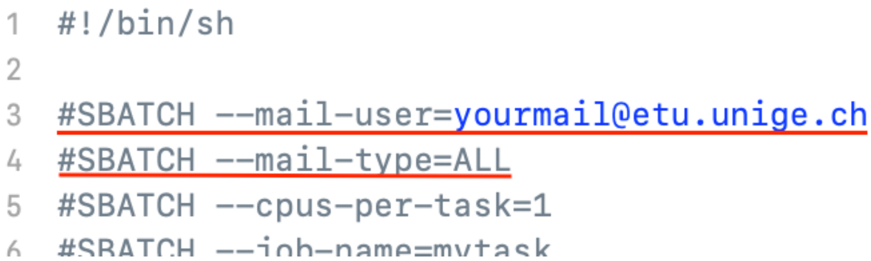
```
scontrol show job JobID
```



```
[hassann7@login2 object_detection]$ scontrol show job 33376747
JobId=33376747 JobName=mytask
   UserId=hassann7(330417) GroupId=hpc_users(5000) MCS_label=N/A
   Priority=17122 Nice=0 Account=latt QOS=normal
   JobState=PENDING Reason=ReqNodeNotAvail,_Reserved_for_maintenance Dependency=(null)
   Requeue=1 Restarts=0 BatchFlag=1 Reboot=0 ExitCode=0:0
   RunTime=00:00:00 TimeLimit=11:59:00 TimeMin=N/A
   SubmitTime=2020-05-26T20:38:40 EligibleTime=2020-05-26T20:38:40
   AccrueTime=2020-05-26T20:38:40
   StartTime=2020-05-28T08:00:00 EndTime=2020-05-28T19:59:00 Deadline=N/A
   SuspendTime=None SecsPreSuspend=0 LastSchedEval=2020-05-26T21:49:58
   Partition=shared-gpu-EL7 AllocNode:Sid=login2:148986
   ReqNodeList=(null) ExcNodeList=(null)
   NodeList=(null) SchedNodeList=gpu008
   NumNodes=1 NumCPUs=1 NumTasks=1 CPUs/Task=1 ReqB:S:C:T=0:0:*:*
   TRES=cpu=1,mem=3000M,node=1,billing=1
   Socks/Node=* NtasksPerN:B:S:C=0:0:*:* CoreSpec=*
   MinCPUsNode=1 MinMemoryCPU=3000M MinTmpDiskNode=0
   Features=(null) DelayBoot=00:00:00
   OverSubscribe=OK Contiguous=0 Licenses=(null) Network=(null)
   Command=/home/users/h/hassann7/tensorflow1/models/research/object_detection/Object_detection_image.sh
   WorkDir=/home/users/h/hassann7/tensorflow1/models/research/object_detection
   StdErr=/home/users/h/hassann7/tensorflow1/models/research/object_detection/slurm-%J.err
   StdIn=/dev/null
   StdOut=/home/users/h/hassann7/tensorflow1/models/research/object_detection/slurm-%J.out
   Power=
   TresPerNode=gpu:1
```

As You can see in the image above this command returns a lot of information. The main information is the *StartTime*: if the cluster is busy or down for a couple of hours or if in your bash file you have chosen a precise generation of node which is busy, your Job will be in the pending list. And if you would like to look take a look at the list of jobs pending or find your job you can do that on this site: https://baobabmaster.unige.ch/iface.

With the help of these two lines in your bash file you will also be receiving a mail which will inform you that the job has been executed and it is finished.



```
#!/bin/sh

#SBATCH --mail-user=yourmail@etu.unige.ch
#SBATCH --mail-type=ALL
#SBATCH --cpus-per-task=1
#SBATCH --job-name=mytask
```

The computations were performed at University of Geneva on the Baobab cluster.

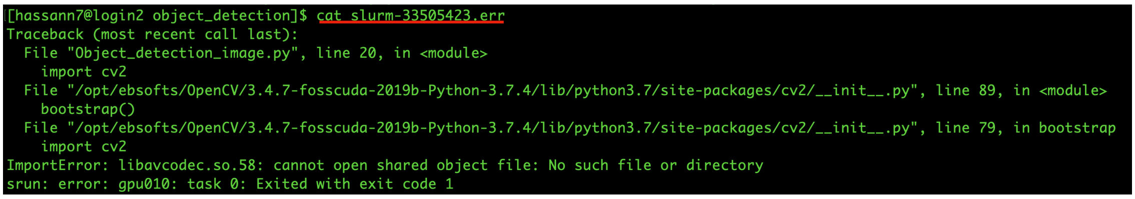
(5) How to check the results and copy back the results to your computer:

1. When ever your code is finish running, anything that your code usually prints will be in the slurm.out file and to check the output of your code you need use the “`cat`" command following the name of your slurm output file and JobID:

```
cat slurm-jobID.out
```

If your code normally gives back a result but output file is empty, you need to check the slurm.err file to see what was the error:

```
cat slurm-jobID.err
```



```
[hassann7@login2 object_detection]$ cat slurm-33505423.err
Traceback (most recent call last):
  File "Object_detection_image.py", line 20, in <module>
    import cv2
  File "/opt/ebsofts/OpenCV/3.4.7-fosscuda-2019b-Python-3.7.4/lib/python3.7/site-packages/cv2/__init__.py", line 89, in <module>
    bootstrap()
  File "/opt/ebsofts/OpenCV/3.4.7-fosscuda-2019b-Python-3.7.4/lib/python3.7/site-packages/cv2/__init__.py", line 79, in bootstrap
    import cv2
ImportError: libavcodec.so.58: cannot open shared object file: No such file or directory
srun: error: gpu010: task 0: Exited with exit code 1
```

2. And the final step is to bring back the results from Baobab to your PC. And for that you need to type the command below on your **PC's** Terminal:

```
scp -r hassann7@baobab2.hpc.unige.ch:tensorflow1 ./
```

Some other commands which you might find useful:

— If you like to erase a file/folder which is on your baobab account all you need to do is type the following command:

```
rm -r object_detection
```

— If you like to check if a module exist on the cluster or check which versions are available you can use the command below:

```
module spider Foo
```

Where "Foo" is the name of the module.

And to find detailed information about a particular package you must specify the version if there is more than one version:

```
module spider Foo/11.1
```

— To quit the cluster and get back to your terminal, all you need to do is type:

```
exit
```

For any more information please check the link below:

https://baobabmaster.unige.ch/enduser/enduser.html

The computations were performed at University of Geneva on the Baobab cluster.