Machine Learning on Puhti

Part 1: Getting started

June 3, 2020
Mats Sjöberg
mats.sjoberg@csc.fi
Webinar: Machine Learning on Puhti

**TODAY**

Part 1: Getting started
- CSC’s services
- Puhti supercomputer
- Available software
- Running jobs on Puhti
- Data storage

**Wed, June 10**

Part 2: Scaling up and using resources efficiently
- Efficient data storage
- GPU utilization
- Multi-GPU and multi-node jobs
- Singularity containers

https://github.com/csc-training/ml-webinar/
What CSC service to use?
What CSC service to use?

- **cPouta (cloud)**
  - Fast parallel storage

- **ePouta (private cloud)**

- **Mahti**
  - Fast parallel storage

- **Puhti**
  - Fast parallel storage

- **Puhti-ai (GPUs)**

- **Allas**
  - Common object storage area

- **Internet**
  - Own workstation, laptop, ...

- **Rahti (container cloud)**

- **Your "own" (virtual) server**
  - Limited number of GPUs
  - Less powerful than Puhti

- **Supercomputer cluster**
  - GPU‐accelerated nodes

- **Container cloud**
  - Easy to use
  - No GPUs yet
What CSC service to use?

Puhti
- Supercomputer cluster
- GPU-accelerated nodes
- Multi-user system
What CSC service to use?

**Pouta**
- Your “own” (virtual) server
- Limited number of GPUs
- Less powerful than Puhti

**Puhti**
- Supercomputer cluster
- GPU-accelerated nodes
- Multi-user system

---

Diagram:

- cPouta (cloud)
- ePouta (private cloud)
- Mahti (fast parallel storage)
- Puhti (fast parallel storage)
- Puhti-ai (GPUs)
- Allas (common object storage area)
- Internet (own workstation, laptop, ...)
- Rahti (container cloud)
What CSC service to use?

**Pouta**
- Your “own” (virtual) server
- Limited number of GPUs
- Less powerful than Puhti

**Puhti**
- Supercomputer cluster
- GPU-accelerated nodes
- Multi-user system

**Rahti**
- Container cloud
- Easy to use
- No GPUs yet
What CSC service to use?

**Pouta**
- Your “own” (virtual) server
- Limited number of GPUs
- Less powerful than Puhti

**Puhti**
- Supercomputer cluster
- GPU-accelerated nodes
- Multi-user system

**Rahti**
- Container cloud
- Easy to use
- No GPUs yet
What CSC service to use?

**Pouta**
- Your “own” (virtual) server
- Limited number of GPUs
- Less powerful than Puhti

**Puhti**
- Supercomputer cluster
- GPU-accelerated nodes
- Multi-user system

**Rahti**
- Container cloud
- Easy to use
- No GPUs yet
Puhti supercomputer
Puhti supercomputer

- *Puhti* has a total of 682 CPU nodes, each with $2 \times 20$ Intel Xeon 2.1 GHz cores
- *Puhti-AI* partition has 80 nodes with 4 GPUs each $\rightarrow$ 320 GPUs in total
- NVIDIA V100 GPUs (Volta) with 32 GB of memory
- Fast network: $2 \times 100$ Gbps links to each node
- Each node has a fast 3.6 TB local NVME disk
Getting access to Puhti

https://docs.csc.fi/computing/overview/

To use Puhti you need to:

• Have a CSC account
• Be member of a CSC project, either by
  • creating a new project, or
  • joining an existing project (ask the PI to add you!)
• Finally, the project needs to have Puhti access

→ MyCSC portal: https://my.csc.fi/
Logging in to Puhti

• Using an ssh client such as OpenSSH or PuTTY
• Basic Linux skills are required!
• More info: https://docs.csc.fi/computing/connecting/

$ ssh <csc_username>@puhti.csc.fi

$ ssh <csc_username>@puhti-login2.csc.fi
Available software
Supported frameworks

We currently support:

- **Python Data** – collection of Python libraries for data analytics and machine learning
- **TensorFlow** – deep learning library for Python
- **PyTorch** – machine learning framework for Python
- **MXNet** – deep learning library for Python
- **RAPIDS** – suite of libraries for data analytics and machine learning on GPUs

https://docs.csc.fi/apps/#data-analytics-and-machine-learning
Example: TensorFlow

- First check the application page for instructions: https://docs.csc.fi/apps/tensorflow/
- Load the default version:
  module load tensorflow
- or specific version:
  module load tensorflow/2.0.0
- **Note:** some modules are *Singularity-based!*
What if some package is missing?

If you are using our module, but a trivial package is missing ...

• install it yourself, e.g.,
  pip install --user <packagename>

• ...or if it might be generally useful, send an email to servicedesk@csc.fi – we can install it for you!
What if some package is missing?

*Note:* you can even upgrade an existing package with:

```bash
pip install --user --upgrade <packagename>
```

but then you need to adjust the order in which packages are accessed by Python:

```bash
export PYTHONPATH=~/.local/lib/python3.7/site-packages/:$PYTHONPATH
```
What if some package is missing?

If you need a specific setup, and our modules are not right for you ...

• use a virtualenv:

```bash
$ module purge
$ python3 -m venv myenv
$ source myenv/bin/activate
$ pip install ...
```

• use conda: https://docs.csc.fi/support/tutorials/conda/

• use singularity containers:
  https://docs.csc.fi/computing/containers/run-existing/

• or if generally useful, send an email to servicedesk@csc.fi
Running jobs on Puhti
Running a job on Puhti

Don’t run heavy computing jobs in the login nodes!

- Puhti uses the *Slurm* batch job system
- Jobs do not run instantly but are put in a *queue*
- Resources (runtime, memory, number of cores) need to be specified
Running a job on Puhti

Batch system

Batch job scheduler places jobs on compute nodes

+ Big memory demand
Running a job on Puhti

Create a job script, for example `run.sh`:

```bash
#!/bin/bash
#SBATCH --account=<project>
#SBATCH --partition=gpu
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=10
#SBATCH --mem=64G
#SBATCH --time=1:00:00
#SBATCH --gres=gpu:v100:1

module load tensorflow/2.0.0
srun python3 myprog.py <options>
```

https://docs.csc.fi/computing/running/creating-job-scripts/
# Running a job on Puhti

Example job script for Singularity-based modules:

```
#!/bin/bash
#SBATCH --account=<project>
#SBATCH --partition=gpu
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=10
#SBATCH --mem=64G
#SBATCH --time=1:00:00
#SBATCH --gres=gpu:v100:1

module load tensorflow/nvidia-20.03-tf2-py3
srun singularity_wrapper exec python3 myprog.py <options>
```
Running a job on Puhti

Submit the job:

`sbatch run.sh`

Check the queue:

`squeue -l -u $USER`

Cancel a job:

`scancel <jobid>`

https://docs.csc.fi/computing/running/submitting-jobs/
Data storage
Data storage on Puhti

- Disk space and *number of files* are limited on Puhti!
  → We want to ensure that the shared (Lustre) filesystem works efficiently for everyone!

- **Useful command:** `csc-workspaces`

<table>
<thead>
<tr>
<th>Owner</th>
<th>Path</th>
<th>Capacity</th>
<th>Number of files</th>
<th>Cleaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>home</td>
<td>/users/&lt;user-name&gt;</td>
<td>10 GiB</td>
<td>100 000 files</td>
<td>No</td>
</tr>
<tr>
<td>projappl</td>
<td>/projappl/&lt;project&gt;</td>
<td>50 GiB</td>
<td>100 000 files</td>
<td>No</td>
</tr>
<tr>
<td>scratch</td>
<td>/scratch/&lt;project&gt;</td>
<td>1 TiB</td>
<td>1 000 000 files</td>
<td>Yes - 90 days</td>
</tr>
</tbody>
</table>

Data quotas can be increased via MyCSC!

[https://docs.csc.fi/computing/disk/](https://docs.csc.fi/computing/disk/)
Using Allas

- store big datasets in Allas, CSC’s object storage
- download them to project scratch prior to computation
- you can also upload trained models (or keep in projappl)

$ module load allas
$ allas-conf
$ cd /scratch/<your-project>
$ swift download <bucket-name> your-dataset.tar

https://docs.csc.fi/data/Allas/
Large number of files

• Many datasets contain a large number of small files
• Shared filesystem (Lustre) performs poorly in this scenario → noticeable slowdowns for all Puhti users!

Consider alternatives:

• packaging your dataset into larger files
• use NVME fast local storage on GPU nodes

More details in webinar part 2!
Thank you!
Don’t forget part 2 of this webinar!

Machine Learning on Puhti
Part 2: Scaling up and using resources efficiently

- Efficient data storage
- GPU utilization
- Multi-GPU and multi-node jobs
- Singularity containers

Time: Wednesday, June 10, 2020 at 14:00-15:00
Place: https://ssl.eventilla.com/event/jJRkz