

GrogRack compute cluster: Getting Started

Bill Arnold

ALCF

arnoldw@anl.gov

December 6, 2023

OVERVIEW

- System Overview
- Account access
- Steps to Login
- Environment setup
- Workflow
- Example programs

GroqRack compute cluster

<https://www.alcf.anl.gov/alcf-ai-testbed>



- 1 GroqRack
- 9 GroqNode servers
- 8 GroqCard accelerators per server node
- (1 GroqChip per card)
- Plus a controller

ACCOUNT ACCESS

- **Step 1 : Request for an ALCF account**

Argonne NATIONAL LABORATORY Argonne Leadership Computing Facility

Support Center | ALCF Resources | ALCF Website

ALCF Account and Project Management

ALCF USER SUPPORT

ACCOUNTS RELATED HELP: accounts@alcf.anl.gov
SUPPORT TICKETS: support@alcf.anl.gov
TELEPHONE: Email accounts@alcf.anl.gov for the phone support number

ALCF HELP DESK ADDRESS:
Argonne Leadership Computing Facility
9700 S. Cass Avenue
Building 240, #2129
Lemont, IL 60439

Account Log In

Username [Forgot username?](#)

CRYPTOCARD Passcode [Token issues?](#)

LOGIN

[Request a new account](#)

Request an ALCF account on our [accounts page](#).

Need an active project account.

Help : accounts@alcf.anl.gov

ACCOUNT ACCESS

- **Step 2 : Login to your ALCF account**
- **Step 3 : Request to Join project**

Join project

Request to become a team member of an existing project.

Available Projects

Click on the name of the project to add and remove proxies and team members.

Project Name ^	Title ↕	PI ↕
groq_training	Groq ALCF AI Testbed Training	Venkatram Vishwanath

Join project under "groq_training".

Request and view systems

Request to use additional systems.

Available Resources

groq.

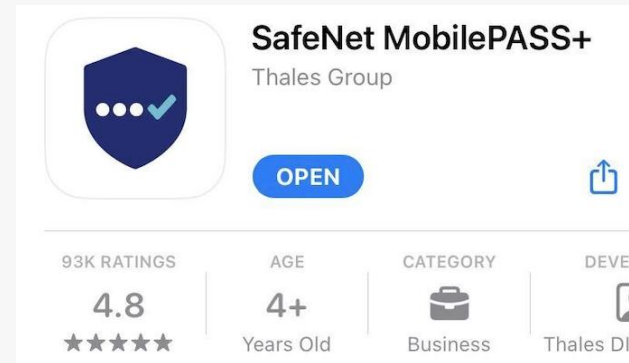
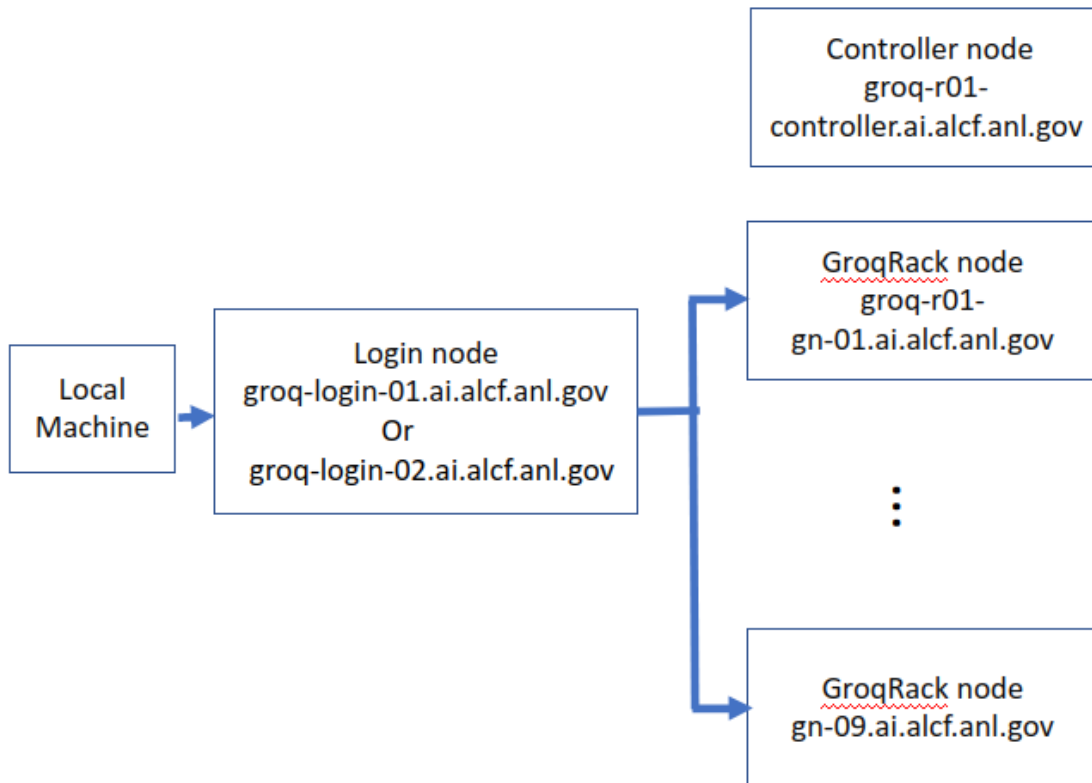
Check for "groq" under Request and view systems

Connection and Login

- **Step 3 : Login to a GroqNode server.**

From local machine to login node :

```
ssh ALCFUserID@groq.ai.alcf.anl.gov  
Password: < MobilePASS+ code >
```



From login node to a GroqNode server :

```
ssh groq-r01-gn-01.ai.alcf.anl.gov
```

Conda environment setup

- **Install conda if not already installed:**

```
rm Miniconda3-latest-Linux-x86_64.sh*
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
bash Miniconda3-latest-Linux-x86_64.sh
# answer y/yes to all prompts
# exit ssh session, then start a new ssh session
exit
```

- **Create a groqflow conda environment and activate it**

```
export PYTHON_VERSION=3.10.12
conda create -n groqflow python=$PYTHON_VERSION
conda activate groqflow
```

Install groqflow

- Clone the groqflow github repo and change current directory to the clone :

```
cd ~/
git clone https://github.com/groq/groqflow.git
cd groqflow
```

- Install groqflow into your groqflow conda environment :

```
conda activate groqflow
# Alter this if you have cloned groqflow to some other location.
cd ~/groqflow
pip install --upgrade pip
pip install -e .
pushd .
cd demo_helpers
pip install -e .
popd
```


Workflow

- Jobs run using PBS Workload Manager and Job Scheduler
- The groqflow framework combines compilation and execution
 - Compiles are cached in `~/.cache/groqflow`
 - Compile artifacts are in the cache dirs
 - Steps may also be done manually

```
(groqflow) $ python distilbert.py 2> /dev/null
```

```
Building "distilbert"
```

- ✓ Exporting PyTorch to ONNX
- ✓ Optimizing ONNX file
- ✓ Checking for Op support
- ✓ Converting to FP16

```
  Compiling model
```

```
  Assembling model
```

Example program – minilm – batch mode

- Path : ~/groqflow/proof_points/natural_language_processing/minilm
- Make a run_minilmv2.sh script :

```
#!/bin/bash
# >>> conda initialize >>>
# !! Contents within this block are managed by 'conda init' !!
__conda_setup="$(${HOME}/miniconda3/bin/conda 'shell.bash' 'hook' 2> /dev/null)"
if [ $? -eq 0 ]; then
    eval "$__conda_setup"
else
    if [ -f "${HOME}/miniconda3/etc/profile.d/conda.sh" ]; then
        . "${HOME}/miniconda3/etc/profile.d/conda.sh"
    else
        export PATH="${HOME}/miniconda3/bin:$PATH"
    fi
fi
unset __conda_setup
# <<< conda initialize <<<
conda activate groqflow
cd ~/groqflow/proof_points/natural_language_processing/minilm
pip install -r requirements.txt
python minilmv2.py
```

Running the inference sample in PBS batch mode

```
qsub -l groq_accelerator=2 run_minilmv2.sh
```

- Output will by default go to two files in the current dir, with names like the following, where the suffix is the job id :

```
$ ls -la run_minilmv2.sh.*
```

```
-rw----- 1 user users 448 Oct 16 18:40 run_minilmv2.sh.e3082  
-rw----- 1 user users 50473 Oct 16 18:42 run_minilmv2.sh.o3082
```

Run a sample using PBS in interactive mode

```
qsub -IV -l walltime=24:00:00 -l groq_accelerator=1
```

```
conda activate groqflow
```

```
cd ~/groqflow/proof_points/natural_language_processing/minilm
```

```
pip install -r requirements.txt
```

```
python minilmv2.py
```

- Note: the number of chips needed is determined during the compile phase. The value can be found here:

```
$ grep num_chips_used ~/.cache/groqflow/minilmv2/minilmv2_state.yaml  
num_chips_used: 1
```


Utility commands

- Qstat : view the PBS queue including running jobs.

```
$ qstat -w
```

Job id	Name	User	Time Use	S	Queue
3276.groq-r01-controller	STDIN	username		0 R	workq

- Man pages are available for PBS commands.

```
# qdel - to delete (cancel) a job:
```

```
man qdel
```

```
# qhold - to hold a job
```

```
man qhold
```

Important directory paths and links

- GroqFlow repo : <https://github.com/groq/groqflow/>
- [AI Testbed User Guide](#)
- [Groq public documentation](#)