

# A Sampling Profiler for JITing in R

Masters Presentation I

Andreas Wälchli

Supervisor: Olivier Flückiger

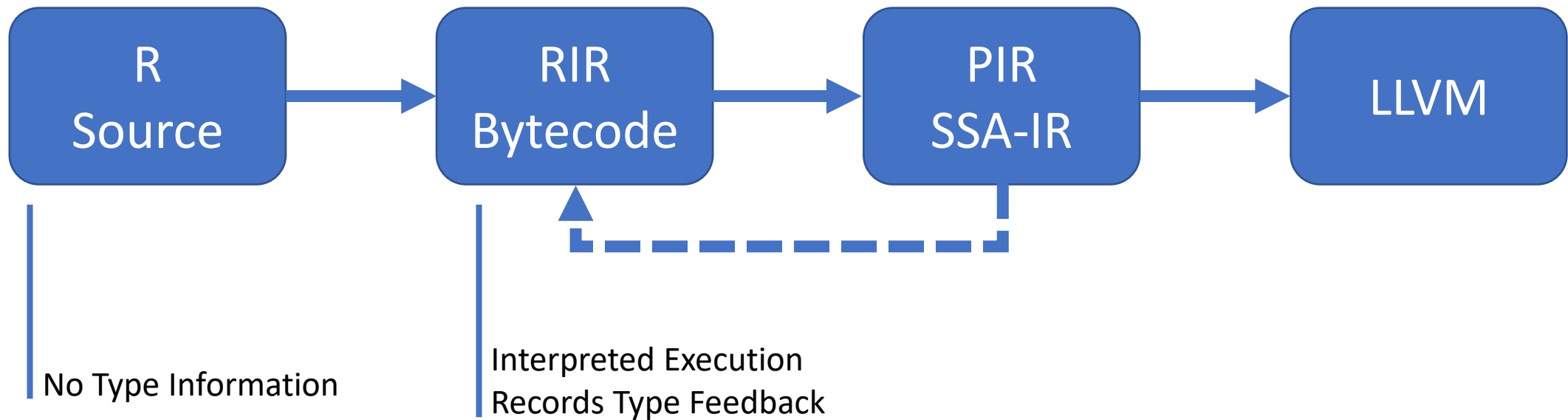
# R and the RIR project

- R
  - Language & Environment
  - Main Use: Statistical Computing & Data Mining
  - Single-Threaded
  - Interpreted Language
  - Dynamic Typing
- RIR Project
  - Implementation of a JIT-Compiler for R

# Compilation with Dynamic Typing

- Problem: no good type information from static analysis
  - need to run code to get type information
  - types may change between executions
- Solution: Warm-Up Phase
  - run code in interpreter
  - record type information during execution
  - compile with collected type information after several runs

# RIR Compilation Pipeline



# Problem: Polluted Type Feedback

- compiler uses type feedback for optimisations
- recorded type feedback is persistent
- pollution by stale feedback prevents better compilation

# DEMO

```
a <- 1L
```

```
f <- function() a+a+a+1L
```

```
f()
```

```
f()
```

```
system.time(for(i in 1:1000000) f())
```

```
a <- 1
```

```
f <- function() a+a+a+1L
```

```
f()
```

```
a <- 1L
```

```
f()
```

```
f()
```

```
system.time(for(i in 1:1000000) f())
```

# Approach

- Take Random Samples (Signal Handler)
- Record Type Feedback
- New Information Better ? → Recompile
  
- Better = Narrower

# DEMO

```
a <- 1L  
f <- function() a+a+a+1L
```

```
f()  
f()
```

```
system.time(for(i in 1:1000000) f())
```

```
rir.disassemble(f)
```

```
a <- 1  
f <- function() a+a+a+1L
```

```
f()  
a <- 1L
```

```
f()  
f()
```

```
system.time(for(i in 1:1000000) f())
```

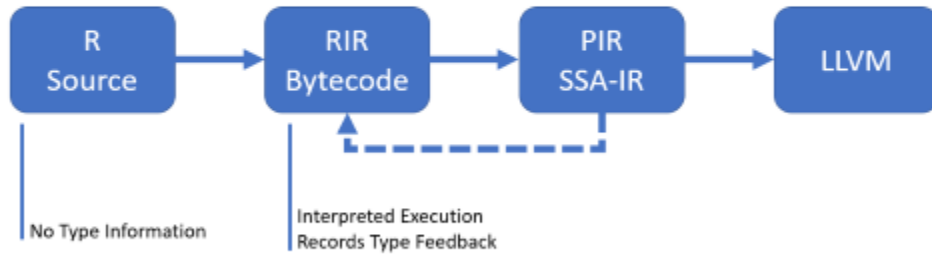
```
rir.disassemble(f)
```



# Research Questions

- Will performance improvements outweigh profiling overhead?
- How can the profiling overhead be minimised?
- How can sampling be implemented safely?
  - syscall problem, PMU

## RIR Compilation Pipeline



## Problem: Polluted Type Feedback

- compiler uses type feedback for optimisations
- recorded type feedback is persistent
- pollution by stale feedback prevents better compilation

## Approach

- Take Random Samples (Signal Handler)
- Record Type Feedback
- New Information Better ? → Recompile
  
- Better = Narrower

## Research Questions

- Will performance improvements outweigh profiling overhead?
- How can the profiling overhead be minimised?
  
- How can sampling be implemented safely?
  - syscall problem, PMU