

MLCommons Earthquake Benchmark

Gregor von Laszewski, Jacques Fleischer, Geoffrey C. Fox

Contents

| | | |
|----------|----------------------------------|----------|
| 1 | Setup | 2 |
| 1.1 | Rivanna | 2 |
| 2 | Cloudmesh | 2 |
| 3 | Report Generation | 3 |
| 4 | Benchmark Terminology | 3 |
| 5 | Results from Rivanna V100 | 3 |
| 5.1 | Best Accuracy | 3 |
| 5.2 | Accuracy Comparison | 3 |
| 5.3 | Time Comparison | 3 |
| 5.4 | Power Consumption | 3 |

MLCommons Earthquake Benchmark

Gregor von Laszewski, Jacques Fleischer, Geoffrey C. Fox

November 28, 2022

Abstract

We report here the results of the MLCommons Science Working group benchmark of the Earthquake code.

done: Describe in detail how to create the report from an existing cloudmesh sbatch run.

Make sure that the README is sufficient for running this example, and that it is properly referred to in this report.

Programming bug is in cloudmesh sbatch. Figure out how to run the dummy and document how to run the dummy sbatch.

1 Setup

Identify, for each experiment, if the YAML file is written into the experiment directory. If this is not the case, we need to fix this. We need to have an output directory for each experiment. Must have the ipynb, the output files, and the YAML file that creates the sbatch submission, and the individual SBATCH run file for that particular experiment. The YAML file takes the original in.yaml file and replaces, in that, all the variables that are being defined by the experiment, and then writes a YAML file. That is the yaml file that we need in that directory.

The code is relatively easy to set up for use on various computers using NVIDIA Cards. The code has been run on A100, V100, K100, P100, and RTX3090. Other cards may be also supported, but the code requires a fair amount of

memory. Therefore, it may not be able to be run on NVIDIA Cards with less than 24GB of memory.

The documentation on how to set up the code is available as part of a README.md file under the System Setup section [1]. Instructions to run the code on the University of Virginia's Rivanna Supercomputer can be found on GitHub [2].

1.1 Rivanna

Running the earthquake code on Rivanna can be done on different filesystems available on the HPC center. This includes localscratch, a local filesystem. The localscratch configuration is best as it uses local memory, leading to faster run-times, and it does not require special permissions like access to the /project network file system.

After running the code on Rivanna using the localscratch configuration, the output files can be found under the mlcommons dir in the home directory:

```
cd ~/mlcommons/benchmarks/  
  ↪ earthquake/latest/  
  ↪ experiments/rivanna/  
  ↪ localscratch  
ls
```

2 Cloudmesh

The code has been enhanced with benchmark functions from Cloudmesh [3]. It also uses a convenient extension to batch scripts called cloudmesh-sbatch allowing more easily to place additional templated parameters into the

SBATCH or LSF shell script parameters when submitting to batch systems. In addition, we have developed a cloudmesh compute cluster workflow management tool that allows us to submit the codes in parallel to multiple supercomputers as well as parallel batch jobs [4, 5].

2 Week Intervals is the wrong label, label must be unique

3 Report Generation

To create the report figures, we assume you have fully run the Earthquake code on Rivanna. Execute the following commands:

```
mkdir ~/cm
cd ~/cm
git clone https://github.com/
    ↪ laszewsk/mlcommons.git
cd mlcommons/benchmarks/
    ↪ earthquake/latest/report

# make fetch only works on linux,
# so run in WSL on windows
make fetch

make images
make
```

4 Benchmark Terminology

Table 1: Terminology of benchmark labels

| Label | Definition |
|---------------|--------------------------------------|
| Year Back | Analysis of the previous year |
| 6 Months Back | Analysis of previous 6 months |
| 3 Months Back | Analysis of the previous 3 months |
| 2 weeks Now | Analysis of 2 weeks |
| 2wk+7AVG | The average of 7 two-week intervals |
| 2wk+13AVG | The average of 13 two-week intervals |
| 2wk+26AVG | The average of 26 two-week intervals |

5 Results from Rivanna V100

Rivanna is a supercomputer located at the University of Virginia. It has a variety of CUDA cards, including an A100, V100, K80. In this benchmark we use its V100.

5.1 Best Accuracy

Just one entry for best accuracy.

5.2 Accuracy Comparison

We compare accuracy.

5.3 Time Comparison

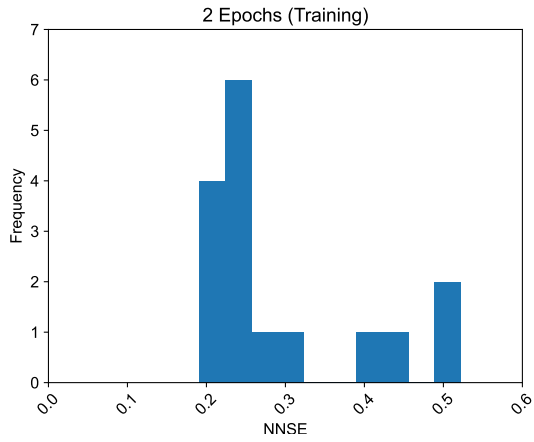
We compare time.

5.4 Power Consumption

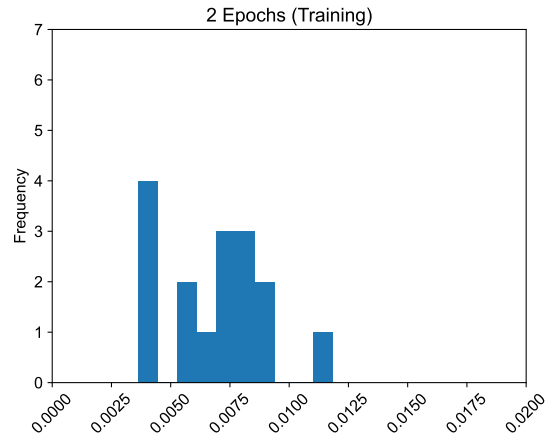
We compare power.

Table 2: To be determined

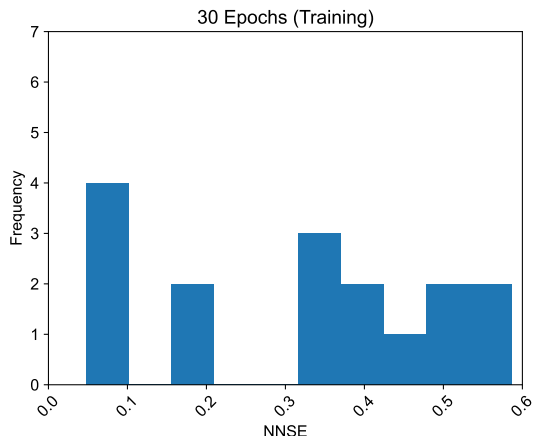
| NNSE | Name |
|----------|---------------|
| 0.191300 | Year Back |
| 0.192700 | 6M 2wk+7AVG |
| 0.197000 | 6M 2wk+13AVG |
| 0.201600 | 6 Months Back |
| 0.232600 | 1Y 2wk+13AVG |
| 0.233000 | 3 Months Back |
| 0.235800 | 1Y 2wk+7AVG |
| 0.243000 | 3M 2wk+7AVG |
| 0.251600 | 1Y 2wk+26AVG |
| 0.251700 | 6M 2wk+26AVG |
| 0.278800 | 3M 2wk+13AVG |
| 0.302500 | 3M 2wk+26AVG |
| 0.405600 | Now 2wk+7AVG |
| 0.429900 | Now 2wk+13AVG |
| 0.506800 | 2 weeks Now |
| 0.521800 | Now 2wk+26AVG |



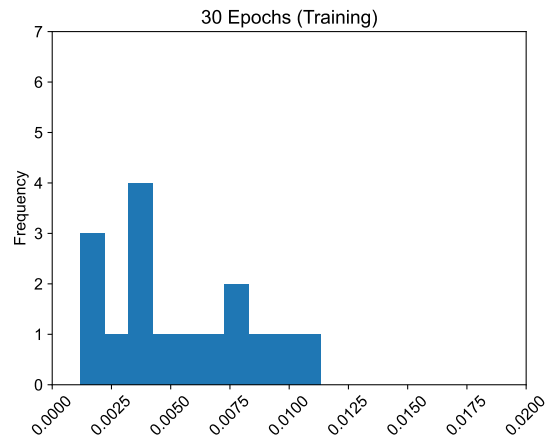
(a) NNSE - 2 epochs



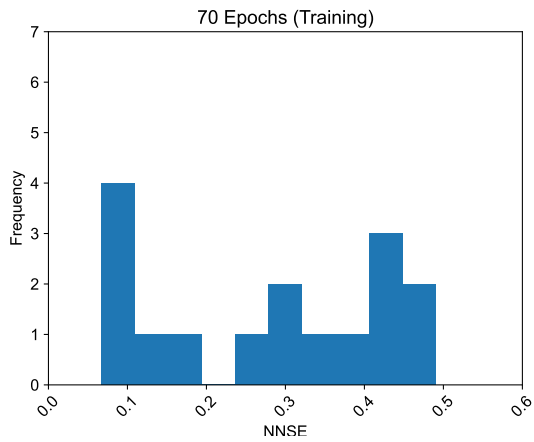
(b) MSE - 2 epochs



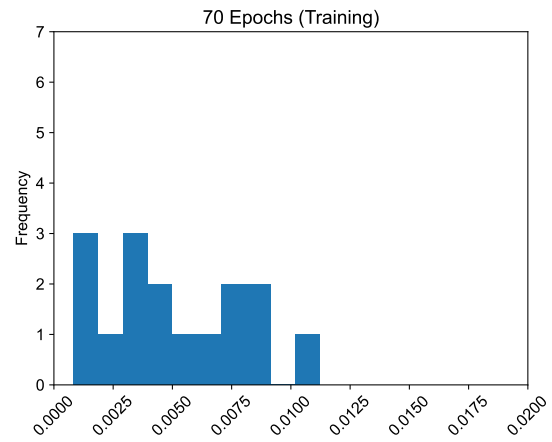
(c) NNSE - 30 epochs



(d) MSE - 30 epochs

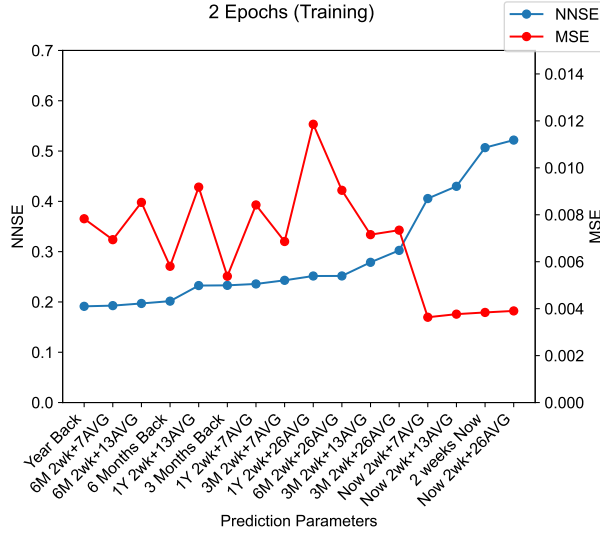


(e) NNSE - 70 epochs

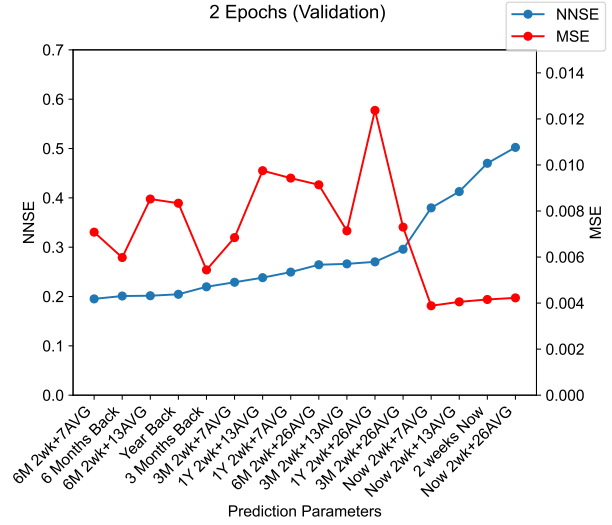


(f) MSE - 70 epochs

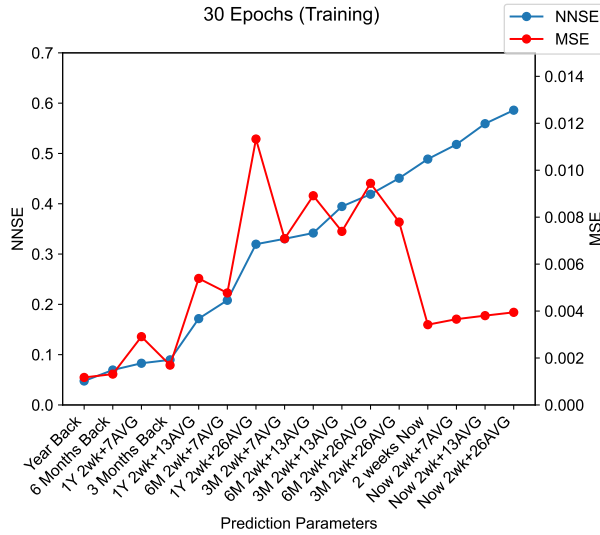
Figure 1: NNSE and MSE values for epochs 2, 30, 70 (training).



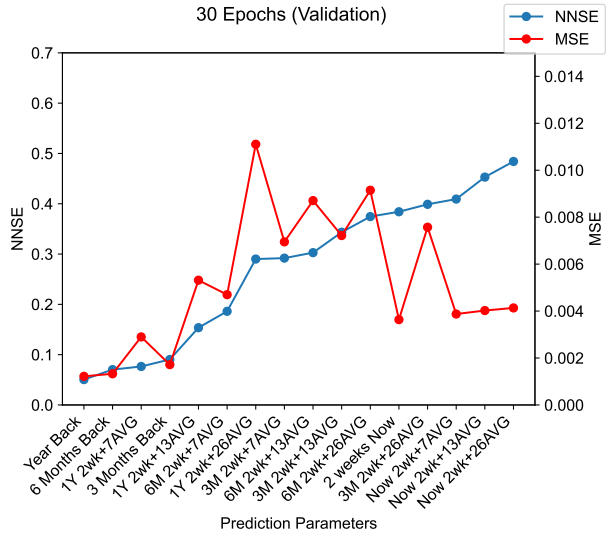
(a) MSE and NNSE - 2 epochs training



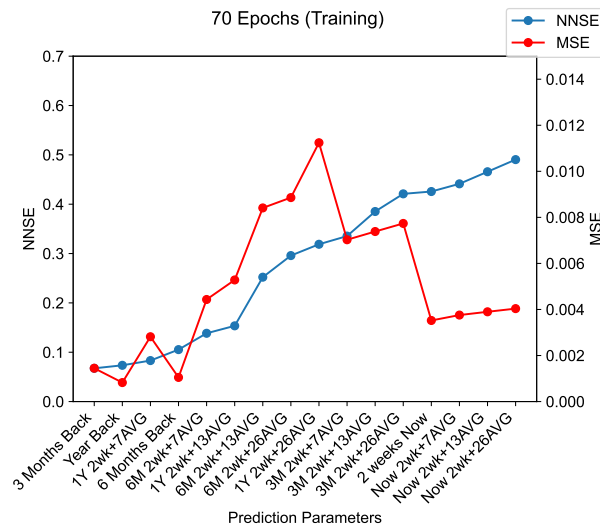
(b) MSE and NNSE - 2 epochs validation



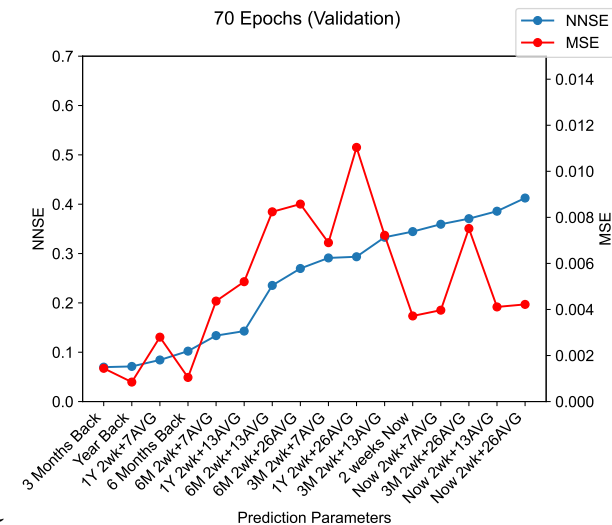
(c) MSE and NNSE - 30 epochs training



(d) MSE and NNSE - 30 epochs validation



(e) MSE and NNSE - 70 epochs training



(f) MSE and NNSE - 70 epochs validation

Figure 2: NNSE and MSE values for epochs 2, 30, 70 (training and validation).

Table 3: To be determined

| NNSE | Name |
|----------|---------------|
| 0.195200 | 6M 2wk+7AVG |
| 0.201000 | 6 Months Back |
| 0.201600 | 6M 2wk+13AVG |
| 0.204500 | Year Back |
| 0.219700 | 3 Months Back |
| 0.228900 | 3M 2wk+7AVG |
| 0.238200 | 1Y 2wk+13AVG |
| 0.249500 | 1Y 2wk+7AVG |
| 0.264400 | 6M 2wk+26AVG |
| 0.266200 | 3M 2wk+13AVG |
| 0.270300 | 1Y 2wk+26AVG |
| 0.295800 | 3M 2wk+26AVG |
| 0.379700 | Now 2wk+7AVG |
| 0.412700 | Now 2wk+13AVG |
| 0.470100 | 2 weeks Now |
| 0.502300 | Now 2wk+26AVG |

Acknowledgements

Continued work was in part funded by the NSF CyberTrain- ing: CIC: CyberTraining for Students and Technologies from Generation Z with the award numbers 1829704 and 2200409 and NIST 60NANB21D151T.

References

- [1] G. C. Fox, G. v. Laszewski, R. Knuuti, T. Butler, and J. Kolessar, “Mlcommons science earthquake benchmark,” GitHub, 2022. [Online]. Available: <https://github.com/laszewski/mlcommons/tree/main/benchmarks/earthquake#readme>
- [2] G. v. Laszewski, R. Knuuti, and J. Fleischer, “Running on rivanna using the simple configuration,” GitHub, 2022. [Online]. Available: <https://github.com/laszewski/mlcommons/tree/main/benchmarks/earthquake/latest/experiments/rivanna#running-on-rivanna-using-the-simple-configuration>
- [3] G. v. Laszewski, “Cloudmesh repositories,” GitHub, 2022. [Online]. Available: <https://github.com/cloudmesh>
- [4] G. von Laszewski, J. P. Fleischer, and G. C. Fox, “Hybrid reusable computational analytics workflow management with cloudmesh,” 2022. [Online]. Available: <https://arxiv.org/abs/2210.16941>
- [5] G. von Laszewski, A. Orlowski, R. H. Otten, R. Markowitz, S. Gandh, A. Chai, G. C. Fox, and W. L. Chang, “Using gas for speedy generation of hybridmulti-cloud auto generated ai services,” in *IEEE COMPSAC 2021: Intelligent and Resilient Computing for a Collaborative World45th Anniversary Conference*. All Virtual: IEEE, Jul. 2021. [Online]. Available: <https://ieeecompsac.computer.org/2021/>