GPULAB INTRODUCTION
TABLE OF CONTENTS

- Why use GPULab?
- What is GPULab?
- JupyterHub on GPULab
- How to use GPULab
- Hands on!
WHY USE GPULAB?

Access to a lot of GPU's:
- 44x GTX 1080 Ti
- 32x Tesla V100 (DGX-2, HGX-2)
- ...

Your pip/conda packages are installed and ready to use!
Choose any Docker image with your packages pre-installed

Isolated Storage
Separated storage per project

Automatic Job Scheduling
Jobs are started in FIFO order
WHAT IS GPULAB?

Thin wrapper around GPU-enabled Docker containers:
- Hides complexities of mounting storage, CPU/GPU isolation, etc.
- No need to install CUDA, Tensorflow, etc. on the machine yourself

Job Scheduler:
- Over multiple machines
- With 1 or more GPU's

Authentication:
- Via IDLab iLab.t or Fed4FIRE accounts
- Concept of ‘projects’ for sharing of resources
JUPYTERHUB ON GPULAB

Available on
https://jupyterhub.ilabt.imec.be
Generates and starts a GPULab job for you
Redirects you to your Jupyter notebook server once started

**Gotchas:**
- Server start will timeout after 5 minutes (ex. No GPU’s available in chosen cluster)
- Job will be cancelled after 1 hour of inactivity in the browser, even if a computation is running!
- Custom docker images must descend from jupyter/base-notebook
USING GPULAB

- Website [https://gpulab.ilabt.imec.be](https://gpulab.ilabt.imec.be) for monitoring/submitting simple jobs
- `gpulab-cli` for submitting jobs from the command line
USING GPULAB
SUBMITTING A JOB VIA THE CLI

`thijs@ibcn055:~$ gpulab-cli submit --project twalcari-test < jupyter-scipy.json`

87914bc6-10ec-11ea-93a1-d7177117bc9b
DEFINING A GPULAB JOB

```json
{
   "jobDefinition": {
      "name": "helloworld",
      "description": "Hello world!",
      "clusterId": 1,
      "resources": {
         "gpus": 1,
         "systemMemory": 2000,
         "cpuCores": 2
      },
      "dockerImage": "nvidia/cuda:10.2-cudnn7-devel",
      "command": "",
      "jobDataLocations": [],
      "portMappings": [],
      "environment": {}
   }
}
```
DEFINING A GPULAB JOB

- Modify an example

- Use the ‘Create job’ function on the website
# Defining a GPULab Job

## Determining the Cluster and Resources to Request: Via the Website

### Clusters

<table>
<thead>
<tr>
<th>GPULab Version</th>
<th>Cluster ID</th>
<th>Comment</th>
<th>GPU model</th>
<th>GPU's</th>
<th>CPU's</th>
<th>Memory (GB)</th>
<th>Slaves</th>
<th>Running Jobs</th>
</tr>
</thead>
<tbody>
<tr>
<td>stable</td>
<td>1</td>
<td>1x 2x GF GTX 1080 Ti</td>
<td>GeForce GTX 1080 Ti</td>
<td>2:16</td>
<td>16:16</td>
<td>31.47/31.47</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>stable</td>
<td>2</td>
<td>1x Tesla V100</td>
<td>Tesla V100-PCIE-32GB</td>
<td>0:10</td>
<td>10:12</td>
<td>23.62/31.62</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>stable</td>
<td>3</td>
<td>1x RTX2080</td>
<td>GeForce RTX 2080 Ti</td>
<td>1:12</td>
<td>12:12</td>
<td>31.66/31.66</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>stable</td>
<td>4</td>
<td>3x 11x GF GTX 1080 Ti</td>
<td>GeForce GTX 1080 Ti</td>
<td>5:28</td>
<td>28:28</td>
<td>244.93/988.58</td>
<td>4</td>
<td>27</td>
</tr>
</tbody>
</table>
DEFINING A GPULAB JOB
DETERMINING THE CLUSTER AND RESOURCES TO REQUEST: VIA THE CLI

```bash
~$ gpulab-cli clusters
```

<table>
<thead>
<tr>
<th>ID</th>
<th>GPU Model</th>
<th>Comment</th>
<th>Slaves</th>
<th>GPUs</th>
<th>CPUs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1x 2x GF GTX 1080 Ti</td>
<td></td>
<td>0</td>
<td>0/0</td>
<td>0/0</td>
</tr>
<tr>
<td>2</td>
<td>1x Tesla V100</td>
<td></td>
<td>0</td>
<td>0/0</td>
<td>0/0</td>
</tr>
<tr>
<td>3</td>
<td>GeForce RTX 2080 Ti</td>
<td>1x RTX2080</td>
<td>1</td>
<td>1/1</td>
<td>12/12</td>
</tr>
<tr>
<td>4</td>
<td>GeForce GTX 1080 Ti</td>
<td>4x 11x GF GTX 1080 Ti</td>
<td>7</td>
<td>19/30</td>
<td>61/93</td>
</tr>
<tr>
<td>5</td>
<td>No GPU, shared CPUs</td>
<td></td>
<td>0</td>
<td>0/0</td>
<td>0/0</td>
</tr>
<tr>
<td>6</td>
<td>Tesla V100-SXM3-32GB</td>
<td>1x HGX-2 - 16xTesla V100</td>
<td>2</td>
<td>3/16</td>
<td>38/96</td>
</tr>
<tr>
<td>7</td>
<td>Tesla V100-SXM3-32GB</td>
<td>UAntwerp: 1x DGX-2 - 16xTesla V100</td>
<td>1</td>
<td>5/16</td>
<td>39/96</td>
</tr>
<tr>
<td>10</td>
<td>Development only - do not use</td>
<td></td>
<td>0</td>
<td>0/0</td>
<td>0/0</td>
</tr>
</tbody>
</table>
STORAGE ON GPULAB

- Use jobDataLocations to set storage mount points

- `/project` is shared NFS storage per project between imec Virtual Wall 2 and Ghent-based GPULab-slaves

- **Gotcha**: this storage is not available in Antwerp, separate storage on UAntwerp DGX machine!

```
"jobDataLocations": [
  {
    "mountPoint": "/project"
  },
  {
    "mountPoint": "/work",
    "sharePath": "/project/work"
  }
]
```
STORAGE ON GPULAB

- `/project_scratch`
  - Project storage on a specific slave
  - Permanent, fast SSD (and large), but not accessible from any other slave
  - Only available on specific slaves!

Currently available on:
- HGX-2 at UGent:
  - 100 TB NVMe storage (`/project_scratch`)
- DGX-2 at UAntwerp:
  - 28 TB NVMe storage (`/project_scratch`)
  - **Gotcha:** `/project` is an alias for `/project_scratch`

```
"jobDataLocations": [
  {
    "mountPoint":
    "/project_scratch"
  },
]
```
STORAGE ON GPULAB
IMPORTING YOUR DATA INTO GPULAB

- Use Jupyter notebooks or SSH-access to your jobs to explore your storage
  - Gotcha: No SCP available
- Start job with SFTP-server

- /project: Swap in a Virtual Wall 2 machine to prepare your data in /groups/ilabt-imec-be/<projectname>
- /project_scratch: Copy from /project to /project_scratch for fast storage on HGX-2 machine


Only applicable for UGent NFS storage
EXPOSING PORTS OF YOUR CONTAINER

- You can define ports to be exposed in `portMappings`.

- Use `containerPort` to specify which port of your container you want to access.

- Host address/port is determined during job scheduling.

- Hosts have no public IPv4 address!
  - Only public IPv6
  - Or private IPv4 (via UGent idlab-VPN)

- **Gotcha:** exposed ports on UAntwerp DGX are only available within UAntwerp IDLab VPN.

```json
"portMappings": [
  {
    "containerPort": 5000
  },
  {
    "containerPort": 5001,
    "hostPort": 5001
  }
]
```

**WARNING:** Job will fail if another container is already mapped to that port!
EXPOSING PORTS OF YOUR CONTAINER
FINDING THE HOST/PORT VIA THE WEBSITE OR CLI
CHECKING THE LOGS OF YOUR CONTAINER ON THE CLI

```
chijs@dibcn055:~ $ gpubl-cli log bdad9168
2019-11-26T14:26:38.810838422Z Executing the command: jupyter notebook
2019-11-26T14:26:42.955435040Z [I 14:26:42.955 NotebookApp] Serving notebooks from local directory: /home/jovyan
2019-11-26T14:26:42.955473356Z [I 14:26:42.955 NotebookApp] The Jupyter Notebook is running at: http://e1e266b4d721:8888/?token=0650d6f6e2f98fb4eb513f4c993c0638a63c541531
2019-11-26T14:26:42.95546481Z [I 14:26:42.955 NotebookApp] Or http://127.0.0.1:8888/?token=0650d6f6e2f98fb4eb513f4c993c0638a63c541531
2019-11-26T14:26:42.955586375Z [I 14:26:42.955 NotebookApp] Use Control-C to stop this server and shut down all kernels
twice to skip confirmation.
2019-11-26T14:26:42.960878486Z [C 14:26:42.960 NotebookApp]
2019-11-26T14:26:42.960896909Z
2019-11-26T14:26:42.960982168Z
2019-11-26T14:26:42.960986520Z
2019-11-26T14:26:42.960911507Z
2019-11-26T14:26:42.960915099Z
2019-11-26T14:26:42.960920383Z
2019-11-26T14:26:46.694747826Z [I 14:26:46.694 NotebookApp] 302 GET / (192.168.124.38) 0.82ms
```

To access the notebook, open this file in a browser:

```
file:///home/jovyan/.local/share/jupyter/runtime/nbserver-6-open.html
```

Or copy and paste one of these URLs:

```
http://e1e266b4d721:8888/?token=0650d6f6e2f98fb4eb513f4c993c0638a63c541531
```

or

```
http://127.0.0.1:8888/?token=0650d6f6e2f98fb4eb513f4c993c0638a63c541531
```
CHECKING THE LOGS OF YOUR CONTAINER ON THE WEBSITE

**Job bdad9168-1058-11ea-93a1-1f5b50ba1116**

**Owner:** djpract

**Name:** Jupyter SciPy

**Status:** RUNNING

Project: F000884-practica

Description: Scipy jupyter notebook server

---

2019-11-26T14:26:30.810588342Z Executing the command: jupyter notebook
2019-11-26T14:26:42.955473156Z I [14:26:42.955 NotebookApp] The Jupyter Notebook is running at:
2019-11-26T14:26:42.955546481Z I [14:26:42.955 NotebookApp] or http://127.0.0.1:8888/?token=0650df66e2f98f6c4
2019-11-26T14:26:42.960878486Z I [14:26:42.960 NotebookApp]
2019-11-26T14:26:42.968898909Z To access the notebook, open this file in a browser:

```
file://home/jovyan/.local/share/jupyter/runtime/nbserver-6-open.html
```

Or copy and paste one of these URLs:

```
http://e1e266b4d721:8888/?token=0650df66e2f98f6c4
```

or

```
http://127.0.0.1:8888/?token=0650df66e2f98f6c4
```
GETTING SSH-ACCESS TO YOUR CONTAINER

Use gpulab-cli ssh <job-id>
1. Develop and test your code locally or in a Jupyter notebook
   - Create your own Docker image with custom software if necessary

2. Scale up to the full dataset once your code is ready

3. Add error-catching and retry mechanisms to your I/O operations

4. Add checkpointing for your intermediate results
GET STARTED!

- **Documentation**: [http://doc.ilabt.imec.be](http://doc.ilabt.imec.be)
- **GPULab**: [https://gpulab.ilabt.imec.be](https://gpulab.ilabt.imec.be)
- **JupyterHub**: [https://jupyterhub.ilabt.imec.be](https://jupyterhub.ilabt.imec.be)

**Support**
- email: helpdesk@ilabt.imec.be

**Mattermost**
- [https://mattermost.ilabt.imec.be](https://mattermost.ilabt.imec.be)

Channels:
- GPULab Support
- JupyterHub support
GET STARTED!
GET AN ACCOUNT

- GPULab uses accounts/projects from the IDLab Testbeds Portal:
  https://account.ilabt.imec.be

- Signup using your University Login

- Request a new project to isolate your file storage
  - or -

- Join the ‘GPULab UA Tutorial’ project for testing:
  https://account.ilabt.imec.be/invite/gpulab ua tut?key=V6Cj0YsbnozUIEhc
GET STARTED!
SETTING UP THE CLI


- Install using `sudo pip3 install gpulab-client-2.0.tar.gz`

- Get your Login Certificate (PEM) from https://account.ilabt.imec.be/

```
thijs@ibcn055:~$ gpulab-cli --cert login_ilabt_imec_be_twalcari@ugent.be.pem clusters
```

```
ID  | GPU Model        | Comment          | Slaves | GPUs | CPUs
--- |------------------|------------------|--------|------|------
1   | stable           | 1x 2x GF GTX 1080 Ti | 0      | 0/0  | 0/0  
2   | stable           | 1x Tesla V100    | 0      | 0/0  | 0/0  
3   | stable           | GeForce RTX 2080 Ti | 1      | 0/1  | 10/12
4   | stable           | GeForce GTX 1080 Ti | 7      | 21/30| 31/93
5   | stable           | No GPU, shared CPUs | 0    | 0/0  | 0/0  
```
embracing a better life