开源之夏校园行
Buddy Compiler 的开源故事

张洪滨 | 中国科学院软件研究所 智能软件研究中心
hongbin2019@iscas.ac.cn

Buddy Compiler

Buddy Compiler 简介和愿景

Buddy Compiler

“

Buddy Compiler 是领域特定的编译器框架，

致力于打造基于 MLIR 和 RISC-V 的软硬件协同设计生态。

我们的目标是实现从 DSL 到 DSA 的编译流程和协同设计。

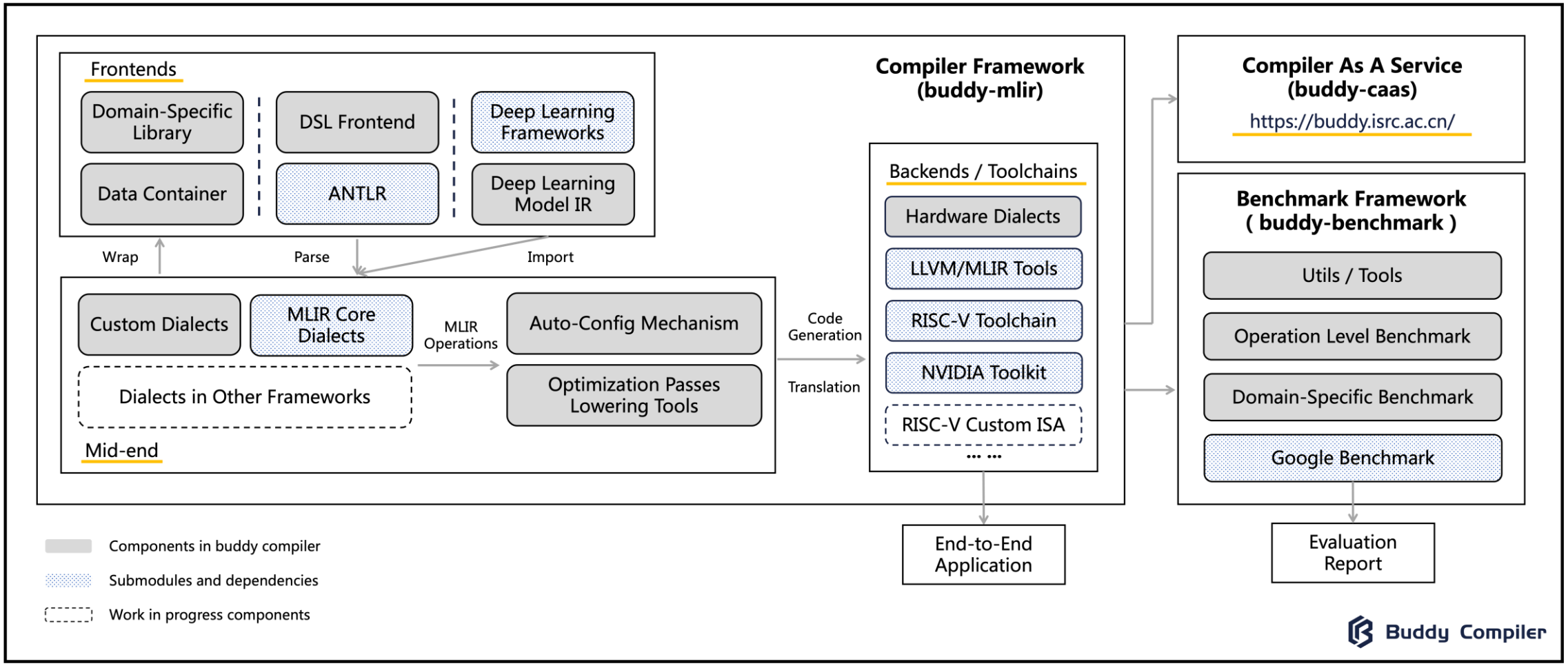
我们的愿景是让领域特定的协同设计不再困难。

”

Buddy Compiler 概览

Buddy Compiler

“Buddy System” for Domain-Specific Compilers | MLIR-Based Compilation Framework for Deep Learning Co-Design



Homepage: https://buddy-compiler.github.io/
GitHub: https://github.com/buddy-compiler

Buddy Compiler 协同设计

Buddy Compiler

“

MLIR 和 RISC-V 是编译器和体系结构协同设计的绝妙搭配!

MLIR 和 RISC-V 都具备模块化和可扩展的特性，

可以组成庞大的协同设计生态。

统一的协同设计生态能够激发出更多的优化机会。

”

MLIR 简介

Buddy Compiler

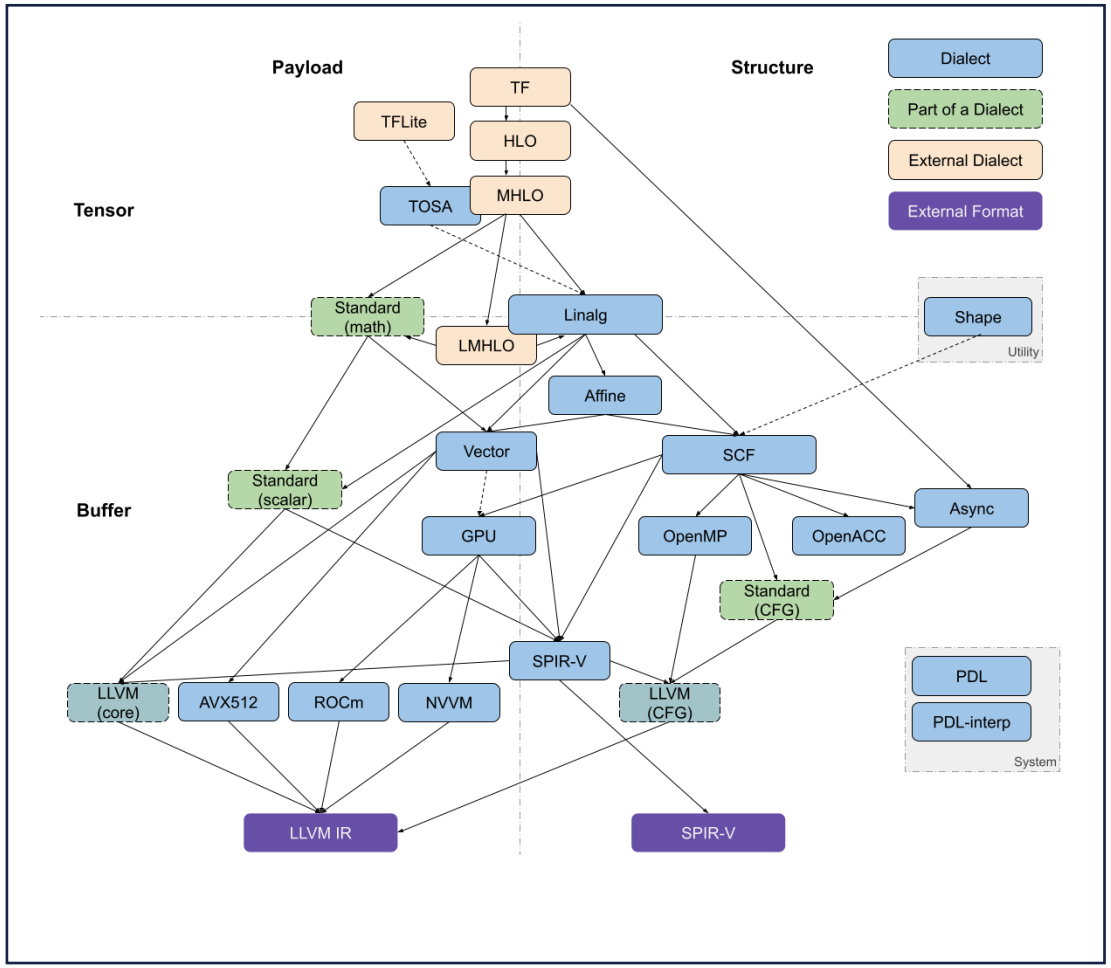
MLIR 概述

MLIR (Multi-Level Intermediate Representation)

是可重用、可扩展的编译器基础设施。

MLIR 生态

- 深度学习框架支持：Torch-MLIR, MHLO-MLIR, ONNX-MLIR
- 端到端编译器：IREE, Buddy Compiler
- 硬件编译器支持：CIRCT
- 面向特定硬件的编译器支持：Triton，TPU-MLIR
- 编译器前端：VAST, Beaver
- 编程语言：Mojo🔥



MLIR：多层中间表示编译基础设施

RISC-V 简介

Buddy Compiler

RISC-V ISA base and extensions[1]

Name	Description	Version	Status[A]	Instruction count
Base				
RVWMO	Weak Memory Ordering	2.0	Ratified	
RV32I	Base Integer Instruction Set, 32-bit	2.1	Ratified	40
RV32E	Base Integer Instruction Set (embedded), 32-bit, 16 registers	2.0	Ratified	40
RV64I	Base Integer Instruction Set, 64-bit	2.1	Ratified	15
RV64E	Base Integer Instruction Set(embedded), 64-bit	2.0	Ratified	
RV128I	Base Integer Instruction Set, 128-bit	1.7	Open	15

Name	Description	Version	Status[A]	Instruction count
Extension				
M	Standard Extension for Integer Multiplication and Division	2.0	Ratified	8 (RV32) 13 (RV64)
A	Standard Extension for Atomic Instructions	2.1	Ratified	11 (RV32) 22 (RV64)
F	Standard Extension for Single-Precision Floating-Point	2.2	Ratified	26 (RV32) 30 (RV64)
D	Standard Extension for Double-Precision Floating-Point	2.2	Ratified	26 (RV32) 32 (RV64)
Zicsr	Control and Status Register (CSR) Instructions	2.0	Ratified	6
Zifencei	Instruction-Fetch Fence	2.0	Ratified	1
G	Shorthand for the IMAFDZicsr_Zifencei base and extensions	—	—	
• • •	• • •			
P	Standard Extension for Packed-SIMD Instructions	0.9.10	Open	
V	Standard Extension for Vector Operations	1.0	Ratified	187[29]
• • •	• • •			

RISC-V®

Specification Status - https://wiki.riscv.org/display/HOME/Specification+Status

[1] The origin image is from Wikipedia - https://en.wikipedia.org/wiki/RISC-V

Buddy Compiler 协同设计

Buddy Compiler

RISC-V®

SIMD Processor	Vector Processor	GPGPU	DSA
(RISC-V P Extension)	(RISC-V V Extension)	(e.g. Ventus)	(e.g. Gemmini)

Buddy Compiler 协同设计

Buddy Compiler

OpenCV

Buddy Compiler Libraries or DSL

Multimodal Representations

Buddy Compiler Domain-Specific Dialects

TensorFlow

PyTorch

OpenXLA

Deep Learning Model Representations

MLIR TOSA / Linalg Dialect

RISC-V®

SIMD Processor
(RISC-V P Extension)

Vector Processor
(RISC-V V Extension)

GPGPU
(e.g. Ventus)

DSA
(e.g. Gemmini)

Buddy Compiler 协同设计

Buddy Compiler

OpenCV

Buddy Compiler Libraries or DSL

Multimodal Representations

Buddy Compiler Domain-Specific Dialects

TensorFlow PyTorch OpenXLA

Deep Learning Model Representations

MLIR TOSA / Linalg Dialect

MLIR Core Dialects

- MemRef Dialect
- Affine Dialect
- SCF Dialect

… …

RISC-V®

SIMD Processor
(RISC-V P Extension)

Vector Processor
(RISC-V V Extension)

GPGPU
(e.g. Ventus)

DSA
(e.g. Gemmini)

Buddy Compiler 协同设计

Buddy Compiler

OpenCV

Buddy Compiler Libraries or DSL

Multimodal Representations

Buddy Compiler Domain-Specific Dialects

TensorFlow PyTorch OpenXLA

Deep Learning Model Representations

MLIR TOSA / Linalg Dialect

Vector Representation

- Vector Dialect
- RVV Dialect
- LLVM VP Intrinsic

MLIR Core Dialects

- MemRef Dialect
- Affine Dialect
- SCF Dialect

… …

RISC-V®

SIMD Processor
(RISC-V P Extension)

Vector Processor
(RISC-V V Extension)

GPGPU
(e.g. Ventus)

DSA
(e.g. Gemmini)

Buddy Compiler 协同设计

Buddy Compiler

OpenCV

Buddy Compiler Libraries or DSL

Multimodal Representations

Buddy Compiler Domain-Specific Dialects

TensorFlow PyTorch OpenXLA

Deep Learning Model Representations

MLIR TOSA / Linalg Dialect

Vector Representation

- Vector Dialect
- RVV Dialect
- LLVM VP Intrinsic

MLIR Core Dialects

- MemRef Dialect
- Affine Dialect
- SCF Dialect

… …

Gemmini Dialects

- Gemmini Operation
- Gemmini Intrinsic Operation
- Custom LLVM Extension

RISC-V®

SIMD Processor
(RISC-V P Extension)

Vector Processor
(RISC-V V Extension)

GPGPU
(e.g. Ventus)

DSA
(e.g. Gemmini)

Buddy Compiler 协同设计

Buddy Compiler

OpenCV | Buddy Compiler Libraries or DSL

Multimodal Representations

Buddy Compiler Domain-Specific Dialects

TensorFlow | PyTorch | OpenXLA

Deep Learning Model Representations

MLIR TOSA / Linalg Dialect

Vector Representation

- Vector Dialect
- RVV Dialect
- LLVM VP Intrinsic

MLIR Core Dialects

- MemRef Dialect
- Affine Dialect
- SCF Dialect

… …

Gemmini Dialects

- Gemmini Operation
- Gemmini Intrinsic Operation
- Custom LLVM Extension

LLVM | RISC-V GNU Toolchain | Emulators

RISC-V®

SIMD Processor
(RISC-V P Extension)

Vector Processor
(RISC-V V Extension)

GPGPU
(e.g. Ventus)

DSA
(e.g. Gemmini)

Buddy Compiler 协同设计

Buddy Compiler

OpenCV | Buddy Compiler Libraries or DSL

Multimodal Representations

Buddy Compiler Domain-Specific Dialects

TensorFlow | PyTorch | OpenXLA

Deep Learning Model Representations

MLIR TOSA / Linalg Dialect

Vector Representation
- Vector Dialect
- RVV Dialect
- LLVM VP Intrinsic

MLIR Core Dialects
- MemRef Dialect
- Affine Dialect
- SCF Dialect

… …

Gemmini Dialects
- Gemmini Operation
- Gemmini Intrinsic Operation
- Custom LLVM Extension

LLVM | RISC-V GNU Toolchain | Emulators

RISC-V®

SIMD Processor (RISC-V P Extension)

Vector Processor (RISC-V V Extension)

GPGPU (e.g. Ventus)

DSA (e.g. Gemmini)

Preprocessing + Deep Learning Workload
- Preprocessing Operation Optimization
- Unified Data Structure to Avoid Copy Overhead
- Potential Operation Fusion Opportunity

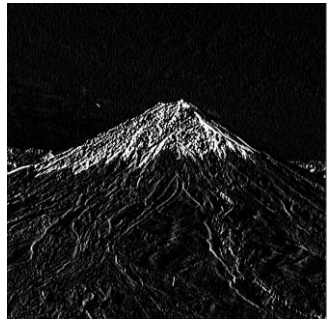
Compiler Passes + Hardware Architecture
- Design Representations for Hardware Features
- Configure Passes by Hardware Information
- Potential Auto-Tuning / DSE Opportunity

Buddy Compiler 领域特定应用支持

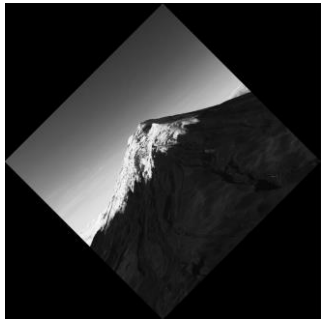
Buddy Compiler

Image Processing

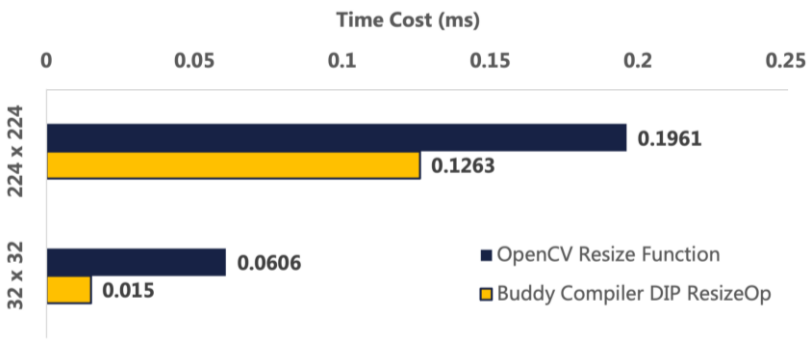
Original Image [1]



Correlation

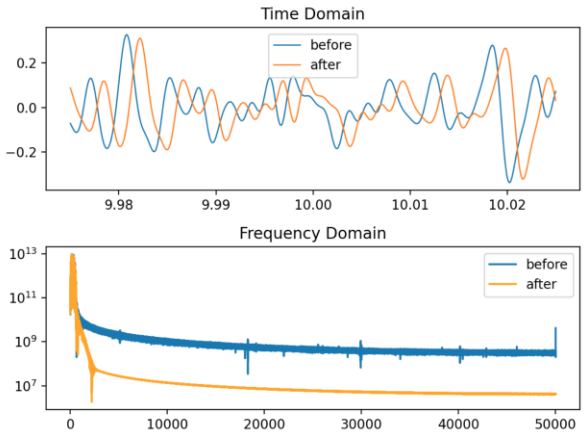


Rotation



Resize Performance Evaluation

Audio Processing[2]



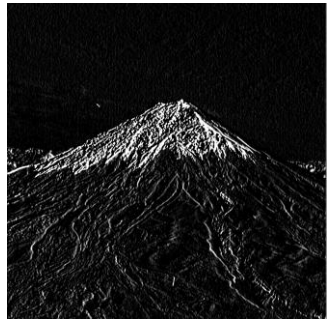
[1] The origin image is from MediaStorm - https://www.ysjf.com/materialLibrary
[2] The origin audio is from NASA ‘s recording of sound on Mars - https://www.nasa.gov/connect/sounds/index.html

Buddy Compiler 领域特定应用支持

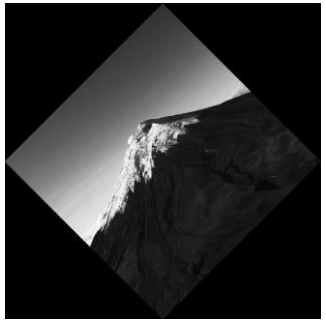
Buddy Compiler

Image Processing

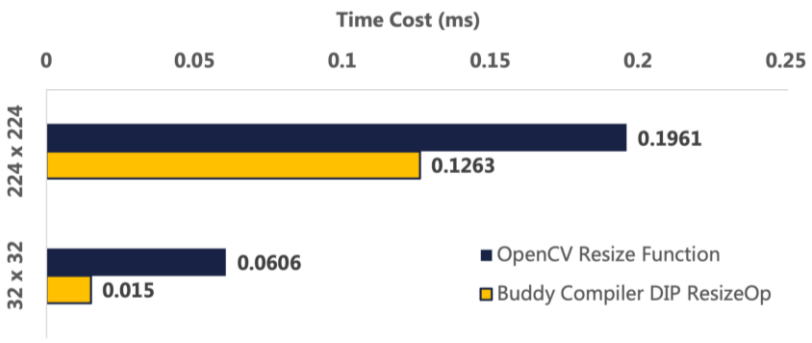
Original Image [1]



Correlation

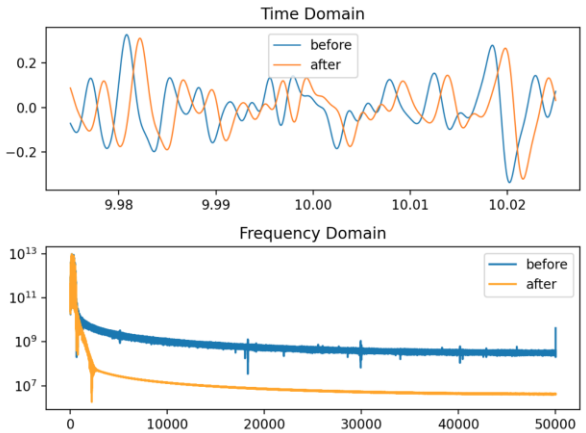


Rotation

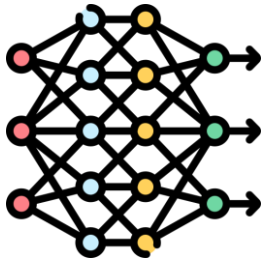


Resize Performance Evaluation

Audio Processing[2]



Deep Learning E2E Application



Data Container

Preprocessing

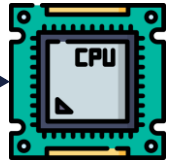
Model MLIR Function

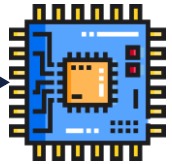
MLIR Core Abstraction

RVV-Specific Support

Gemmini-Specific Support

… …





[1] The origin image is from MediaStorm - https://www.ysjf.com/materialLibrary
[2] The origin audio is from NASA ‘s recording of sound on Mars - https://www.nasa.gov/connect/sounds/index.html

Buddy Compiler 社区现状

Buddy Compiler

Buddy Compiler 社区

开发者分布：中国 CN 印度 IN 俄罗斯 RU 喀麦隆 CM

团队分支：Core Team | Domain-Specific Compiler | MLIR + RISC-V Ecosystem | Optimization | Buddy-CAAS

Buddy MLIR 仓库（https://github.com/buddy-compiler/buddy-mlir）

Star： 216 | Fork：80

开发者：

Buddy Benchmark 仓库（https://github.com/buddy-compiler/buddy-benchmark）

Star： 25 | Fork：20

开发者：

Buddy Compiler 社区服务（https://buddy.isrc.ac.cn/）

Buddy Compiler

ves MLIR users and developers to configure the pass pipeline and demonstrate on multiple backends quic

Buddy Compiler

We ❤ MLIR Playground

RISC-V X86

Choose a Backend

Example Dir: MLIRLinalg

File: linalg-matmul.mlir

Code Input

Choose an Example or Insert Your Code

```
func.func @matmul(%a : memref<?x?xf32>, %b : memref<?x?xf32>, %c : memref<?x?xf32>) {
  linalg.matmul
    ins(%a, %b: memref<?x?xf32>, memref<?x?xf32>)
   outs(%c:memref<?x?xf32>)
  return
}

func.func @main(){
  // Set up dims.
  %cM = arith.constant 4 : index
  %cN = arith.constant 4 : index
  %cK = arith.constant 4 : index

  // Set Init Value.
  %cf1 = arith.constant 1.0 : f32

  %A = memref.alloc(%cM, %cK) : memref<?x?xf32>
  %B = memref.alloc(%cK, %cN) : memref<?x?xf32>
  %C = memref.alloc(%cM, %cN) : memref<?x?xf32>
```

Configure the Lowering Pass Pipeline

Lower Box | Translate Box | Compile Box | Link Box | Execute Box

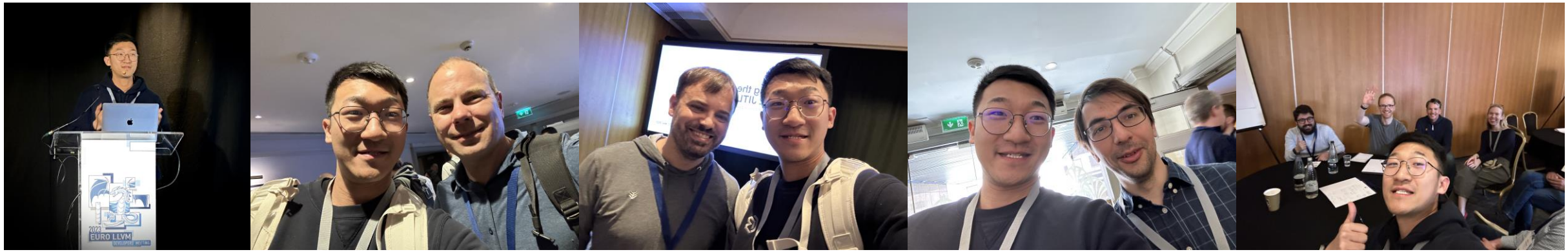
-convert-linalg-to-loops | -lower-affine | -convert-scf-to-cf | -convert-vector-to-llvm | -convert-memref-to-llvm | -convert-arith-to-llvm | -convert-func-to-llvm | -reconcile-unrealized-casts

Result Output

Buddy Compiler @ EuroLLVM 2023

Buddy Compiler

“软硬件协同设计终究是“人”的协同，让软件开发者和硬件开发者互相理解是协同设计的关键…”

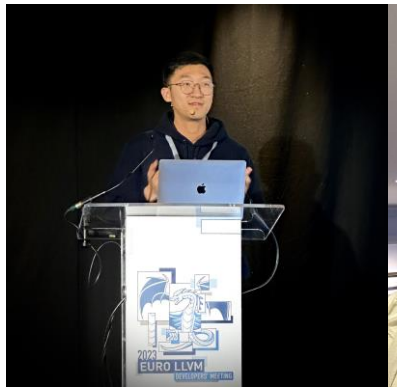


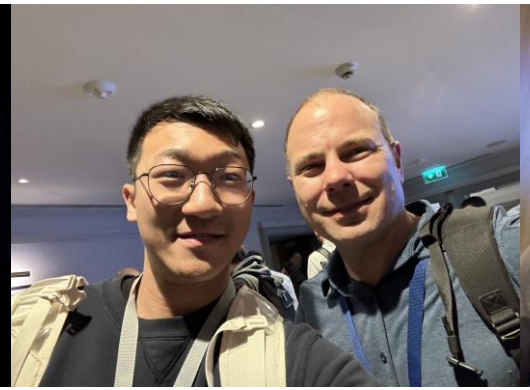
Buddy Compiler @ EuroLLVM 2023

Buddy Compiler

“软硬件协同设计终究是“人”的协同，让软件开发者和硬件开发者互相理解是协同设计的关键…”

“中间表示不是目的，编译优化才是目的。我们要从优化出发，反过来思考需要怎样的中间表示…”





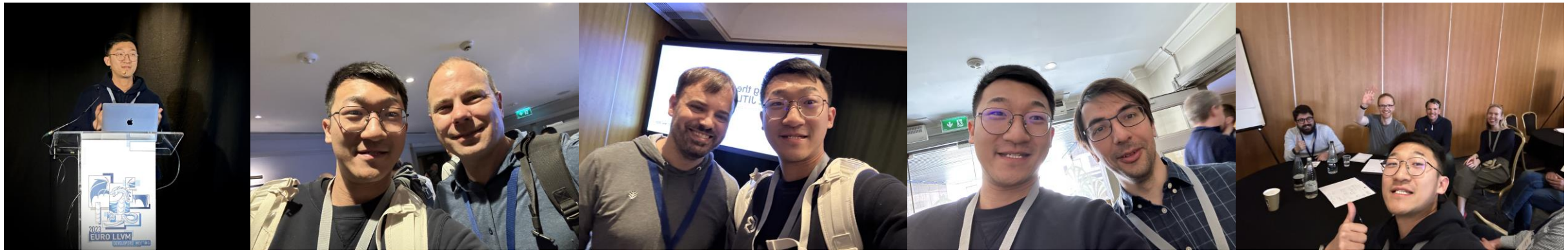

Buddy Compiler @ EuroLLVM 2023

Buddy Compiler

“软硬件协同设计终究是“人”的协同，让软件开发者和硬件开发者互相理解是协同设计的关键...”

“中间表示不是目的，编译优化才是目的。我们要从优化出发，反过来思考需要怎样的中间表示...”

“我们一定可以找到统一各架构向量扩展的方式，罗马不是一天建成的...”



Buddy Compiler @ EuroLLVM 2023

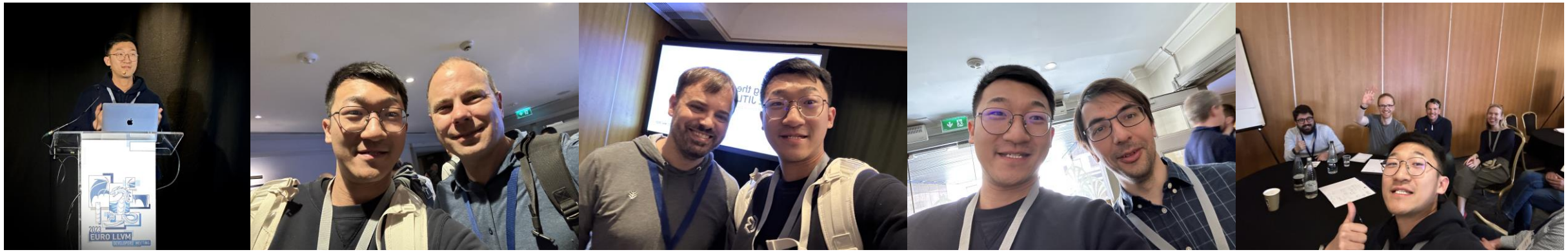
Buddy Compiler

“软硬件协同设计终究是“人”的协同，让软件开发者和硬件开发者互相理解是协同设计的关键…”

“中间表示不是目的，编译优化才是目的。我们要从优化出发，反过来思考需要怎样的中间表示…”

“我们一定可以找到统一各架构向量扩展的方式，罗马不是一天建成的…”

“开源活动的目的就是希望吸引开发者来持续贡献，很高兴看到你在 MLIR 生态中的工作和研究…”



Buddy Compiler @ LFX Mentorship

Buddy Compiler

THE LINUX FOUNDATION

RISC-V | Vector Programming | Instruction Set Architecture

RISC-V Mentorship: MLIR Convolution Vectorization

Mentees

Prathamesh Tagore

- DIP Dielect Corr2D Operation
- Vectorization Pass
- Performance Evaluation

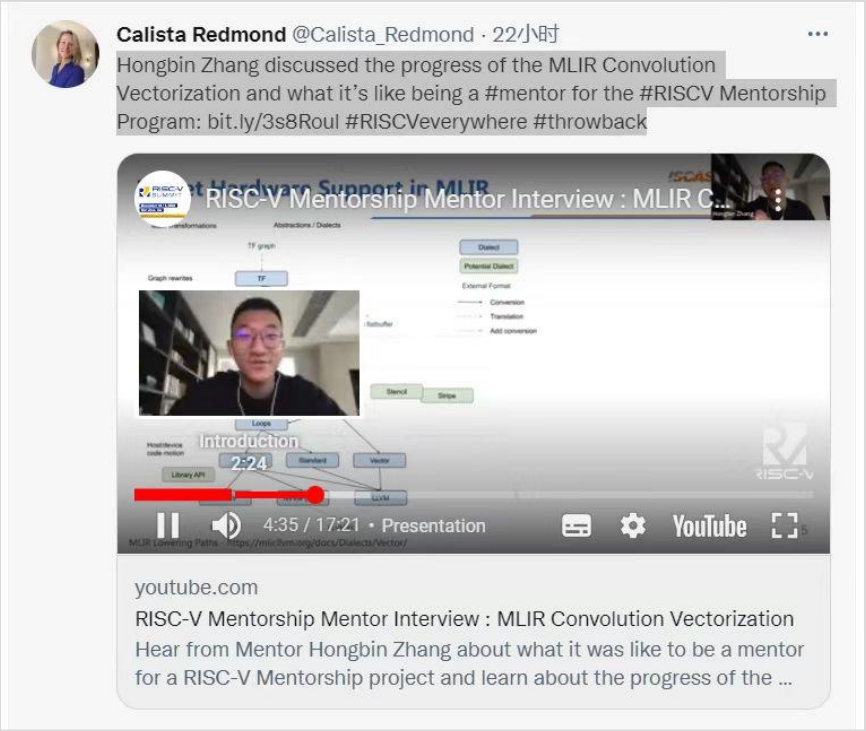
Joe Wu

- Convolution Operation Optimization

Ahmat Hamdan

- MemRef Descriptor Implementation and Test
- Pooling Operation Optimization

Mentors



https://www.youtube.com/watch?v=1pe3G3UQRkQ&t=142s

Buddy Compiler @ OSPP Summer

Buddy Compiler

Buddy Compiler 在开源之夏 2022 活动中

提供 5 个项目席位均顺利结题

- 面向 MLIR 的编译器前端设计与实现
- Adding Morphological Transformations in DIP Dialect.
- libcxx-simd 库的多平台测试框架搭建
- 数字语音处理领域多层编译抽象的设计与实现
- 基于MLIR的通用矩阵乘法和卷积向量优化

Buddy Compiler 在开源之夏 2023 活动中

提供 11 个项目，欢迎申报！

- 面向异构计算平台的任务自动调度模块设计与实现
- 面向 Systolic Array 加速器的软硬件协同设计
- 图像语音处理计算负载的编译优化
- 基于RISC-V向量拓展的深度学习模型推理优化
- 基于C++标准库experimental/simd的 OpenCV后端移植与优化
- PyTorch 2.0 对接：Buddy Compiler 作为 TorchDynamo 的编译器后端
- 图像语音多模态深度学习的推理性能优化
- MLIR 向量化 Benchmark 的设计与实现
- 深度学习框架/编译器的 Benchmark 的设计与实现
- Gemmini Dialect 指令和激活函数的补充与优化
- 图像编码/解码在 Buddy Compiler 上的移植与适配

https://summer-ospp.ac.cn/org/orgdetail/8d995d4c-b188-4690-9a53-c022dc7c19e3?lang=zh

Buddy Compiler

谢 谢

hongbin2019@iscas.ac.cn