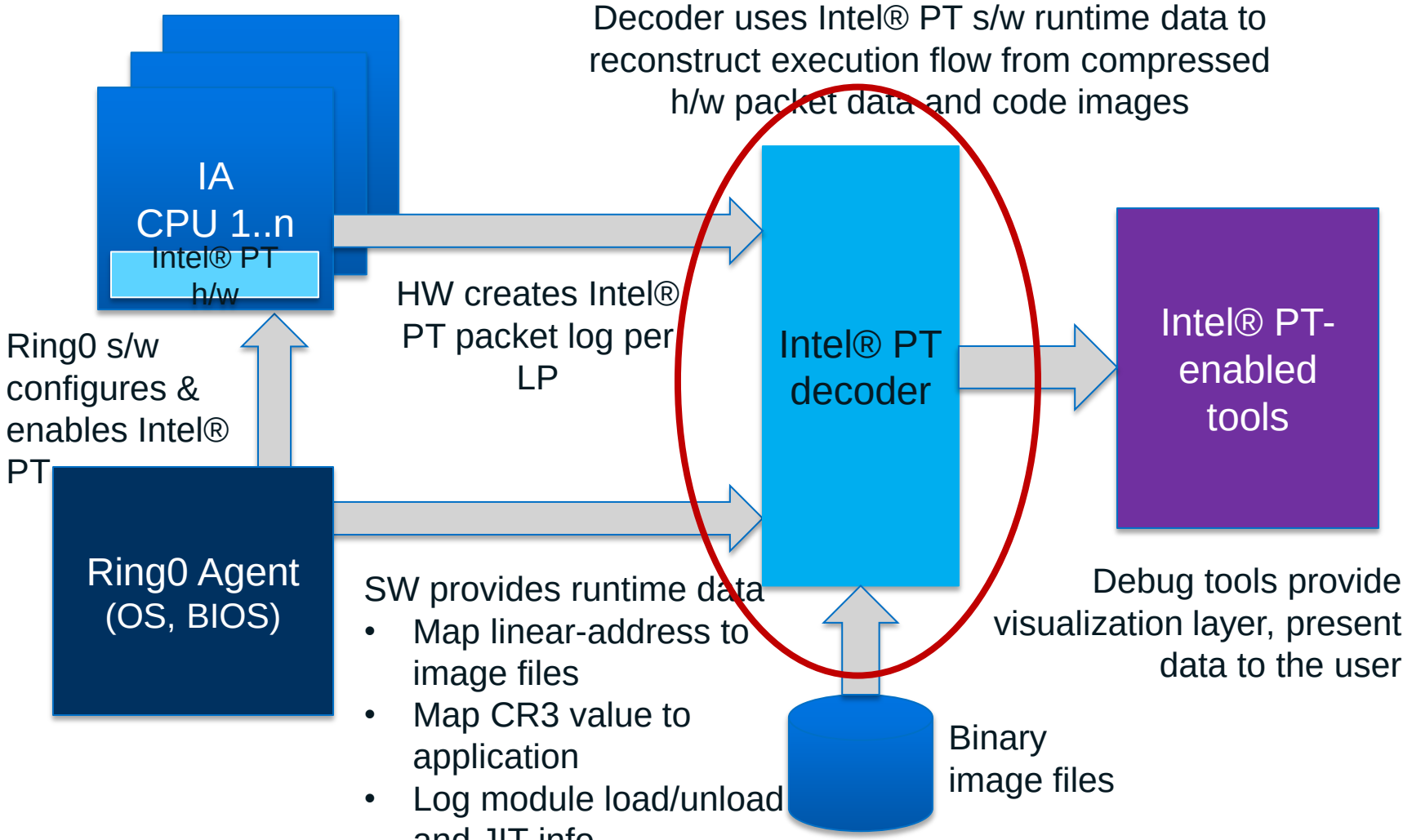

Intel® Processor Trace
What it is and how to decode it
Markus Metzger, Intel GmbH

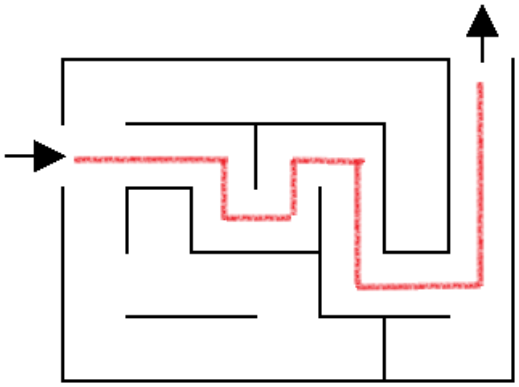
# Intel® Processor Trace
# What is it? How can I use it?

- New h/w feature for execution tracing
  - Compact trace format
  - Low trace collection overhead
  - Trace contains time stamps
- Usage models:
  - Crash debug: how did I get here?
  - Analysis: provide context around samples
  - Latency debug: what was the code doing around time X?

# Intel® Processor Trace H/W and S/W components



# Decoding Intel® Processor Trace Control Flow



- **PSB**: packet stream boundary
  - Unique pattern for synchronizing onto the
- **TNT**: taken/not-taken packet
  - One bit for each conditional jump
- **TIP**: target ip packet
  - One packet for each indirect jump or call
- **FUP**: flow update packet
  - Indicate IP for other packets
  - **FUP+TIP**: asynchronous control transfer

# Decoding Intel® Processor Trace Execution State Information

- **MODE.EXEC**: execution mode packet
  - Execution mode, i.e. 16bit, 32bit, 64bit
- **MODE.TSX**: tsx state packet
  - Transactional execution state changes
- **PIP**: paging information packet
  - Paging changes, e.g. switching processes
- **TIP.PGE/TIP.PGD**: trace generation enable/disable
  - Indicate start/stop of tracing
- **TSC**: time stamp counter
  - Timing information

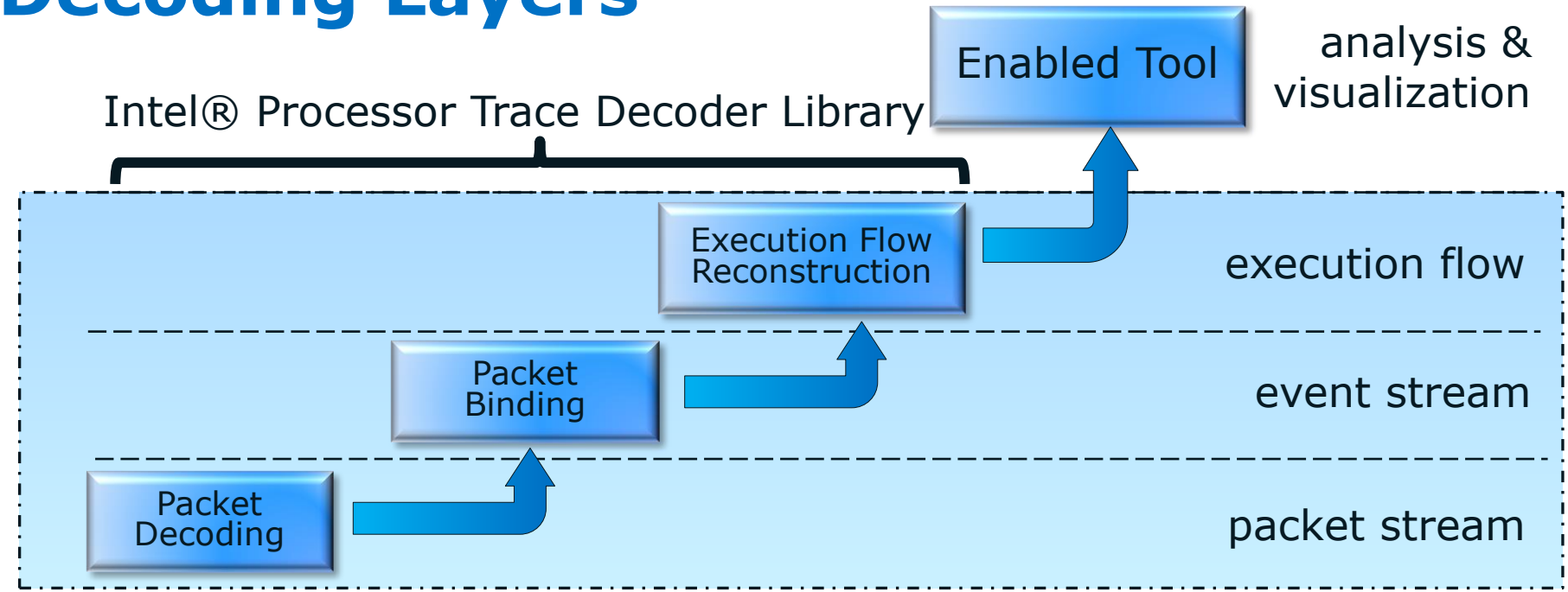
# Decoding Intel® Processor Trace Packet Binding

- Packets bind together to describe events
  - **MODE.EXEC+TIP**: change exec mode after this branch
  - **MODE.TSX+FUP**: tsx state change at this IP
- Packets may bind to packet combinations
  - **MODE.TSX(abort)+FUP+TIP**: (async) transaction abort
- Context sensitive packet semantics
  - **TIP**: binds to next indirect jump or call
  - **FUP+TIP**: asynchronous jump
- Bindings and semantics specified for each packet in SDM: http://download-software.intel.com/sites/default/files/319433-015.pdf.

# Intel® Processor Trace Decoder Library

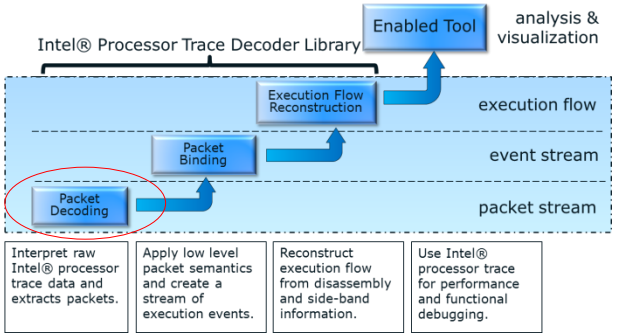
- Intel's reference package of s/w building blocks:
  - Packet encoding/decoding
  - Execution event reconstruction (packet binding)
  - Execution flow reconstruction
  - Samples
    - packet dumper
    - instruction flow dumper (using XED disassembler)
  - Testing
    - tools for creating and running tests
    - test suite covering sample tools
- S/W enabling ahead of Si.
- Written in C – runs on Linux, Windows, and OS X.
- Available at http://01.org under 3-clause BSD license.

# Intel® Processor Trace Decoder Library Decoding Layers



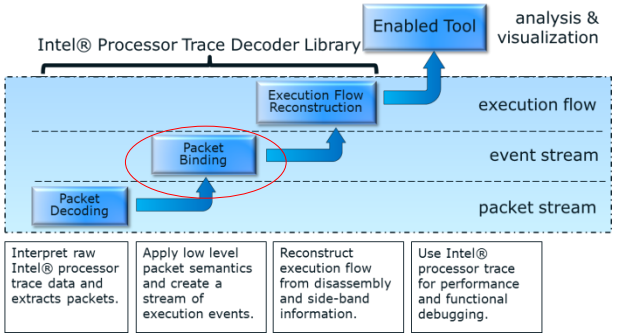

| | | | |
|---|---|---|---|
| Interpret raw Intel® processor trace data and extracts packets. | Apply low level packet semantics and create a stream of execution events. | Reconstruct execution flow from disassembly and side-band information. | Use Intel® processor trace for performance and functional debugging. |

# Intel® Processor Trace Decoder Library Packet Decoding



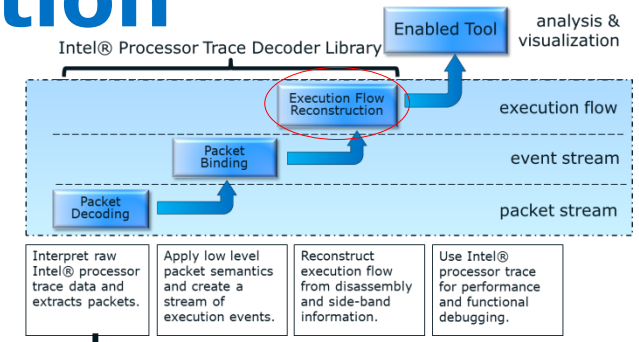

- Decode a single Intel® PT packet into a C struct:
  - Intel® PT opcode
  - Intel® PT payload
  - Payload interpretation where necessary
- Intel® PT packet dumper built on top.

# Intel® Processor Trace Decoder Library Packet Binding




- Combine Intel® PT packets to extract semantics.
- Keep track of time.
- Read ahead to signal upcoming events.
- Query interface for
  - Conditional branches
  - Indirect branches
  - Events
  - Time

# Intel® Processor Trace Decoder Library Execution Flow Reconstruction




- Fast, special-purpose instruction decoder.
- Reconstruct the execution flow based on
  - Instruction analysis
  - Event stream (query interface)
- Interface to iterate over instructions providing:
  - Instruction virtual address
  - Raw instruction bytes and size
  - Execution mode
  - Coarse classification
  - Add. information like enable/disable, speculation, ...

# https://01.org
# Processor Trace Decoder Library

# Intel® Processor Trace in GDB

```
(gdb) record function-call-history /cli
12          fib inst 101,111    at src/fib.c:3,7
13            fib       inst 112,124    at src/fib.c:3,8
14          fib inst 125,129    at src/fib.c:7
15            fib       inst 130,142    at src/fib.c:3,8
16          fib inst 143,147    at src/fib.c:7,8
17        fib   inst 148,152    at src/fib.c:7,8
18      fib     inst 153,157    at src/fib.c:7
19        fib   inst 158,168    at src/fib.c:3,7
20          fib inst 169,179    at src/fib.c:3,7
21            fib       inst 180,185    at src/fib.c:3,4
```

- Current implementation supports BTS
  - Print trace on instruction level
  - Print trace on function level (show call structure)
  - Limited reverse execution and replay
- Patches being discussed at gdb mailing list

# Legal Disclaimer & Optimization Notice

INFORMATION IN THIS DOCUMENT IS PROVIDED "AS IS". NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO THIS INFORMATION INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

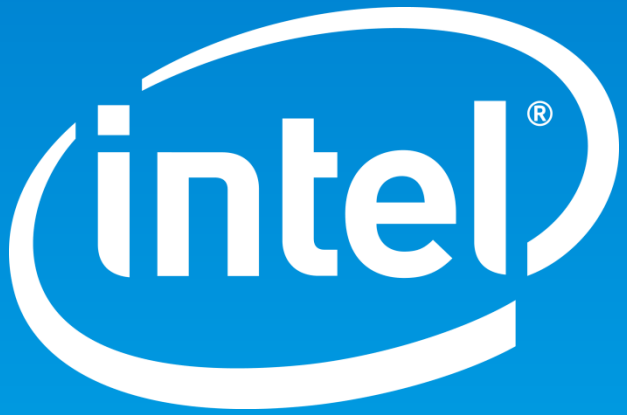
Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.



**Optimization Notice**

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Notice revision #20110804


intel