

# Artifact Overview

## Getting Started Guide

1. Install [Docker](#) on a Linux machine
2. Clone the [artifact repo](#) and cd into it

```
git clone https://github.com/jialin-li/linear-veribetrkv-artifact.git
cd linear-veribetrkv-artifact
```
3. Get the artifact docker image
  - Option 1: Pull from dockerhub and rename the image

```
docker pull lsixseven/linear-eval
docker tag lsixseven/linear-eval linear-eval
```
  - Option 2: Build the image locally

```
docker build -t linear-eval .
```
4. Create a directory for the container to write results to, make sure to use absolute path in all the commands

```
mkdir /tmp/linear-eval
```
5. Run the basic test and wait for the container to exit (~ 3 minutes)

```
docker run --rm -v /tmp/linear-eval:/home/root/output linear-eval
scripts/run-experiments.sh basic-test
```
6. Check the basic test result at `/tmp/linear-eval/basic-log`

The log should show the verification result of a module written in dynamic frames and the same module in linear types. All procedures should be verified.
7. Tada! You just ran the artifact container and collected output from it :) Take a look at the Step by Step instructions on the next page!

## Step by Step Instructions

### Reproduce the Result

To reproduce all the experiments, run the command below. The experiments run single-threaded and take 3 to 4 hours to complete. We do not recommend running other workloads while the experiments are running, as that will affect the verification time of the code. If you want to run the experiments in the background, add `-d` right after `docker run`.

```
docker run --rm -v /tmp/linear-eval:/home/root/output linear-eval
scripts/run-experiments.sh all
```

Once finished, the output can be collected at `/tmp/linear-eval`, and figures from the paper are under `/tmp/linear-eval/figures`. Figure 6 is `loc-breakdown.png`, Figure 7 is `module-vertime.png`, and Figure 8 is `method-diff-cdf.png`. The error message examples described in the paper can be found at `/tmp/linear-eval/error-log`.

### Run Individual Experiments

To run experiments separately, replace `all` from the command to `loc` to run just the `loc` experiment ( $\sim 10$  minutes), to `vertime` to run just the verification time experiments ( $\sim 2-3$  hours), or `error-msg` to show the error message examples ( $\sim 5$  minutes).

```
docker run --rm -v /tmp/linear-eval:/home/root/output linear-eval
scripts/run-experiments.sh loc
```

```
docker run --rm -v /tmp/linear-eval:/home/root/output linear-eval
scripts/run-experiments.sh vertime
```

```
docker run --rm -v /tmp/linear-eval:/home/root/output linear-eval
scripts/run-experiments.sh error-msg
```

### Smaller Input

The actual verification time experiments take a couple hours, but there is a `vertime-small` option ( $\sim 15$  minutes) that only runs on the committer module. The resulting figures are based solely on the committer module: the new Figure 7 only has an entry for `I0`, and the new Figure 8 shows the CDF of all committer methods over 5s. Results can be found at `/tmp/linear-eval/figures`

```
docker run --rm -v /tmp/linear-eval:/home/root/output linear-eval
scripts/run-experiments.sh vertime-small
```

If you read the `/tmp/linear-eval/vtime-log` file, there will be many messages for "file x does not exist in vertime", this is expected, as we are only testing a subset of the implementation.

### Interactive Testing

To interactively play with the system, run the following command:

```
docker run -it --rm -v /tmp/linear-eval:/home/root/output linear-eval
```

To verify `Impl/CacheImpl.i.dfy` inside the container, `cd` into the `veribetrkv` directory and run `tools/local-dafny.sh Impl/NodeImpl.i.dfy /compile:0 /trace`

### Supported Claims

1. Proof to code ratio is less for linearized implementation compared to the dynamic frames version, consistent with Figure 6 in the paper.
2. The total time for type checking and verification of each module in linearized VeribetrKV should be less than the time for the dynamic frames version, consistent with Figure 7.

3. For methods that take longer than 5 seconds to verify, over 85% of the linearized methods should verify faster than the dynamic frames version. The 25% and 50% labels in the CDF graph should show roughly the same amount (within 10%) <sup>1</sup> of verification time speed up compared to what is shown in Figure 8.
4. Error messages in the `error-log` is consistent with the behavior described in Section 3.6

## Not Supported Claims

None

---

<sup>1</sup>The exact verification time for each method varies on different machines. The reproduced Figure 7 and 8 are unlikely to be exactly the same as the paper, hence roughly the same amount.