Темная магия аппаратной трассировки приложений

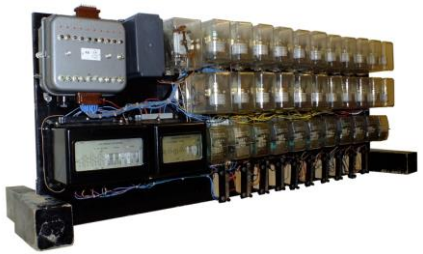
Артём Кашканов

Я – Инженер энтузиаст.

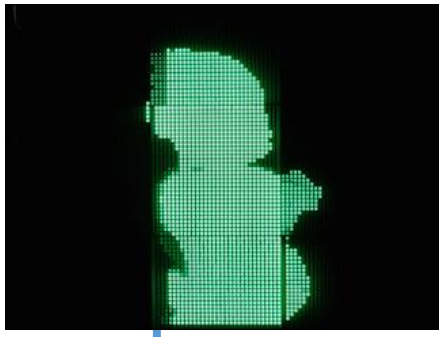
SCAN ME

SCAN ME

Релейный контроллер автополива



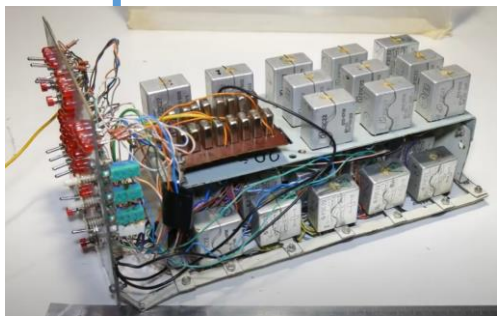
ГИП 10000 на стероидах



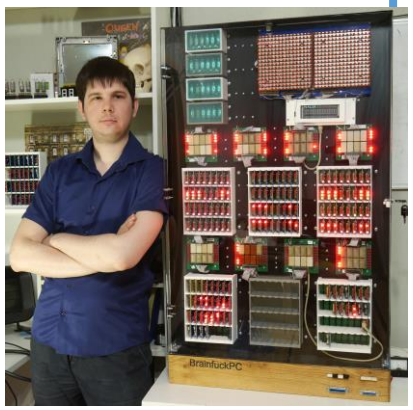
Экспериментальная декатронная ячейка

Реставрация машинки Robotron S6130

2008 2015 2017 2019 2020 2021 2022 2023 2024 2025

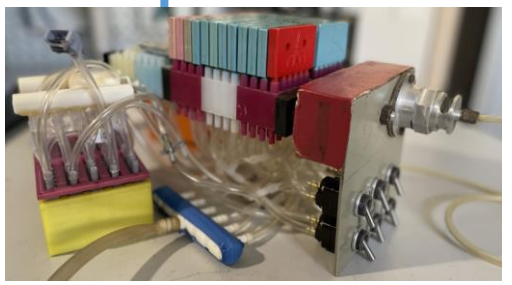


Релейный калькулятор РЦВМ-1



Релейный компьютер BrainfuckPC

Самодельные Электролюминесцентные индикаторы

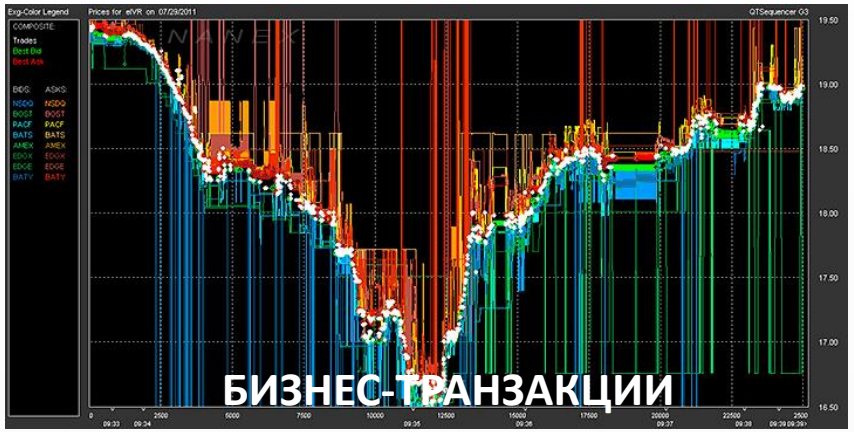


Процессор на пневмонике

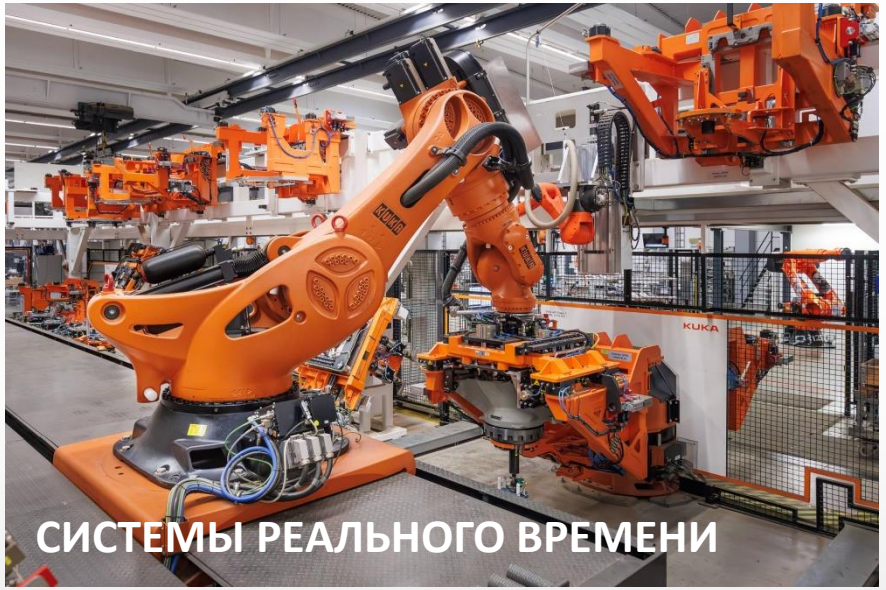
Эмулятор лампового компьютера.

Стабильность периодических событий

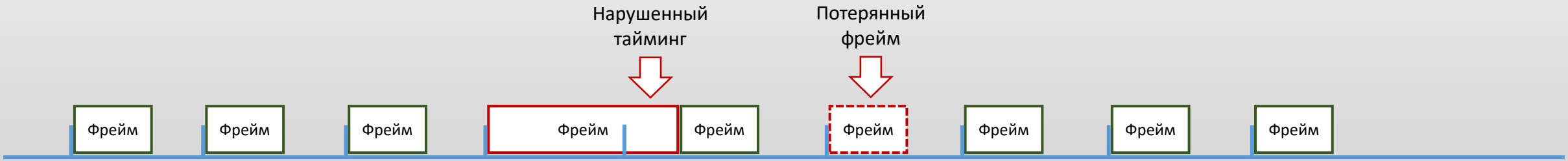
Нестабильность ФПС =
Недовольный игрок



Медленная реакция на транзакцию =
Потеря денег

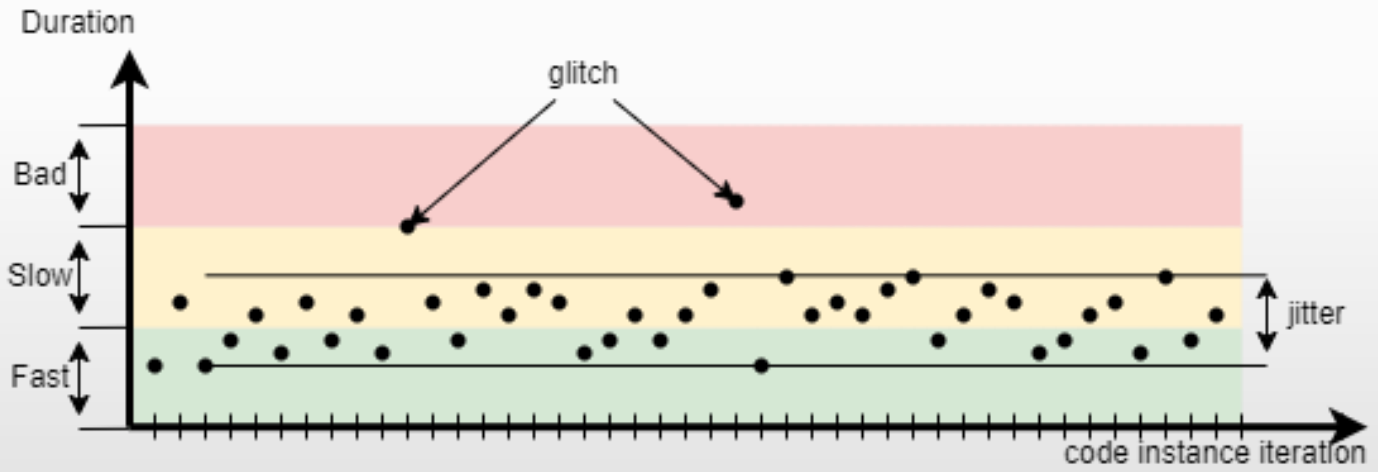


Датчик не обработан вовремя =
Повреждение детали. Или робота



Джиттер и Глитчи

- Джиттер – девиация времени исполнения участка кода
- Глитч – резкий выброс времени исполнения



Как найти Джиттер и Глитчи

Статистический самплинг

+ Идеален для анализа установившихся процессов
- Дает агрегированную картину

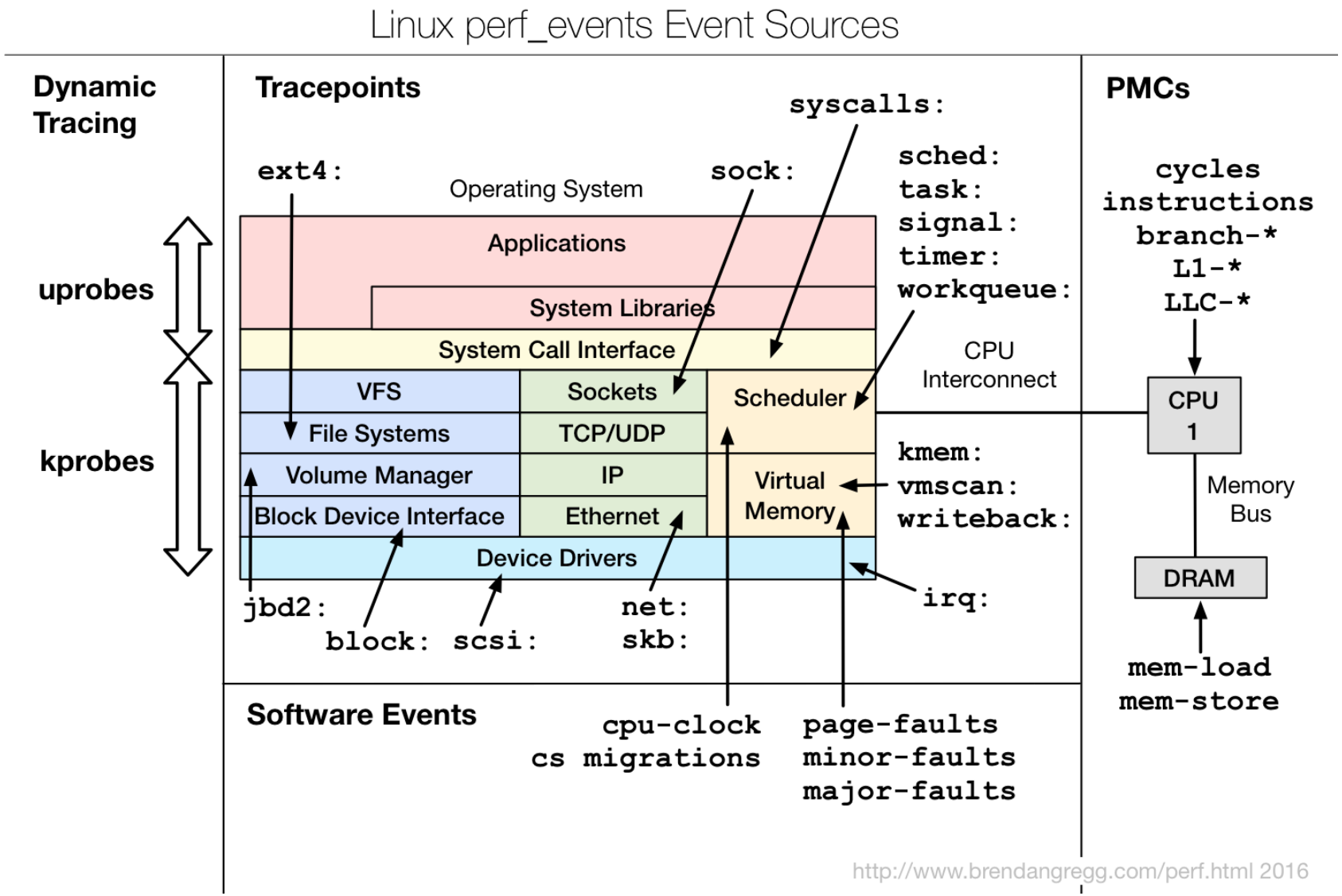
Инструментация кода

+ Отличное решение для отладки
- Требуется модификация кода

Аппаратная трассировка

+ Полная и точная информация
- Большой оверхед. Или нет?

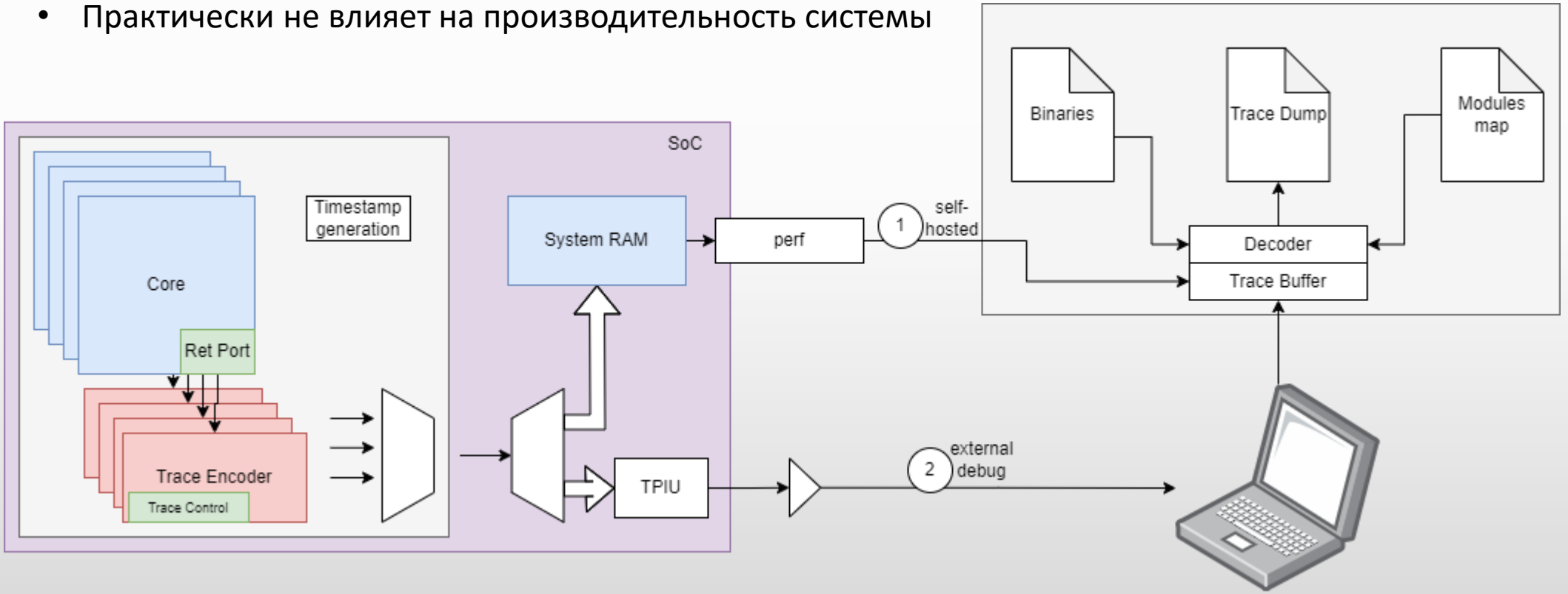
Linux perf events overview



http://www.brendangregg.com/perf.html 2016

Аппаратная трассировка

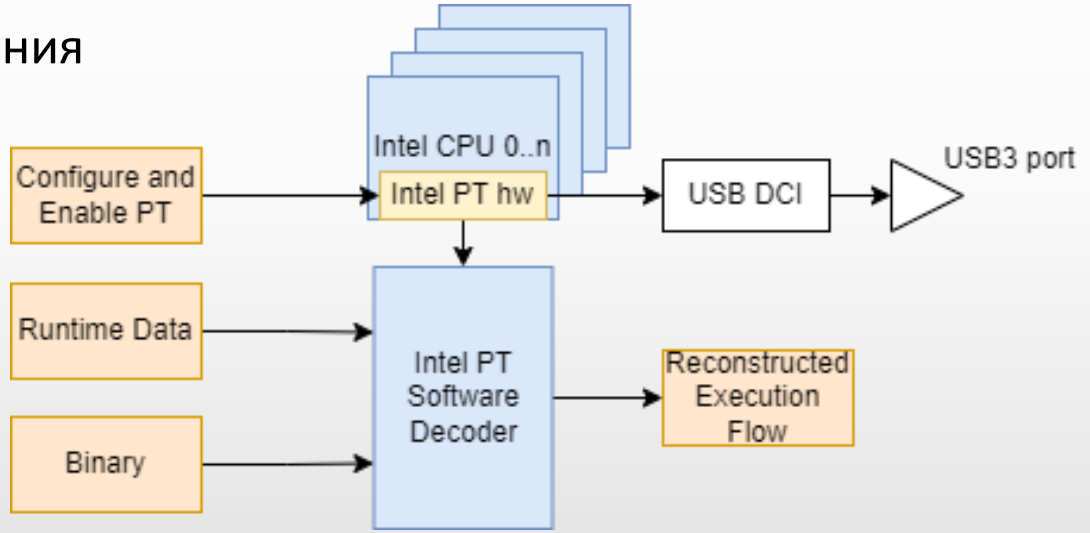
- Мощный, но узко-специалилированный инструмент анализа производительности
  - Записывает поток исполнения программы с наносекундными метками времени
  - Не требуется модификация кода
  - Практически не влияет на производительность системы



Intel ® processor Trace (PT)

- Аппаратный блок для записи полного потока исполнения с метками времени и малым влиянием на систему (Intel® 6+ gen)

- Собирает:
  - Косвенные и условные переходы, прерывания и исключения
  - Асинхронные события, транзакции
  - Входы и выходы в виртуальные машины
  - Изменения частоты, режимов энергопотребления
  - PEBS события от Core PMU блока
- Опции фильтрации:
  - По уровням привилегий (User/Kernel/Both)
  - По виртуальным адресным пространствам
  - По диапазонам IP адресов
- События доступны с точностью до 1 клоктика*



- Intel® Processor Trace in the Intel® Software Developer Manual, Volume 3B, Chapter 33

Применение

- **Поиск коротко-живущих, случайных просадок производительности**
  - Анализ причин возникновения глитчей, джиттера, которые невозможно поймать инструментами статистического анализа (семплированием)
  - Анализ влияния ядра ОС, прерываний, гипервизора на работу приложений
- **Анализ старта/завершения приложений**, операционной системы, Uefi и др.
- **Postmortem анализ падения приложения**, драйверов, системы
  - Предоставляет полноценный стек вызовов функций, в том числе с метриками производительности

Применение

- Проверка целостности исполняемого кода
  - [ASPLOS’17] GRIFIN: Guarding Control Flows Using Intel Processor Trace
- Отладка состояния гонки
  - [SOSP’17] Lazy Diagnosis of In-Production Concurrency Bugs
- Обратная отладка
  - [OSDI’18] REPT: Reverse Debugging of Failures in Deployed Software
  - [ATC’20] Reverse Debugging of Kernel Failures in Deployed Systems
- Фаззинг
  - [CCS’21] HyperFuzzer: An Efficient Hybrid Fuzzer for Virtual CPUs

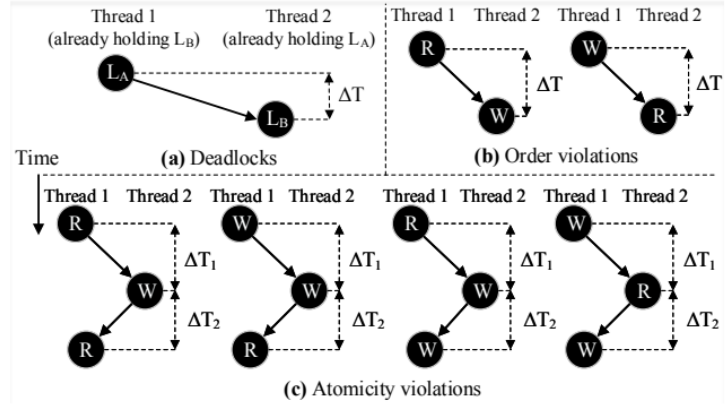


Figure 1: Patterns of concurrency bugs and target events

Поддержка в Linux Perf

- Доступна с kernel 4.1. Полноценно – с 4.9
  - System-wide сбор контекста, доступно для non-root пользователя
  - В kernel 5.1-5.14 есть баг с потерей трассы при работающих виртуальных машинах

Full Trace Mode

Непрерывный сбор трассы пока не кончится место на диске (50-500MBps/core)

Ring Buffer Mode

Данные собираются в кольцевой буфер, до срабатывания внешнего события

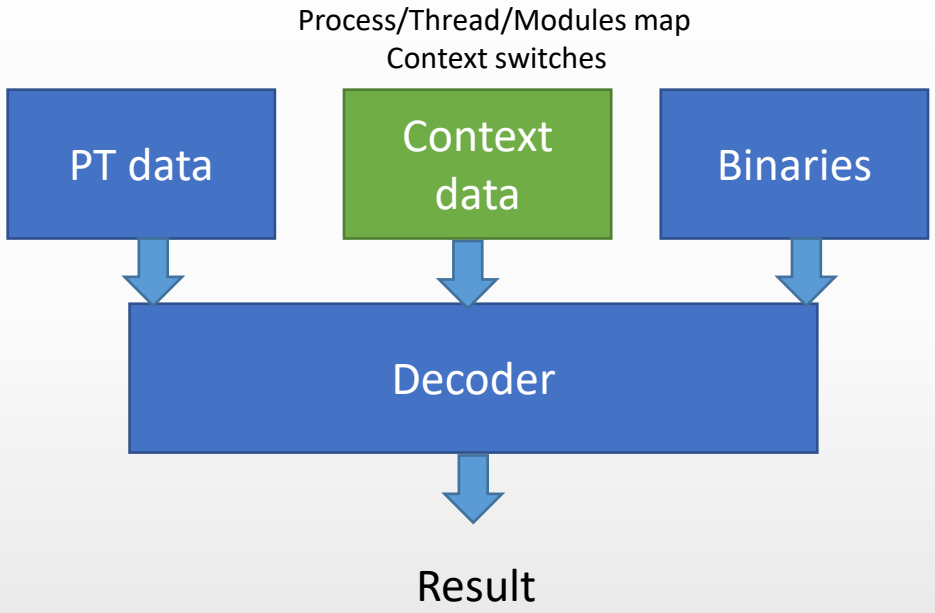
Sampling Mode

На каждый сампл от PMU счетчиков, небольшой объем PT данных

```
perf record -e intel_pt// -- ls
```

perf report –D --header

```
0 0 0x65b0 [0x30]: PERF_RECORD_AUXTRACE size: 0x160  offset: 0  ref: 0x434ca35767e2  idx: 0  tid: 3916  cpu: 0
.
. ... Intel Processor Trace data: size 352 bytes
.  00000000:  02 82 02 82 02 82 02 82 02 82 02 82 02 82 02 82 PSB
.  00000010:  00 00 00                                        PAD
.  00000013:  99 20                                           MODE.TSX TXAbort:0 InTX:0
.  00000015:  99 01                                           MODE.Exec 64
.  00000017:  7d d8 8a 07 b1 ff ff 00                         FUP 0xffffb1078ad8
.  0000001f:  00                                              PAD
.  00000020:  59 c9 00 00 00 00                               MTC 0xc9
.  00000026:  02 43 00 c4 75 00 00 00                         PIP 0x3ae200 (NR=0)
.  0000002e:  00 00 00 00 00 00 00 00                         PAD
.  00000036:  02 c8 ff ff ff ff ff 00                         VMCS 0xffffffffff
.  0000003e:  00 00 00 00 00 00 00 00                         PAD
.  00000046:  19 54 de 54 a3 4c 43 00                         TSC 0x434ca354de54
.  0000004e:  00 00 00 00 00 00 00 00                         PAD
.  00000056:  02 73 48 ce 00 24 00 00                         TMA CTC 0xce48 FC 0x24
.  0000005e:  00 00                                           PAD
.  00000060:  02 03 2a 00                                     CBR 0x2a
.  00000064:  02 23 00                                        PSBEND
.  00000067:  71 da 8a 07 b1 ff ff 00                         TIP.PGE 0xffffb1078ada
.  0000006f:  00                                              PAD
.  00000070:  4d 79 6b 01 b1                                  TIP 0xb1016b79
.  00000075:  2d 3e 71                                        TIP 0x713e
.  00000078:  4d a1 f2 00 b1                                  TIP 0xb100f2a1
.  0000007d:  ac 00 00                                        TNT NTNTTN (6)
.  00000080:  2d ff f5                                        TIP 0xf5ff
.  00000083:  d4                                              TNT TNTNTN (6)
.  00000084:  06                                              TNT T (1)
.  00000085:  59 ca                                           MTC 0xca
.  00000087:  f6                                              TNT TTTNTT (6)
.  00000088:  04                                              TNT N (1)
.  00000089:  4d 10 e0 06 b1 00 00                            TIP 0xb106e010
.  00000090:  4d 4c f6 00 b1                                  TIP 0xb100f64c
.  00000095:  2d d2 71                                        TIP 0x71d2
```

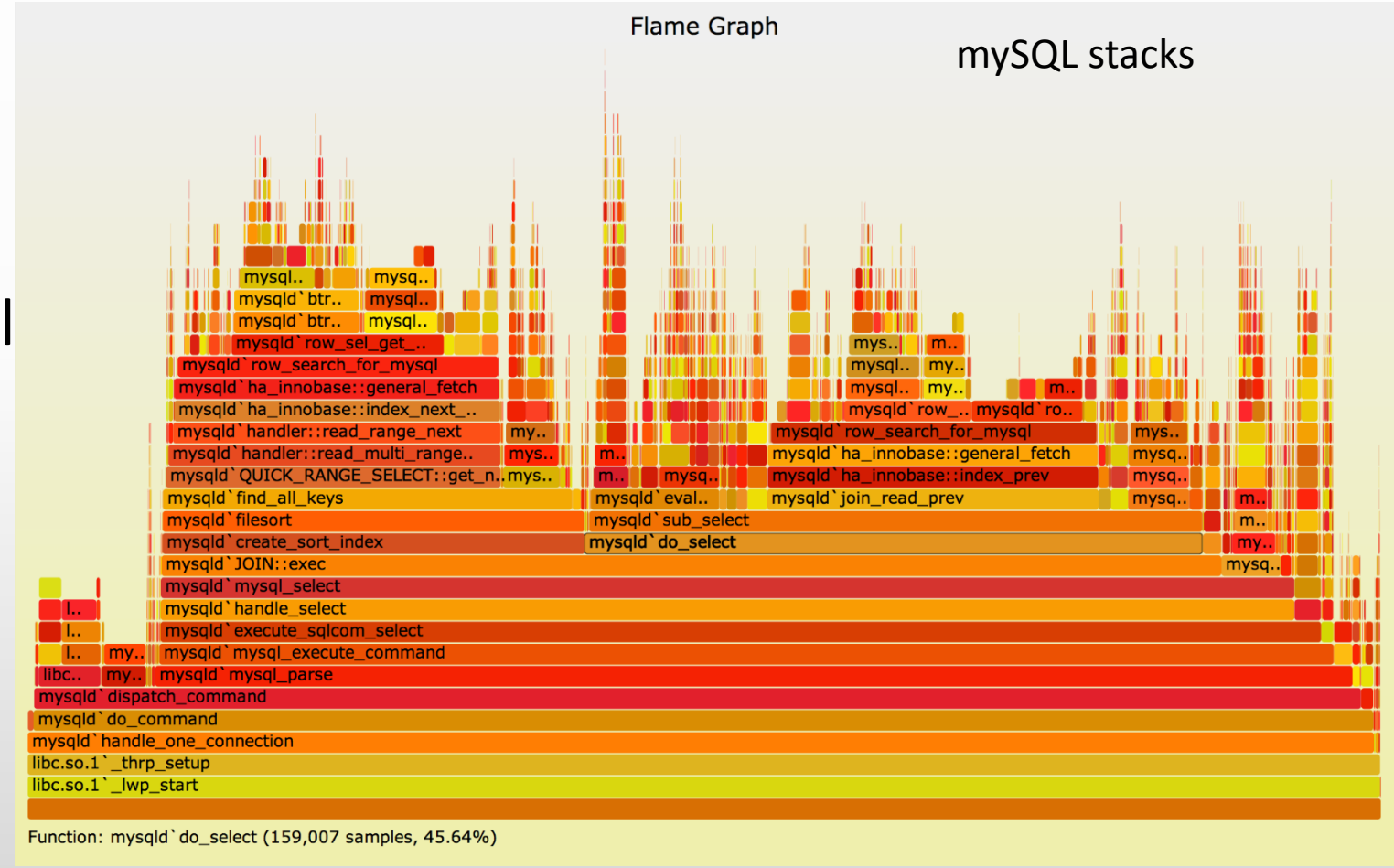


perf script --call-trace

```
Instruction trace error type 1 time 20532.573714140 cpu 3 pid 3916 tid 3919 ip 0 code 8: Lost trace data
          matrix  3919 [003] 20532.573715279: /home/matrix/matrix )          irq_work_interrupt
          matrix  3919 [003] 20532.573715785:  psb offs: 0xb0108
          matrix  3919 [003] 20532.573715785: ([kernel.kallsyms] )          native_write_msr
          matrix  3918 [005] 20532.573715942:  psb offs: 0xa8110
          matrix  3920 [000] 20532.573722041:  psb offs: 0xf4150
          matrix  3917 [001] 20532.573768946: /home/matrix/matrix )             page_fault
          matrix  3917 [001] 20532.573769946: /home/matrix/matrix )             page_fault
          matrix  3920 [000] 20532.573771279: /home/matrix/matrix )                 call_function_interrupt
          matrix  3918 [005] 20532.573771279: /home/matrix/matrix )          call_function_interrupt
          matrix  3918 [005] 20532.573819613: /home/matrix/matrix )          call_function_interrupt
          matrix  3917 [001] 20532.573819613: /home/matrix/matrix )             call_function_interrupt
          matrix  3920 [000] 20532.573819613: /home/matrix/matrix )                 call_function_interrupt
          matrix  3918 [005] 20532.573822946: /home/matrix/matrix )          page_fault
          matrix  3918 [005] 20532.573823946: /home/matrix/matrix )          page_fault
          matrix  3917 [001] 20532.573825279: /home/matrix/matrix )             call_function_interrupt
          matrix  3920 [000] 20532.573825279: /home/matrix/matrix )                 call_function_interrupt
          matrix  3917 [001] 20532.573839156:  psb offs: 0xb4150
          matrix  3918 [005] 20532.573848784:  psb offs: 0xac0d0
          matrix  3920 [000] 20532.573856190:  psb offs: 0xf8110
          matrix  3920 [000] 20532.573938946: /home/matrix/matrix )                 page_fault
          matrix  3920 [000] 20532.573939946: /home/matrix/matrix )                 page_fault
          matrix  3917 [001] 20532.573941613: /home/matrix/matrix )             call_function_interrupt
          matrix  3918 [005] 20532.573941613: /home/matrix/matrix )          call_function_interrupt
```

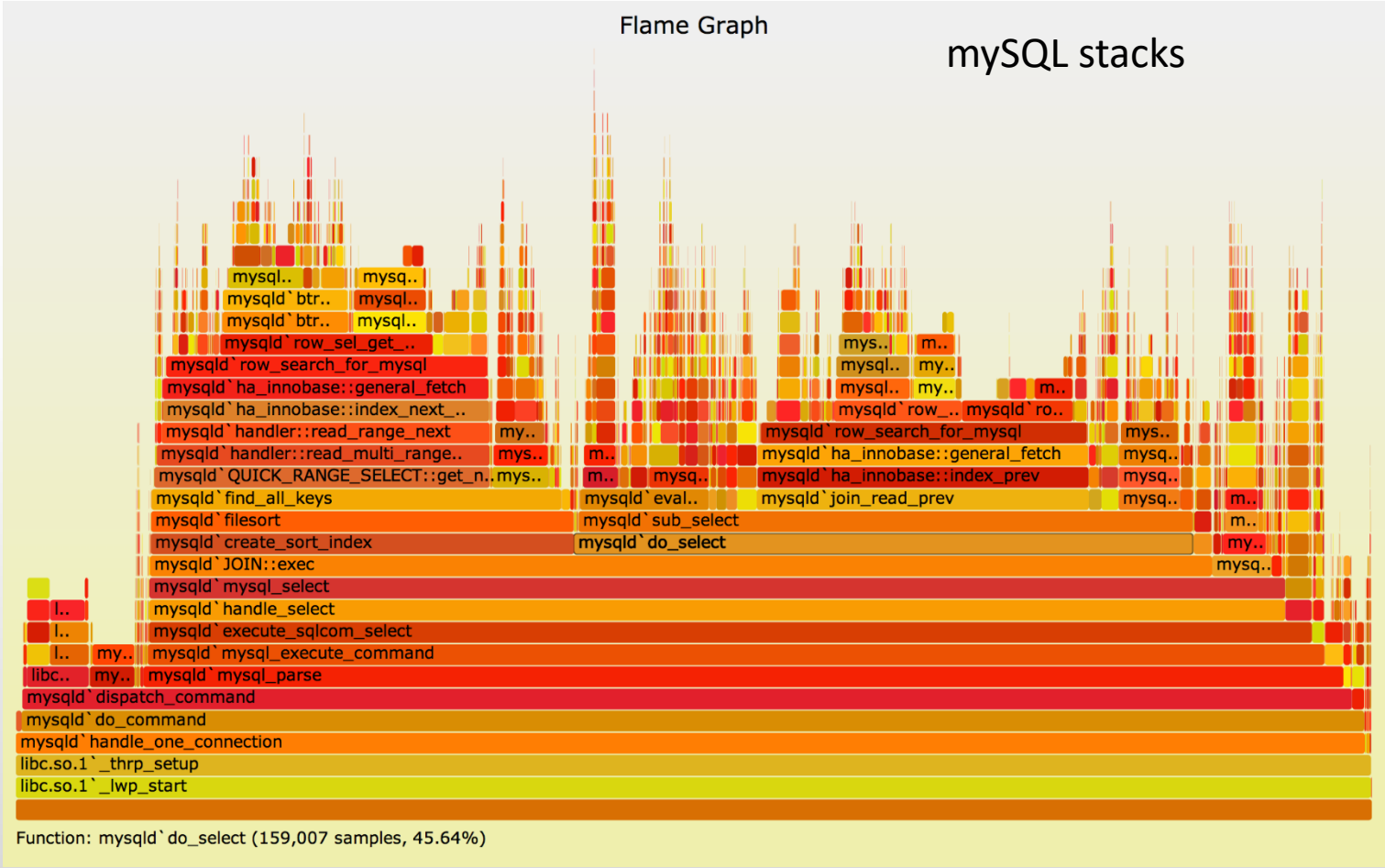
PT flamegraph

- Интерактивный SVG для визуализации стеков
- Подходит для анализа коротких процессов
- Сбор по функциямUser+Kernel
- Поддержка C/C++, java, javascript



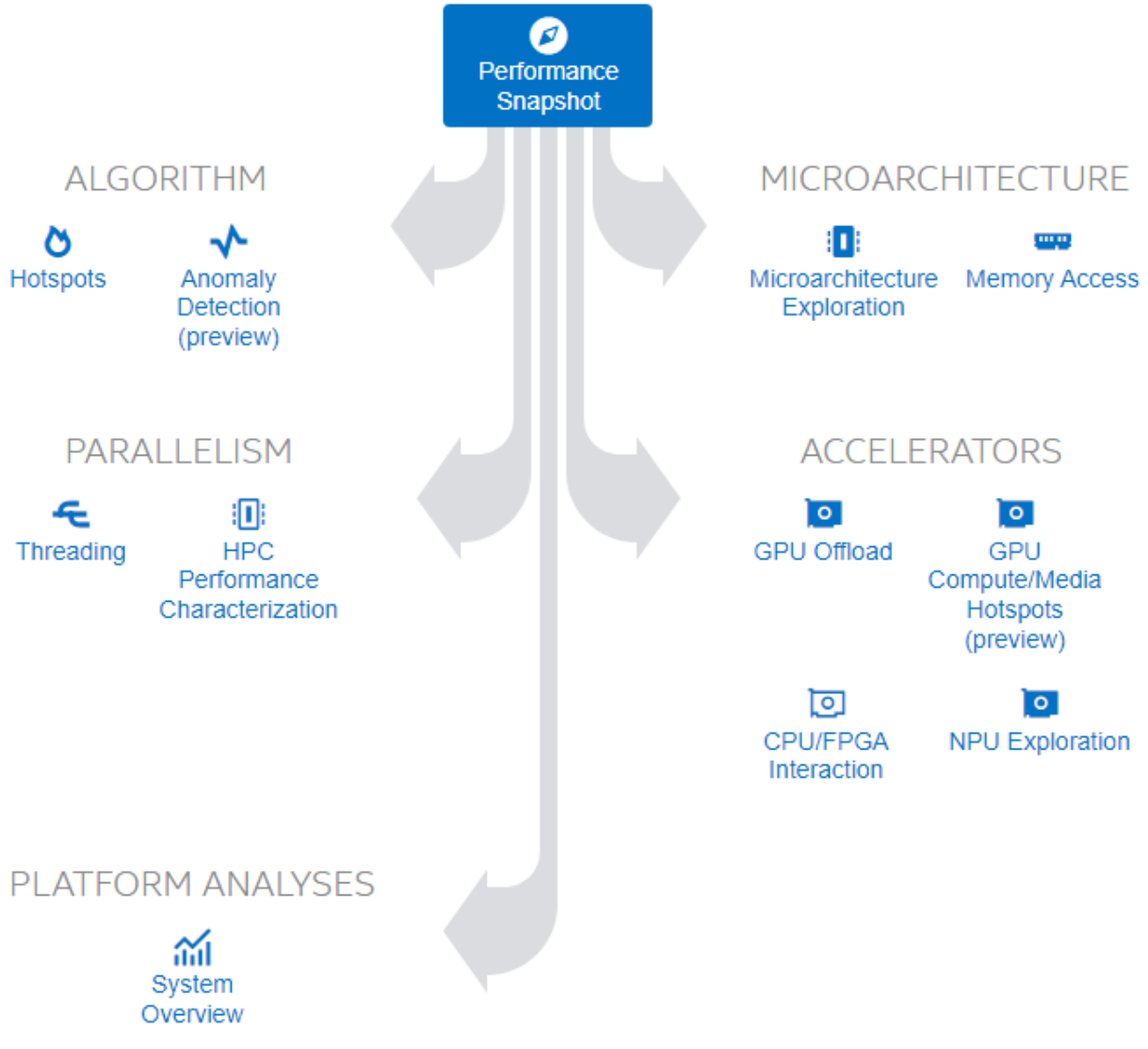
PT flamegraph

- `perf record -e intel_pt// -- app`
- `perf script --itrace=i1usg | ./stackcollapse-perf.pl > workload.folded`
- `./flamegraph.pl workload.folded > workload.svg`



Intel VTune Profiler

- Мощный бесплатный инструмент анализа производительности x86
- Анализ
  - эффективности алгоритмов,
  - параллелизма,
  - микроархитектурных оптимизаций



Anomaly Detection analysis in VTune

- Анализ проблемного процесса
- Требует инструментацию кода

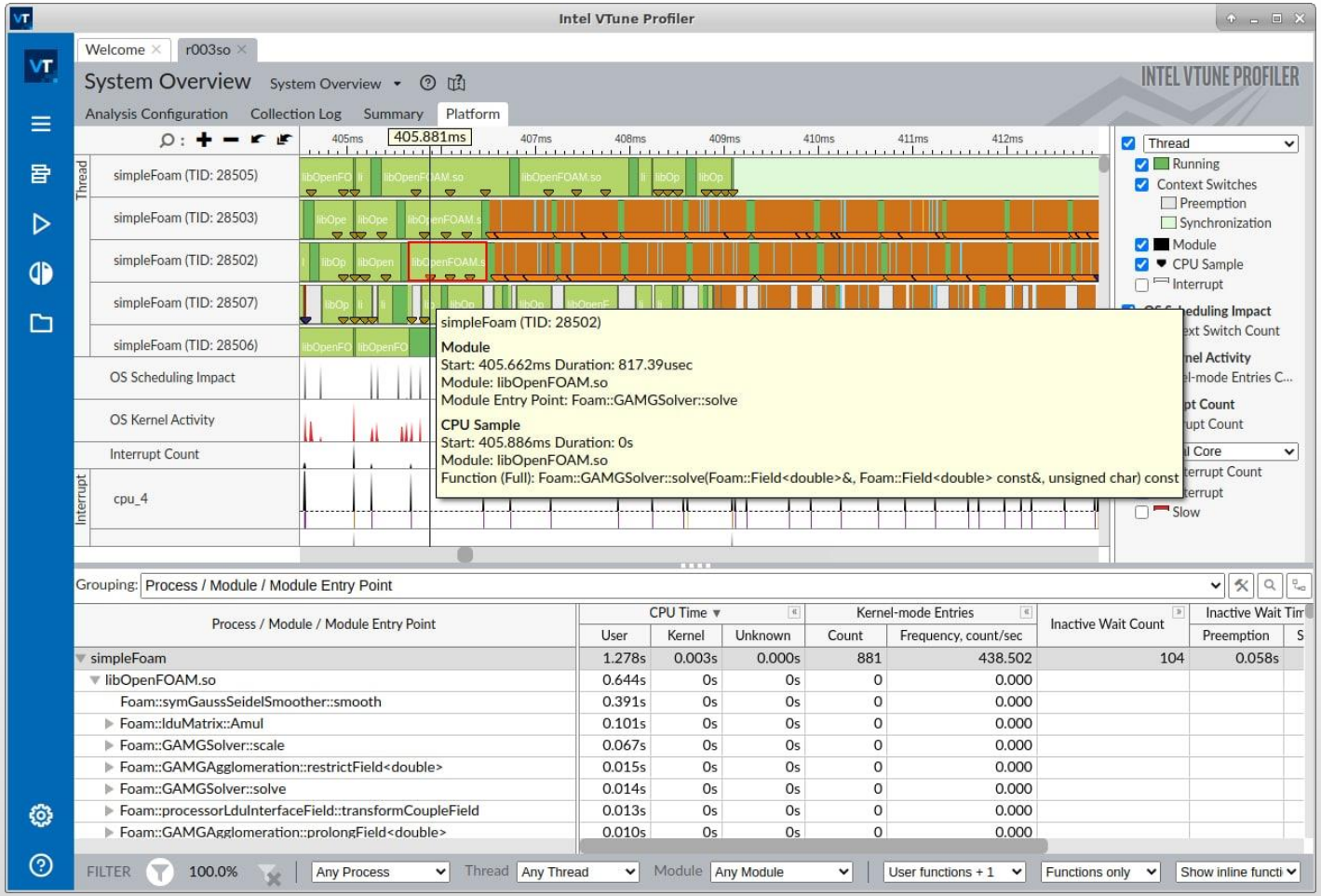
```
__itt_pt_region region =
     __itt_pt_region_create("name");
for(…;…;…) {
   __itt_mark_pt_region_begin(region);
   //… code, processing your task …
   __itt_mark_pt_region_end(region);
}
```

Code Region Of Interest / Code Region Of Interest (Instance) / Function / Call Stack	Instructions Retired	Call Count	Total Iteration Count	Elapsed Time	CPU Time		Wait Time	Inactive Time
					Kernel	User		
core_worker_evloop	20,268,375	329,210	489,045	98.659ms	42.631ms	16.882ms	4.724ms	0ms
25883	3,082	54	64	1.318ms	0.029ms	0.002ms	1.269ms	0ms
60215	3,110	55	65	1.240ms	0.030ms	0.004ms	1.209ms	0ms
276245	3,082	54	64	1.175ms	0.014ms	0.002ms	1.143ms	0ms
498819	3,082	54	64	1.005ms	0.016ms	0.003ms	0.988ms	0ms
558496	448,129	10,543	16,527	1.009ms	0.024ms	1.328ms	0.080ms	0ms
26851	447,762	10,530	16,480	1.057ms	0.014ms	0.769ms	0.035ms	0ms
484307	252,503	3,872	5,682	1.088ms	0.619ms	0.181ms	0ms	0ms
484306	452,450	10,641	16,607	1.049ms	0.020ms	0.750ms	0ms	0ms

Code Region Of Interest / Code Region Of Interest (Instance) / Function / Call Stack	Instructions Retired	Call Count	Total Iteration Count	Elapsed Time	CPU Time		Wait Time	Inactive Time
					Kernel	User		
436053	242,474	3,706	5,450	0.997ms	0.566ms	0.176ms	0ms	0ms
[kernel activity]	173	0	0		0.566ms	0ms	0ms	0ms
↖ write ← tcp_send ← buf_tcp_write ← _worker_event_wi	84	0	0		0.399ms	0ms	0ms	0ms
↖ __read ← tcp_recv ← buf_tcp_read ← _worker_event_r	85	0	0		0.137ms	0ms	0ms	0ms
↖ epoll_pwait ← [Loop at line 218 in event_wait] ← event_	1	0	0		0.024ms	0ms	0ms	0ms
↖ [Loop at line 128 in hashtable_get] ← hashtable_get ← i	1	0	0		0.005ms	0ms	0ms	0ms
↖ __tz_convert ← _klog_write ← [Loop at line 786 in twem	1	0	0		0.002ms	0ms	0ms	0ms
↖ clock_gettime ← _gettime ← duration_snapshot ← time_	1	0	0		0.000ms	0ms	0ms	0ms

System Overview HW Tracing in Vtune Profiler

- Анализ поведения всей системы, с разбивкой по процессам, тредам, модулям
- Активность прерываний по ядрам
- Таймлайн исполнения приложения с точностью до модуля



Вместо заключения

CFP FPGA Conference

- Intel Processor Trace мощный инструмент анализа производительности
  - ARM Coresight ETM
  - RISC-V N-Trace & E-Trace

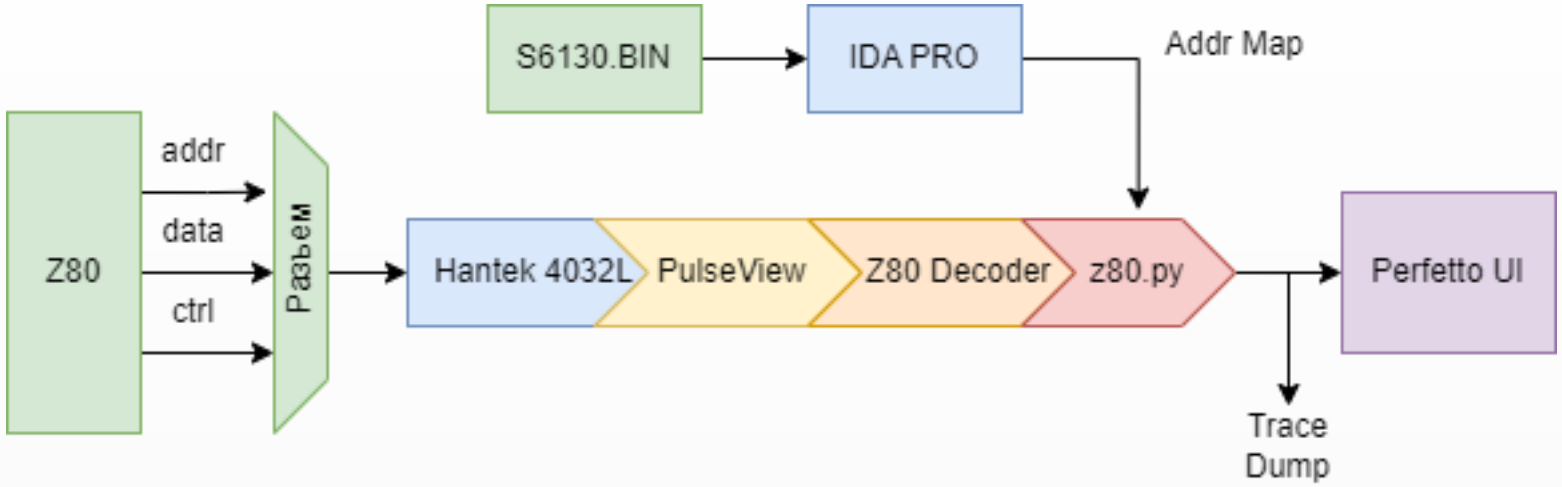
} Ваши друзья на других платформах

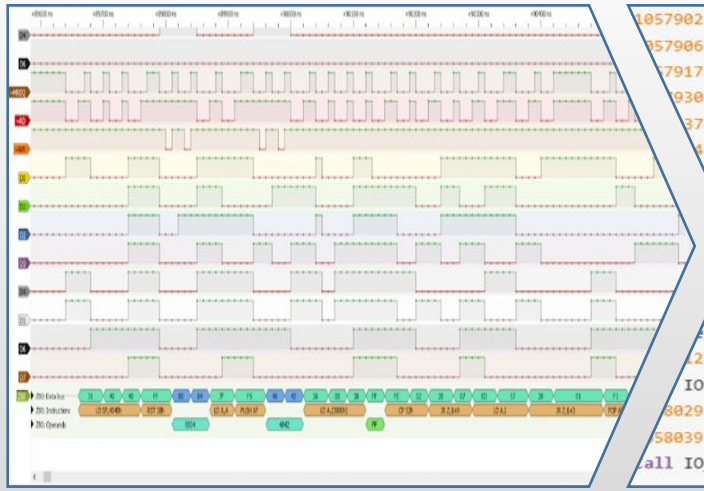
- Linux perf + flamegraph - инструменты, которые всегда под рукой
- Больше трассировки с Anomaly Detection & System Overview HW Tracing Intel VTune Profiler 2022.4

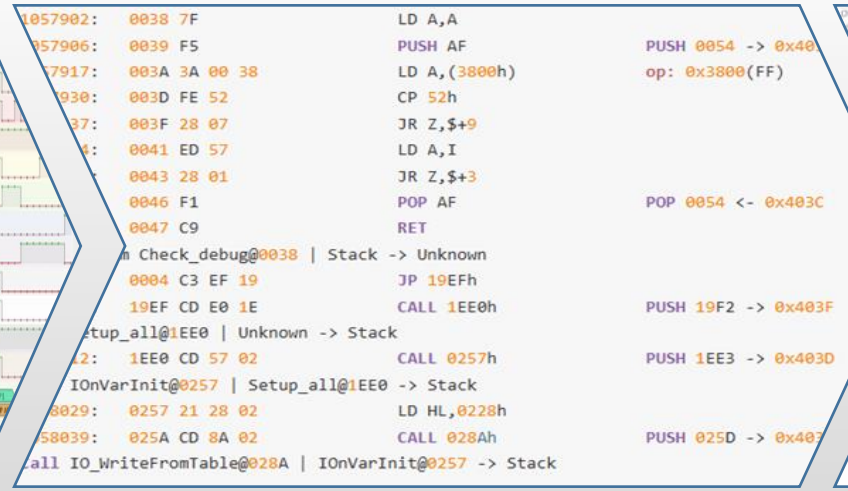
Useful Links

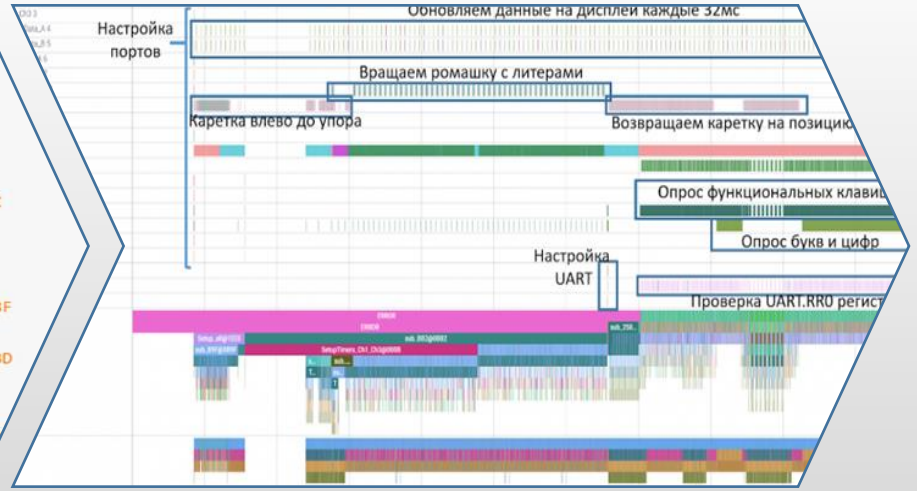
- https://www.brendangregg.com/perf.html
- Performance Monitoring in the Intel® Software Developer Manual, Volume 3B, Chapter 18
- https://perfmon-events.intel.com/
- https://easyperf.net/blog/2019/02/09/Top-Down-performance-analysis-methodology
- https://man7.org/linux/man-pages/man1/perf-intel-pt.1.html
- https://www.brendangregg.com/FlameGraphs/cpuflamegraphs.html
- https://www.intel.com/content/www/us/en/develop/documentation/vtune-help/top/analyze-performance/algorithm-group/anomaly-detection-analysis.html
- https://github.com/intel/ittapi
- https://www.brendangregg.com/FlameGraphs/cpuflamegraphs.html
- https://halobates.de/blog/p/329
- https://github.com/brendangregg/FlameGraph.git

Instruction Tracing on Z80









Collect

Decode

Visualize

Диаграмма старта машинки