

linaroforge

Forge hands-on

Rudy Shand
Field Application Engineer



Cheat Sheet

Training material

`tar -xf linaro-forge-training.tar.gz`

Training content

`asc_linaro_forge` (presenter slides)

`asc_handson` (hands-on slides)

`linaro-forge-training.tar.gz` (examples)

Forge Client (On local machine)

Install Forge client <https://www.linaroforge.com/downloadForge>

Running with a batch script

`./<FORGE_TRAINING>/scripts/submit-job.sh`

Forge commands

<code>ddt --connect</code>	# Reverse connect
<code>ddt --offline</code>	# Run DDT without GUI
<code>map --profile</code>	# Profile without GUI
<code>perf-report</code>	# Generate Performance Report

VSC-5 queue commands

<code>sbatch <jobscript></code>	# Submit a job to queue
<code>squeue -u <user></code>	# Check job queue
<code>scancel <job-id></code>	# Remove job from queue

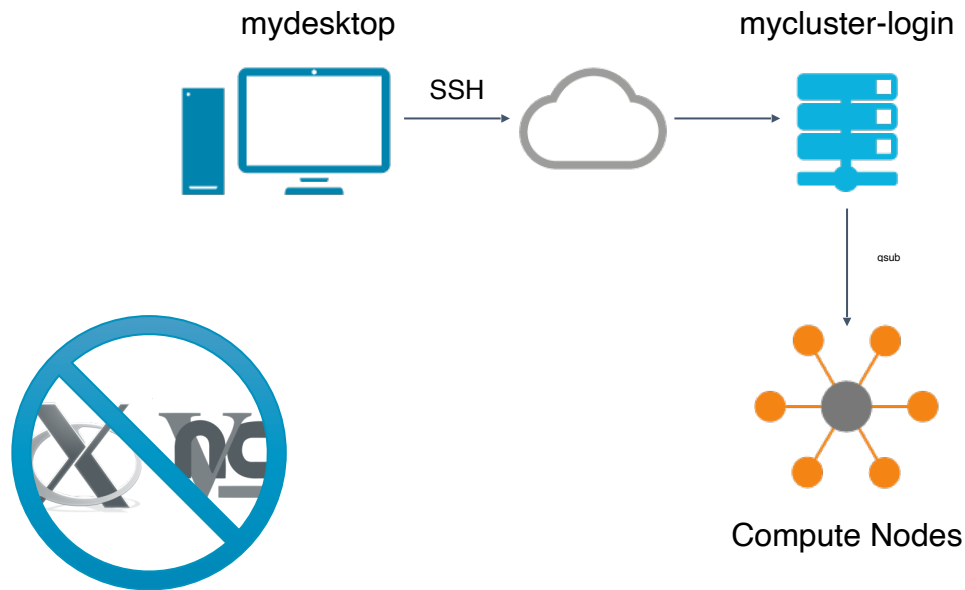
Guides

[Forge userguide](#)

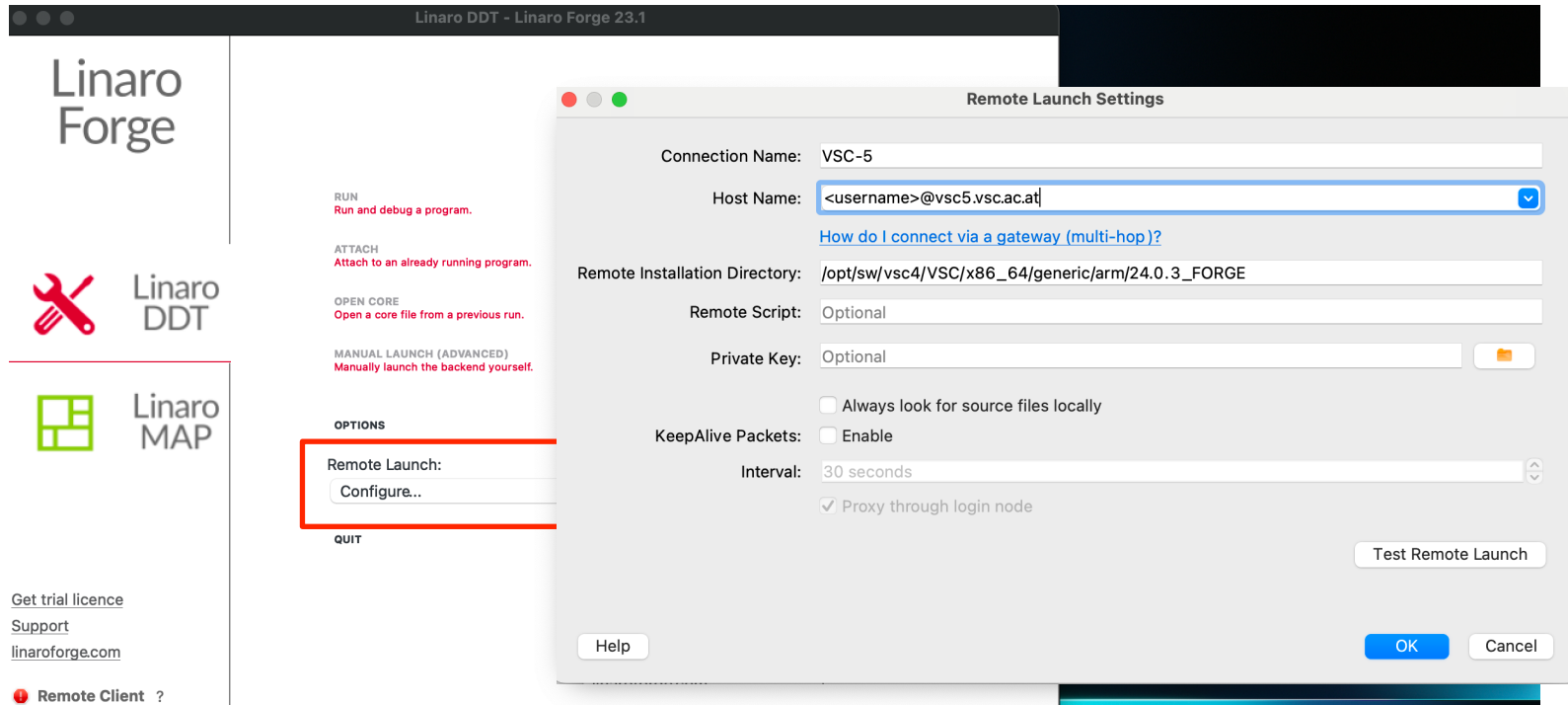
[vsc-5 forge guide](#)

The Forge GUI and where to run it

DDT provides a powerful GUI that can be run in a variety of configurations.



Remote connection to AWS



Setting up the environment

1. Untar the linaro-forge-examples

```
tar -xf linaro-forge-training.tar.gz
```

2. Set the path to the forge training folder

```
export FORGE_TRAINING=<path_to_training_folder>
```

3. Load appropriate modules

```
module load openmpi/4.1.4-gcc-9.5.0-rbertc2
```

```
module load arm/25.0.4_FORGE
```

Post-mortem debugging with a core file

1. Make the example, which creates core files

```
cd $FORGE_TRAINING/correctness/core-files  
make -f core.makefile
```

2. Hook up the Forge client

Select the appropriate remote launch option in the Forge client

3. Open the core files

Linaro DDT —> OPEN CORE

Select div-by-zero as the application

Select both core files

DDT GUI enables post-mortem debugging

Simple DDT correctness example

1. *build simple example*

```
cd <FORGE_TRAINING>/debug  
make
```

2. *Launch the simple application using the DDT client*

```
sbatch <FORGE_TRAINING>/scripts/submit-gpu.sh
```

Accept the connection within the DDT client
Press connect

3. *Verify the correctness of the application*

Set a breakpoint on line 32 and 43
Run to line 32
Single step process 0 (until MPI_Waitall)
Single step process 3 (which releases process 0)
Run all processes
Check sparklines to verify correct MPI rank setup
Drag and drop source, rank, target to Evaluations Window
Run to line 43
Check that sendbuf in the locals tab contains the value 42

4. *Run in offline mode*

```
sbatch <FORGE_TRAINING>/scripts/submit_cpu.sh
```

DDT debug examples

1. *build memory_debugging and split examples*

```
cd <FORGE_TRAINING>/debug  
make
```

2. *Launch applications using DDT*

```
sbatch <FORGE_TRAINING>/scripts/submit-gpu.sh
```

3. *split*

```
mpirun -n 16 ./split
```

4. *memory_debugging*

```
mpirun -n 1 ./memory_debugging  
mpirun -n 4 ./memory_debugging
```

MAP profiling example

Worked Example: https://docs.linaroforge.com/23.1.1/html/forge/worked_examples_appendix/mmult/analyze.html

1. Build the multi example

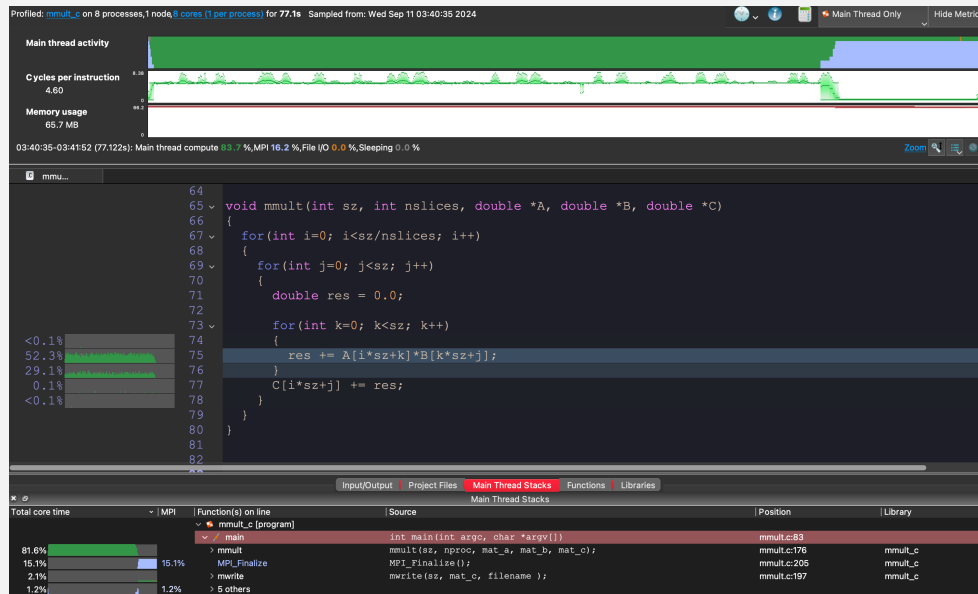
```
cd <FORGE_TRAINING>/performance
make -f mmult_py.makefile
```

2. Create and activate venv

```
python3 -m venv .venv
. .venv/bin/activate
pip install --upgrade pip
pip3 install numpy scipy mpi4py
```

3. Offline profile

```
< Uncomment the map part of the script first >
sbatch <FORGE_TRAINING>/scripts/submit-gpu.sh
```



Thank you

rudy.shand@linaro.org

support@forge.linaro.com

Go to www.linaroforge.com