OCTOBER 3
P4
WORKSHOP 2024

P4HIR: Towards Bridging P4C with MLIR

Bili Dong (Google)

Motivation

- The general problem of compiler construction for domain-specific accelerators

- The emergence of machine learning compilers
  - ML frameworks: TensorFlow / PyTorch / Triton / etc.
  - ML accelerators: GPU / TPU / xPU

- MLIR as a common technology used in building machine learning compilers
  - MLIR doesn’t seem to be machine learning specific
  - Can we leverage MLIR in building P4 compilers (P4C)?

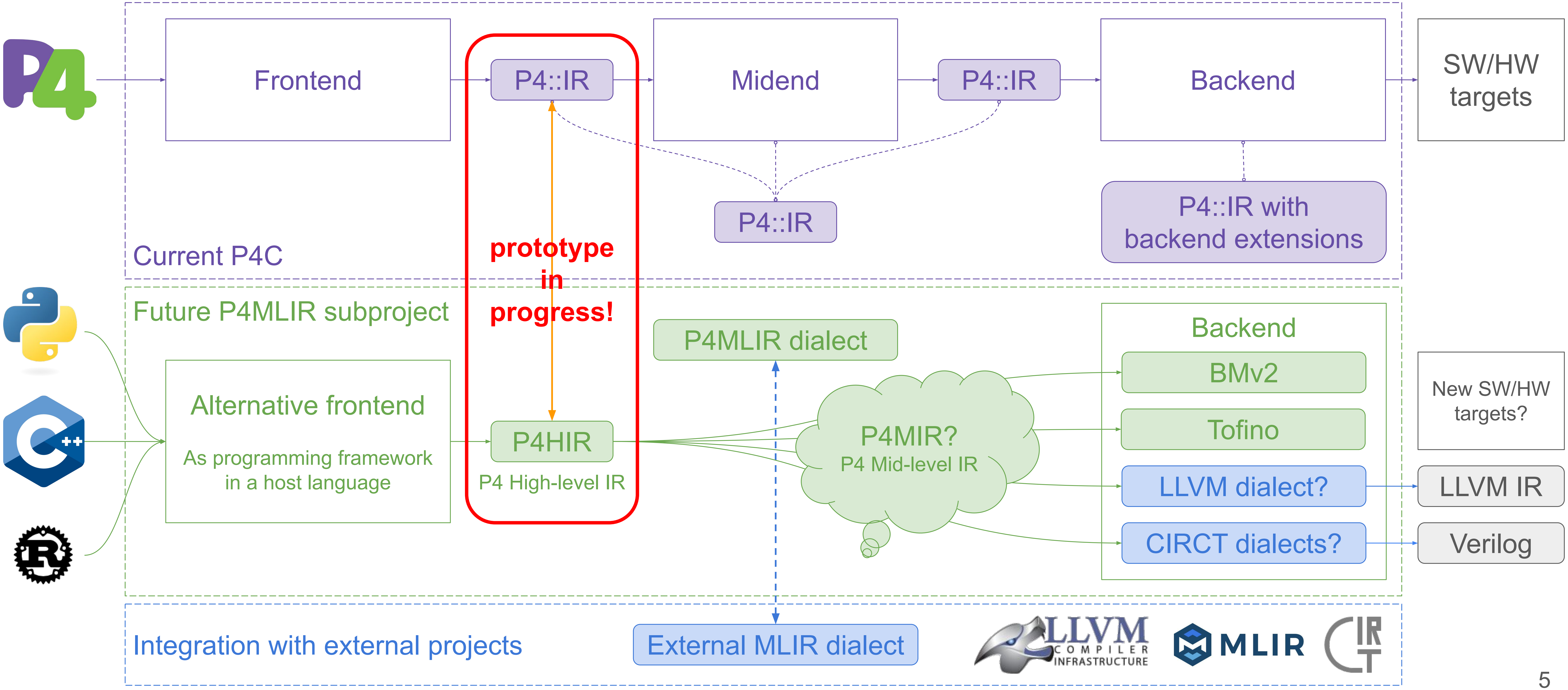
What is MLIR?

- MLIR = Multi-Level Intermediate Representation (https://mlir.llvm.org/)
- A general framework for building compilers
  - Extensible through user-defined IR libraries (type system, operation, etc.) called **dialects**
  - Provides a toolbox of common compiler infrastructure (diagnostics, testing, etc.)
  - Provides a core library of reusable dialects and transformations
  - Allows different levels of abstraction to freely co-exist (mixture of dialects, progressive lowering)
- Widely used in building machine learning compilers
  - Triton (blog post) from OpenAI
  - PyTorch (Torch-MLIR) from Meta
  - TensorFlow & OpenXLA from Google
- Also used in other domains
  - CIRCT for hardware design (EDA tools)

Why consider using MLIR in P4C?

- Reuse MLIR common compiler infrastructure
- Enable extending P4C in novel ways through the MLIR integration
- Encourage communication between P4 and MLIR open source communities

How to bridge P4C with MLIR? A proposal



Summary

- Overall goal is to get P4 community interested in using MLIR in P4C
- Full P4C MLIR integration will be a huge project far beyond any personal effort
- Defining a single dialect is relatively straightforward, and might be a useful starting point
- We plan to build a P4HIR prototype as an experimental project:

https://github.com/p4lang/project-ideas/issues/20

Further discussions are welcome!

Thank You