EBPF

Observability In Deep

Kabilesh P R
Co-Founder, Mydbops

October 3rd, 2020
Mydbops Database Meetup -7

Mydbops

About Me

- Interested in Open Source technologies
- Building Solutions on MySQL, Cloud & MongoDB
- Active Tech Speaker/Blogger
- Co-Founder, Mydbops

Kabilesh PR

Mydbops

Mydbops Services

Focuses on Top Opensource database MySQL, MongoDB and PostgreSQL

Mydbops

Our Clients

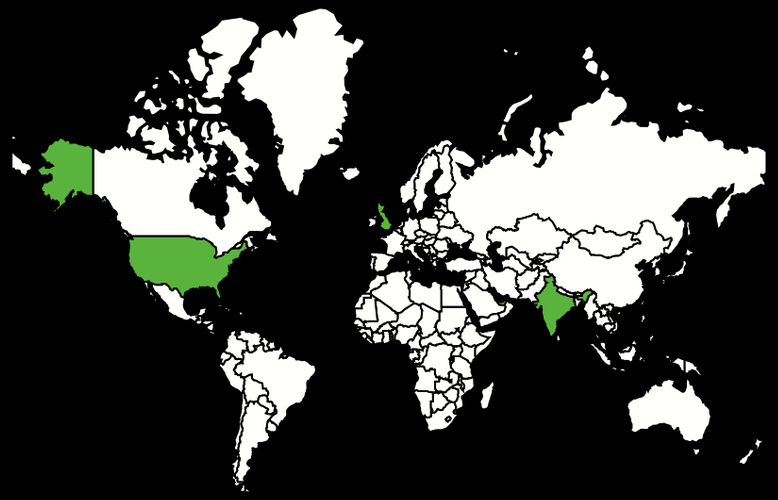
400 + Clients In 4 Yrs. of Operations

Flipkart

SWIGGY

yulu

NYKAA



DREAM11

Mydbops

Agenda

BPF & EBPF

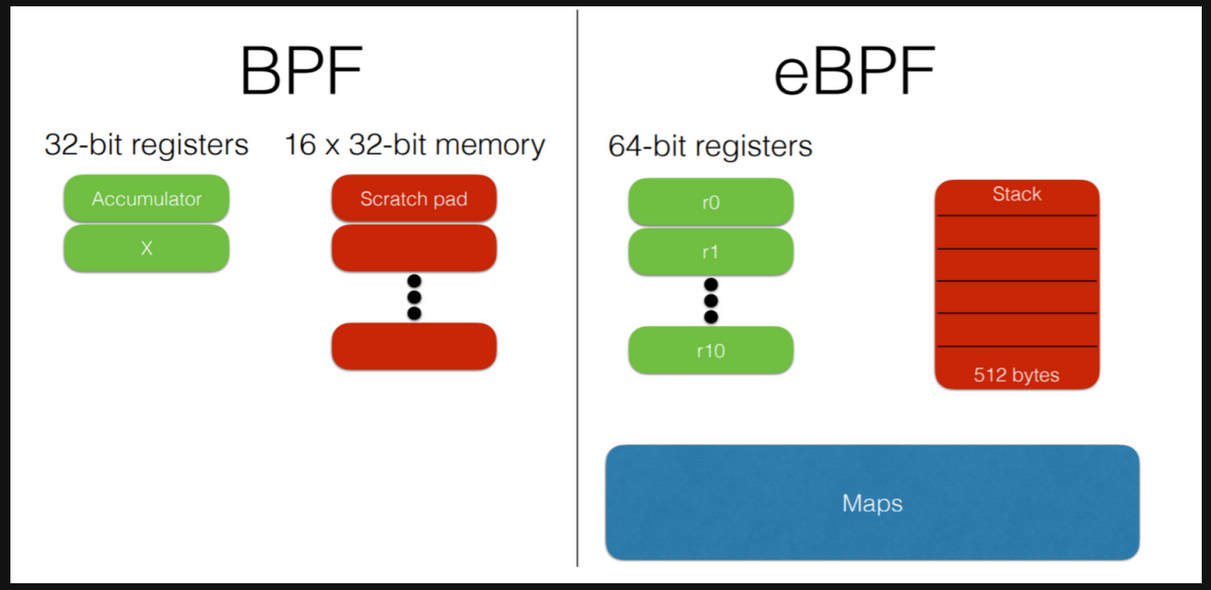
BPF Internals

Tools in action

Mydbops

BPF Vs EBPF

- BPF(Berkely Packet Filter) an "in-kernal VM" for packet filter developed in 1992 by Steven McCanne and Van Jacobson
- Introduced with the Linux Kernel 2.1.75 in 1997
- EBPF(Extended BPF), has been there from 2014 with LinuX kernel from version 3.15...
- It's no more a acronym, Since it not only does packet filtering



Mydbops

Pre BPF Analysis

- Mostly based on the procfs (/proc) -- post-process manner

1. uptime
2. dmesg -T | tail
3. vmstat 1
4. mpstat -P ALL 1
5. pidstat 1
6. iostat -xdz 1
7. free -m
8. sar -n DEV 1
9. sar -n TCP,ETCP 1
10. top

Mydbops

What is BPF

eBPF is a revolutionary technology that can run sandboxed programs in the Linux kernel without changing kernel source code or loading kernel modules. By making the Linux kernel programmable, infrastructure software can leverage existing layers, making them more intelligent and feature-rich without continuing to add additional layers of complexity to the system (Dynamic tracing)

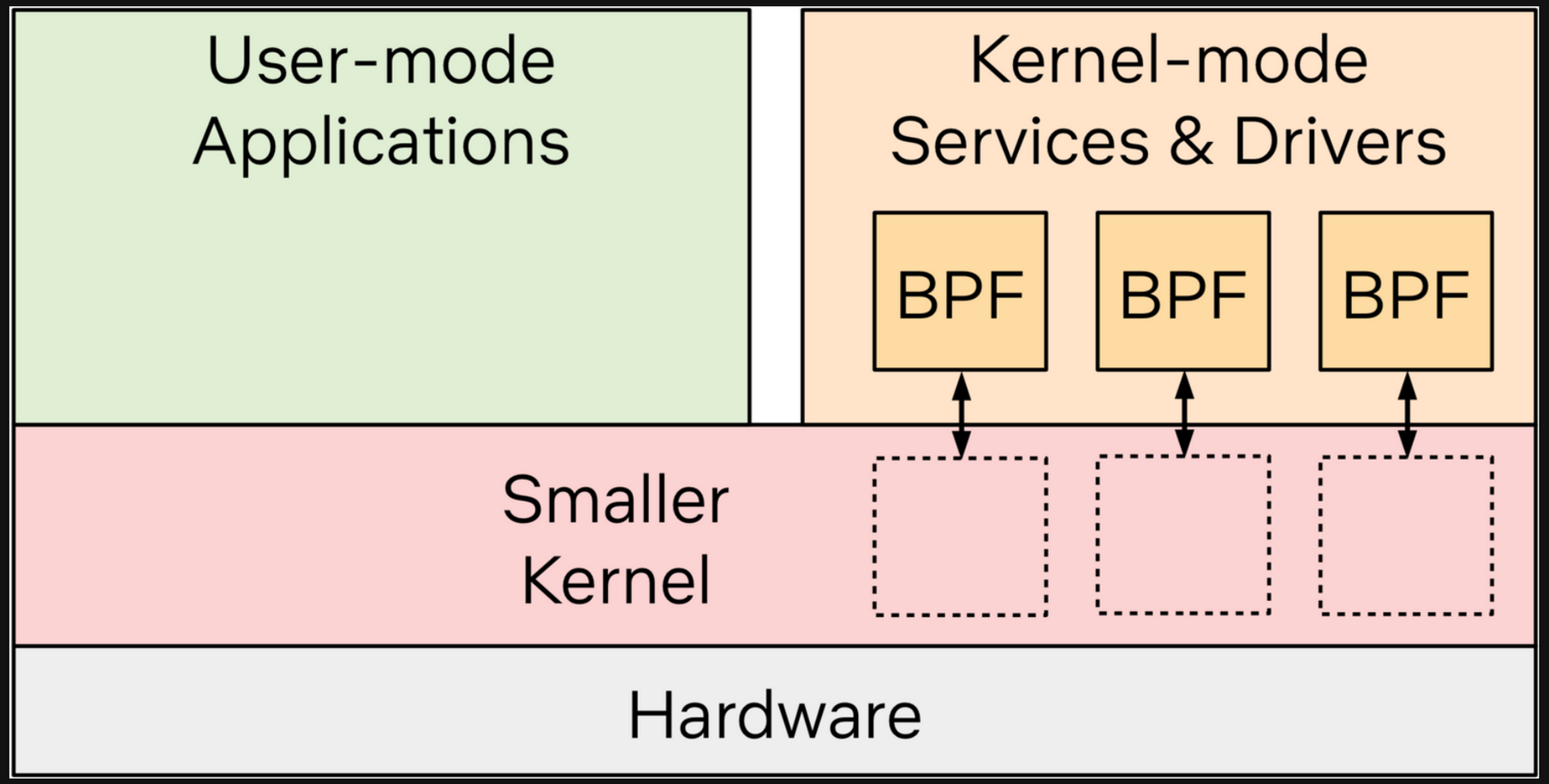
A user can interact with the eBPF kernel using a bpf() system call whose prototype is:

```
int bpf(int cmd, union bpf_attr *attr, unsigned int size);
```

eBPF can prepare user information in kernel context and transfer only needed information to user space. So far, support of kprobes, tracepoints, uprobes and perf_events filtering using eBPF have been implemented in the upstream kernel

Mydbops

Microkernel model

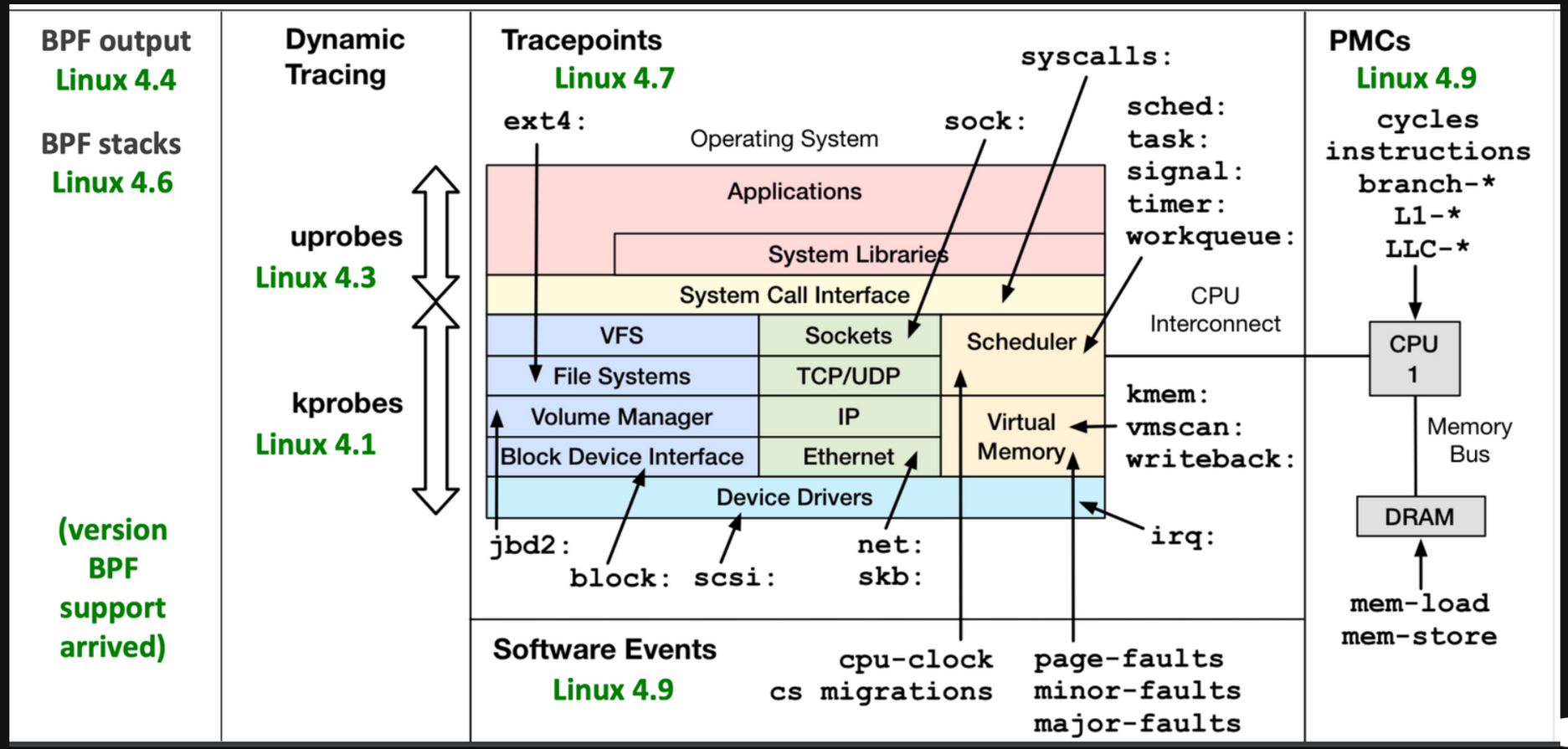


update to modify the data).

Mydbops

Linux Kernel support for eBPF

Development is being actively done later is better



Mydbops

Improvements with Kernel Versions

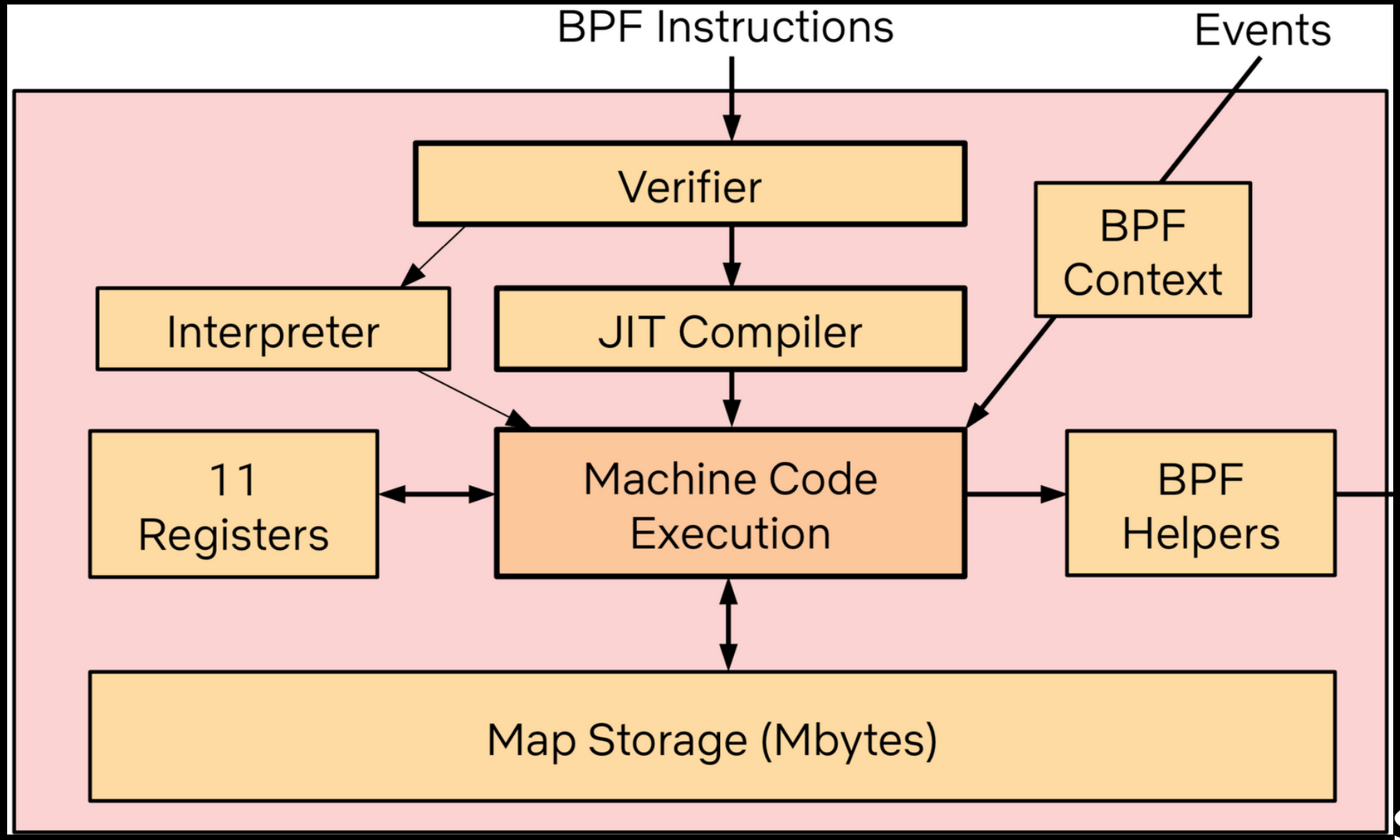
bpf2bpf function calls	4.16	`cc8b0b92a169`
BPF used for monitoring socket RX/TX data	4.17	`4f738adba30a`
BPF attached to raw tracepoints	BPF attached to `bind()` system call	4.17 `4fbac77d2d09`
BPF attached to `bind()` system call	BPF Type Format (BTF)	4.18 `69b693f0aefa`
BPF Type Format (BTF)	AF_XDP	4.18 `fbfc504a24f5`
AF_XDP	bpfilter	4.18 `d2ba09c17a06`
bpfilter	End.BPF action for seg6local LWT	4.18 `004d4b274e2a`
End.BPF action for seg6local LWT	BPF attached to LIRC devices	4.18 `f4364dcfc86d`
BPF attached to LIRC devices	BPF socket reuseport	4.19 `2dbb9b9e6df6`
	BPF flow dissector	4.20 `d58e468b1112`
	BPF cgroup sysctl	5.2 `7b146cebe30c`
	BPF raw tracepoint writable	5.2 `9df1c28bb752`

https://github.com/iovisor/bcc/blob/master/docs/kernel-versions.md

Mydbops

BPF Internals

BPF Internals

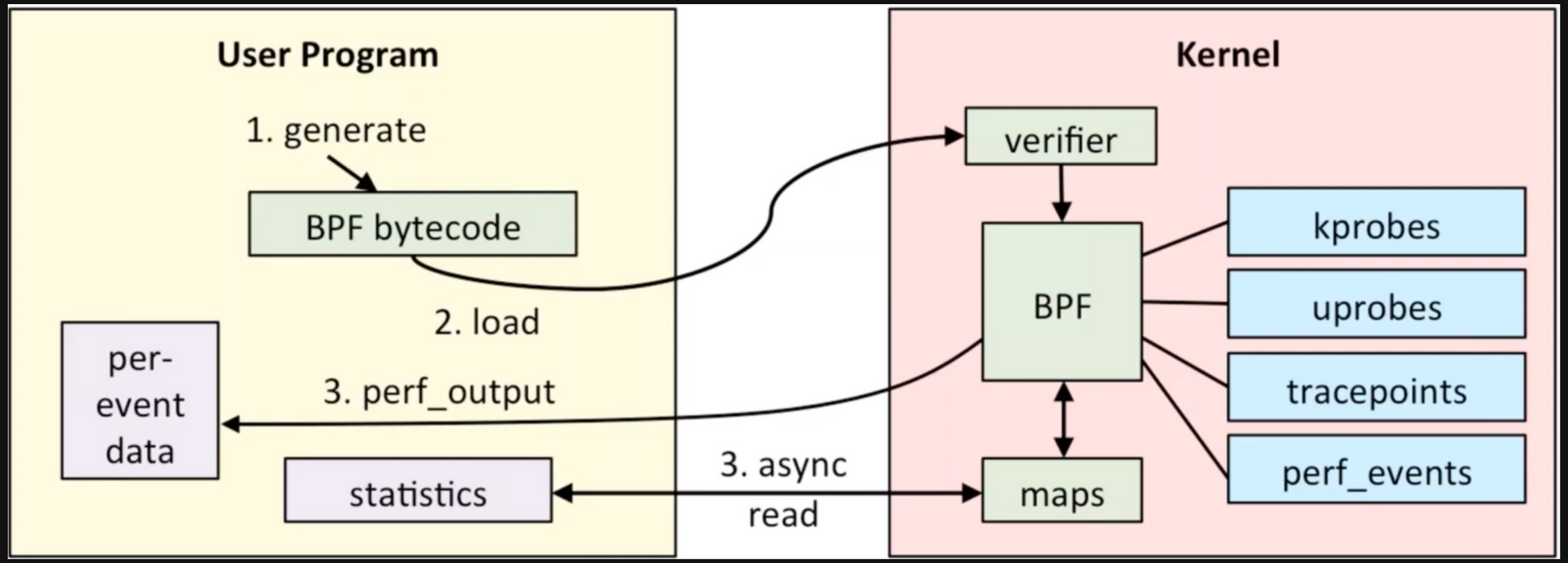


Mydbops

BPF Internals

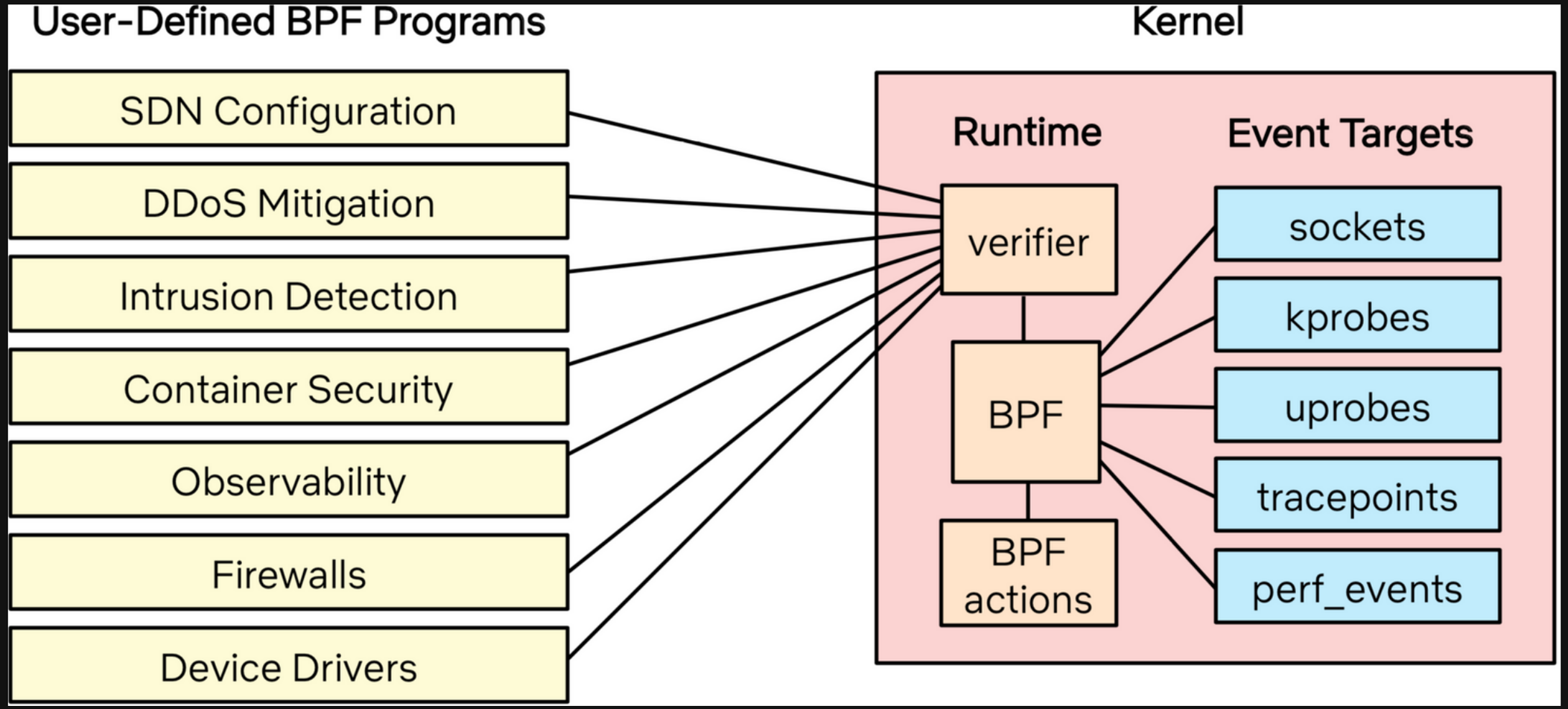
- BPF instruction are verified against end-loop and sanity (Verifier)
- Interpretor/Compiler uses GCC/LVM compiler
- Just-in-time (JIT) compiler translates eBPF bytecode into a host system's assembly code
- Registers to store instructions
- eBPF uses map as generic key/value data structure for data transfer between Kernel and user space

BPF Internals



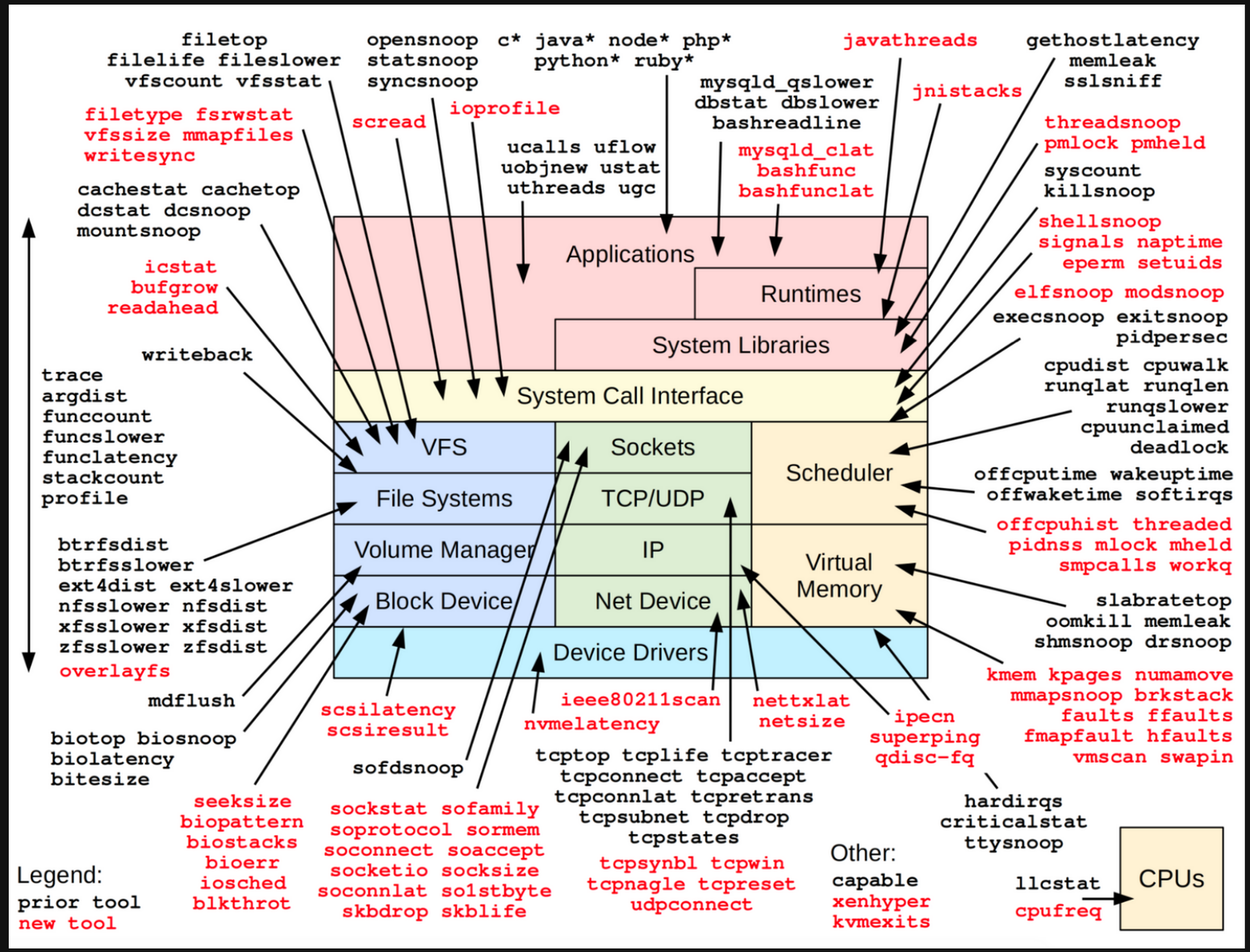
Mydbops

Current State of BPF



Mydbops

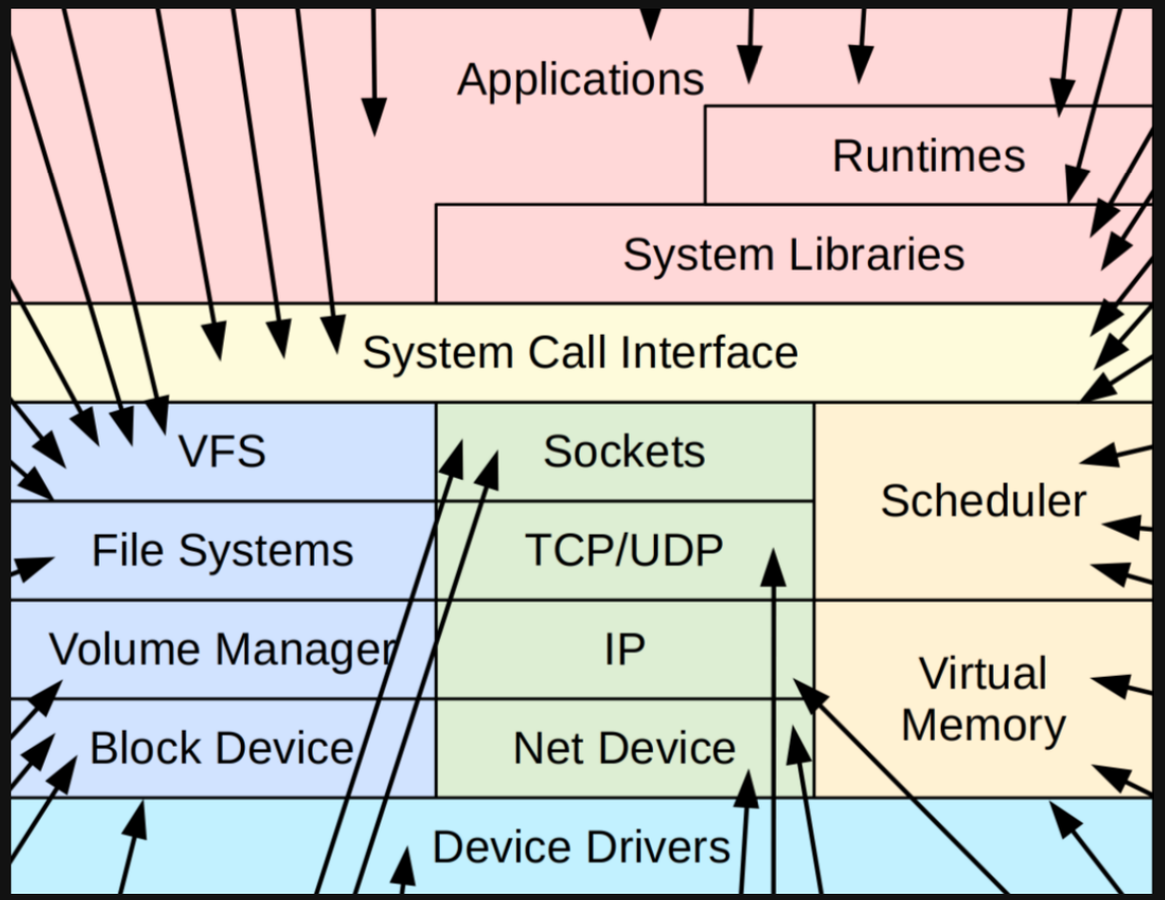
Tooling's around BPF



Mydbops

Closer view of Sub-Systems

- **CPU (Scheduling)**
- **Memory**
- **Disks**
- **File Systems**
- **Networking**
- **Applications**
- **Kernel**
- **Hypervisors**
- **Containers**



Mydbops

Installation

- Newer Kernal is better, Great to have > 4.4
- https://github.com/iovisor/bcc/blob/master/INSTALL.md (choose according to the version)

Ubuntu Package

Ubuntu Packages Source packages and the binary packages produced from them can be found at packages.ubuntu.com.

```
#sudo apt-get install bpfcc-tools linux-headers-$(uname -r)
```

The tools are installed in /sbin (/usr/sbin in Ubuntu 18.04) with a -bpfcc extension. Try running sudo opensnoop-bpfcc.

Mydbops

Installation

Iovisor:

BPF Compiler Collection (BCC). BCC is a toolkit for creating efficient kernel tracing and manipulation programs, and includes several useful tools

Installation Ubuntu

```
#sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys
4052245BD4284CDD
#echo "deb https://repo.iovisor.org/apt/$(lsb_release -cs) $(lsb_release
-cs) main" | sudo tee /etc/apt/sources.list.d/iovisor.list
#sudo apt-get update
#sudo apt-get install bcc-tools libbcc-examples linux-headers-$(uname -
r)
```

Installation

Installation Centos8

With Centos8, the package comes with the default repo

#yum install bcc-tools

Location: /usr/share/bcc/tools

Tools In Action

GUI tools

- Cloud-Flare has developed a open-source exporter around EPBF tool to ingest data to prometheus and Grafana for visualisation
- PMM also supports EBPF with external exporter support
- Netflix has their own GUI tool called Vector built on top of EBPF
- Many more enhancements are in progress

Mydbops

References

- https://ebpf.io https://www.iovisor.org/technology/ebpf
- http://www.brendangregg.com/blog/2019-01-01/learn-ebpf-tracing.html http://www.brendangregg.com/ebpf.html
- https://netflixtechblog.com/introducing-vector-netflixs-on-host-performance-monitoring-tool-c0d3058c3f6f
- https://github.com/iovisor/bcc

Mydbops

Thank You

Reach Us : Info@mydbops.com