A. Artifact Appendix

A.1 Abstract

This artifact description contains information about the complete workflow required to set up the simulations of NoC design space exploration (DSE).

We describe how the software process the trace form 7 SNN applications. In addition, the scripts implements 3 mapping methods for all these traces. At the end, the script drives NoC's DSE process by driving the simulator.

A.2 Artifact check-list (meta-information)

- Algorithm: Better-History Simulated Annealing (BHSA)
- Compilation: gcc 7.5.0
- Binary: will be compiled on a target platform
- Data set: FSDD, MNIST, N-MNIST, Smooth-320, Smooth-1280, MLP-2048
- Run-time environment: Ubuntu18.04, no root, python 3.6, Matlab (R2018a)
- Metrics: Average Packet Latency (cycles per packet)
- Output: NoC design parameters
- Experiments: We tested seven applications (only 140,000 packets for each application): FSDD, MNIST, etc. Three mapping methods: average, random, sequence. Two NoC network structures: 8x8, 16x16.
- How much disk space required (approximately)?: ĩGb
- How much time is needed to complete experiments (approximately)?: About 15 minutes for each experiment.
- Publicly available?: Yes
- Code licenses?: MIT licenses
- Workflow framework used?: Yes

A.3 Description

A.3.1 How to access

Our benchmarks, source code, scripts are available on `http://doi.org/10.5281/zenodo.3762948`

A.4 Installation

Download flex and bison and unzip to ./booksim2-master/src .
Enter the flex, bison directory and execute separately:

- $./configure$
- make
- make install

Run

- ./booksim ./src/examples/cmesh...

A.5 Experiment workflow

Figure.1 shows the flow of extracting traces and processing of three RC models. It contains 3 stages, using Pychram ($readnp.py$), Matlab ($RC\_Trace.m$), and Booksim2 tools to support the corresponding operations.

Figure.2 shows the flow of extracting traces and processing of four RC models. It contains 2 stages, using Matlab ($MLP\_Trace.m$) and Booksim2 tools to support the operations

- The file ($readnp.py$) receives the output (10 files) from the Brain2 simulator as input. Extract the connection relationship of the corresponding neuron ($connection\_1000x1000.txt$), the fire situation of the neuron ($index.txt, time.txt$)
- The file ($RC\_Trace.m$)receives these file generated by the script ($readnp.py$), and generates the corresponding routing trace file according to the mapping strategy.
- The file ($MLP\_Trace.m$)receives the file come from CARLsim , and generates the corresponding routing trace file according to the mapping strategy.
- Booksim similator configures the corresponding NoC size ($8\times*8, 16\times16$), and receives the above router file as input. Then use the BHSA algorithm to explore the design space

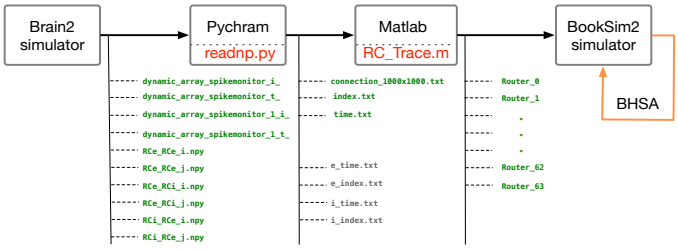

Figure 1. Processing flow for RC model.

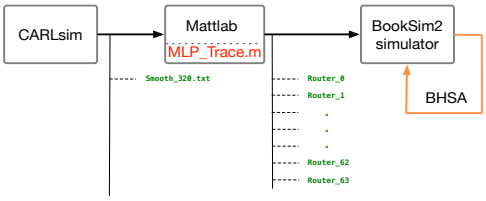

Figure 2. Processing flow for MLP model.

A.6 Evaluation and expected result

The trace format generated by the mapped spikes of the neurons in the SNN is as follows:

```
824 1
824 2
824 3
824 6
824 8
824 9
824 11
```

Figure 3. FSDD trace format of Router_0 in 16×16 NoC. The first column is the transmission time, the second column is the destination node.

After executing the following command:

```
$ python BHSA.py | tee log
```

the intermediate results of the entire search process will be saved to the *log* file. Finally, a set of NoC parameters and the corresponding transmission latency will be output on the console.One of the running examples is shown in the figure:One of the running examples is shown in the figure:

```
best:  1 4 5 1 2 6 2 3311.98
```

Figure 4. Running result.

As shown in the Figure.4, the optimal configuration found by DSE is $< 1, 4, 5, 1, 2, 6, 2 >$, and the average transmission delay corresponding to this configuration is 3311.98 cycles.