VM Analysis and Hardware Trace Reconstruction

Suchakrapani Datt Sharma

Dec 12, 2016

École Polytechnique de Montréal

Laboratoire **DORSAL**

Agenda

Introduction

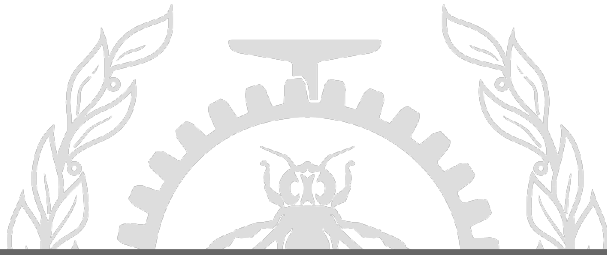
- Research Updates

New Investigations

- Intel PT
  - Advanced Analysis of VMs
- Trace Reconstruction Issues
  - Failed & Incorrect Decoding
  - Kernel Assisted Reconstruction

Upcoming

- Kernel Patches

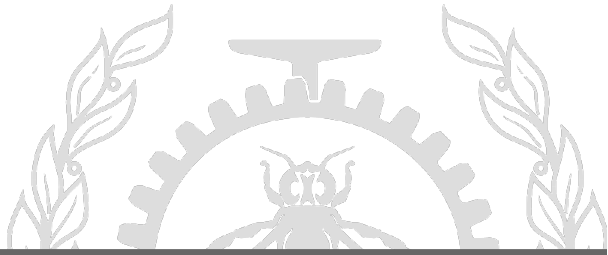


Introduction

Research Focus : Hardware tracing on Intel and ARM for low overhead and high accuracy tracing and profiling

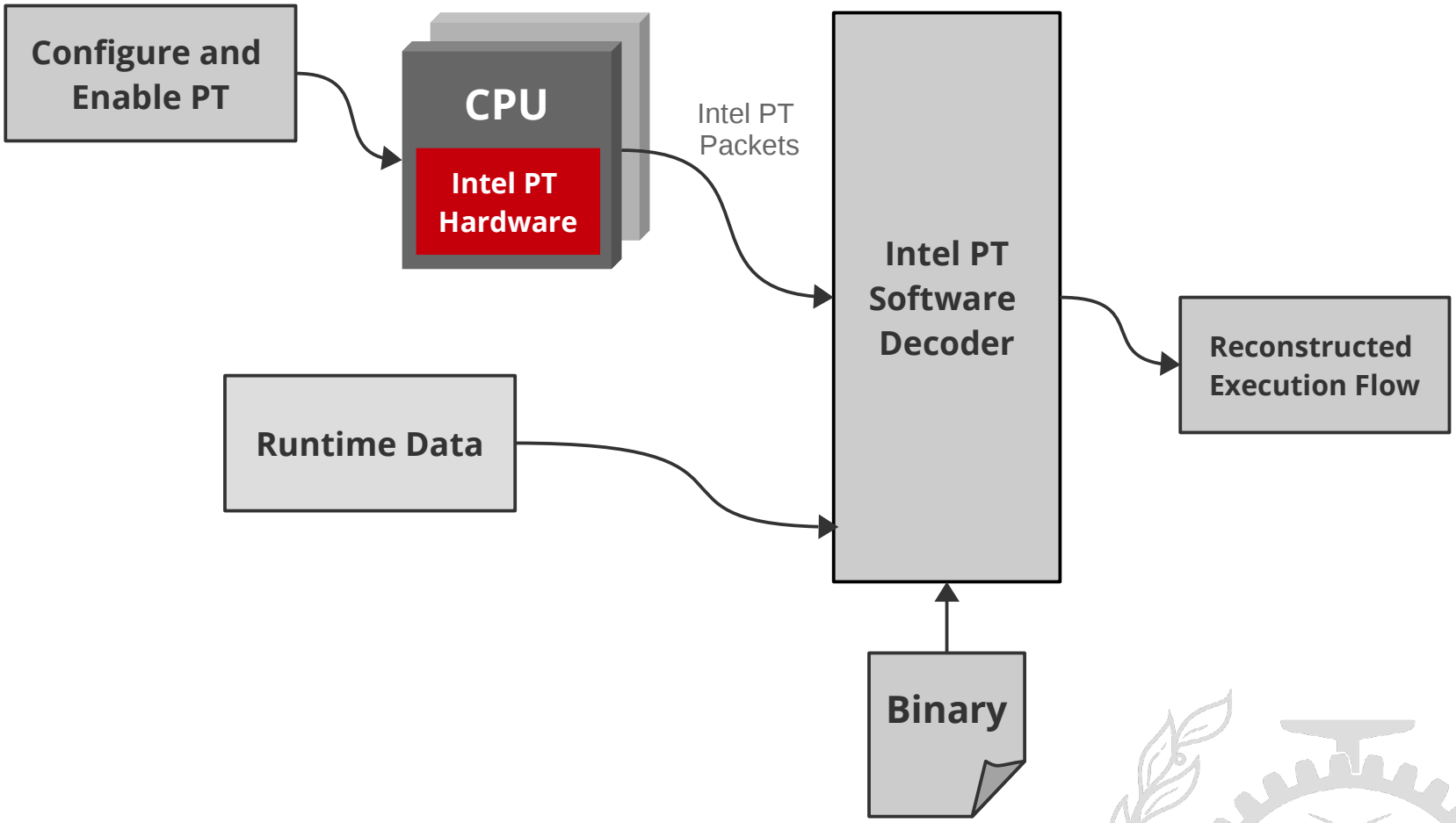
Research Updates

- Advanced VM analysis with hardware tracing
- FlowJIT : A robust hardware trace reconstruction technique



Introduction

Hardware Tracing with PT



Based on, Andi Kleen's Presentation (TracingSummit 2015)

Intel PT

Hardware Trace Packets (Perf)

```
. ... Intel Processor Trace data: size 8544 bytes
.  00000000:  02 82 02 82 02 82 02 82 02 82 02 82 02 82 PSB
.  00000010:  00 00 00 00 00 00                         PAD
.  00000016:  19 ba 39 4d 7b 89 5e 04                   TSC 0x45e897b4d39ba
.  0000001e:  00 00 00 00 00 00 00 00                   PAD
.  00000026:  02 73 57 64 00 1c 00 00                   TMA CTC 0x6457 FC 0x1c
.  0000002e:  00 00                                     PAD
.  00000030:  02 03 27 00                               CBR 0x27
.  00000034:  02 23                                     PSBEND
.  00000036:  59 8b                                     MTC 0x8b
.  00000038:  59 8c                                     MTC 0x8c
.
.  00000304:  f8                                        TNT TTTTNN (6)
.  00000305:  06 00 00                                  TNT T (1)
.  00000308:  4d e0 3c 6d 9c                            TIP 0x9c6d3ce0
.  0000030d:  1c 00 00                                  TNT TTN (3)
.  00000310:  2d f0 3c                                  TIP 0x3cf0
.  00000313:  06                                        TNT T (1)
.  00000314:  59 2e                                     MTC 0x2e
.  00000316:  94                                        TNT NNTNTN (6)
.  00000317:  a8                                        TNT NTNTNN (6)
.  00000318:  a6                                        TNT NTNNTT (6)
```

Intel PT

Timing

```
. ... Intel Processor Trace data: size 8544 bytes
.  00000000:  02 82 02 82 02 82 02 82 02 82 02 82 02 82 02 82 PSB
.  00000010:  00 00 00 00 00 00                               PAD
.  00000016:  19 ba 39 4d 7b 89 5e 04                         TSC 0x45e897b4d39ba
.  0000001e:  00 00 00 00 00 00 00 00                         PAD
.  00000026:  02 73 57 64 00 1c 00 00                         TMA CTC 0x6457 FC 0x1c
.  0000002e:  00 00                                           PAD
.  00000030:  02 03 27 00                                     CBR 0x27
.  00000034:  02 23                                           PSBEND
.  00000036:  59 8b                                           MTC 0x8b
.  00000038:  59 8c                                           MTC 0x8c
.
.  00000304:  f8                                              TNT TTTTNN (6)
.  00000305:  06 00 00                                        TNT T (1)
.  00000308:  4d e0 3c 6d 9c                                  TIP 0x9c6d3ce0
.  0000030d:  1c 00 00                                        TNT TTN (3)
.  00000310:  2d f0 3c                                        TIP 0x3cf0
.  00000313:  06                                              TNT T (1)
.  00000314:  59 2e                                           MTC 0x2e
.  00000316:  94                                              TNT NNTNTN (6)
.  00000317:  a8                                              TNT NTNTNN (6)
.  00000318:  a6                                              TNT NTNNTT (6)
```

Intel PT

Conditional Branches

```
. ... Intel Processor Trace data: size 8544 bytes
.  00000000:  02 82 02 82 02 82 02 82 02 82 02 82 02 82 PSB
.  00000010:  00 00 00 00 00 00                         PAD
.  00000016:  19 ba 39 4d 7b 89 5e 04                   TSC 0x45e897b4d39ba
.  0000001e:  00 00 00 00 00 00 00 00                   PAD
.  00000026:  02 73 57 64 00 1c 00 00                   TMA CTC 0x6457 FC 0x1c
.  0000002e:  00 00                                     PAD
.  00000030:  02 03 27 00                               CBR 0x27
.  00000034:  02 23                                     PSBEND
.  00000036:  59 8b                                     MTC 0x8b
.  00000038:  59 8c                                     MTC 0x8c
.
.  00000304:  f8                                        TNT TTTTNN (6)
.  00000305:  06 00 00                                  TNT T (1)
.  00000308:  4d e0 3c 6d 9c                            TIP 0x9c6d3ce0
.  0000030d:  1c 00 00                                  TNT TTN (3)
.  00000310:  2d f0 3c                                  TIP 0x3cf0
.  00000313:  06                                        TNT T (1)
.  00000314:  59 2e                                     MTC 0x2e
.  00000316:  94                                        TNT NNTNTN (6)
.  00000317:  a8                                        TNT NTNTNN (6)
.  00000318:  a6                                        TNT NTNNTT (6)
```

Intel PT

Indirect Branches

```
. ... Intel Processor Trace data: size 8544 bytes
.  00000000:  02 82 02 82 02 82 02 82 02 82 02 82 02 82 PSB
.  00000010:  00 00 00 00 00 00                         PAD
.  00000016:  19 ba 39 4d 7b 89 5e 04                   TSC 0x45e897b4d39ba
.  0000001e:  00 00 00 00 00 00 00 00                   PAD
.  00000026:  02 73 57 64 00 1c 00 00                   TMA CTC 0x6457 FC 0x1c
.  0000002e:  00 00                                     PAD
.  00000030:  02 03 27 00                               CBR 0x27
.  00000034:  02 23                                     PSBEND
.  00000036:  59 8b                                     MTC 0x8b
.  00000038:  59 8c                                     MTC 0x8c
.
.  00000304:  f8                                        TNT TTTTNN (6)
.  00000305:  06 00 00                                  TNT T (1)
.  00000308:  4d e0 3c 6d 9c                            TIP 0x9c6d3ce0
.  0000030d:  1c 00 00                                  TNT TTN (3)
.  00000310:  2d f0 3c                                  TIP 0x3cf0
.  00000313:  06                                        TNT T (1)
.  00000314:  59 2e                                     MTC 0x2e
.  00000316:  94                                        TNT NNTNTN (6)
.  00000317:  a8                                        TNT NTNTNN (6)
.  00000318:  a6                                        TNT NTNNTT (6)
```

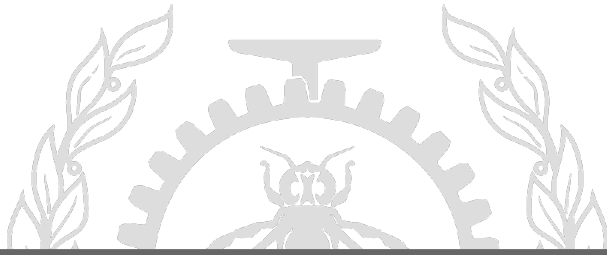
Intel PT

VM Analysis

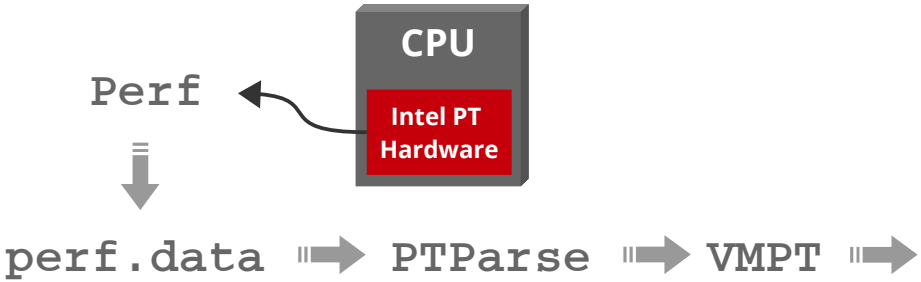
- Resource consumption and process analysis
  - **PTParse**[1] : Extract PT data from Perf
  - **VMPT**[2] : Format PT data as XML *bundles*
- *Bundle* contains PIP (CR3 value / NR bit), VMCS and TSC packets in XML
  - VMCS Base Register → Associated VM
  - CR3 → Process
  - NR → VM Entry/Exit
- Decoder + TraceCompass View

Thanks to Hani & Geneviève!

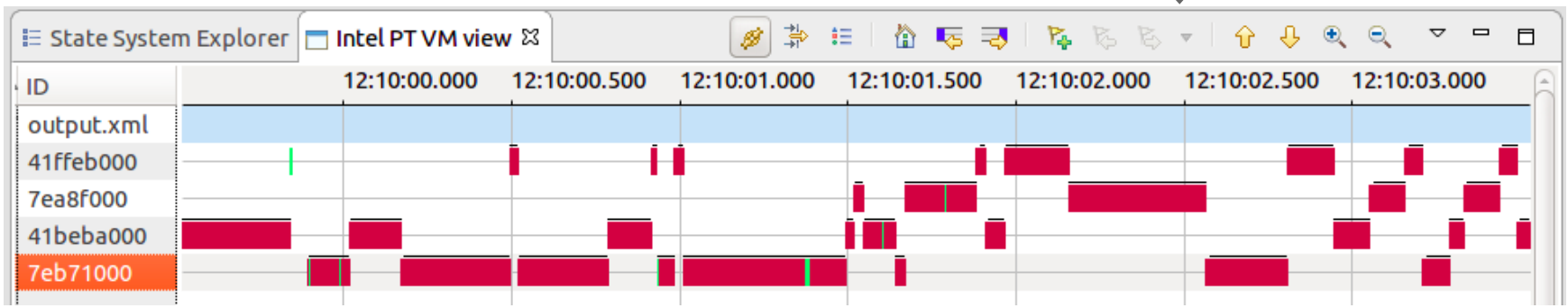
[1] https://github.com/tuxology/dorsal/tree/master/ptparse
[2] https://github.com/tuxology/vmpt

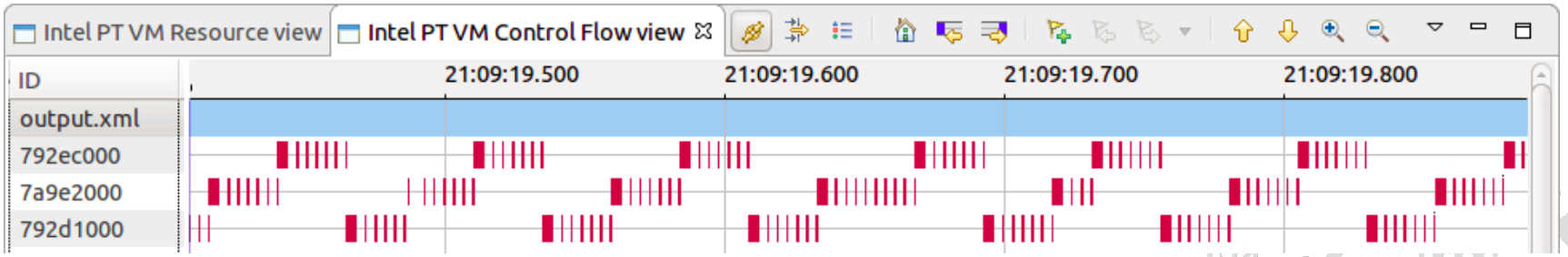


Intel PT




```
<bundle>
    <PIP> 792ec000 </PIP>
    <NR> 1 </NR>
    <VMCS> 7eb71000 </VMCS>
    <TSC> 2342353646 </TSC>
</bundle>
```

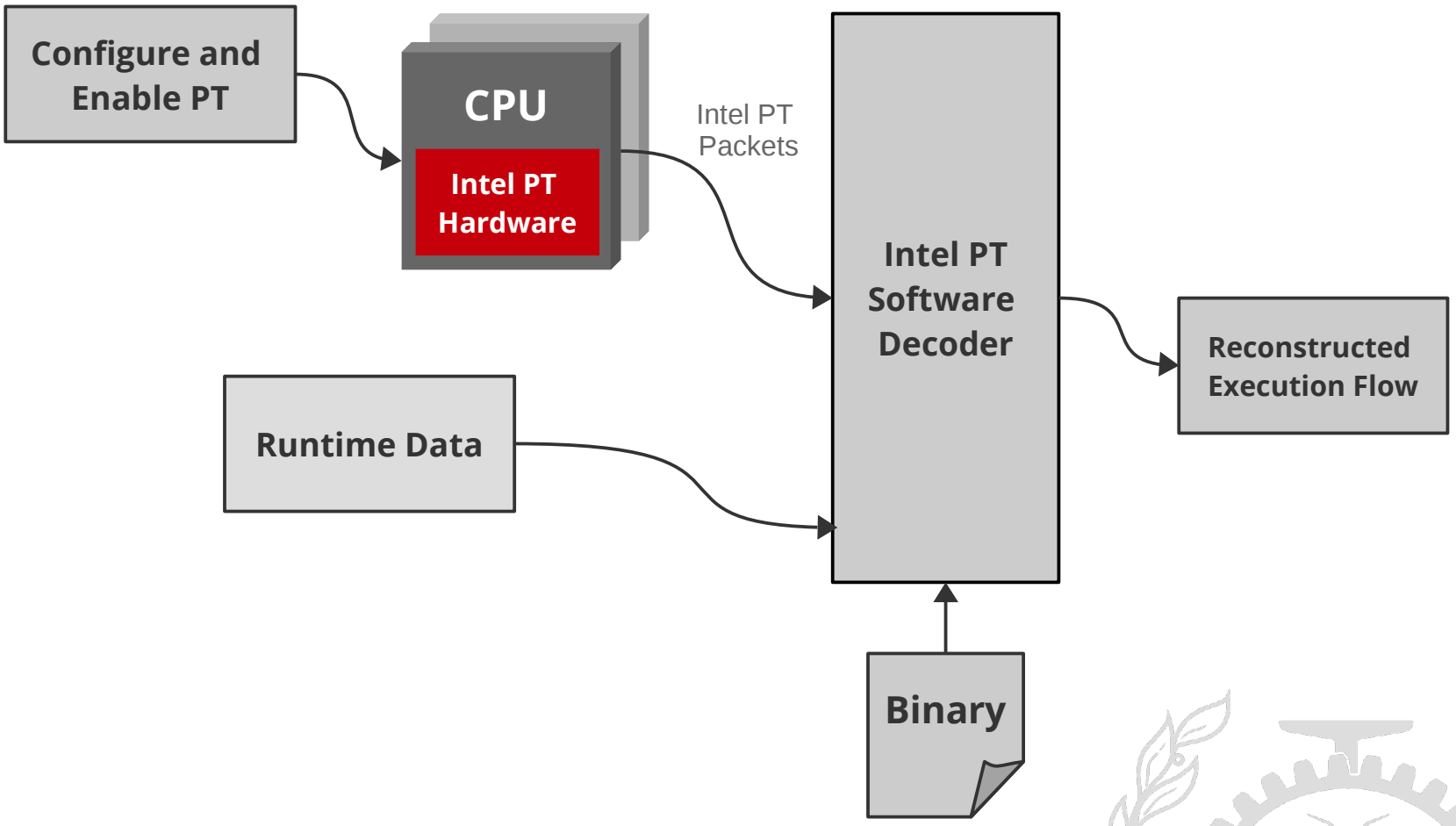




There's a glitch in the matrix though..

Intel PT

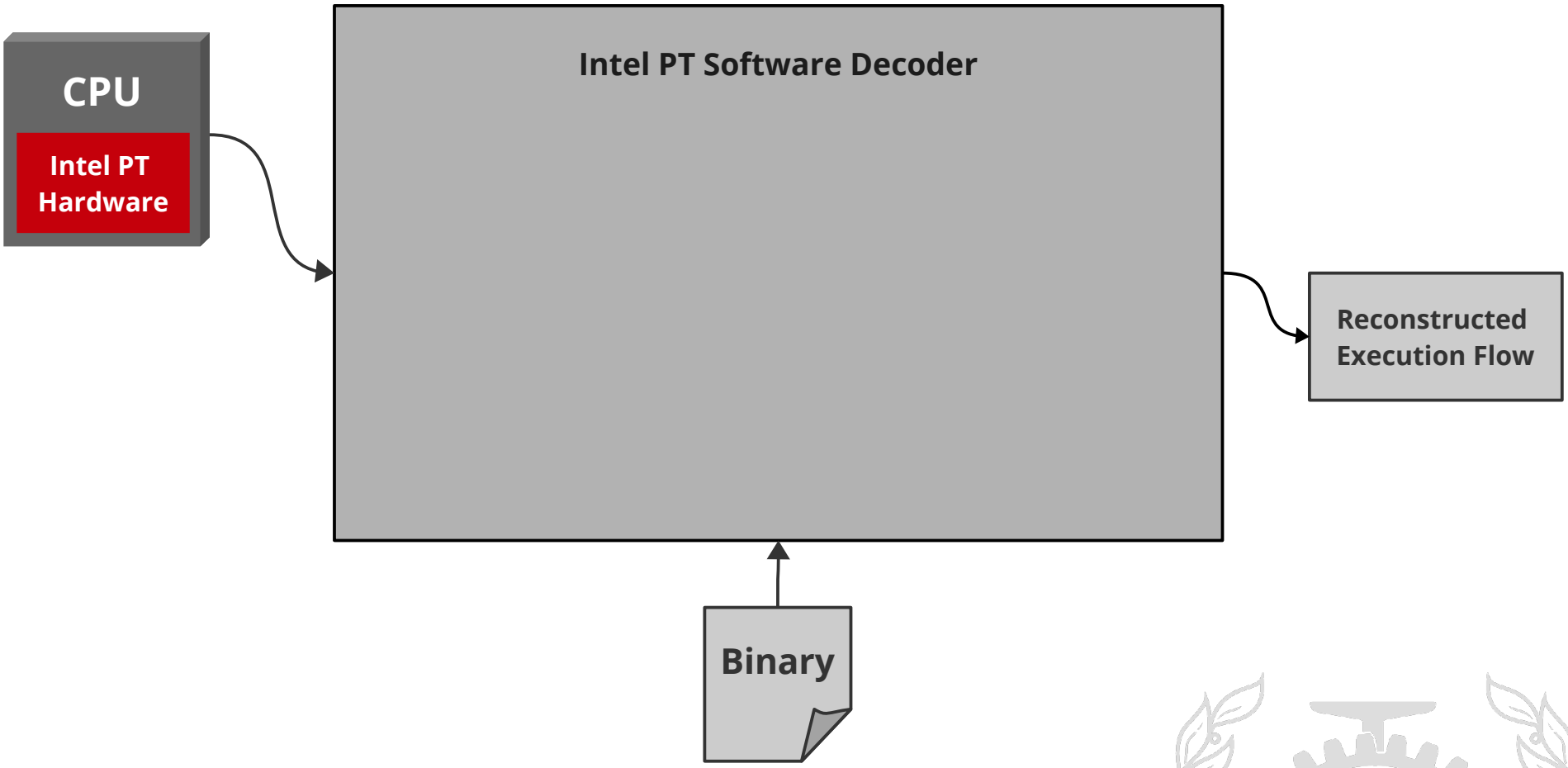
Limitations



Based on, Andi Kleen's Presentation (TracingSummit 2015)

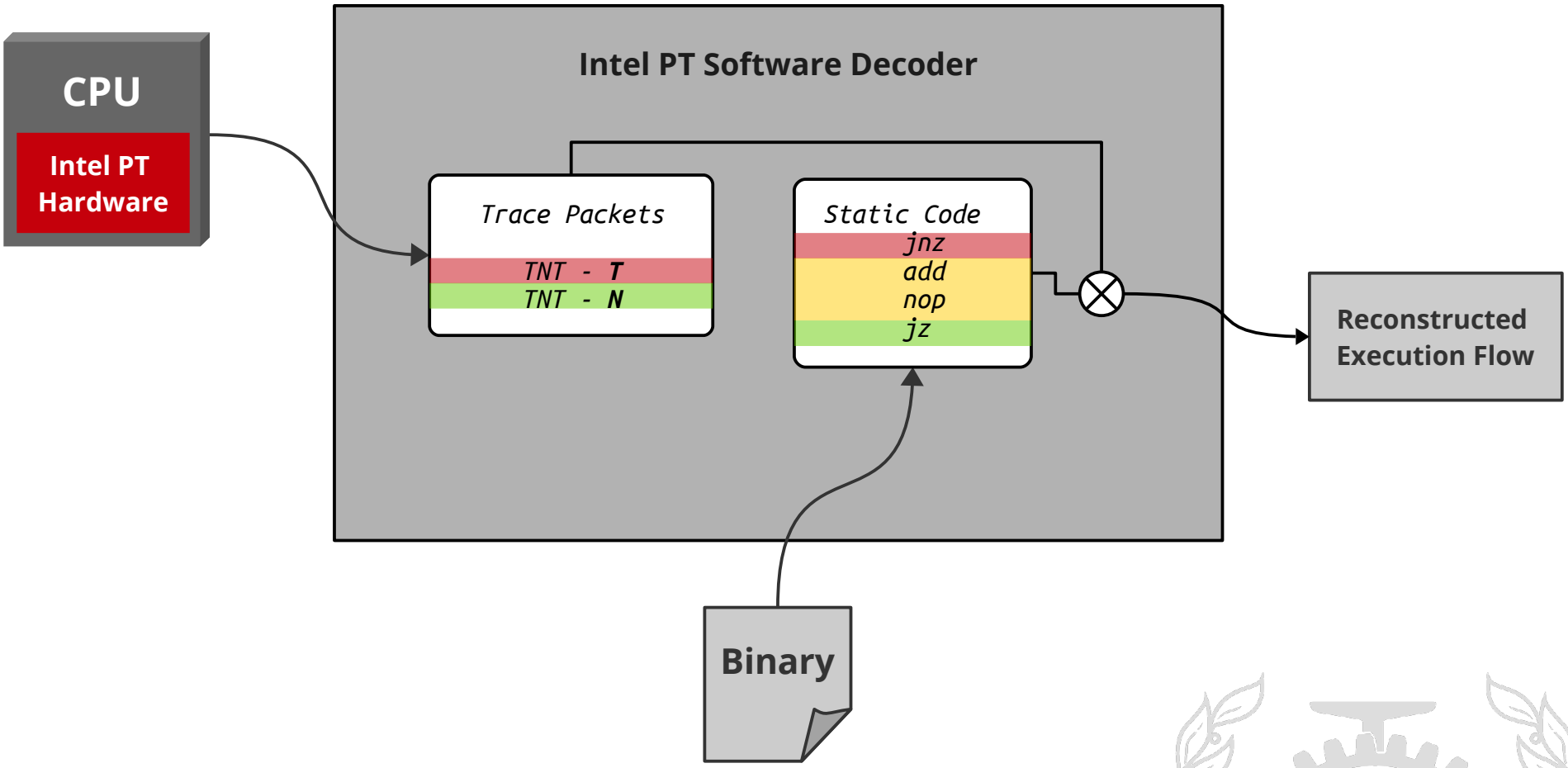
Intel PT

Limitations



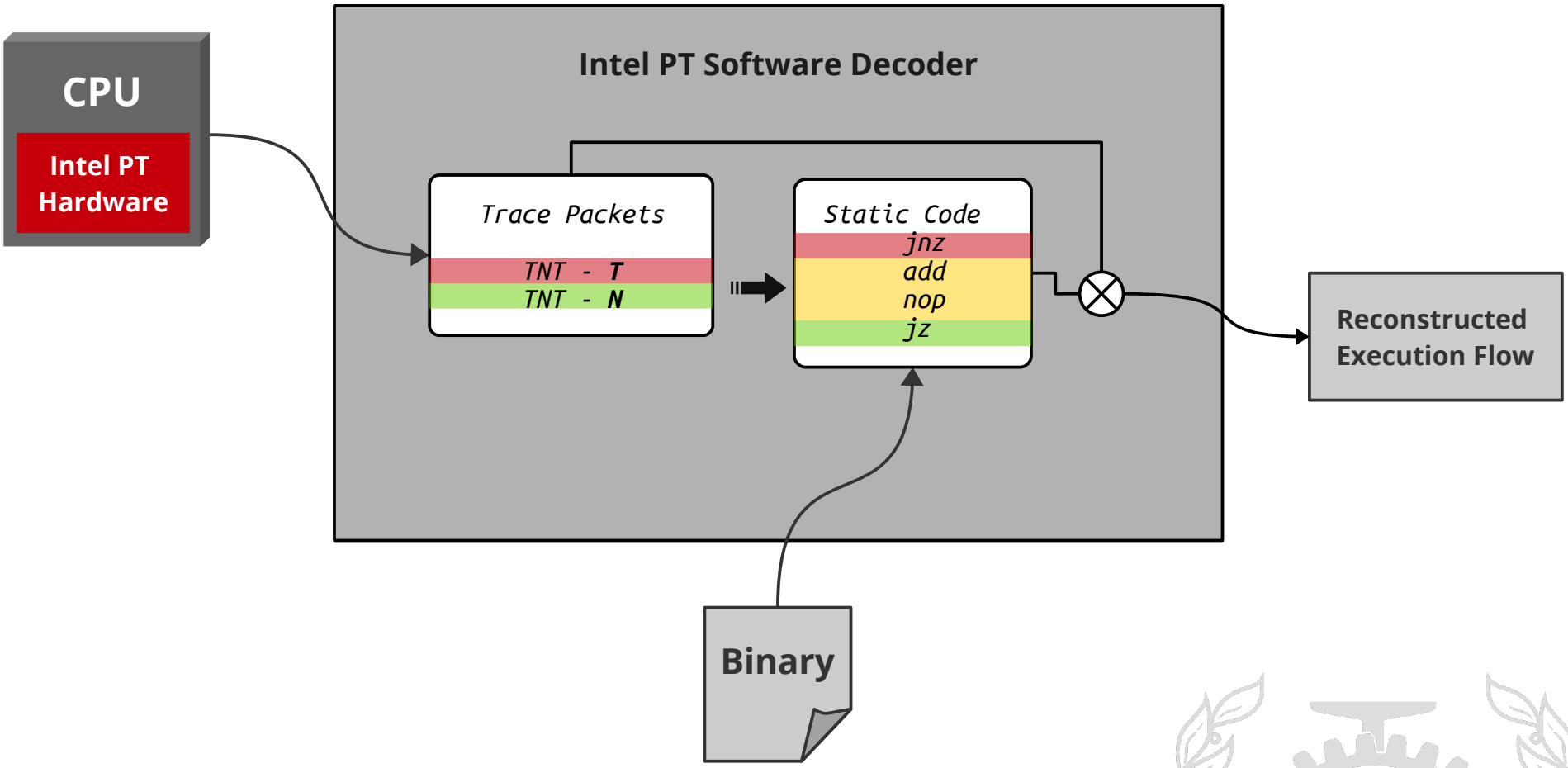
Intel PT

Limitations - JIT Code



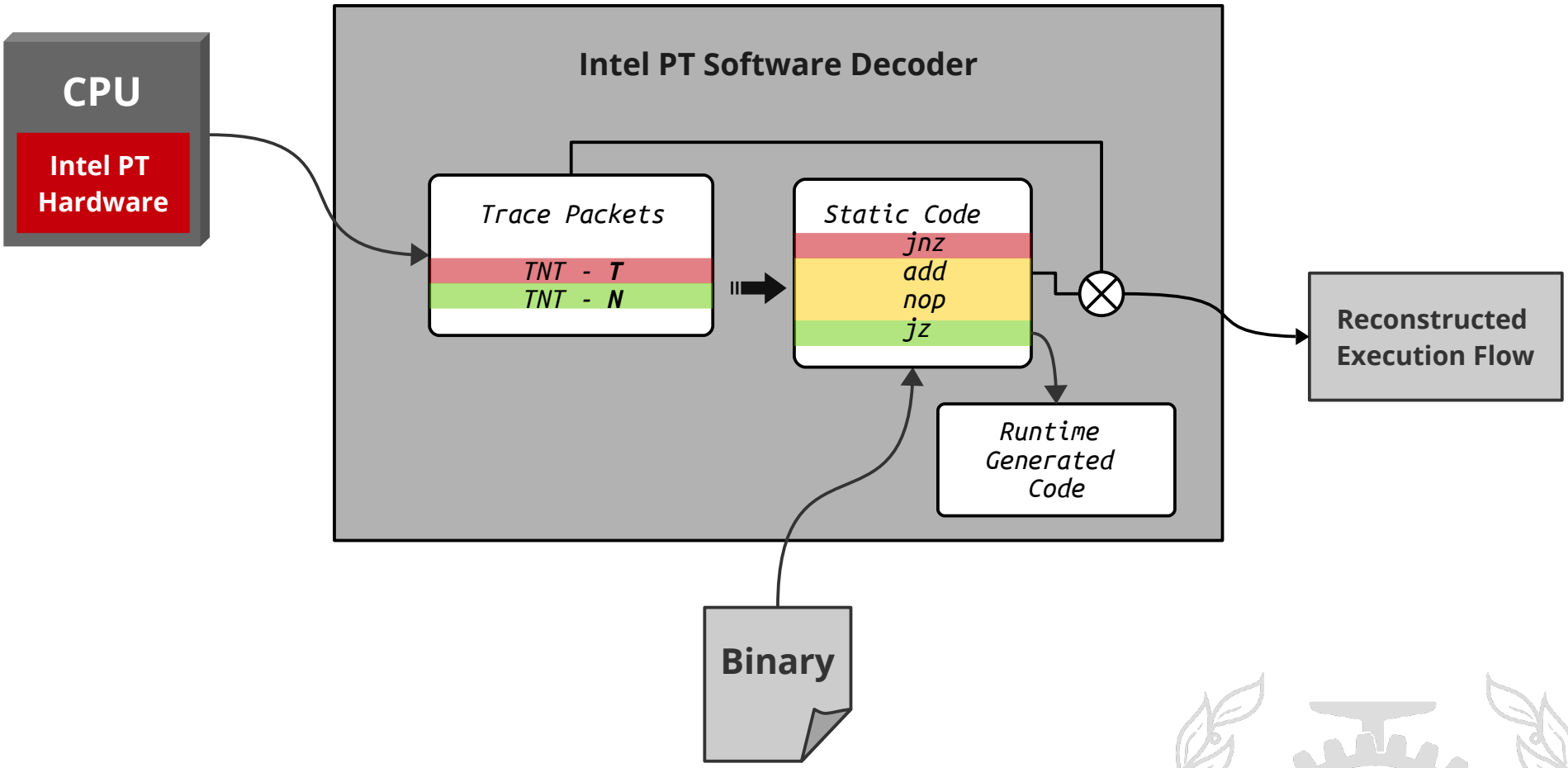
Intel PT

Limitations - JIT Code



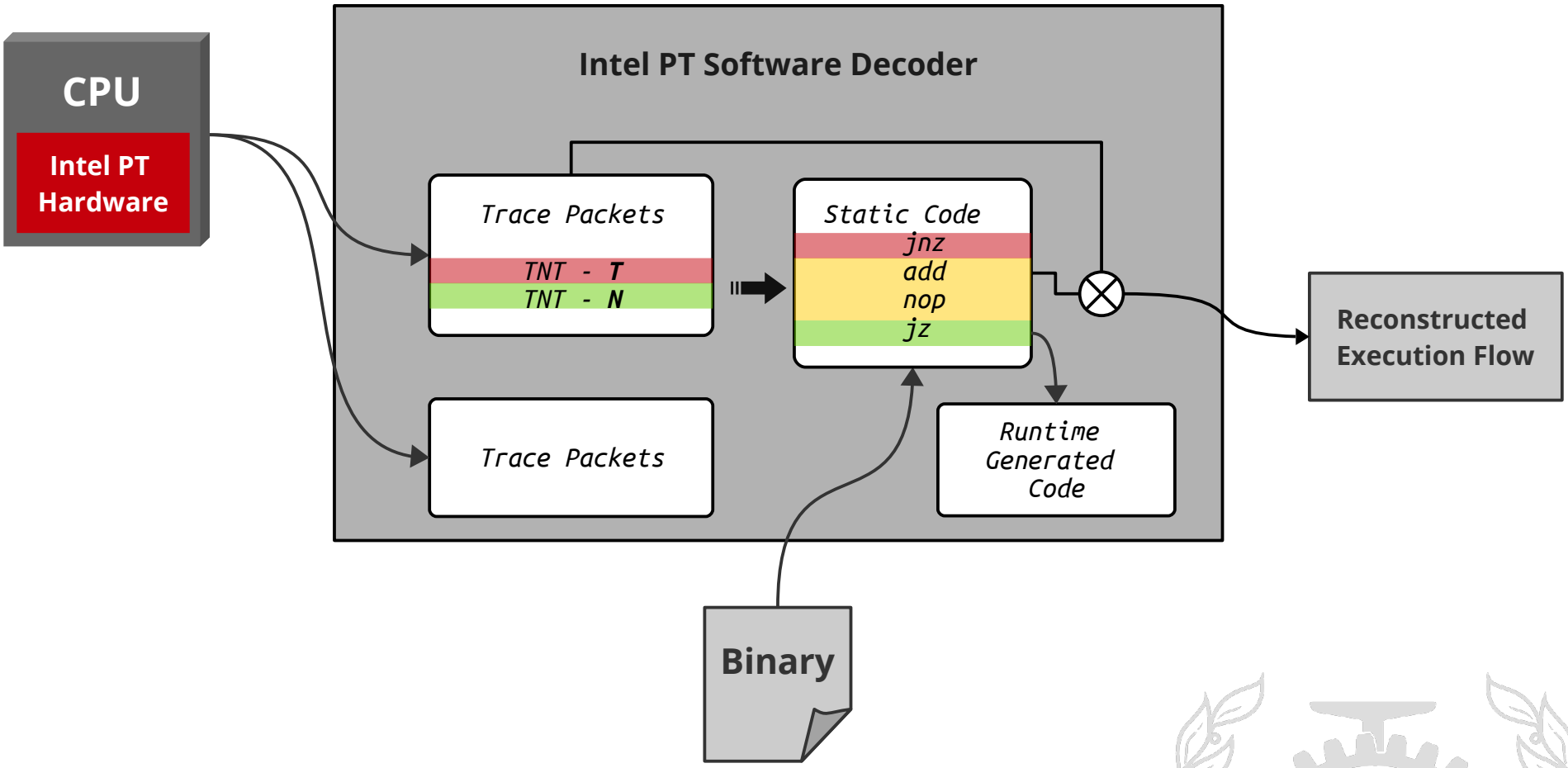
Intel PT

Limitations - JIT Code



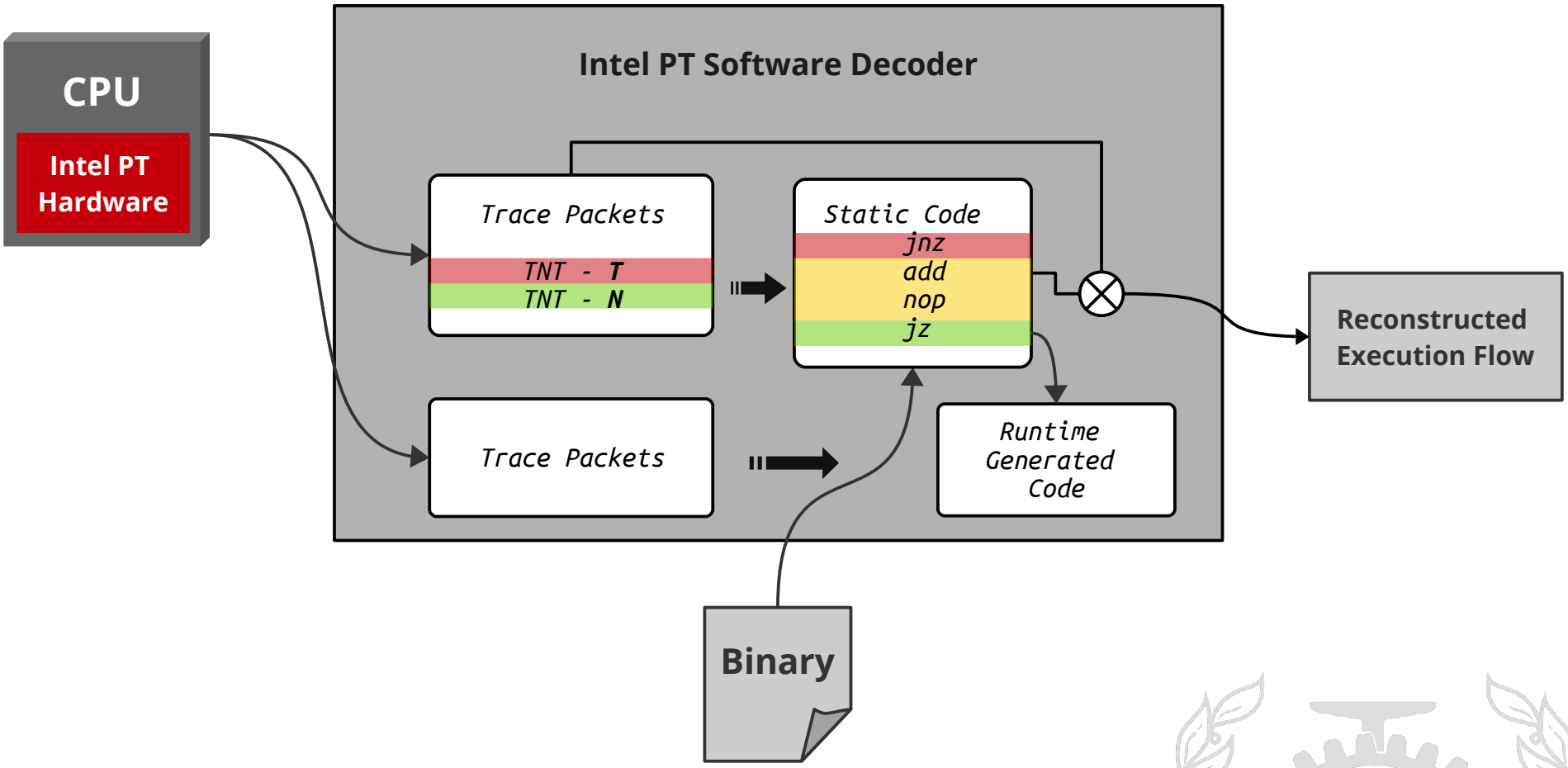
Intel PT

Limitations - JIT Code



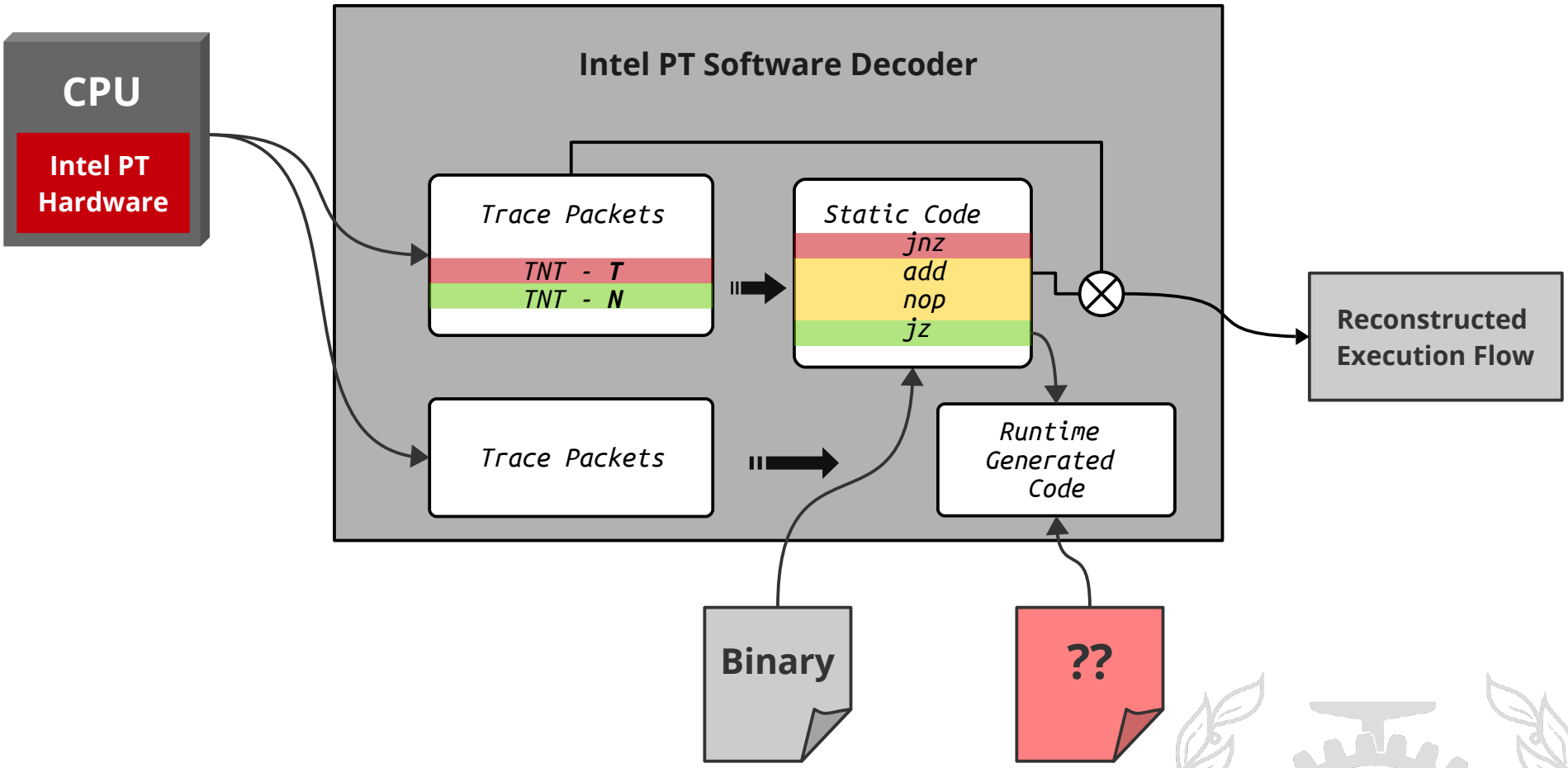
Intel PT

Limitations - JIT Code



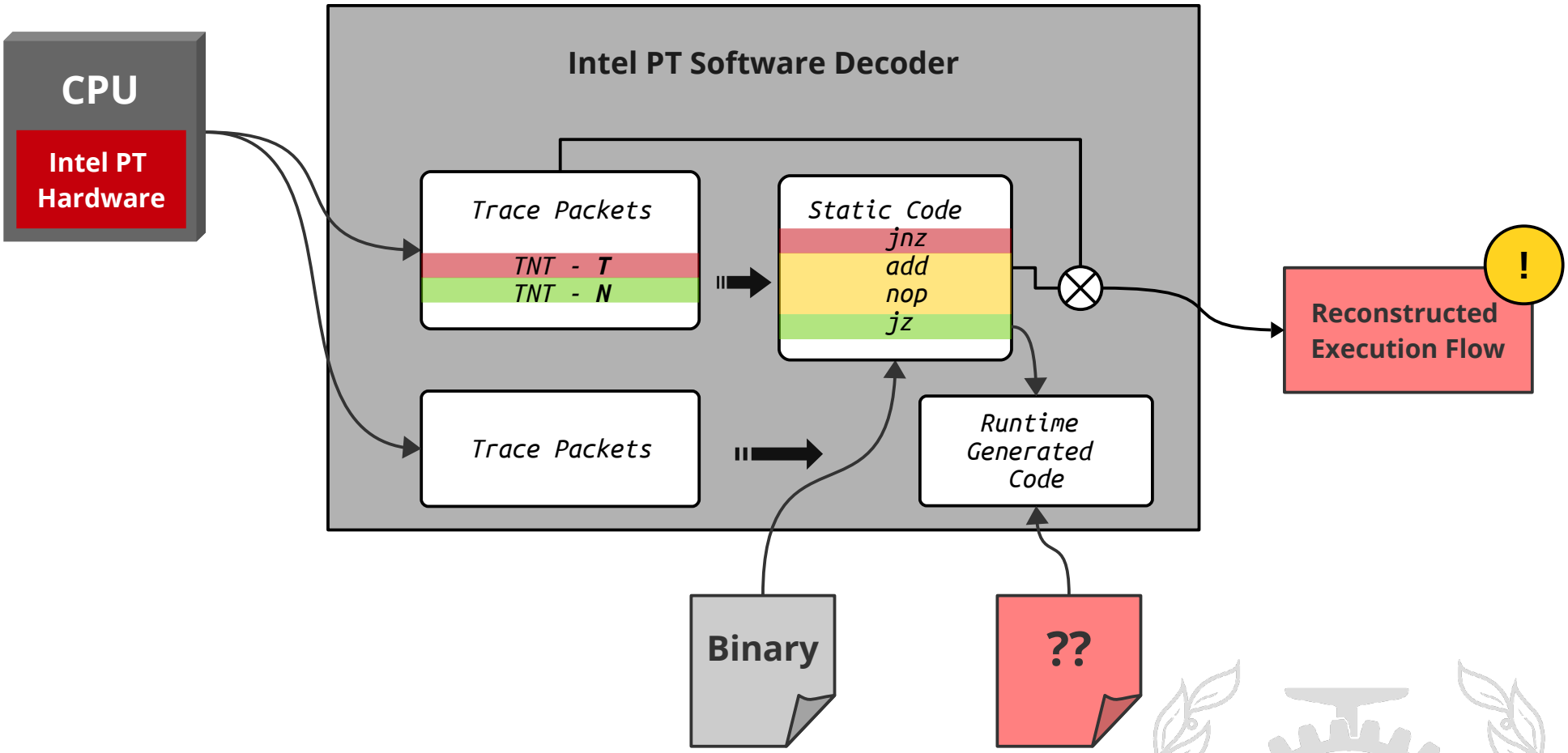
Intel PT

Limitations - JIT Code



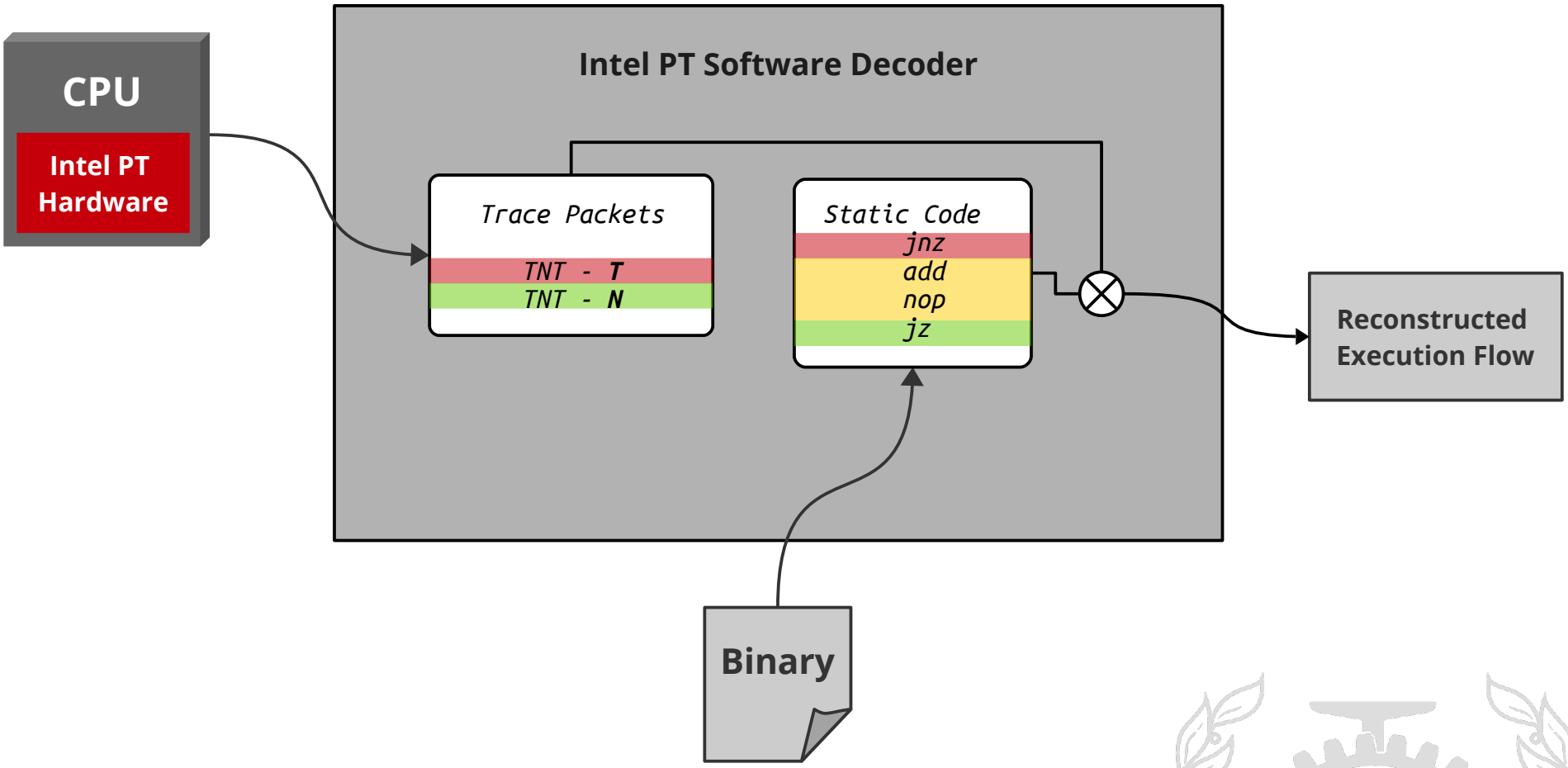
Intel PT

Limitations - JIT Code



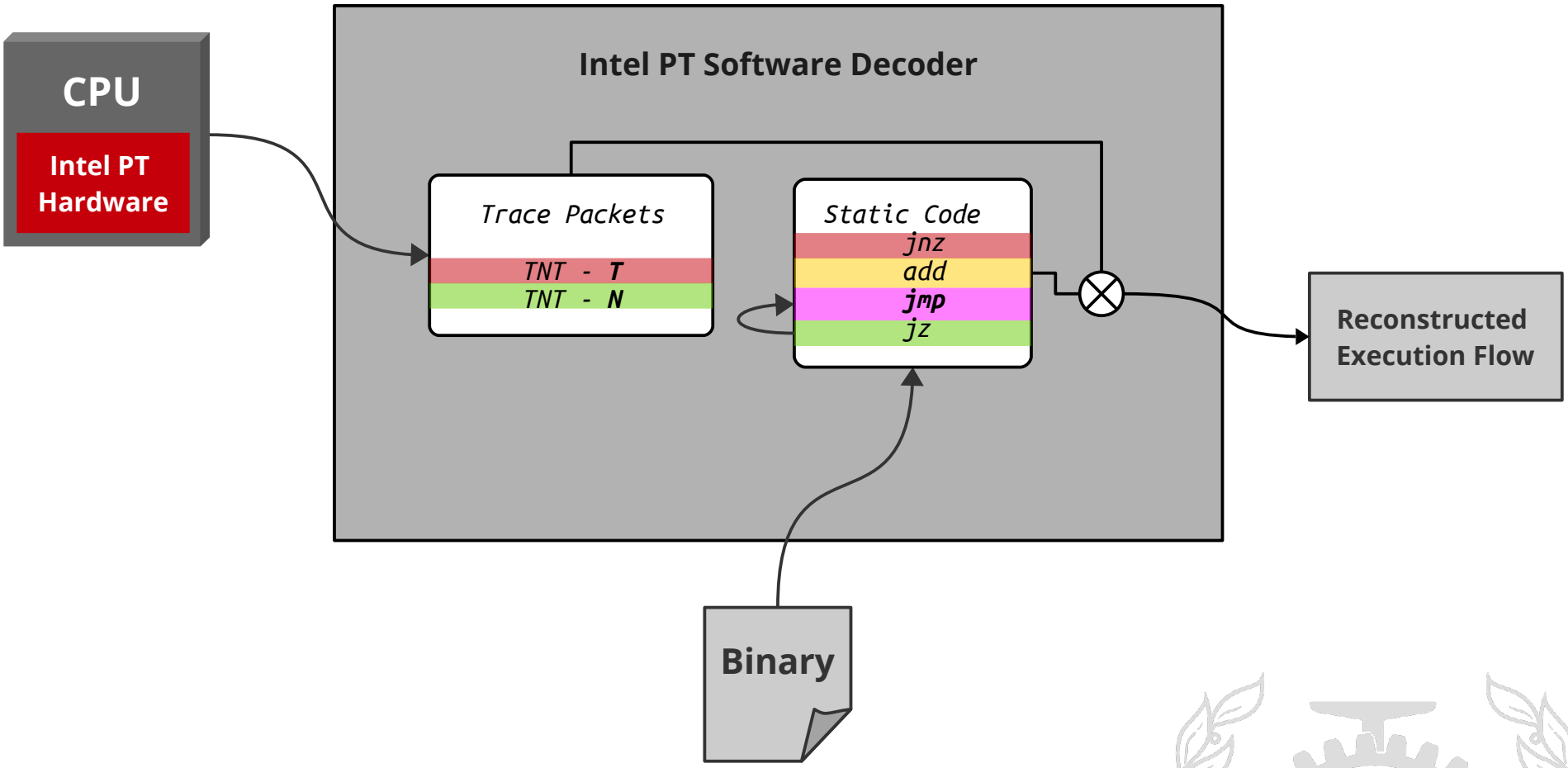
Intel PT

Limitations - Self-modifying Code



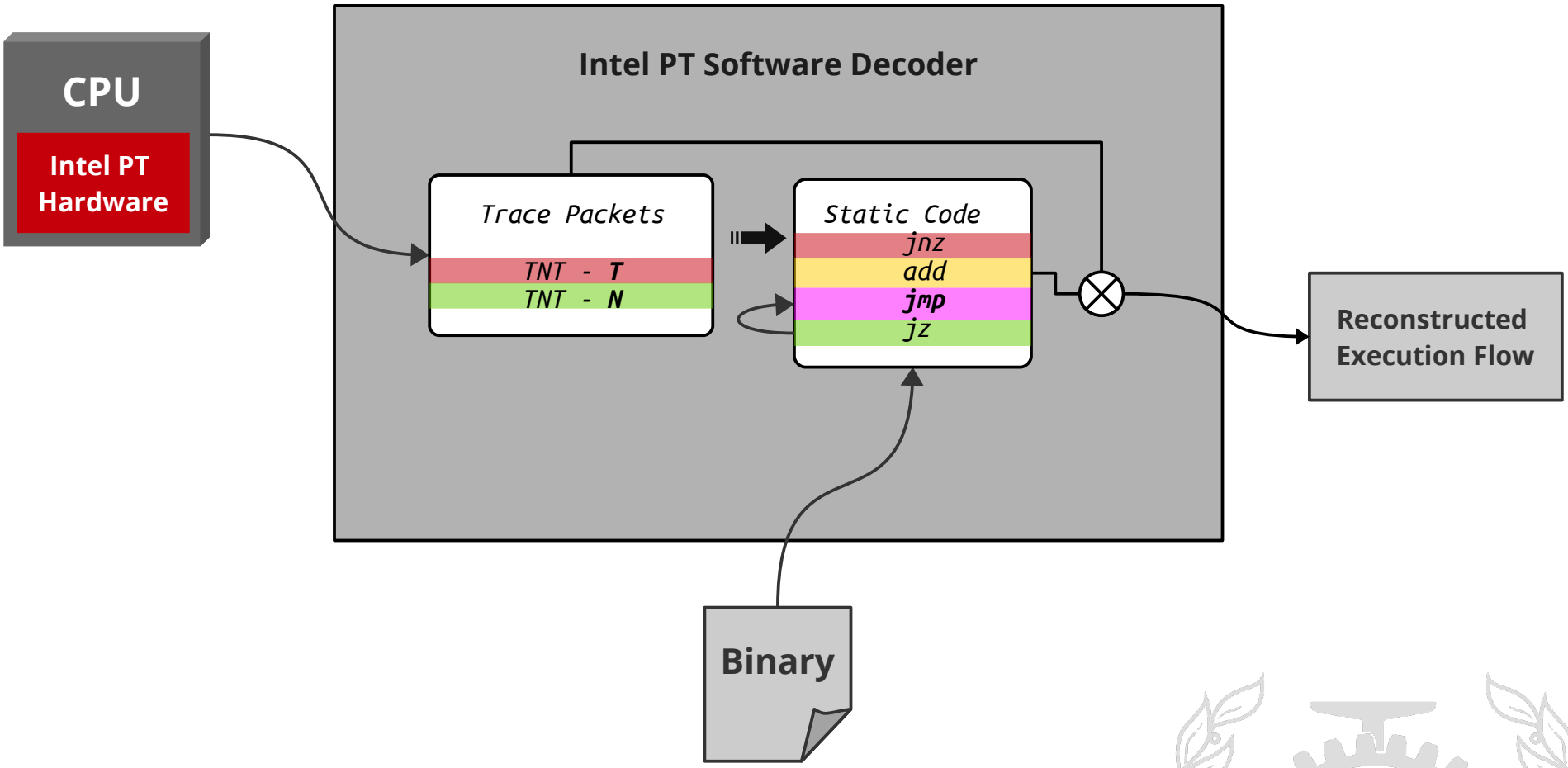
Intel PT

Limitations - Self-modifying Code



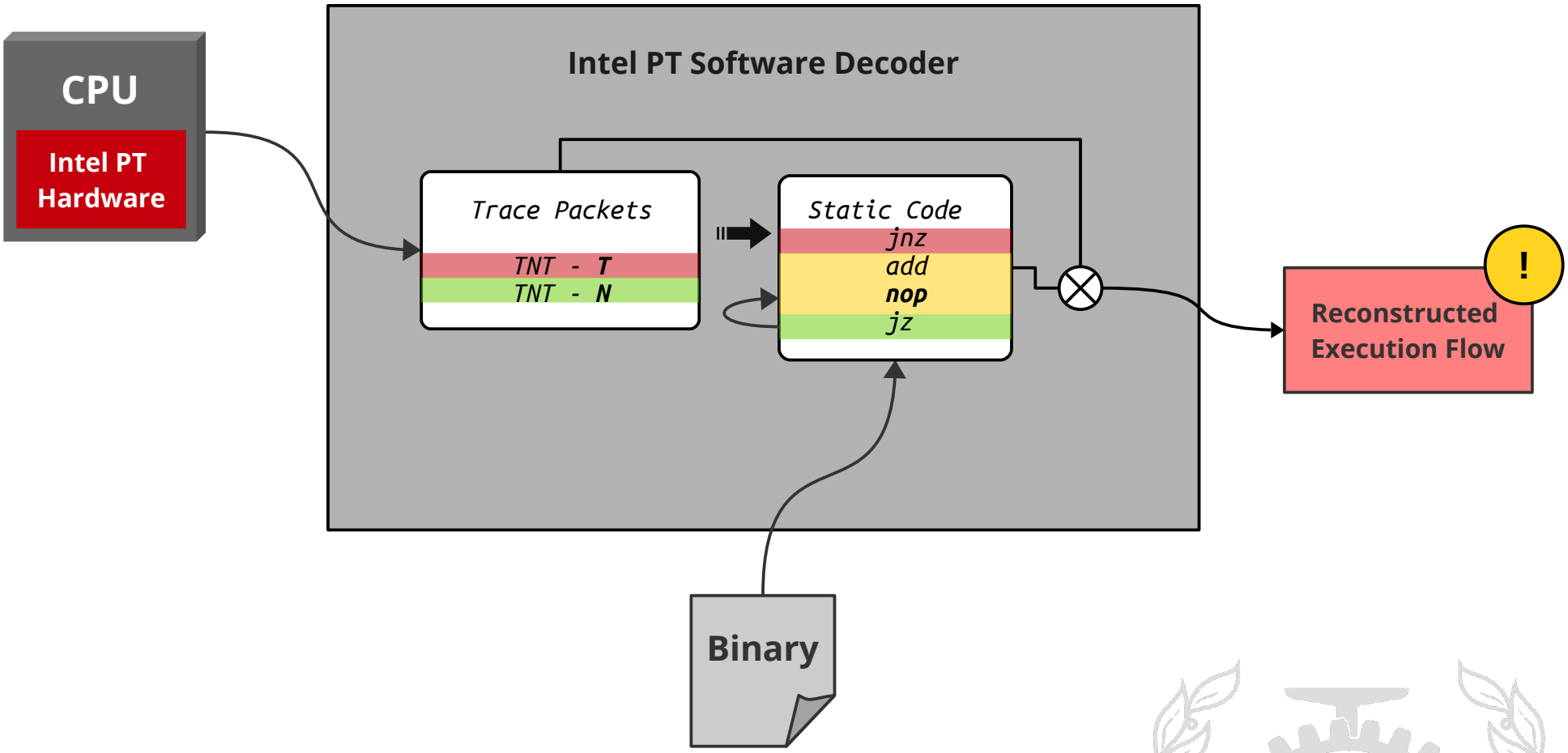
Intel PT

Limitations - Self-modifying Code



Intel PT

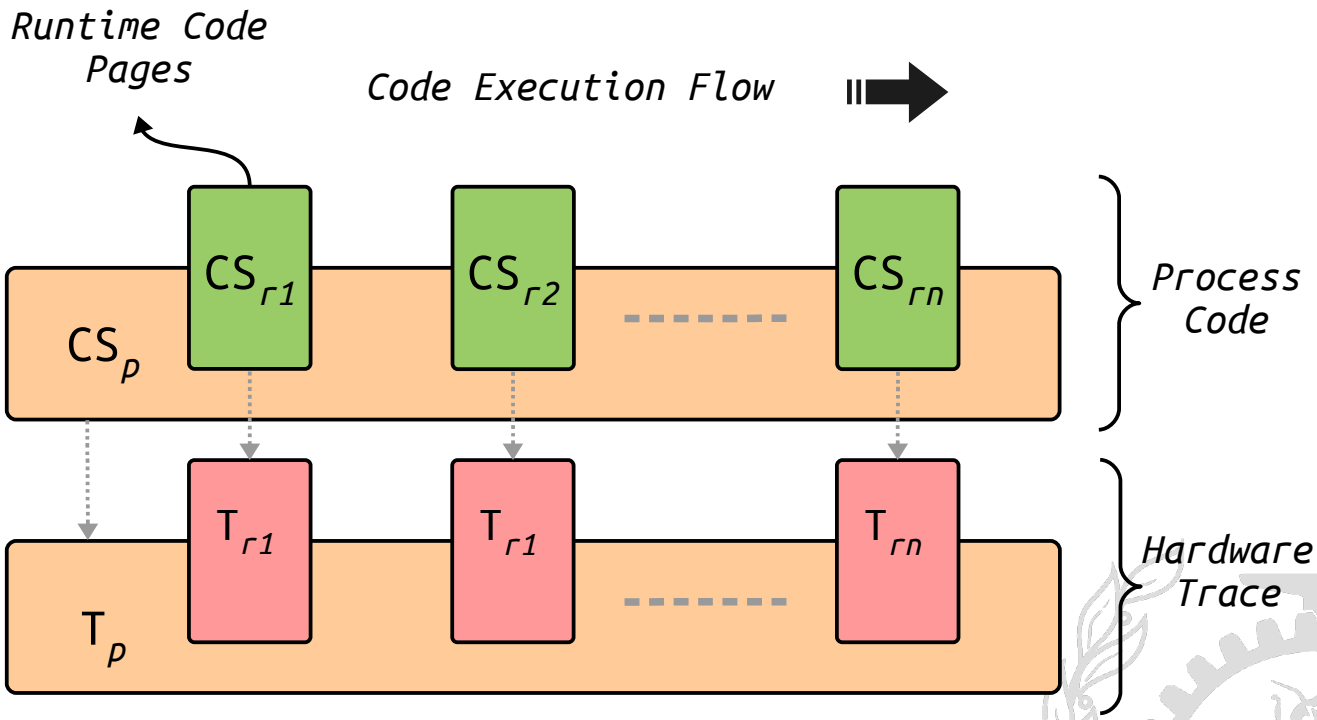
Limitations - Self-modifying Code



FlowJIT

Trace Reconstruction

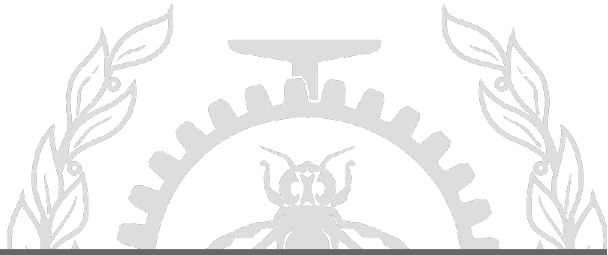
- JIT code (such as eBPF) allocates memory for code-cache
- We define dynamic Code Sections (CS_r) - pages corresponding to code-cache executing in process



FlowJIT

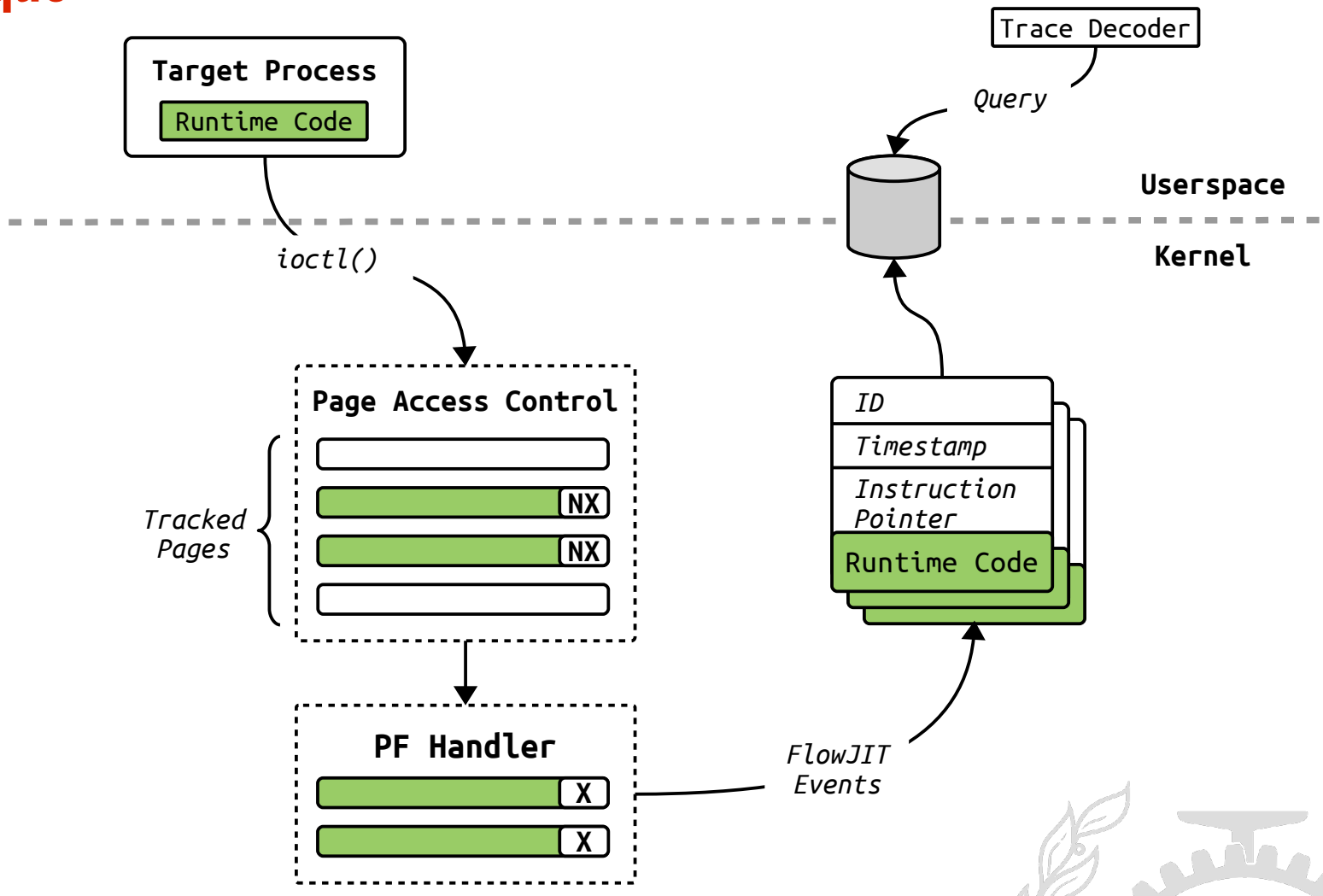
Technique

- In-kernel tracking of runtime generated/modified code pages
  - Compilers use **`malloc()`** and **`mprotect()`**
  - FlowJIT intercepts and modifies exec bits
- Synthetic page faults at execution
  - Intercept tracked pages and re-flip exec bits
  - Record IP, Timestamp, complete page data as a FlowJIT event and copy to disk.
- Events indexed and queried by IP. At decode time, query by failed decoding IP



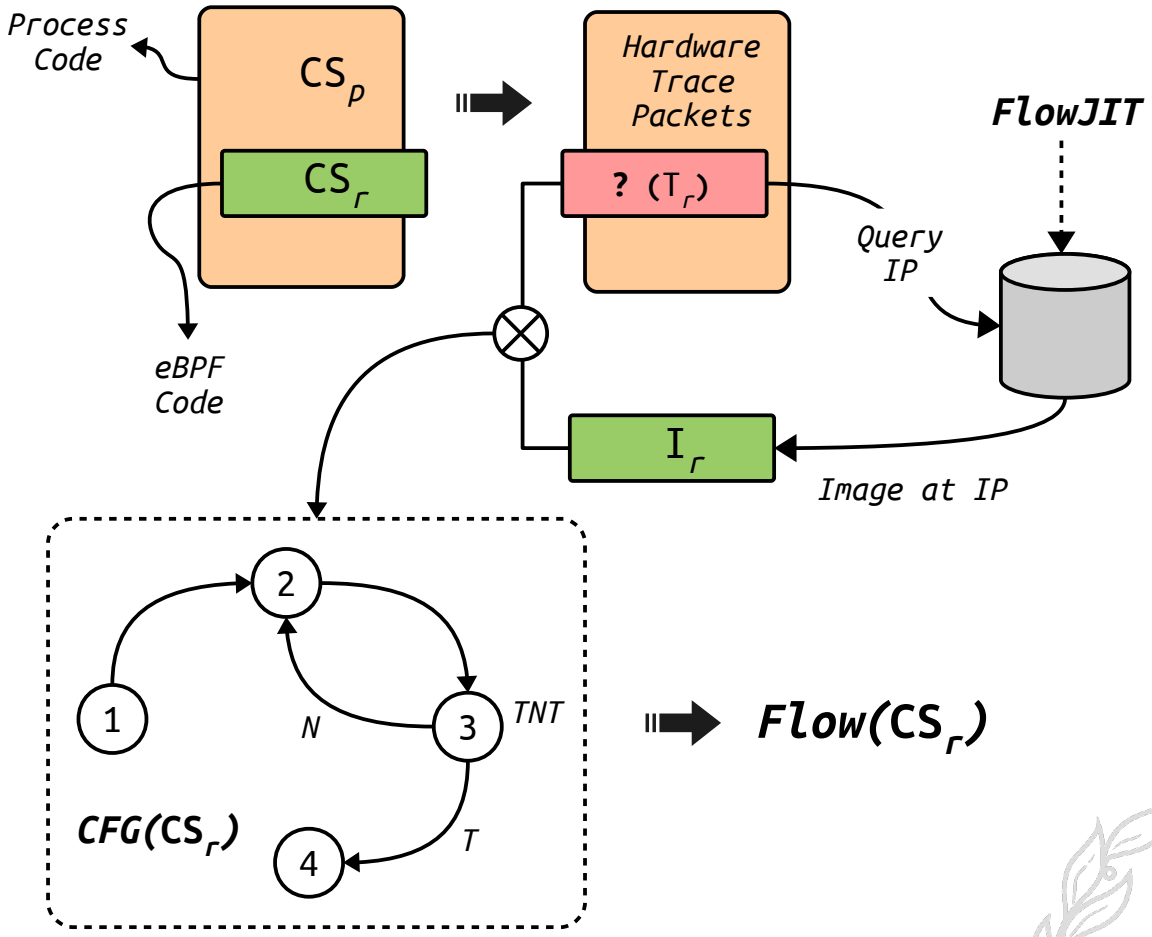
FlowJIT

Technique

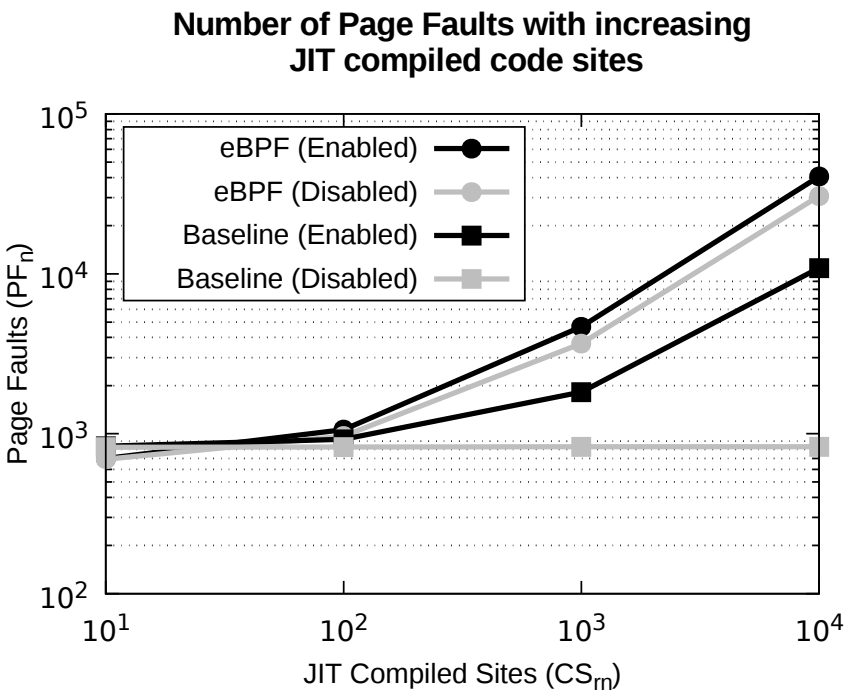


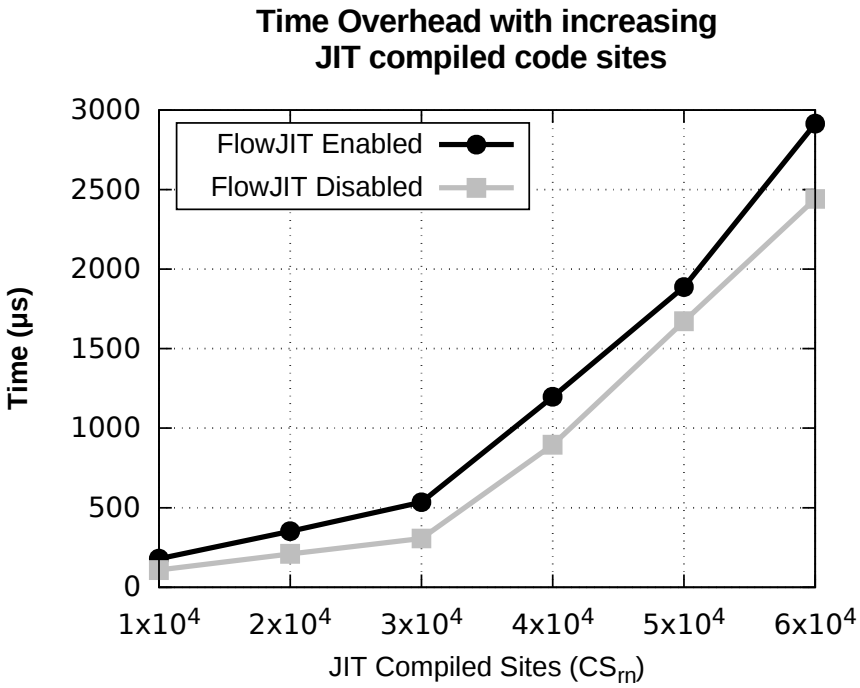
FlowJIT

Usecase : eBPF JIT Code



Experiments

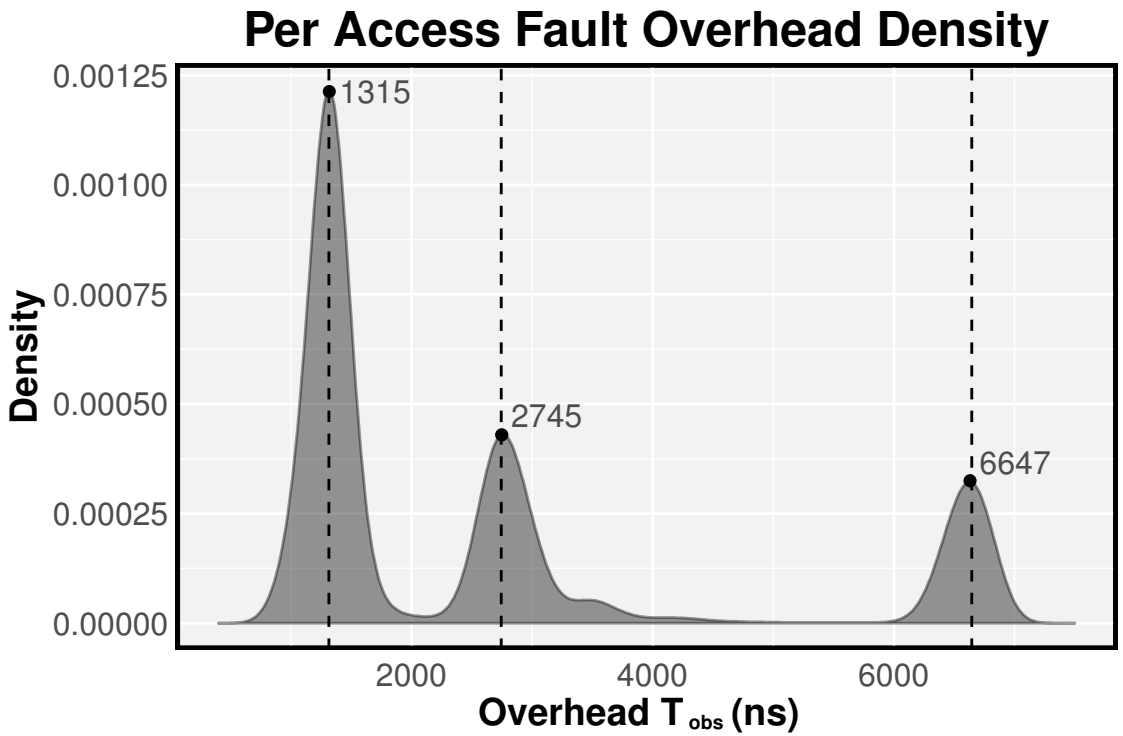




Experiments

T = T(Tracking Initiation) + **T(Access Change) + T(Page Fault)**

20K executions of JIT compiled sites



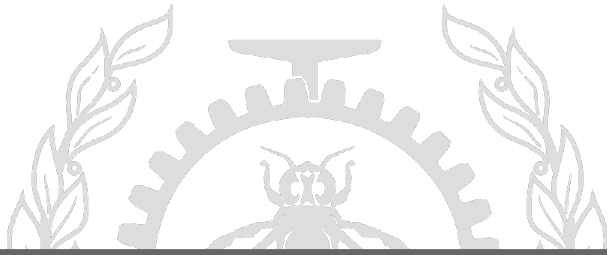
Upcoming

FlowJIT Patch

- An initial version of kernel patch against v4.7 available
  - https://github.com/tuxology/flowjit
- Enhance patch and work on Perf
  - Probably link FlowJIT events to PT data in Perf's aux buffer?

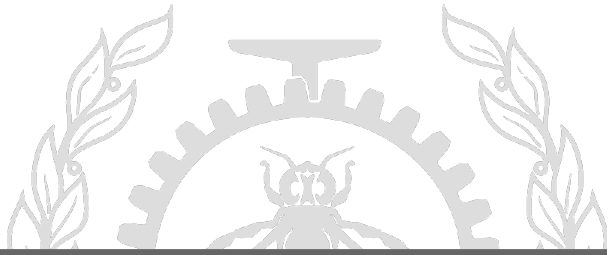
Future Work

- Extend self-modifying code to provide better code security and application robustness
- Extend FlowJIT to support ARM through Perf



Outcomes

- *Low Overhead Hardware-Assisted Virtual Machine Analysis and Profiling*, IEEE CCSNA'16 Globecom Workshops
- *Hardware Trace Reconstruction of Runtime Compiled Code* [Submitted]
- *Hardware-Assisted Instruction Profiling and Latency Detection*, Journal of Engineering, IET



“Education never ends, Watson. It is a series of lessons, with the greatest for the last.”

~ Arthur Conan Doyle

Questions?

suchakrapani.sharma@polymtl.ca