# StableHLO

Eugene Burmako, 11/17/2022
Web Machine Learning Working Group

# Introduction

- StableHLO is a backward compatible ML compute opset.
- Developed as a portability layer between ML frameworks and ML compilers.
- Aiming for adoption by a wide variety of ML frameworks including TensorFlow, JAX and PyTorch, and ML compilers including XLA and IREE.

# Scope of StableHLO

Artifacts that the StableHLO community produces, maintains and evolves

1. Operation set
2. C++ and Python bindings (through MLIR)
3. Serialization format
4. Conformance suite *

* Work in progress, ETA: H1 2023

# Why StableHLO?

There are related opsets in the community, e.g. MIL, ONNX, TFLite, TOSA.

1. Reduced instruction set
2. Support in JAX, PyTorch and TensorFlow
3. Strong track record on the server, ongoing work on mobile
4. Open governance

# Reduced instruction set

| | |
|---|---|
| Control Flow | case, if, while |
| Data Movement | broadcast_in_dim, concatenate, gather, pad, reshape, reverse, scatter, slice, sort, transpose |
| Distribution | after_all, all_gather, all_reduce, all_to_all, collective_permute, infeed, outfeed, recv, reduce_scatter, replica_id, send |
| Elementwise | abs, add, and, atan2, bitcast_convert, cbrt, ceil, clamp, compare, complex, convert, cosine, count_leading_zeros, divide, exponential, exponential_minus_one, floor, imag, is_finite, log, log_plus_one, logistic, map, maximum, minimum, multiply, negate, not, or, popcnt, power, real, reduce_precision, remainder, round_nearest_afz, round_nearest_even, rsqrt, select, shift_left, shift_right_arithmetic, shift_right_logical, sign, sine, sqrt, subtract, tanh, xor |
| Extensibility | custom_call, get_tuple_element, tuple |
| Miscellaneous | cholesky, constant, fft, iota, optimization_barrier, resize, rng, rng_bit_generator, triangular_solve |
| Modularity | call, func, module, return |
| Quantization | uniform_dequantize, uniform_quantize |
| Reduction | convolution, dot_general, reduce, reduce_window, select_and_scatter |

# Support in ML frameworks

- StableHLO can be produced by JAX, TensorFlow and PyTorch.
- Strong internal alignment at Google on support for StableHLO as the plan of record.
- Ongoing work on using StableHLO on mobile: adding to the TFLite flatbuffer schema, producing in the TFLite converter.

# Support in ML compilers

- Within the XLA compiler, HLO has been successfully used at Google scale both for training and inference for 5+ years.
- IREE is addressing some of the most important emerging ML use cases and supports StableHLO as one of its canonical inputs.

# OpenXLA

A community-driven, open source ML compiler ecosystem, using the best of XLA & MLIR.

Overview | Repositories 5 | Discussions | Projects | Packages | People

## Pinned

**xla** Public

A community-driven and modular open source compiler for ML.

127 5

**stablehlo** Public

Backward compatible ML compute opset inspired by HLO/MHLO

MLIR 59 16

**community** Public

Stores documents and resources used by the OpenXLA developer community

44 1

# Relationship with related opsets

- There are related opsets in the community, e.g. MIL, ONNX, TFLite, TOSA.
- We want to learn and collaborate - there are a lot of good ideas out there, and we're looking to evolve StableHLO beyond what's there in HLO.
- On our side, we can offer production-grade support for lowering from JAX, PyTorch and TensorFlow.
- There is ongoing work on connecting StableHLO to many of these opsets.

openxla / stablehlo Public

Notifications | Fork 16 | Star 59

Code | Issues 129 | Pull requests 36 | Actions | Projects | Security | Insights

main | 1 branch | 0 tags | Go to file | Code

GleasonK Improve CHLO code coverage for verifiers, type inference, reify (#551) ... ✓ 4aff405 yesterday 240 commits

| | | |
|---|---|---|
| .github | Build StableHLO as a part of LLVM in CI (#457) | 8 days ago |
| build_tools | Integrate LLVM at llvm/llvm-project@0b94525ddcfc (#515) | 2 days ago |
| docs | Add spec for ScatterOp (#330) | yesterday |
| stablehlo | Improve CHLO code coverage for verifiers, type inference, reify (#551) | yesterday |
| .clang-format | Refactor the lint action into a build script, temporarily update .cla... | 2 months ago |
| .gitignore | Add spec for SortOp (#310) | last month |
| CMakeLists.txt | Revert #457 embedded build change (#545) | 2 days ago |
| CODE_OF_CONDUCT.md | Initial commit | 4 months ago |
| CONTRIBUTING.md | Initial commit | 4 months ago |
| LICENSE | Initial commit | 4 months ago |
| README.md | Format markdown files consistently with MLIR-HLO (#128) | 2 months ago |

## About

Backward compatible ML compute opset inspired by HLO/MHLO

- Readme
- Apache-2.0 license
- Code of conduct
- 59 stars
- 17 watching
- 16 forks

## Contributors 12

## Languages

MLIR 54.2% C++ 41.1%