



Hewlett Packard
Enterprise

Using SmartSim on ARCHER2

Holly Judge

24th October 2024

First experience with SmartSim on ARCHER2

- How can we install SmartSim?
- How should we run SmartSim?
 - Direct use of the batch system
 - Use of resources for the database and HPC application
- Where can we run SmartSim?
 - Use of CPUs/GPUs?
- How to build and run a sample application that uses SmartSim



Installing SmartSim on ARCHER2

- SmartSim and SmartRedis python packages can be installed in a python environment – this environment is based on the ARCHER2 pytorch installation.

```
module load pytorch/1.13.0a0
PYVENV_ROOT=/work/group/group/account/smartsim-eke
PYVENV_SITEPKGS=${PYVENV_ROOT}/lib/python3.9/site-packages
mkdir -p ${PYVENV_ROOT}
python -m venv --system-site-packages ${PYVENV_ROOT}
extend-venv-activate ${PYVENV_ROOT}
source ${PYVENV_ROOT}/bin/activate
```

module load pytorch/1.13.1-gpu
for GPU usage

- The SmartRedis C++ and Fortran libraries need to be installed.

```
git clone https://github.com/CrayLabs/SmartRedis.git
cd SmartRedis CC=cc CXX=CC FC=ftn
make lib-with-fortran
pip install .
```

- To build Redis, RedisAI, the ML backends:

for SmartSim versions < v0.8

```
export TORCH_CMAKE_DIR=$(python -c 'import torch;print(torch.utils.cmake_prefix_path)')
smart build --device {cpu,gpu} --torch_dir $TORCH_CMAKE_DIR --no_tf -v
```

SmartSim helloworld batch example

```
from smartsim import Experiment
exp = Experiment(name="getting-started", launcher="slurm")

run_settings = exp.create_run_settings(exe="echo", exe_args="hello!", run_command="srun")

run_settings.set_nodes(1)
run_settings.set_cpus_per_task(1)
run_settings.set_tasks(2)
run_settings.set_tasks_per_node(2)

# batch settings for archer2

sbatch_settings = exp.create_batch_settings(nodes=1, time="00:10:00", queue="standard", account="y02-coe", batch_args= {"qos": "standard"}
)

# run the model

M1 = exp.create_model("tutorial-model", run_settings=run_settings, batch_settings=sbatch_settings)
exp.start(M1, block=True, summary=True)
```

SmartSim database

- An in-memory database that uses Redis and RedisAI to provide a distributed database and access to ML backends.
- Stores the tensors, the machine learning model and scripts.
- Holds and executes ML models written in python on CPU or GPU.

```
from smartsim import Experiment

exp = Experiment("batch-db-on-slurm", launcher="slurm")
db_cluster = exp.create_database(db_nodes=1,
                                 db_port=6780,
                                 batch=True,
                                 time="00:10:00",
                                 interface="hsn0",
                                 account="y02-coe",
                                 queue="standard")

db_cluster.set_batch_arg("qos", "standard")
exp.start(db_cluster)

print(f"Orchestrator launched on nodes: {db_cluster.hosts}")

exp.stop(db_cluster)
```

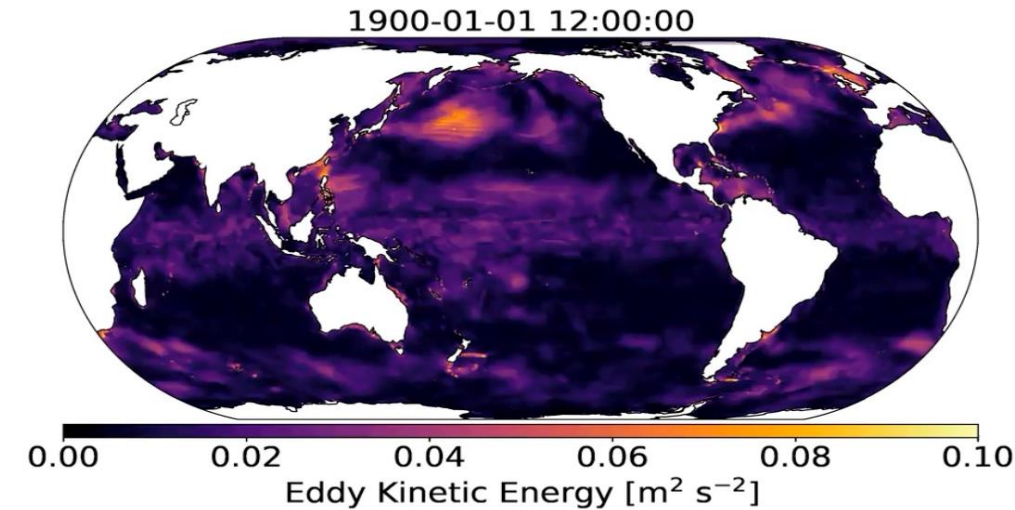


MOM6 application

- MOM6+pytorch example for predicting the Eddy kinetic energy.
- Needs to be built including SmartRedis libraries.
- The driver_OM4_025.py script is used to launch SmartSim
- Set run and batch settings in the **driver script** for the MOM6 application and the database to allow these to run on ARCHER2.

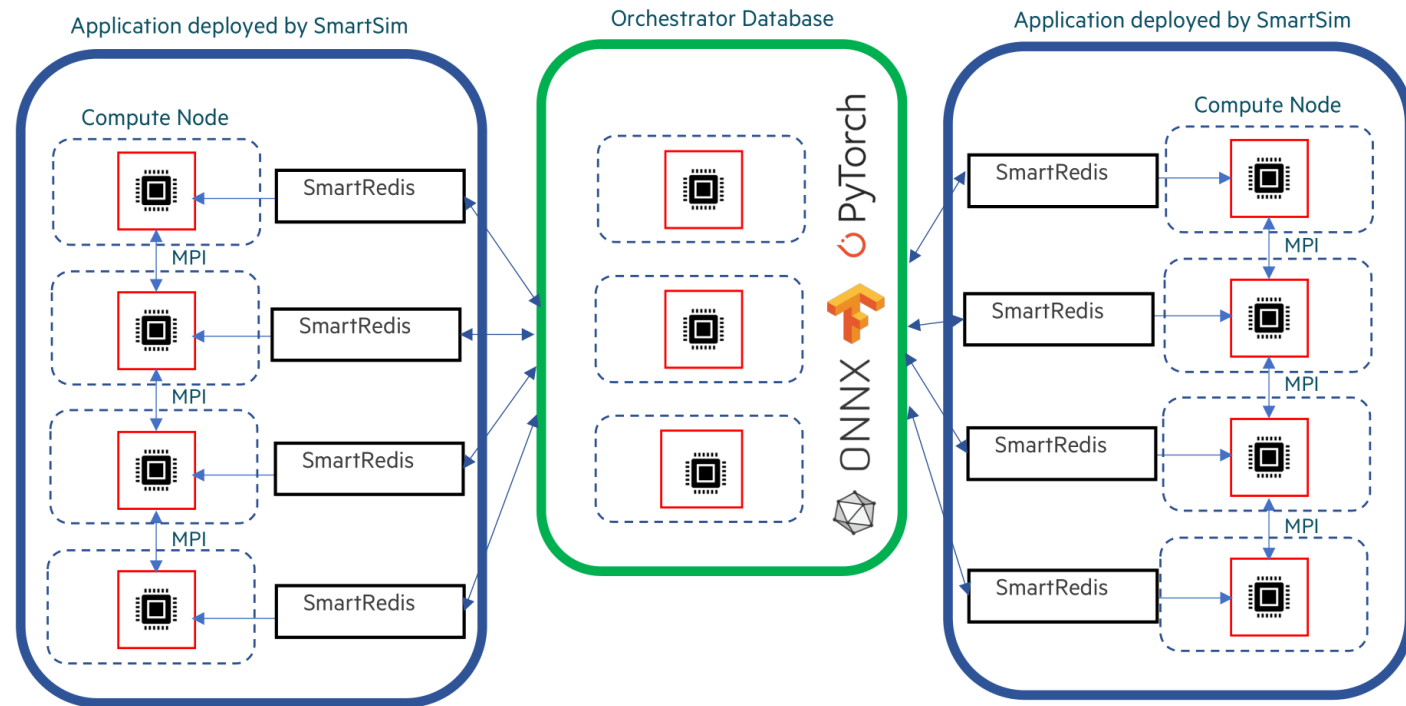
```
ensemble_batch_settings = experiment.create_batch_settings(  
    nodes    = ensemble_size*nodes_per_member,  
    time     = "01:00:00",  
    account  = "y02-coe",  
    queue    = "standard",  
    batch_args = {"qos":"standard"},  
ensemble_batch_settings.add_preamble(  
"export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/path/to/smartredis/install/lib")
```

```
orchestrator = exp.create_database(  
    port    = 6780,  
    interface = "hsn0",  
    db_nodes = 1,  
    time     = "01:00:00",  
    account  = "y02-coe",  
    batch    = True)  
orchestrator.set_batch_arg("partition", "gpu")  
orchestrator.set_batch_arg("qos", "gpu-shd")  
orchestrator.set_batch_arg("gpus","1")  
orchestrator.batch_settings.add_preamble("module load pytorch/1.13.1-gpu")
```



MOM6 on ARCHER2

- MOM6 application runs on CPUs only.
- Database can be run on CPU or GPU – using relevant pytorch.
- Here we have been able to run the database on the CPUs and also on the ARCHER2 GPU development platform by building the redis DB on top of our own rocm pytorch build.
- The driver script creates one job script for MOM6 and one for the database and launches these.



Database job script (generated by SmartSim)

```
#!/bin/bash

#SBATCH --job-name=orchestrator-D4WEP4PQNYA5
#SBATCH --nodes=1
#SBATCH --time=2:00:00
#SBATCH --account=y02-coe
#SBATCH --cpus-per-task=8
#SBATCH --partition=gpu
#SBATCH --qos=gpu-shd
#SBATCH --gpus=1

module load pytorch/1.13.1-gpu

srun --ntasks=1 --ntasks-per-node=1 --cpus-per-task=8 /path/to/smartsim/env/python -m smartsim._core.entrypoints.redis +ifname=hsn0
+command /path/to/smartsim/_core/bin/redis-server /path/to/smartsim/_core/config/redis6.conf --loadmodule
/path/to/smartsim/_core/lib/redisai.so
--port 6780
```

Uses the Smartsim python environment and redis module

+ifname: the network interface DB should listen on

+command: settings for launching redis and redisai

Port: specified port no.

MOM6 job script (generated by SmartSim)

```
#!/bin/bash

#SBATCH --job-name=MOM-D4WEPFAQV0DJ
#SBATCH --qos=standard
#SBATCH --nodes=4
#SBATCH --time=2:00:00
#SBATCH --partition=standard
#SBATCH --account=y02-coe

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/path/to/SmartRedis/install/lib

srun --export ALL,SSDB=10.253.98.5:6780,SSKEYIN=MOM_0,SSKEYOUT=MOM_0 --ntasks-per-node=120 --ntasks=480
/path/to/MOM6/MOM6
```

SSDB: Address of the redis database and port

SSKEYIN: the prefix attached to tensors, datasets, models, etc.
retrieved from the database

SSKEYOUT: the prefix that is attached to tensors, datasets, etc.
sent from client to database

Driver script launch

- The driver script launches batch jobs from within the python driver script.
 - Manages execution of the components
- Note:
 - Two jobs queue separately, can be a waste of resources
 - Need to change batch settings within driver script

```
python driver_OM4_025.py clustered
=== Launch Summary ===
Experiment: AI-EKE-MOM6
Experiment Path: /path/to/exp/NCAR_ML_EKE/driver/AI-EKE-MOM6
Launcher: slurm
Database Status: launching
```

```
=== Ensembles ===
MOM
Members: 1
Batch Launch: True
Batch Command: sbatch
Batch arguments:
qos = standard
nodes = 4
time = 2:00:00
partition = standard
account = y02-coe
```

```
=== Database ===
Shards: 1
Port: 6780
Network: hsn0
Batch Launch: True
Batch Command: sbatch
Batch arguments:
nodes = 1
time = 2:00:00
account = y02-coe
cpus-per-task = 8
partition = gpu
qos = gpu-shd
gpus = 1
```

```
14:16:10 In01 SmartSim[127842] INFO Orchestrator launched as a batch
14:16:10 In01 SmartSim[127842] INFO While queued, SmartSim will wait for Orchestrator to run
14:16:10 In01 SmartSim[127842] INFO CTRL+C interrupt to abort and cancel launch
14:16:36 In01 SmartSim[127842] INFO MOM(7815050): New
14:16:41 In01 SmartSim[127842] INFO MOM(7815050): New
14:16:46 In01 SmartSim[127842] INFO MOM(7815050): Running
14:16:51 In01 SmartSim[127842] INFO MOM(7815050): Running
```

```
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
7838941 gpu orchestr hjhpe R 0:47 1 nid200002
7838946 standard MOM-D4Z0 hjhpe PD 0:00 4 (Priority)
```



MOM6 hetjob usage

```
#!/bin/bash
```

```
#SBATCH --job-name=MOM6-het  
#SBATCH --time=02:00:00  
#SBATCH --partition=standard
```

CPU hetjob for MOM6 application

```
#SBATCH --account=y02-coe  
#SBATCH --qos=standard  
#SBATCH --nodes=4  
#SBATCH --ntasks-per-node=120
```

```
#SBATCH hetjob
```

GPU hetjob for database

```
#SBATCH --account=y02-coe  
#SBATCH --partition=gpu  
#SBATCH --qos=gpu-shd  
#SBATCH --gpus=1  
#SBATCH --cpus-per-task=8
```

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/path/to/SmartRedis/install/lib  
source /path/to/smartsim/pyenv/activate
```

```
python driver_OM4_025_het.py clustered
```

- An alternative to letting the driver script launch the batch jobs.
- Using a modified driver script:
 - turn off batch launching.
 - add in the relevant --het-groups in the run settings for srun to use.

```
orchestrator.set_run_arg("het-group" , "1")
```

Recap

- Demonstrated how build and run SmartSim examples on ARCHER2 with no prior experience.
- Most effort was in building the MOM6 application and adding the ARCHER2 configuration into the driver script.
- We have been able to run the SmartSim database on the ARCHER2 CPUs - and on the GPU development platform.



Thank you

