Piranha: A GPU Platform for Secure Computation

Jean-Luc Watson, Sameer Wagh, and Raluca Ada Popa,
University of California, Berkeley

https://www.usenix.org/conference/usenixsecurity22/presentation/watson

This artifact appendix is included in the Artifact Appendices to the Proceedings of the 31st USENIX Security Symposium and appends to the paper of the same name that appears in the Proceedings of the 31st USENIX Security Symposium.

August 10–12, 2022 • Boston, MA, USA

978-1-939133-31-1

A Artifact Appendix

A.1 Abstract

Piranha is an open-source platform (https://github.com/ucbrise/piranha) for accelerating multi-party computation protocols using the GPU. Our artifact for Piranha consists of the prototype platform, along with scripts needed to replicate our experiments from Section 6. In particular, our artifact supports 3 LSSS protocols and a protocol-agnostic Neural Network training library. We demonstrate that such GPU acceleration can speed complex MPC applications like NN training by orders of magnitude compared to the CPU, achieves meaningful training results, and does so in a significantly more memory-efficient manner than prior work.

Our artifact is a combination of the Piranha platform and associated test scripts that can be used to validate these claims, specifically in replicating Figures 4-7 and Tables 2-4 in our paper, by executing the relevant micro- and end-to-end benchmarks. Piranha requires that each party participating in the protocol provision a machine with a GPU, along with access to the NVIDIA CUDA toolkit. The expected result of evaluation is to be able to replicate each major paper figure.

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

- **Program:** C++-based MPC GPU acceleration platform, with associated protocol-independent neural network library.
- **Run-time environment:** Cloud-based GPU cluster, with parties each executing on dedicated GPUs. A separate control server dispatches test runs to each party.
- **Execution:** Script-based for individual figure and table replication.
- **Metrics:** Computation and communication time, communication cost, training/test accuracy, and GPU memory footprint.
- **Output:** Data replicating paper figures and tables.
- **Experiments:** Secure training and inference passes, basic accelerated operation benchmarks, memory footprint measurement under various changes
- **How much time is needed to prepare workflow (approximately)?:** Provisioning a cluster of GPUs and a control server to run experiments might require 1-2 hours to set up.
- **How much time is needed to complete experiments (approximately)?:** Multi-party computation remains very slow compared to state-of-the-art plaintext. Expect almost no hands-on time but at least 40 hours of compute-time without evaluating the largest network (VGG16) and upwards of 400 compute-hours for VGG to replicate Figure 5.
- **Publicly available (explicitly provide evolving version reference)?:** Available at https://github.com/ucbrise/piranha/tree/main
- **Code licenses (if publicly available)?:** Piranha is licensed under the MIT License
- **Archived (explicitly provide DOI or stable reference)?:** https://github.com/ucbrise/piranha/commit/ddfb646f6f0e37e20194e4437e0d8e303fd89e4c

A.3 Description

The Piranha artifact consists of (1) the implementation of the device layer for integer-based GPU acceleration, (2) three LSSS protocols with varying party setups, (3) a neural network training library, and (4) a set of experimental runtime scripts to replicate the experiments we show in the manuscript.

For artifact evaluation, we assume a cloud-based cluster of compute- and GPU-provisioned machines, for MP-SPDZ and Piranha benchmarks, respectively. For LAN and WAN experiments, a replicator will provision 4 machines in a local (1ms latency) network and 2 additional machines in a different network (60ms latency), each acting as one party in a Piranha computation. Finally, a separate machine should act as a control server that runs the replication script.

A.3.1 Hardware dependencies

Piranha requires that each machine be provisioned with an NVIDIA GPU, and was initially evaluated on NVIDIA Tesla V100 GPUs.

A.3.2 Software dependencies

In a standalone installation, Piranha depends on installed GPU drivers, the NVIDIA CUDA Toolkit libraries, as well as CUTLASS and gtest.

A.3.3 Data sets

Piranha uses common ML datasets – MNIST, CIFAR10, and Tiny ImageNet; you may use scripts provided in the artifact to download and format the MNIST and CIFAR10 datasets we use for training.

A.3.4 Models

The artifact includes as part of the neural network training library a set of common neural network architectures in JSON format that can be executed using Piranha.

A.3.5 Security, privacy, and ethical concerns

N/A. All evaluation is done between parties that the evaluator directly controls; there is no interaction with third parties.

A.4 Installation

To install the artifact, clone the repository and follow the most up-to-date installation instructions at https://github.com/ucbrise/piranha. Install the required software dependencies, and, if using Piranha to perform training or inference, download the desired datasets with the provided scripts. Build the needed integer kernels with CUTLASS and compile the project for a given fixed point precision and multi-party protocol using the step-by-step instructions provided.

A.5 Evaluation and expected results

The primary evaluation script can be found at `experiments->run_experiment.py`, which will connect to and spawn execution on each of the GPU-provisioned parties as needed. Use the script to individually recreate figures and tables from the paper. You can choose to focus on the faster results first, leaving the extremely long VGG16-based results until you've verified the first set.

The main claims of the paper are that (1) GPU acceleration can improve MPC runtime in a protocol-agnostic manner far above CPU performance, (2) enabling realistic machine learning training, and (3) that Piranha accomplishes this while providing much better memory efficiency. Figure 4 and Table 4 support the first, showing a large improvement in runtime. Table 2 and Figures 5 and 6 show that real training results can be achieved in a reasonable amount of time, while Table 3 and Figure 7 demonstrate the benefits of Piranha's memory-conscious approach on memory footprint, thus achieving larger batch sizes. Reproducing these results involves running individual iterations or full training epochs for every table or figure in the paper, and comparing the output values or figures to those in the manuscript.

A.6 Version

Based on the LaTeX template for Artifact Evaluation V20220119.