MAMBO: Dynamic Binary Modification on RISC-V

1st Open-Source RISC-V Software Workshop
Munich, 28 June 2024

John Alistair Kressel
Igor Wodiany
Mikel Luján

University of Manchester
<firstname>.<lastname>@manchester.ac.uk

MAMBO
=
First optimised DBM framework for RISC-V

What is DBM?

(Also, DBI and DBT)

Valgrind

QEMU

(And MAMBO)

What is DBM?

Dynamic – Working at runtime

Binary – Natively compiled user-space code

Modification – Alteration of applications

+

Instrumentation – Inserting additional functionality

Translation – Translating one instruction set into another

MANCHESTER
1824
The University of Manchester

DBM Use Cases

DBM Use Cases

Program analysis

Callgrind (Valgrind)

Memory error detection

Memcheck (Valgrind), Dr. Memory (DynamoRIO), Memcheck (MAMBO)

Dynamic binary translation

QEMU, Apple Rosetta, TANGO

Why MAMBO?

MANCHESTER 1824
The University of Manchester

Why MAMBO?

Optimized for RISC-V 64-bit, ARM 32-bit & ARM 64-bit
- Low overhead
- **Only available DBM optimized for RISC-V**

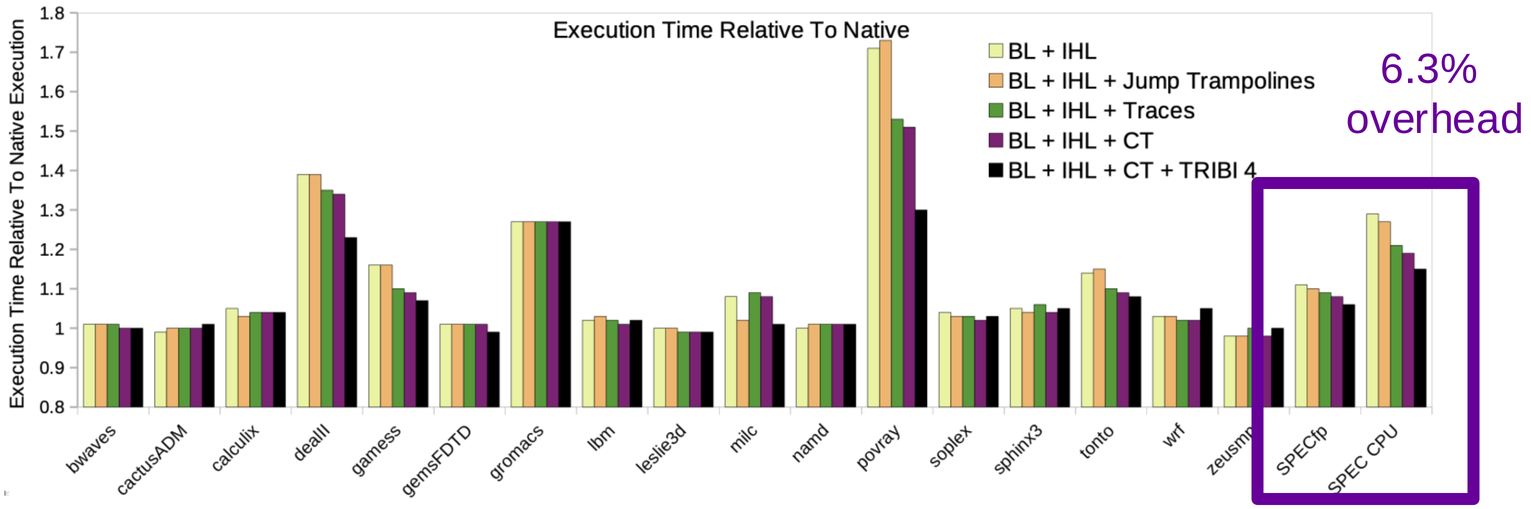
Low complexity
- Relatively small codebase (~20k LoC)

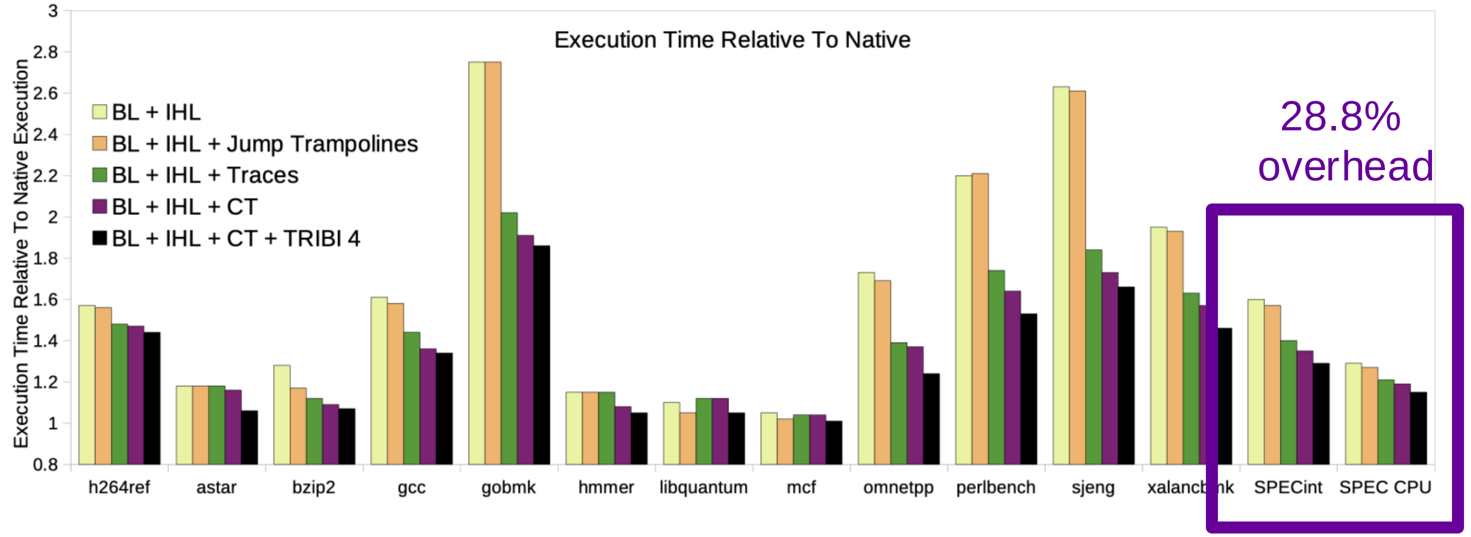
Simple plugin API
- Architecture agnostic helper functions for portable plugins

Not a toy!

Why MAMBO on RISC-V?

Why MAMBO on RISC-V?



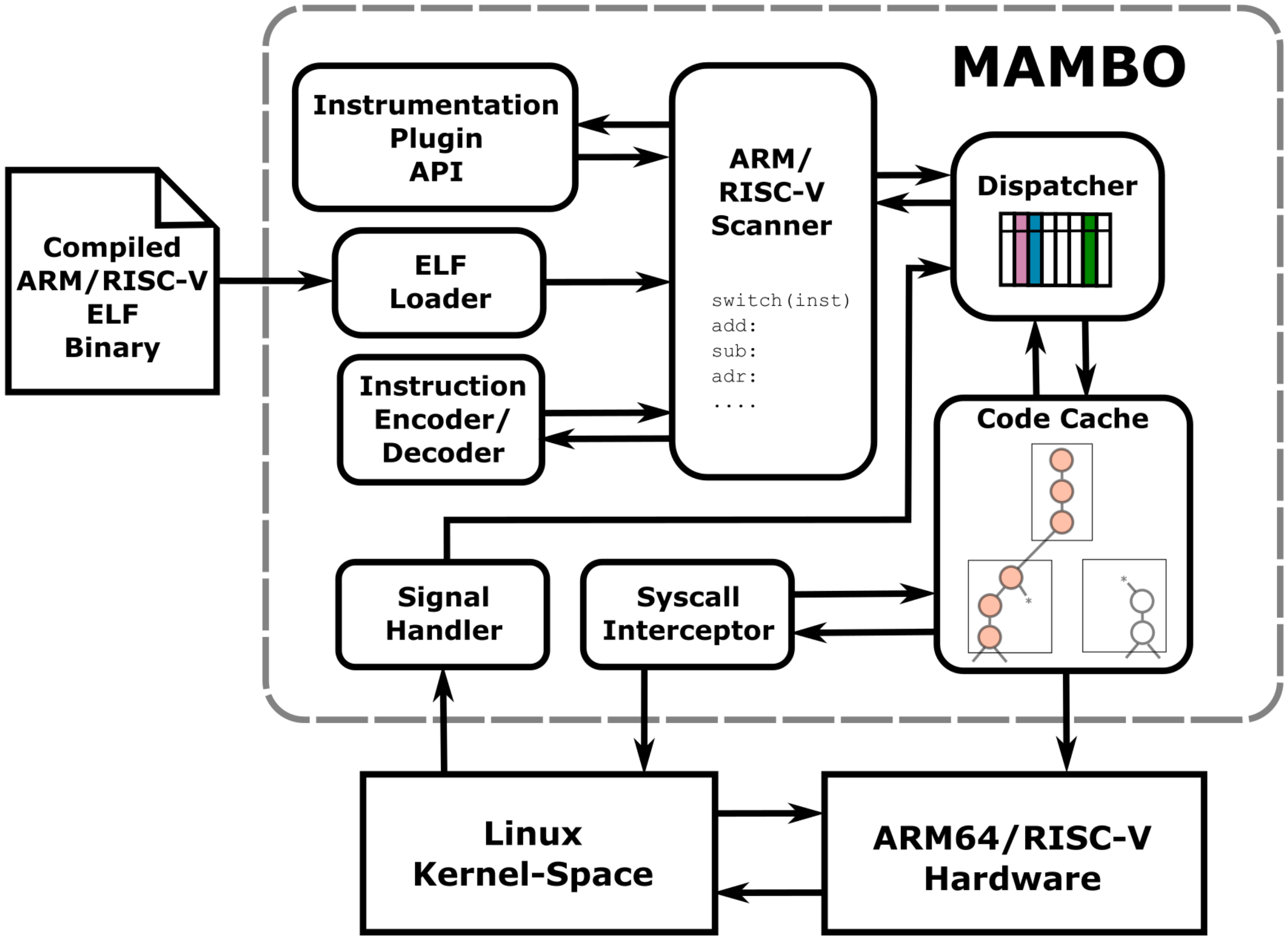


Slowdown relative to native execution for SPEC CPU2006 – RISC-V 64GC.

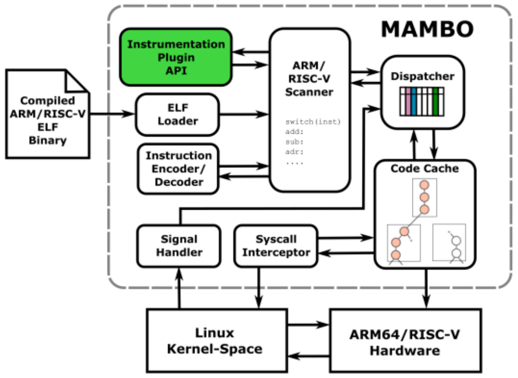
Kressel et al. Evaluating the Impact of Optimizations for Dynamic Binary Modification on 64-bit RISC-V.

MAMBO Architecture

MAMBO Architecture



Introduction to MAMBO plugin API



MANCHESTER
1824
The University of Manchester

Example Use Cases

Example Use Cases

Code analysis

Building CFG

Code generation

Insertion of new functionality

Code modification

Reimplementation of library functions

Code instrumentation

Performance counters and metrics

Runtime event handling

Tracking thread creation/destruction

MANCHESTER
1824
The University of Manchester

Event Driven Programming Model

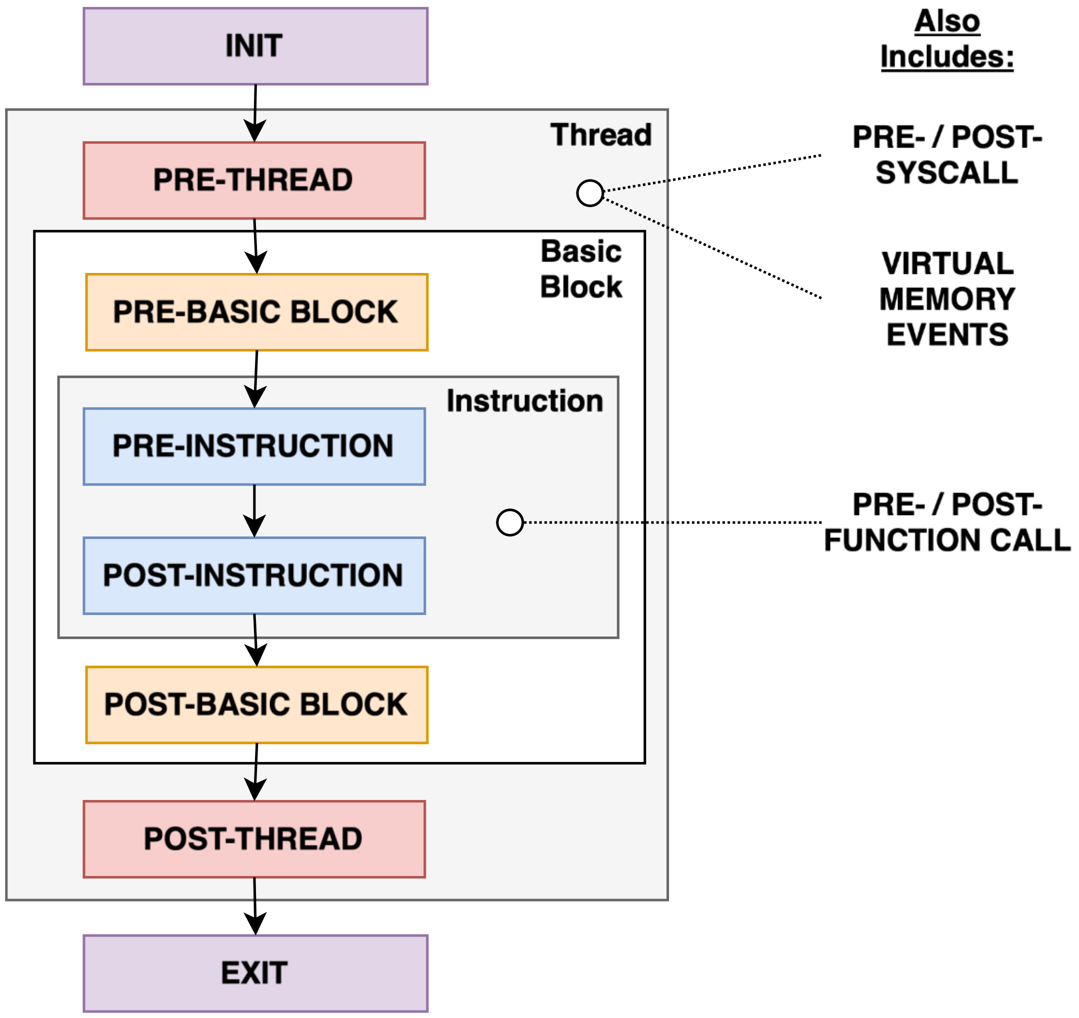
Event Driven Programming Model

User defined functions are registered as callbacks

MAMBO executes callbacks when the event is encountered

Fine-grained (e.g., per-instruction), medium-grained (e.g., per-basic block) and coarse-grained (e.g., per-thread) instrumentation

Event Driven Programming Model



MAMBO API

MAMBO API

Callback registering functions

Code analysis

Instrumentation functions

Various helper functions

Both architecture dependent and independent functions

MANCHESTER
1824
The University of Manchester

Lessons Learned from RISC-V Port

MANCHESTER
1824
The University of Manchester

Lessons Learned From RISC-V Port

Limited range of direct branches

Restrictions placed on atomic Load Reserved (LR) and Store Conditional (SC)

Thread pointer as a general-purpose register (x4)

MAMBO Roadmap

MANCHESTER
1824
The University of Manchester

MAMBO Roadmap

Foster an open-source community
- Collaborations and contributions welcome

Improve Documentation

More tools
- Data race detector

Keep up with RISC-V (and ARM)

Current research projects
- Cybersecurity
- Binary lifting

CODE OPEN SOURCE ON GITHUB

BEEHIVE-LAB / MAMBO

(APACHE 2.0 LICENSE)

MANCHESTER 1824
The University of Manchester

Thanks!

UK Research and Innovation

ROYAL ACADEMY OF ENGINEERING

THE ROYAL SOCIETY

EPSRC
Engineering and Physical Sciences Research Council

MoatE (10017512) and Soteria (75243)

Hardware Donations

SiFive

Deep Computing