

Brendan Gregg



Foreword by **Alexei Starovoitov**,
creator of the new BPF

BPF Performance Tools

BPF Performance Tools

Table of Contents

Cover

Half Title

Title Page

Copyright Page

Contents

Foreword

Preface

Acknowledgments

About the Author

Part I: Technologies

1 Introduction

1.1 What Are BPF and eBPF?

1.2 What Are Tracing, Snooping, Sampling, Profiling, and Observability?

1.3 What Are BCC, bpftrace, and IO Visor?

1.4 A First Look at BCC: Quick Wins

1.5 BPF Tracing Visibility

1.6 Dynamic Instrumentation: kprobes and uprobes

1.7 Static Instrumentation: Tracepoints and USDT

1.8 A First Look at bpftrace: Tracing open()

1.9 Back to BCC: Tracing open()

1.10 Summary

2 Technology Background

2.1 BPF Illustrated

Table of Contents

2.2 BPF

2.3 Extended BPF (eBPF)

- 2.3.1 Why Performance Tools Need BPF
- 2.3.2 BPF Versus Kernel Modules
- 2.3.3 Writing BPF Programs
- 2.3.4 Viewing BPF Instructions: bpftool
- 2.3.5 Viewing BPF Instructions: bpftrace
- 2.3.6 BPF API
- 2.3.7 BPF Concurrency Controls
- 2.3.8 BPF sysfs Interface
- 2.3.9 BPF Type Format (BTF)
- 2.3.10 BPF CO-RE
- 2.3.11 BPF Limitations
- 2.3.12 BPF Additional Reading

2.4 Stack Trace Walking

- 2.4.1 Frame PointerBased Stacks
- 2.4.2 debuginfo
- 2.4.3 Last Branch Record (LBR)
- 2.4.4 ORC
- 2.4.5 Symbols
- 2.4.6 More Reading

2.5 Flame Graphs

- 2.5.1 Stack Trace
- 2.5.2 Profiling Stack Traces
- 2.5.3 Flame Graph
- 2.5.4 Flame Graph Features
- 2.5.5 Variations

2.6 Event Sources

2.7 kprobes

- 2.7.1 How kprobes Work
- 2.7.2 kprobes Interfaces
- 2.7.3 BPF and kprobes
- 2.7.4 kprobes Additional Reading

Table of Contents

2.8 uprobes

- 2.8.1 How uprobes Work
- 2.8.2 Uprobes Interfaces
- 2.8.3 BPF and uprobes
- 2.8.4 uprobes Overhead and Future Work
- 2.8.5 uprobes Additional Reading

2.9 Tracepoints

- 2.9.1 Adding Tracepoint Instrumentation
- 2.9.2 How Tracepoints Work
- 2.9.3 Tracepoint Interfaces
- 2.9.4 Tracepoints and BPF
- 2.9.5 BPF Raw Tracepoints
- 2.9.6 Additional Reading

2.10 USDT

- 2.10.1 Adding USDT Instrumentation
- 2.10.2 How USDT Works
- 2.10.3 BPF and USDT
- 2.10.4 USDT Additional Reading

2.11 Dynamic USDT

2.12 PMCs

- 2.12.1 PMC Modes
- 2.12.2 PEBS
- 2.12.3 Cloud Computing

2.13 perf_events

2.14 Summary

3 Performance Analysis

3.1 Overview

- 3.1.1 Goals
- 3.1.2 Activities
- 3.1.3 Multiple Performance Issues

3.2 Performance Methodologies

- 3.2.1 Workload Characterization

Table of Contents

3.2.2 Drill-Down Analysis

3.2.3 USE Method

3.2.4 Checklists

3.3 Linux 60-Second Analysis

3.3.1 uptime

3.3.2 dmesg | tail

3.3.3 vmstat 1

3.3.4 mpstat -P ALL 1

3.3.5 pidstat 1

3.3.6 iostat -xz 1

3.3.7 free -m

3.3.8 sar -n DEV 1

3.3.9 sar -n TCP,ETCP 1

3.3.10 top

3.4 BCC Tool Checklist

3.4.1 execsnoop

3.4.2 opensnoop

3.4.3 ext4slower

3.4.4 biolatency

3.4.5 biosnoop

3.4.6 cachestat

3.4.7 tcpconnect

3.4.8 tcpaccept

3.4.9 tcpretrans

3.4.10 runqlat

3.4.11 profile

3.5 Summary

4 BCC

4.1 BCC Components

4.2 BCC Features

4.2.1 Kernel-Level Features

4.2.2 BCC User-Level Features

4.3 BCC Installation

Table of Contents

4.3.1 Kernel Requirements

4.3.2 Ubuntu

4.3.3 RHEL

4.3.4 Other Distributions

4.4 BCC Tools

4.4.1 Highlighted Tools

4.4.2 Tool Characteristics

4.4.3 Single-Purpose Tools

4.4.4 Multi-Purpose Tools

4.5 funccount

4.5.1 funccount Examples

4.5.2 funccount Syntax

4.5.3 funccount One-Liners

4.5.4 funccount Usage

4.6 stackcount

4.6.1 stackcount Example

4.6.2 stackcount Flame Graphs

4.6.3 stackcount Broken Stack Traces

4.6.4 stackcount Syntax

4.6.5 stackcount One-Liners

4.6.6 stackcount Usage

4.7 trace

4.7.1 trace Example

4.7.2 trace Syntax

4.7.3 trace One-Liners

4.7.4 trace Structs

4.7.5 trace Debugging File Descriptor Leaks

4.7.6 trace Usage

4.8 argdist

4.8.1 argdist Syntax

4.8.2 argdist One-Liners

4.8.3 argdist Usage

4.9 Tool Documentation

Table of Contents

- 4.9.1 Man Page: opensnoop
- 4.9.2 Examples File: opensnoop

4.10 Developing BCC Tools

4.11 BCC Internals

4.12 BCC Debugging

- 4.12.1 printf() Debugging
- 4.12.2 BCC Debug Output
- 4.12.3 BCC Debug Flag
- 4.12.4 bpflist
- 4.12.5 bpftool
- 4.12.6 dmesg
- 4.12.7 Resetting Events

4.13 Summary

5 bpftrace

5.1 bpftrace Components

5.2 bpftrace Features

- 5.2.1 bpftrace Event Sources
- 5.2.2 bpftrace Actions
- 5.2.3 bpftrace General Features
- 5.2.4 bpftrace Compared to Other Observability Tools

5.3 bpftrace Installation

- 5.3.1 Kernel Requirements
- 5.3.2 Ubuntu
- 5.3.3 Fedora
- 5.3.4 Post-Build Steps
- 5.3.5 Other Distributions

5.4 bpftrace Tools

- 5.4.1 Highlighted Tools
- 5.4.2 Tool Characteristics
- 5.4.3 Tool Execution

5.5 bpftrace One-Liners

5.6 bpftrace Documentation

Table of Contents

5.7 bpftrace Programming

- 5.7.1 Usage
- 5.7.2 Program Structure
- 5.7.3 Comments
- 5.7.4 Probe Format
- 5.7.5 Probe Wildcards
- 5.7.6 Filters
- 5.7.7 Actions
- 5.7.8 Hello, World!
- 5.7.9 Functions
- 5.7.10 Variables
- 5.7.11 Map Functions
- 5.7.12 Timing vs `read()`

5.8 bpftrace Usage

5.9 bpftrace Probe Types

- 5.9.1 tracepoint
- 5.9.2 usdt
- 5.9.3 kprobe and kretprobe
- 5.9.4 uprobe and uretprobe
- 5.9.5 software and hardware
- 5.9.6 profile and interval

5.10 bpftrace Flow Control

- 5.10.1 Filter
- 5.10.2 Ternary Operators
- 5.10.3 If Statements
- 5.10.4 Unrolled Loops

5.11 bpftrace Operators

5.12 bpftrace Variables

- 5.12.1 Built-in Variables
- 5.12.2 Built-ins: pid, comm, and uid
- 5.12.3 Built-ins: kstack and ustack
- 5.12.4 Built-ins: Positional Parameters
- 5.12.5 Scratch

Table of Contents

5.12.6 Maps

5.13 bpftrace Functions

5.13.1 printf()

5.13.2 join()

5.13.3 str()

5.13.4 kstack() and ustack()

5.13.5 ksym() and usym()

5.13.6 kaddr() and uaddr()

5.13.7 system()

5.13.8 exit()

5.14 bpftrace Map Functions

5.14.1 count()

5.14.2 sum(), avg(), min(), and max()

5.14.3 hist()

5.14.4 lhist()

5.14.5 delete()

5.14.6 clear() and zero()

5.14.7 print()

5.15 bpftrace Future Work

5.15.1 Explicit Address Modes

5.15.2 Other Additions

5.15.3 ply

5.16 bpftrace Internals

5.17 bpftrace Debugging

5.17.1 printf() Debugging

5.17.2 Debug Mode

5.17.3 Verbose Mode

5.18 Summary

Part II: Using BPF Tools

6 CPUs

6.1 Background

6.1.1 CPU Fundamentals

Table of Contents

6.1.2 BPF Capabilities

6.1.3 Strategy

6.2 Traditional Tools

6.2.1 Kernel Statistics

6.2.2 Hardware Statistics

6.2.3 Hardware Sampling

6.2.4 Timed Sampling

6.2.5 Event Statistics and Tracing

6.3 BPF Tools

6.3.1 execsnoop

6.3.2 exitsnoop

6.3.3 runqlat

6.3.4 runqlen

6.3.5 runqslower

6.3.6 cpudist

6.3.7 cpufreq

6.3.8 profile

6.3.9 offcputime

6.3.10 syscount

6.3.11 argdist and trace

6.3.12 funccount

6.3.13 softirqs

6.3.14 hardirqs

6.3.15 smpcalls

6.3.16 llcstat

6.3.17 Other Tools

6.4 BPF One-Liners

6.4.1 BCC

6.4.2 bpftrace

6.5 Optional Exercises

6.6 Summary

7 Memory

Table of Contents

7.1 Background

7.1.1 Memory Fundamentals

7.1.2 BPF Capabilities

7.1.3 Strategy

7.2 Traditional Tools

7.2.1 Kernel Log

7.2.2 Kernel Statistics

7.2.3 Hardware Statistics and Sampling

7.3 BPF Tools

7.3.1 oomkill

7.3.2 memleak

7.3.3 mmapnoop

7.3.4 brkstack

7.3.5 shmsnoop

7.3.6 faults

7.3.7 ffaulst

7.3.8 vmscan

7.3.9 drsnoop

7.3.10 swapin

7.3.11 hfaulst

7.3.12 Other Tools

7.4 BPF One-Liners

7.4.1 BCC

7.4.2 bpftrace

7.5 Optional Exercises

7.6 Summary

8 File Systems

8.1 Background

8.1.1 File Systems Fundamentals

8.1.2 BPF Capabilities

8.1.3 Strategy

8.2 Traditional Tools

Table of Contents

8.2.1 df

8.2.2 mount

8.2.3 strace

8.2.4 perf

8.2.5 fatrace

8.3 BPF Tools

8.3.1 opensnoop

8.3.2 statsnoop

8.3.3 syncsnoop

8.3.4 mmapfiles

8.3.5 scread

8.3.6 fmapfault

8.3.7 filelife

8.3.8 vfsstat

8.3.9 vfscount

8.3.10 vfssize

8.3.11 fsrwstat

8.3.12 fileslower

8.3.13 filetop

8.3.14 writesync

8.3.15 filetype

8.3.16 cachestat

8.3.17 writeback

8.3.18 dcstat

8.3.19 dcsnoop

8.3.20 mountsnoop

8.3.21 xfsslower

8.3.22 xfsdist

8.3.23 ext4dist

8.3.24 icstat

8.3.25 bufgrow

8.3.26 readahead

8.3.27 Other Tools

Table of Contents

8.4 BPF One-Liners

8.4.1 BCC

8.4.2 bpftrace

8.4.3 BPF One-Liners Examples

8.5 Optional Exercises

8.6 Summary

9 Disk I/O

9.1 Background

9.1.1 Disk Fundamentals

9.1.2 BPF Capabilities

9.1.3 Strategy

9.2 Traditional Tools

9.2.1 iostat

9.2.2 perf

9.2.3 blktrace

9.2.4 SCSI Logging

9.3 BPF Tools

9.3.1 biolatency

9.3.2 biosnoop

9.3.3 biotop

9.3.4 bitesize

9.3.5 seeksize

9.3.6 biopattern

9.3.7 biostacks

9.3.8 bioerr

9.3.9 mdflush

9.3.10 iosched

9.3.11 scsilatency

9.3.12 scsiresult

9.3.13 nvmlatency

9.4 BPF One-Liners

9.4.1 BCC

Table of Contents

9.4.2 bpftrace

9.4.3 BPF One-Liners Examples

9.5 Optional Exercises

9.6 Summary

10 Networking

10.1 Background

10.1.1 Networking Fundamentals

10.1.2 BPF Capabilities

10.1.3 Strategy

10.1.4 Common Tracing Mistakes

10.2 Traditional Tools

10.2.1 ss

10.2.2 ip

10.2.3 nstat

10.2.4 netstat

10.2.5 sar

10.2.6 nicstat

10.2.7 ethtool

10.2.8 tcpdump

10.2.9 /proc

10.3 BPF Tools

10.3.1 sockstat

10.3.2 sofamily

10.3.3 soprotocol

10.3.4 soconnect

10.3.5 soaccept

10.3.6 socketio

10.3.7 socksize

10.3.8 sormem

10.3.9 soconnlat

10.3.10 so1stbyte

10.3.11 tcpconnect

10.3.12 tcpaccept

Table of Contents

- 10.3.13 tcplife
- 10.3.14 tcptop
- 10.3.15 tcpsnoop
- 10.3.16 tcpretrans
- 10.3.17 tcpsynbl
- 10.3.18 tcpwin
- 10.3.19 tcpnagle
- 10.3.20 udpconnect
- 10.3.21 gethostlatency
- 10.3.22 ipecn
- 10.3.23 superping
- 10.3.24 qdisc-fq
- 10.3.25 qdisc-cbq, qdisc-cbs, qdisc-codel, qdisc-fq_codel, qdisc-red, and qdisc-tbf
- 10.3.26 netsize
- 10.3.27 nettxlat
- 10.3.28 skbdrop
- 10.3.29 skblife
- 10.3.30 ieee80211scan
- 10.3.31 Other Tools

10.4 BPF One-Liners

- 10.4.1 BCC
- 10.4.2 bpftrace
- 10.4.3 BPF One-Liners Examples

10.5 Optional Exercises

10.6 Summary

11 Security

11.1 Background

- 11.1.1 BPF Capabilities
- 11.1.2 Unprivileged BPF Users
- 11.1.3 Configuring BPF Security
- 11.1.4 Strategy

11.2 BPF Tools

Table of Contents

- 11.2.1 execsnoop
- 11.2.2 elfsnoop
- 11.2.3 modsnoop
- 11.2.4 bashreadline
- 11.2.5 shellsnoop
- 11.2.6 ttysnoop
- 11.2.7 opensnoop
- 11.2.8 eperm
- 11.2.9 tcpconnect and tcpaccept
- 11.2.10 tcpreset
- 11.2.11 capable
- 11.2.12 setuids

11.3 BPF One-Liners

- 11.3.1 BCC
- 11.3.2 bpftrace
- 11.3.3 BPF One-Liners Examples

11.4 Summary

12 Languages

12.1 Background

- 12.1.1 Compiled
- 12.1.2 JIT Compiled
- 12.1.3 Interpreted
- 12.1.4 BPF Capabilities
- 12.1.5 Strategy
- 12.1.6 BPF Tools

12.2 C

- 12.2.1 C Function Symbols
- 12.2.2 C Stack Traces
- 12.2.3 C Function Tracing
- 12.2.4 C Function Offset Tracing
- 12.2.5 C USDT
- 12.2.6 C One-Liners

12.3 Java

Table of Contents

- 12.3.1 libjvm Tracing
- 12.3.2 jnistacks
- 12.3.3 Java Thread Names
- 12.3.4 Java Method Symbols
- 12.3.5 Java Stack Traces
- 12.3.6 Java USDT Probes
- 12.3.7 profile
- 12.3.8 offcputime
- 12.3.9 stackcount
- 12.3.10 javastat
- 12.3.11 javathreads
- 12.3.12 javacalls
- 12.3.13 javaflow
- 12.3.14 javagc
- 12.3.15 javaobjnew
- 12.3.16 Java One-Liners

12.4 Bash Shell

- 12.4.1 Function Counts
- 12.4.2 Function Argument Tracing (bashfunc.bt)
- 12.4.3 Function Latency (bashfuncat.bt)
- 12.4.4 /bin/bash
- 12.4.5 /bin/bash USDT
- 12.4.6 bash One-Liners

12.5 Other Languages

- 12.5.1 JavaScript (Node.js)
- 12.5.2 C++
- 12.5.3 Golang

12.6 Summary

13 Applications

13.1 Background

- 13.1.1 Application Fundamentals
- 13.1.2 Application Example: MySQL Server
- 13.1.3 BPF Capabilities

Table of Contents

13.1.4 Strategy

13.2 BPF Tools

13.2.1 execsnoop

13.2.2 threadsnoop

13.2.3 profile

13.2.4 threaded

13.2.5 offcputime

13.2.6 offcpuhist

13.2.7 syscount

13.2.8 ioprofile

13.2.9 libc Frame Pointers

13.2.10 mysqld_qslower

13.2.11 mysqld_clat

13.2.12 signals

13.2.13 killsnoop

13.2.14 pmlock and pmheld

13.2.15 napttime

13.2.16 Other Tools

13.3 BPF One-Liners

13.3.1 BCC

13.3.2 bpftrace

13.4 BPF One-Liners Examples

13.4.1 Counting libpthread Conditional Variable Functions for One Second

13.5 Summary

14 Kernel

14.1 Background

14.1.1 Kernel Fundamentals

14.1.2 BPF Capabilities

14.2 Strategy

14.3 Traditional Tools

14.3.1 Ftrace

14.3.2 perf sched

Table of Contents

14.3.3 slabtop

14.3.4 Other Tools

14.4 BPF Tools

14.4.1 loads

14.4.2 offcputime

14.4.3 wakeuptime

14.4.4 offwaketime

14.4.5 mlock and mheld

14.4.6 Spin Locks

14.4.7 kmem

14.4.8 kpages

14.4.9 memleak

14.4.10 slabratetop

14.4.11 numamove

14.4.12 workq

14.4.13 Tasklets

14.4.14 Other Tools

14.5 BPF One-Liners

14.5.1 BCC

14.5.2 bpftrace

14.6 BPF One-Liners Examples

14.7 Challenges

14.8 Summary

15 Containers

15.1 Background

15.1.1 BPF Capabilities

15.1.2 Challenges

15.1.3 Strategy

15.2 Traditional Tools

15.2.1 From the Host

15.2.2 From the Container

15.2.3 systemd-cgtop

Table of Contents

- 15.2.4 kubectl top
- 15.2.5 docker stats
- 15.2.6 /sys/fs/cgroups
- 15.2.7 perf

15.3 BPF Tools

- 15.3.1 runqlat
- 15.3.2 pidnss
- 15.3.3 blkthrot
- 15.3.4 overlayfs

15.4 BPF One-Liners

15.5 Optional Exercises

15.6 Summary

16 Hypervisors

16.1 Background

- 16.1.1 BPF Capabilities
- 16.1.2 Suggested Strategies

16.2 Traditional Tools

16.3 Guest BPF Tools

- 16.3.1 Xen Hypercalls
- 16.3.2 xenhyper
- 16.3.3 Xen Callbacks
- 16.3.4 cpustolen
- 16.3.5 HVM Exit Tracing

16.4 Host BPF Tools

- 16.4.1 kvmexits
- 16.4.2 Future Work

16.5 Summary

Part III: Additional Topics

17 Other BPF Performance Tools

- 17.1 Vector and Performance Co-Pilot (PCP)
 - 17.1.1 Visualizations

Table of Contents

- 17.1.2 Visualization: Heat Maps
- 17.1.3 Visualization: Tabular Data
- 17.1.4 BCC Provided Metrics
- 17.1.5 Internals
- 17.1.6 Installing PCP and Vector
- 17.1.7 Connecting and Viewing Data
- 17.1.8 Configuring the BCC PMDA
- 17.1.9 Future Work
- 17.1.10 Further Reading
- 17.2 Grafana and Performance Co-Pilot (PCP)
 - 17.2.1 Installation and Configuration
 - 17.2.2 Connecting and Viewing Data
 - 17.2.3 Future Work
 - 17.2.4 Further Reading
- 17.3 Cloudflare eBPF Prometheus Exporter (with Grafana)
 - 17.3.1 Build and Run the ebpf Exporter
 - 17.3.2 Configure Prometheus to Monitor the ebpf_exporter Instance
 - 17.3.3 Set Up a Query in Grafana
 - 17.3.4 Further Reading
- 17.4 kubectl-trace
 - 17.4.1 Tracing Nodes
 - 17.4.2 Tracing Pods and Containers
 - 17.4.3 Further Reading
- 17.5 Other Tools
- 17.6 Summary
- 18 Tips, Tricks, and Common Problems
 - 18.1 Typical Event Frequency and Overhead
 - 18.1.1 Frequency
 - 18.1.1.2 Action Performed
 - 18.1.3 Test Yourself
 - 18.2 Sample at 49 or 99 Hertz
 - 18.3 Yellow Pigs and Gray Rats

Table of Contents

18.4 Write Target Software

18.5 Learn Syscalls

18.6 Keep It Simple

18.7 Missing Events

18.8 Missing Stacks Traces

18.8.1 How to Fix Broken Stack Traces

18.9 Missing Symbols (Function Names) When Printing

18.9.1 How to Fix Missing Symbols: JIT Runtimes (Java, Node.js, ...)

18.9.2 How to Fix Missing Symbols: ELF binaries (C, C++, ...)

18.10 Missing Functions When Tracing

18.11 Feedback Loops

18.12 Dropped Events

Part IV: Appendixes

A: bpftrace One-Liners

B: bpftrace Cheat Sheet

C: BCC Tool Development

D: C BPF

E: BPF Instructions

Glossary

A

B

C

D

E

F

G

H

I

Table of Contents

J

K

L

M

N

O

P

R

S

T

U

V

W

Z

Bibliography

Index