Common Intermediate Representation(Common IR)

Venkat Pullela, OpenNets
Pranav Dixit, Amazon
Karthik Pullela, UCLA
Sahil Gupta, RIT

What is Intermediate Representation (IR)

Intermediate representation(Data Structure) used by all compiler stages in P4C

IR consists of foundational primitives used to build Data Pipelines

P4C front end translates P4 program in to Abstract Syntax Tree(AST)

Language constructs in P4 programs converted to IR nodes

Same IR is used by all frontend and midend stages in P4C

Frontend and midend passes reduce IR for consumption of backend

Extending P4C to support NPL

Created NPL Lexer and Parser for Network Programming Language(NPL)

Used Open Definition from NPLang.org

NPL disaggregates bulky primitives for efficiency

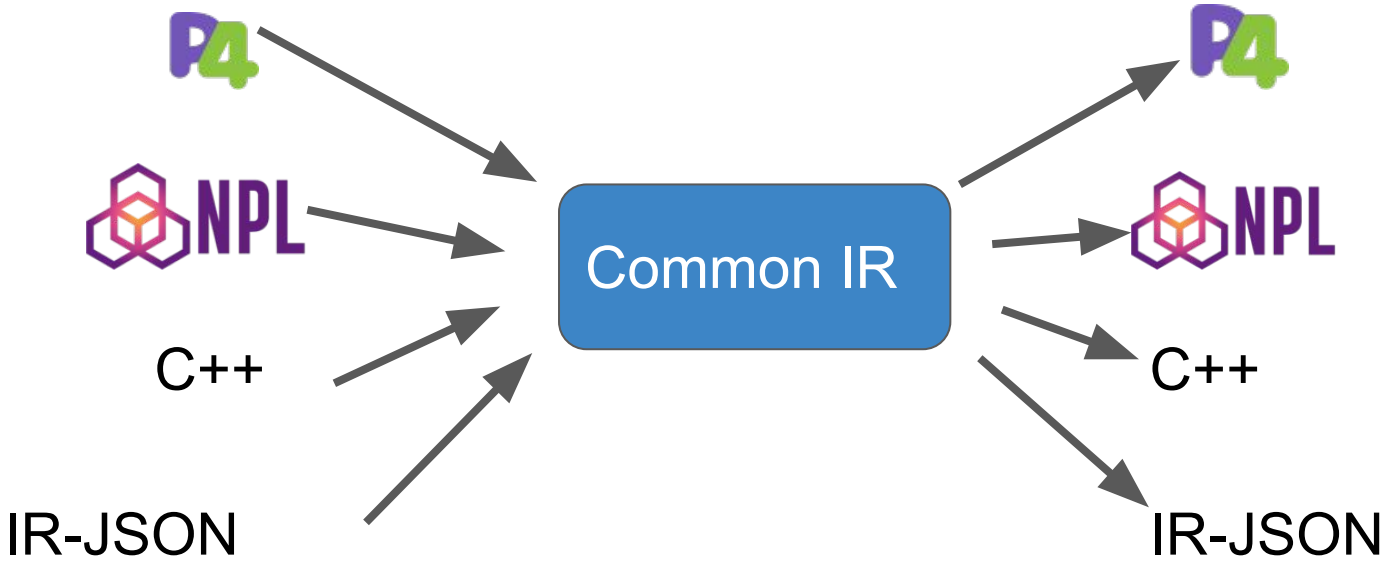
Leverage as much of existing IR nodes as possible

- Struct, Assignment, If

Added new IR Nodes to support NPL specific constructs

- Begin/Break/Continue Parser, Overlays

Abstraction Level



IR is at the right abstraction level to interoperate across architectures

Takeaways

IR Nodes abstraction at the level of common building blocks of HW Dataplanes

IR can be classified into three levels as it gets reduced through compiler stages

Language, Processing, Hardware Primitives

Language constructs are one level of abstraction higher

P4 Match-Action Table = Lookup + Action + Profile + Counter + Meter ...

Takeaways 2

IR Node semantics need to be same across languages

Programming model of languages may differ

Possible to translate between languages using CommonIR

Possible to integrate and interoperate between languages

IR can be used for building libraries for Modularity and Composability

Thank You

venkat.pullela@opennets.com
www.OpenNets.com