

Practical exercises

Connecting to LUMI

Project: project_465002763

Setting up ssh key pair:

<https://docs.lumi-supercomputer.eu/firststeps/SSH-keys/>

Logging in:

<https://docs.lumi-supercomputer.eu/firststeps/loggingin/>



Set up

```
git clone https://github.com/CrayLabs/SmartSim-workshops.git
```

```
cd SmartSim-workshops/
```

```
cd LUMI_2026
```

```
cd exercise_01/
```

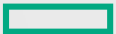
```
module use /pfs/lustref1/flash/project_465002763/shared/modulefiles
```

```
module load smartsim
```



Exercise 1

- Add SmartRedis to an application
- Create a driver script for running SmartSim



Exercise 1 set up

1) Set your language of choice. In smartsim_101.py :

```
# Language options: c, cpp, fortran, python  
language = "cpp"
```

2) Build your application (for c, cpp and Fortran):

```
cd src/<language>  
chmod +x make_app.sh  
./make_app.sh
```



Exercise 1

In the driver script, `smartsim_101.py` there will be missing lines of code to be added, e.g.

```
exp = #EXERCISE
```

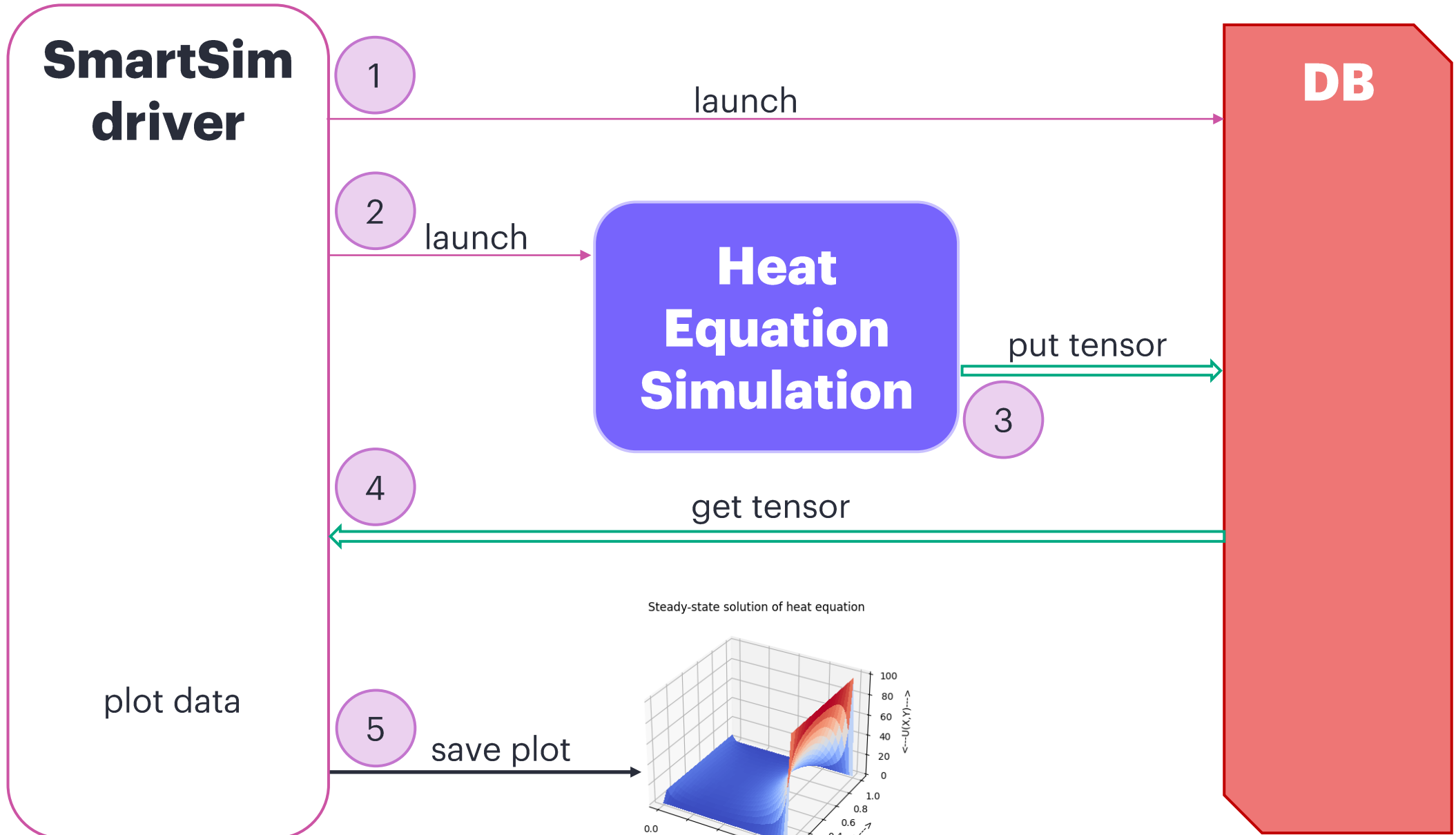
These should be completed to run the example. See <https://www.craylabs.org/docs/overview.html>

In the application, `src/<language>/fd2d_heat_steady_test.<lng>` you will also need to add code to:

- Start a SmartRedis Client (cluster mode should be False)
- Define array dimensions (not required in python)
- Set tensor keys (`steady_state_u` etc.)
- Send data to the orchestrator



SmartSim steady state heating

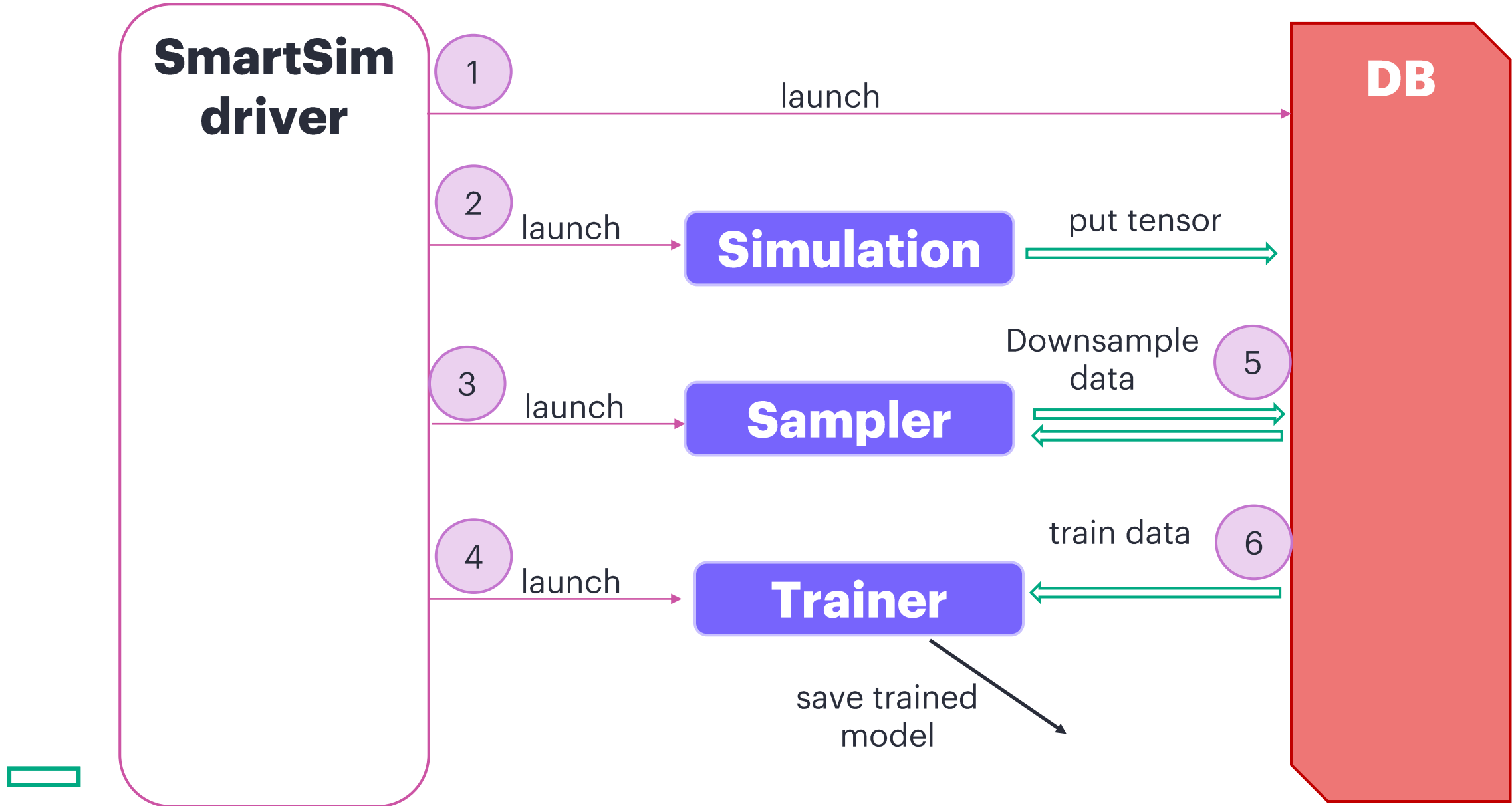


Exercise 2

- Run a SmartSim example with online training
- The orchestrator launches multiple components

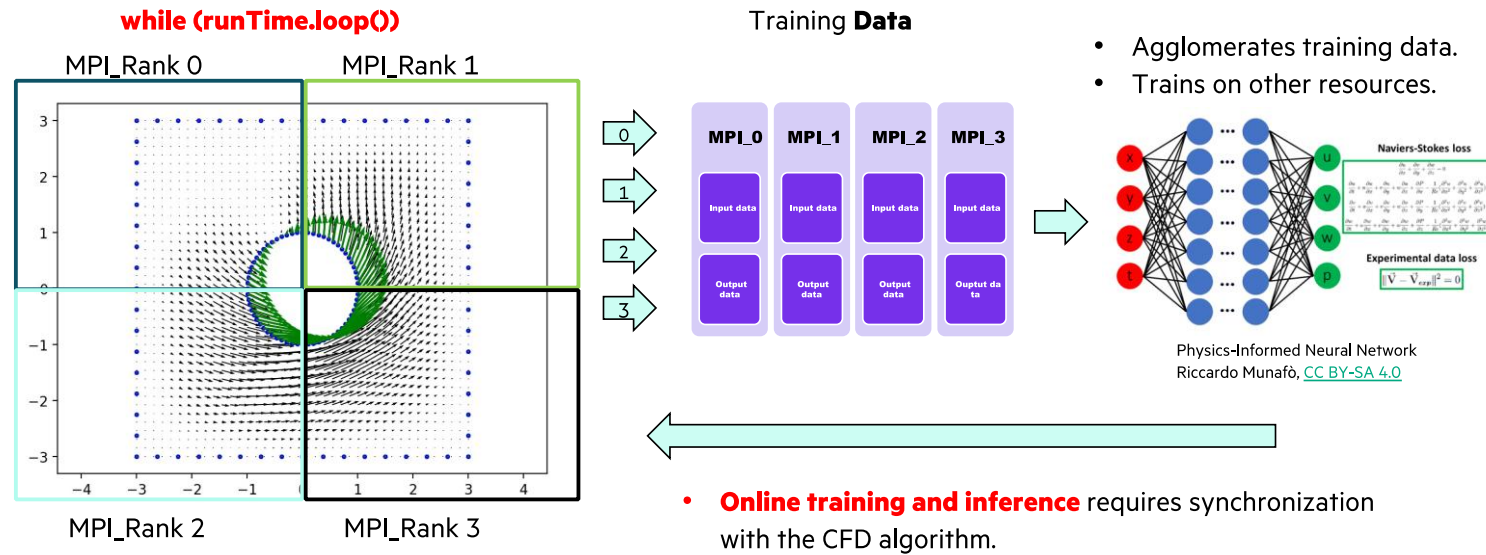


SmartSim online training



Backup Exercise OpenFOAM

ONLINE ML MESH MOTION



- Examine the main loop of `ml_model_training.py` and `networks/PINN.py`
- Run with the default settings (next page)
- Change `PINN.py` to include your own constraint and/or PDE to solve in the interior
- Run the example with your custom physics constraint

Setup on Lumi

```
module use /appl/local/csc/modulefiles
```

```
module load openfoam/v2412
```

```
# One-time setup
```

```
git clone https://github.com/CrayLabs/openfoam-smartsim.git
```

```
cd openfoam-smartsim
```

```
./Allwmake
```

```
# To run the case
```

```
cd run/meshMotion
```

```
srun -N 1 --tasks-per-node=56 --reservation=smartsim --gpus-per-node=8 -p dev-g --account=project_465002763 -t  
01:00:00 --pty bash
```

```
python smartsim_driver.py
```



Setup on Linux

[Assuming that SmartSim has already been installed and OpenFOAM are installed]

```
git clone https://github.com/CrayLabs/SmartRedis.git
```

```
git clone -b workshop_example https://github.com/CrayLabs/openfoam-smartsim.git
```

```
pushd SmartRedis
```

```
make lib
```

```
export SMARTREDIS_INSTALL_PATH=$PWD/install
```

```
popd
```

```
pip install nvidia-physicsnemo "Cython"
```

```
pip install nvidia-physicsnemo.sym --no-build-isolation
```

```
cd openfoam-smartsim
```

```
./Allwmake
```

```
cd run/meshMotion
```

```
python smartsim_driver.py
```

