KVM Forum
2019

Efficient Performance Monitoring in the Cloud with Virtual Performance Monitoring Units (PMUs)

Contributors

Wei Wang, Kan Liang, Like Xu, Andi Kleen, Guang Zeng, Danmei Wei

Presenter

Sean Christopherson

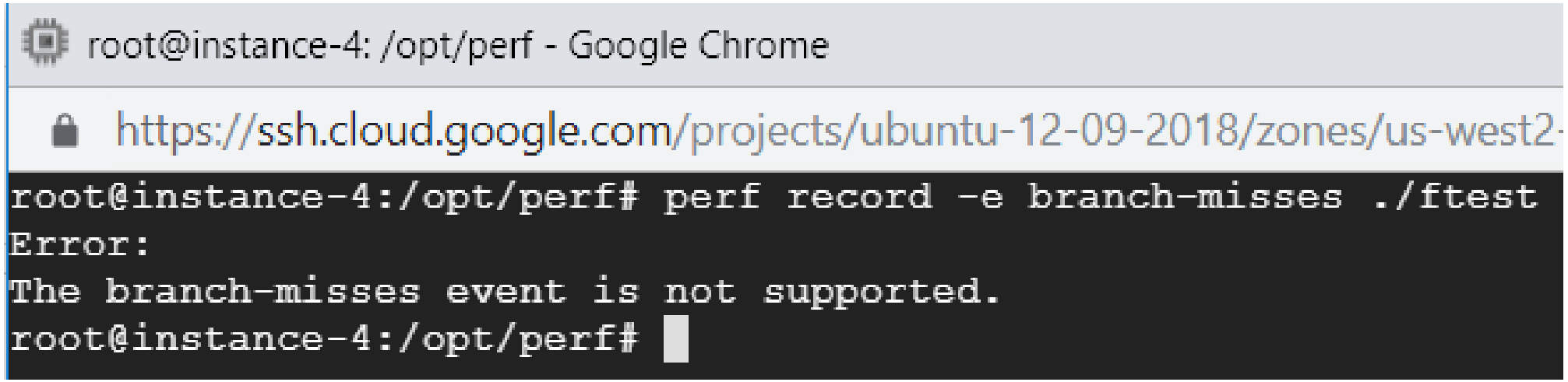
Agenda

- Project Goals
- Background
- Our Solutions
- Test Results
- Current Status
- Future Works

Part 1: Project Goals

Project Goals

Virtual PMUs are usually disabled in today's clouds



*Other names and brands may be claimed as the property of others.

Project Goals

Virtual PMUs are usually disabled in today's clouds

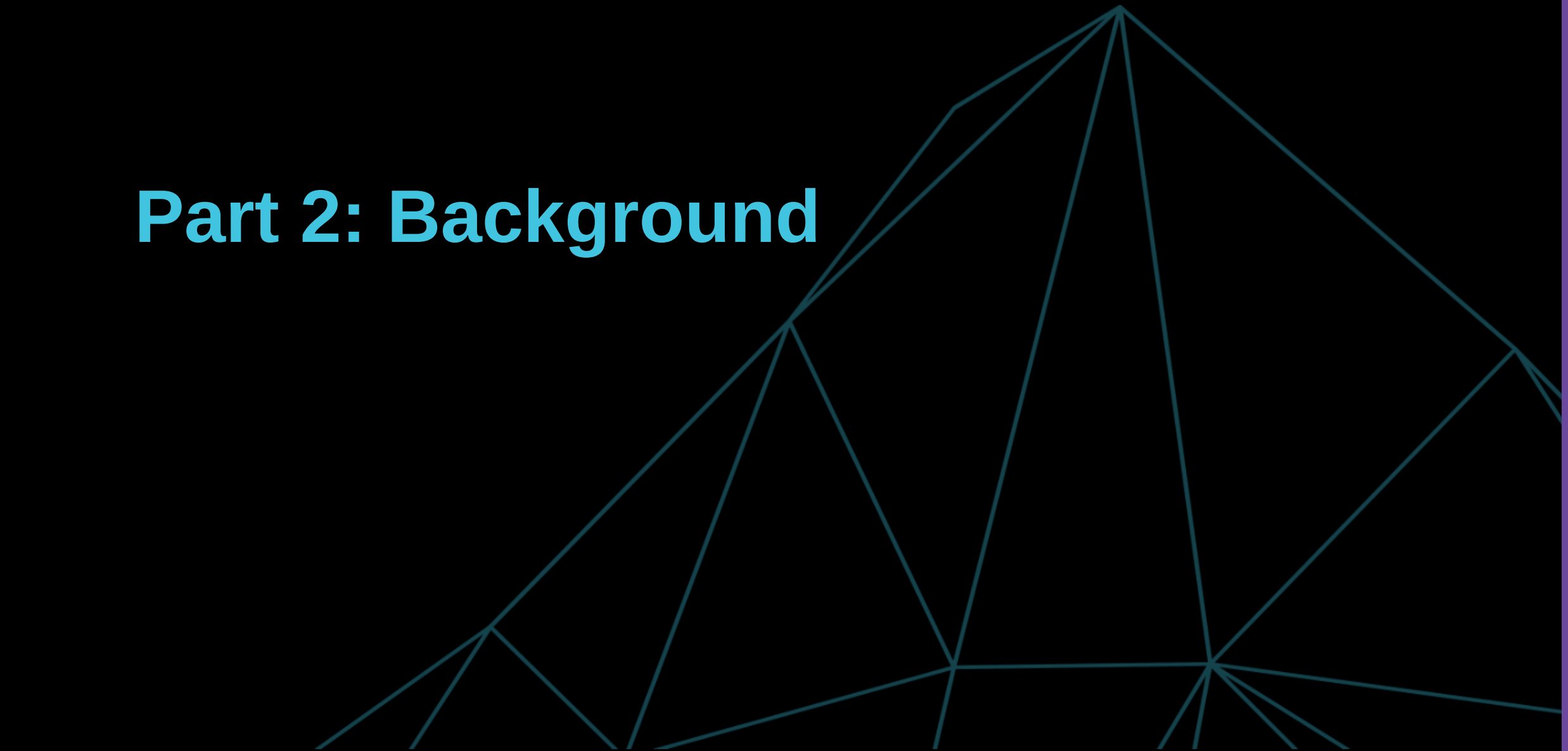
- inaccurate profiling results
- lack of advanced PMU features (e.g., LBR and PEBS)

Many cloud vendors (e.g., Google*, Alibaba*, Tencent*, Huawei*, Baidu*) have a strong interest in making PMUs usable in their cloud productions

What we did

- Reduced PMU virtualization overhead to generate more accurate profiling results
- Added support for LBR and PEBS virtualization in KVM

Part 2: Background



Performance Monitoring Units

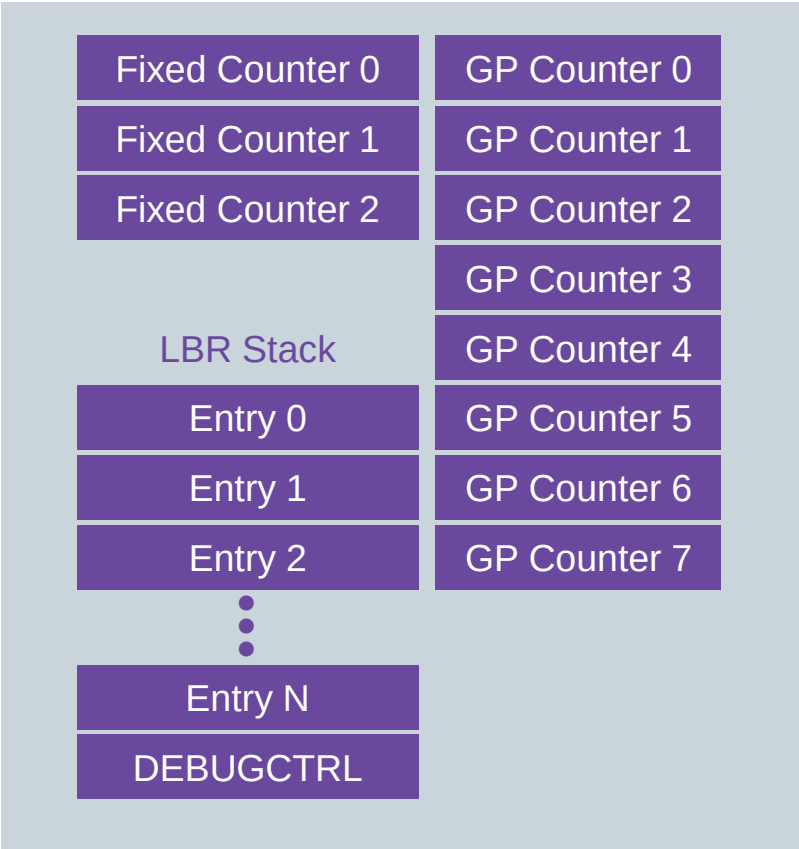
Each **Fixed Function Counter** counts a specific event

- Fixed counter 0: Instruction retired
- Fixed counter 1: Unhalted core cycles
- Fixed counter 2: Reference cycles

Last Branch Records

- Stack of MSRs that records branch sources and destinations
- Enabled via DEBUGCTRL MSR
- Usually takes a PMU counter to do branch sampling

Each logical CPU has its own **PMU**



General Purpose Counters can be configured to count any supported event

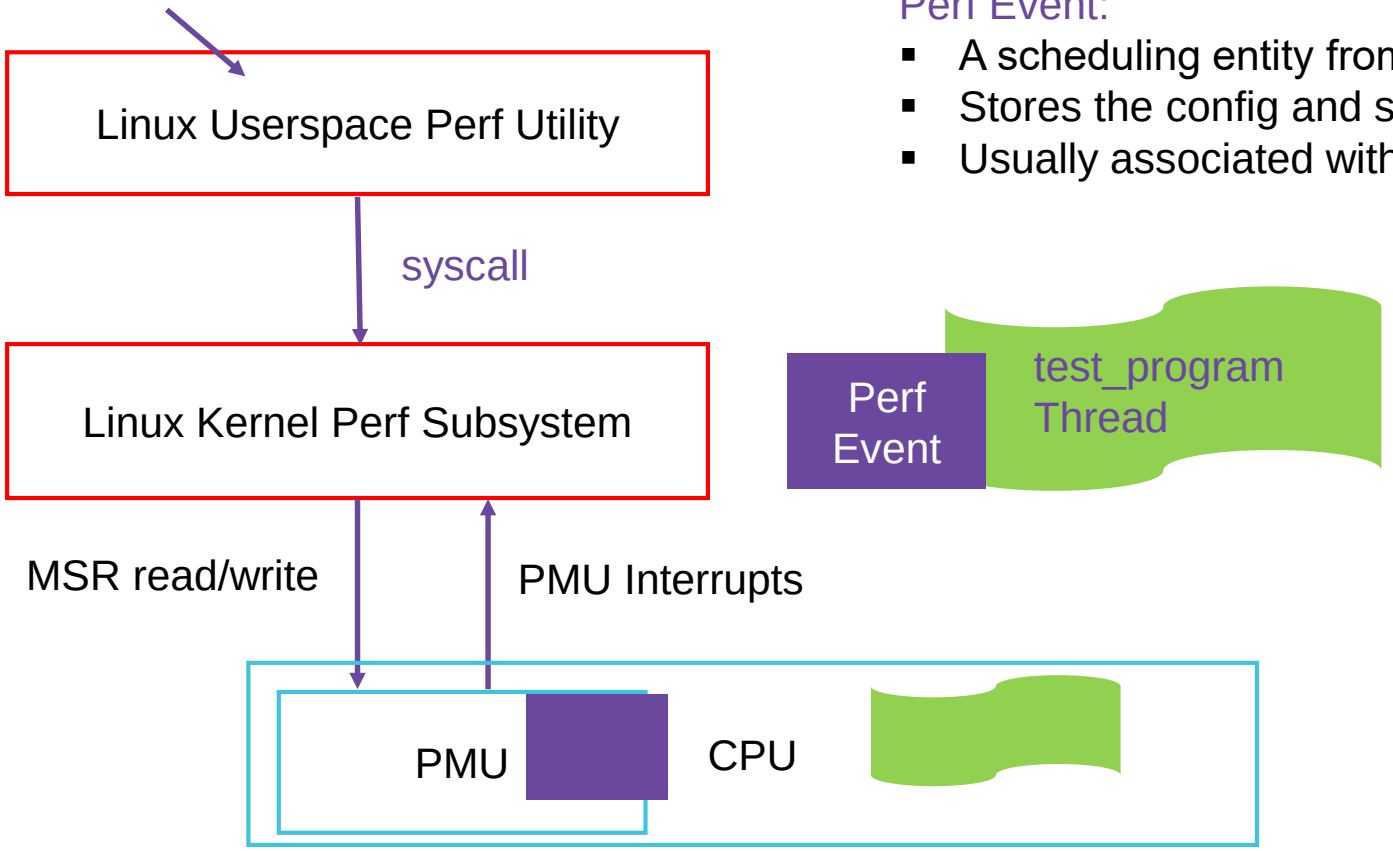
- Unhalted core cycles
- Instruction retired
- Branch instruction retired
- …

PMU can be configured to generate **Performance Monitoring Interrupts** after N events

- Sampling
- Histograms
- …

PMU Usage in Native Linux*

$ perf record -e branch-misses ./test_program

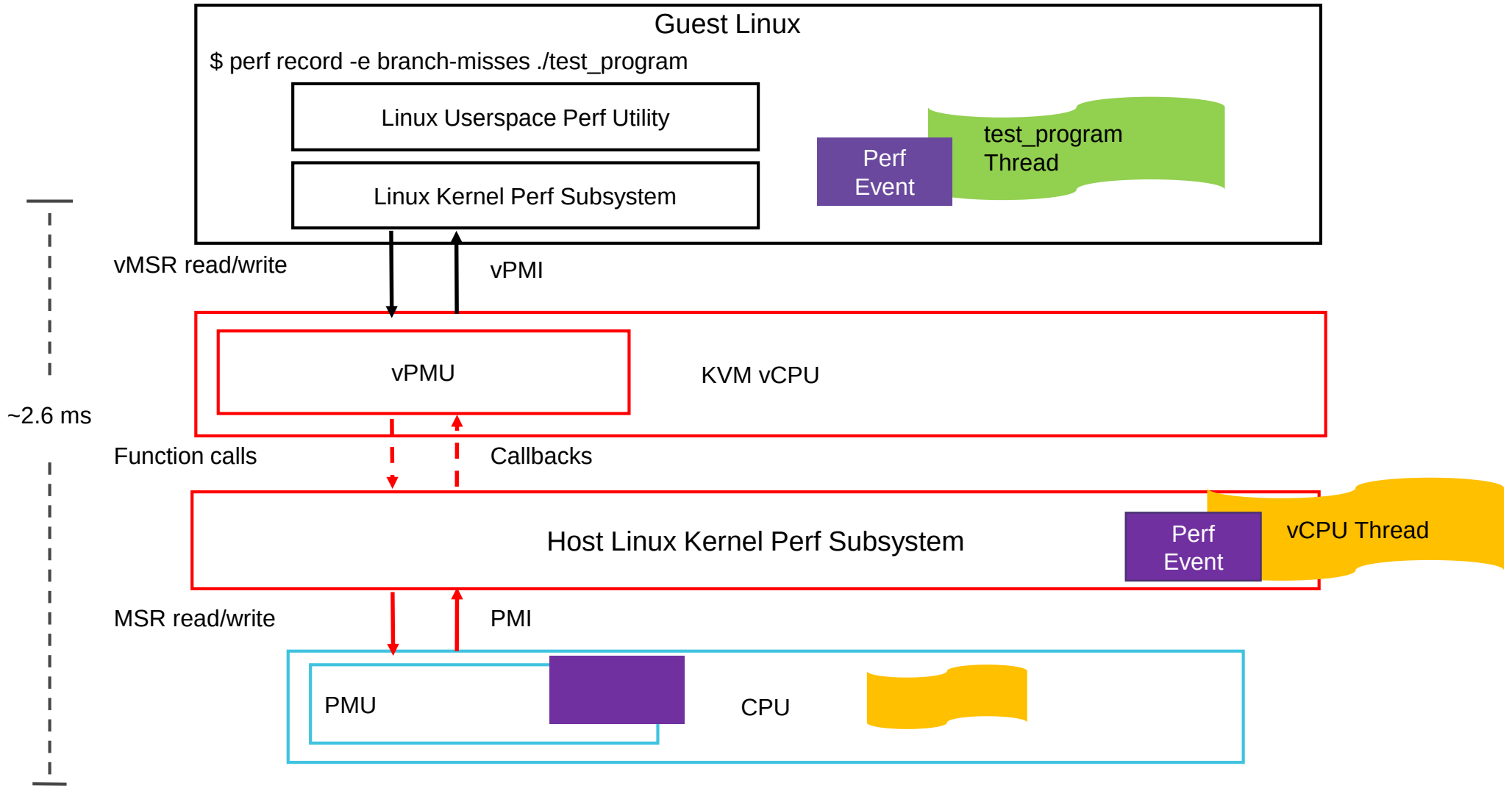


Perf Event:

- A scheduling entity from PMU's point of view
- Stores the config and state data
- Usually associated with one or more PMU counters

*Other names and brands may be claimed as the property of others.

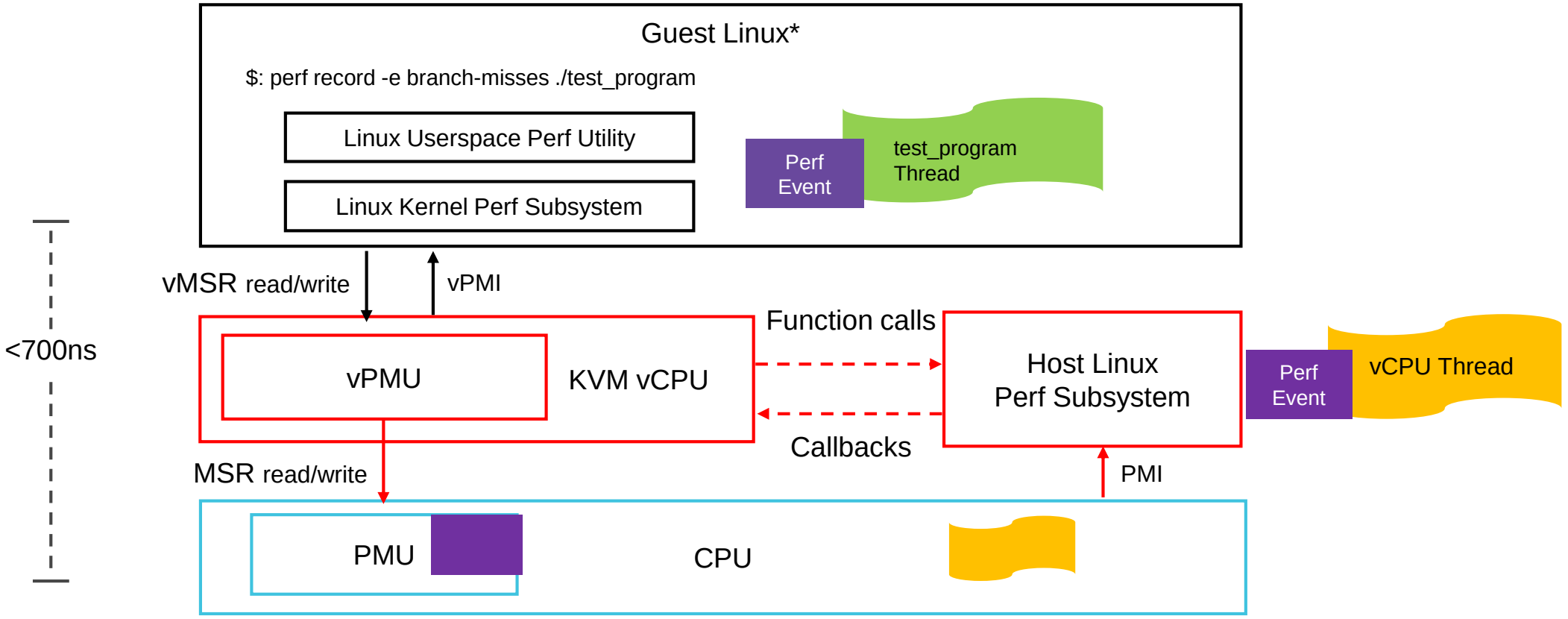
PMUs Usage in Linux* KVM Guest



*Other names and brands may be claimed as the property of others.

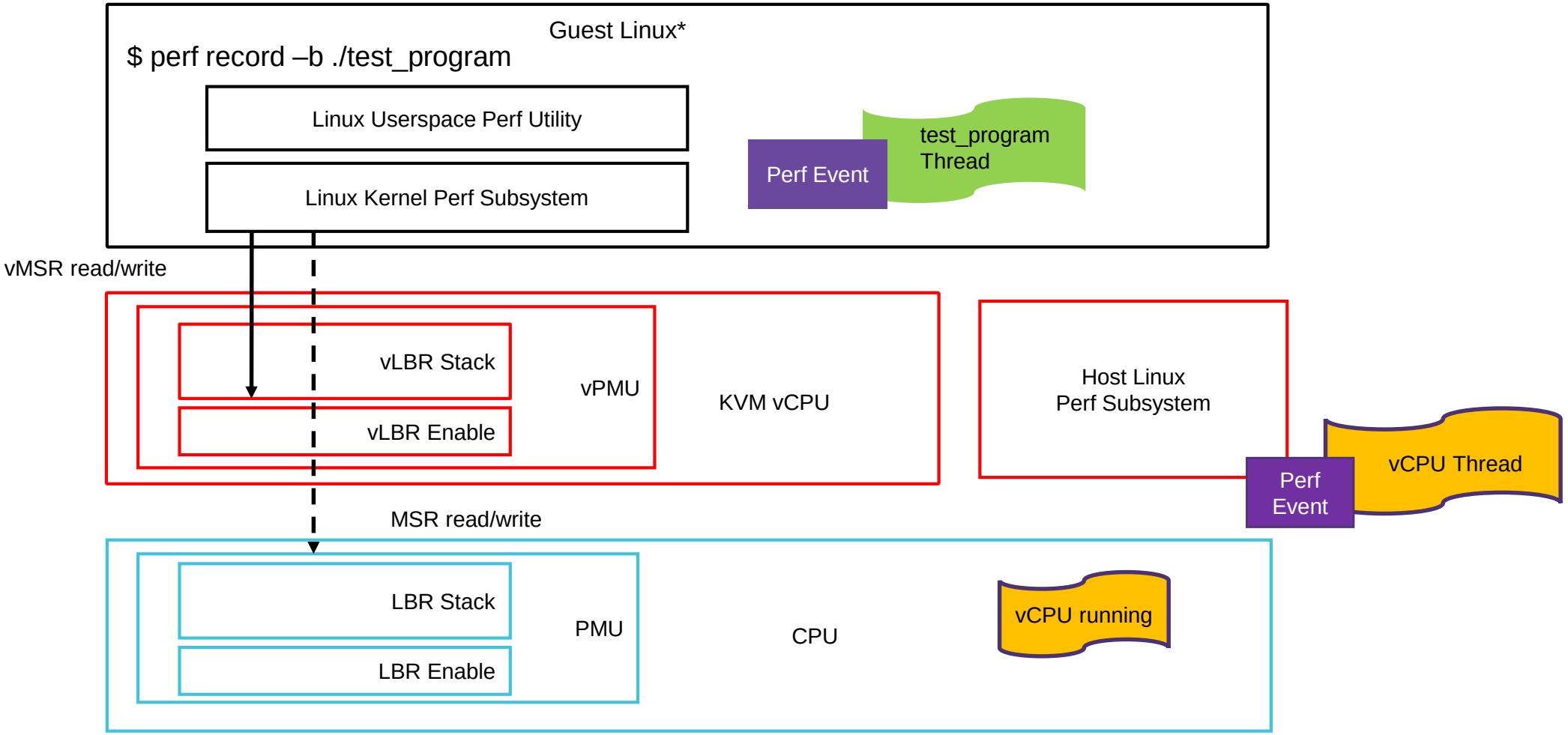
Part 3: Our Solutions

vPMU Working Model Optimization



*Other names and brands may be claimed as the property of others.

Last Branch Records (LBR) Virtualization



*Other names and brands may be claimed as the property of others.

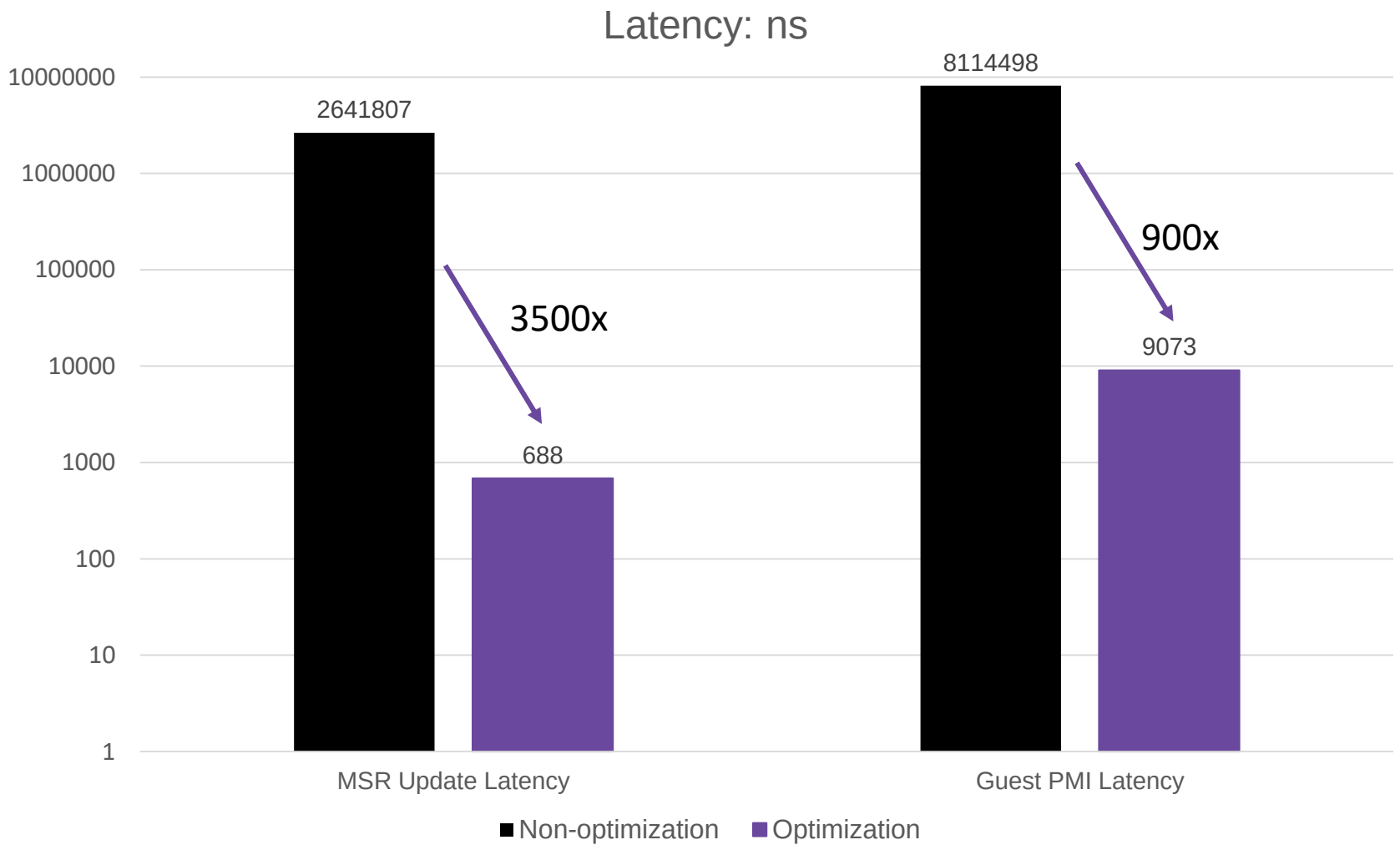
Part 4: Test Results

Test Environment

- CPU: Intel® Xeon® Processor E5-2699 v4 @ 2.20GHz
- Host and Guest Kernel: Linux* 4.19.0, booted with “nowatchdog”
- VM Configuration: 4 vCPUs, 8G memory

*Other names and brands may be claimed as the property of others.

Latency Comparison (Logarithmic)



Branch Miss Sampling

$ perf record -e branch-misses ./test_program

Test on Host:

```
48.19%  ftest   ftest            [.] main
21.21%  ftest   ftest            [.] bar
18.44%  ftest   ftest            [.] foo
11.64%  ftest   ftest            [.] qux
 0.40%  ftest   libc-2.23.so     [.] __random
 0.02%  ftest   libc-2.23.so     [.] __random_r
...
```

Test in Guest with optimization:

```
46.18%  ftest   ftest            [.] main
22.29%  ftest   ftest            [.] bar
20.29%  ftest   ftest            [.] foo
10.47%  ftest   ftest            [.] qux
 0.36%  ftest   libc-2.23.so     [.] random
 0.24%  ftest   libc-2.23.so     [.] random_r
...
```

Test in Guest without optimization:

```
82.64%  ftest   [kernel.kallsyms]  [k] perf_event_task_tick
 6.15%  ftest   [kernel.kallsyms]  [k] scheduler_tick
 4.62%  ftest   [kernel.kallsyms]  [k] trigger_load_balance
 2.20%  ftest   [kernel.kallsyms]  [k] raise_softirq
 1.32%  ftest   [kernel.kallsyms]  [k] nohz_balance_exit_idle
 0.66%  ftest   [kernel.kallsyms]  [k] run_posix_cpu_timers
...
```

Perf run doesn't complete due to the large vPMU overhead
Results gathered via stopping the run via "ctrl-c"

Last Branch Recording (LBR) Tests

$ perf record --call-graph lbr ./ftest

	Children	Self	Command	Shared Object	Symbol
+	99.99%	0.00%	ftest	ftest	[.] _start
+	99.99%	0.00%	ftest	libc-2.23.so	[.] __libc_start_main
+	99.99%	13.27%	ftest	ftest	[.] main
+	39.73%	36.65%	ftest	ftest	[.] qux
+	38.72%	16.57%	ftest	ftest	[.] bar
+	29.44%	10.73%	ftest	ftest	[.] foo
+	20.71%	8.37%	ftest	libc-2.23.so	[.] __random
+	13.39%	12.97%	ftest	libc-2.23.so	[.] __random_r
+	9.23%	1.32%	ftest	ftest	[.] random@plt

Host Results

	Children	Self	Command	Shared Object	Symbol
+	100.00%	0.00%	ftest	ftest	[.] _start
+	99.99%	0.00%	ftest	libc-2.23.so	[.] __libc_start_main
+	99.99%	13.23%	ftest	ftest	[.] main
+	38.51%	16.57%	ftest	ftest	[.] foo
+	37.32%	37.26%	ftest	ftest	[.] qux
+	27.69%	12.28%	ftest	ftest	[.] bar
+	19.09%	9.43%	ftest	libc-2.23.so	[.] random
+	10.88%	1.44%	ftest	ftest	[.] random@plt
+	9.66%	9.65%	ftest	libc-2.23.so	[.] random_r

Guest Results

Part 5: Current Status

Current Status

- vPMU optimization
  - https://lkml.org/lkml/2018/11/1/937 (full optimization, NAK'd)
  - https://lkml.org/lkml/2019/10/27/834 (intermediate step)
- LBR
  - https://lkml.org/lkml/2019/8/6/215
- PEBS
  - https://lkml.org/lkml/2019/10/27/53

Part 6: Future Works

Future Works

- Continue to upstream the patches
- Support arch v5 PMU features

Q&A

Thank You!

Disclaimers

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

The products and services described may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request. No product or component can be absolutely secure.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting www.intel.com/design/literature.htm

Intel, the Intel logo, and Xeon are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others