

# Enterprise and the Cloud

# Brendan Gregg



# Systems Performance

---

Second Edition

# Systems Performance

## Table of Contents

Cover

Half Title Page

Title Page

Copyright Page

Contents at a Glance

Contents

Preface

Acknowledgments

About the Author

1 Introduction

1.1 Systems Performance

1.2 Roles

1.3 Activities

1.4 Perspectives

1.5 Performance Is Challenging

1.5.1 Subjectivity

1.5.2 Complexity

1.5.3 Multiple Causes

1.5.4 Multiple Performance Issues

1.6 Latency

1.7 Observability

1.7.1 Counters, Statistics, and Metrics

# **Table of Contents**

1.7.2 Profiling

1.7.3 Tracing

1.8 Experimentation

1.9 Cloud Computing

1.10 Methodologies

1.10.1 Linux Perf Analysis in 60 Seconds

1.11 Case Studies

1.11.1 Slow Disks

1.11.2 Software Change

1.11.3 More Reading

1.12 References

## **2 Methodologies**

2.1 Terminology

2.2 Models

2.2.1 System Under Test

2.2.2 Queueing System

2.3 Concepts

2.3.1 Latency

2.3.2 Time Scales

2.3.3 Trade-Offs

2.3.4 Tuning Efforts

2.3.5 Level of Appropriateness

2.3.6 When to Stop Analysis

2.3.7 Point-in-Time Recommendations

2.3.8 Load vs. Architecture

2.3.9 Scalability

2.3.10 Metrics

2.3.11 Utilization

# **Table of Contents**

2.3.12 Saturation

2.3.13 Profiling

2.3.14 Caching

2.3.15 Known-Unknowns

## **2.4 Perspectives**

2.4.1 Resource Analysis

2.4.2 Workload Analysis

## **2.5 Methodology**

2.5.1 Streetlight Anti-Method

2.5.2 Random Change Anti-Method

2.5.3 Blame-Someone-Else Anti-Method

2.5.4 Ad Hoc Checklist Method

2.5.5 Problem Statement

2.5.6 Scientific Method

2.5.7 Diagnosis Cycle

2.5.8 Tools Method

2.5.9 The USE Method

2.5.10 The RED Method

2.5.11 Workload Characterization

2.5.12 Drill-Down Analysis

2.5.13 Latency Analysis

2.5.14 Method R

2.5.15 Event Tracing

2.5.16 Baseline Statistics

2.5.17 Static Performance Tuning

2.5.18 Cache Tuning

2.5.19 Micro-Benchmarking

2.5.20 Performance Mantras

## **2.6 Modeling**

# **Table of Contents**

- 2.6.1 Enterprise vs. Cloud
- 2.6.2 Visual Identification
- 2.6.3 Amdahls Law of Scalability
- 2.6.4 Universal Scalability Law
- 2.6.5 Queueing Theory

## **2.7 Capacity Planning**

- 2.7.1 Resource Limits
- 2.7.2 Factor Analysis
- 2.7.3 Scaling Solutions

## **2.8 Statistics**

- 2.8.1 Quantifying Performance Gains
- 2.8.2 Averages
- 2.8.3 Standard Deviation, Percentiles, Median
- 2.8.4 Coefficient of Variation
- 2.8.5 Multimodal Distributions
- 2.8.6 Outliers

## **2.9 Monitoring**

- 2.9.1 Time-Based Patterns
- 2.9.2 Monitoring Products
- 2.9.3 Summary-Since-Boot

## **2.10 Visualizations**

- 2.10.1 Line Chart
- 2.10.2 Scatter Plots
- 2.10.3 Heat Maps
- 2.10.4 Timeline Charts
- 2.10.5 Surface Plot
- 2.10.6 Visualization Tools

## **2.11 Exercises**

## **2.12 References**

# **Table of Contents**

## **3 Operating Systems**

### **3.1 Terminology**

### **3.2 Background**

#### **3.2.1 Kernel**

#### **3.2.2 Kernel and User Modes**

#### **3.2.3 System Calls**

#### **3.2.4 Interrupts**

#### **3.2.5 Clock and Idle**

#### **3.2.6 Processes**

#### **3.2.7 Stacks**

#### **3.2.8 Virtual Memory**

#### **3.2.9 Schedulers**

#### **3.2.10 File Systems**

#### **3.2.11 Caching**

#### **3.2.12 Networking**

#### **3.2.13 Device Drivers**

#### **3.2.14 Multiprocessor**

#### **3.2.15 Preemption**

#### **3.2.16 Resource Management**

#### **3.2.17 Observability**

### **3.3 Kernels**

#### **3.3.1 Unix**

#### **3.3.2 BSD**

#### **3.3.3 Solaris**

### **3.4 Linux**

#### **3.4.1 Linux Kernel Developments**

#### **3.4.2 systemd**

#### **3.4.3 KPTI (Meltdown)**

#### **3.4.4 Extended BPF**

# **Table of Contents**

## **3.5 Other Topics**

3.5.1 PGO Kernels

3.5.2 Unikernels

3.5.3 Microkernels and Hybrid Kernels

3.5.4 Distributed Operating Systems

## **3.6 Kernel Comparisons**

## **3.7 Exercises**

## **3.8 References**

3.8.1 Additional Reading

## **4 Observability Tools**

### **4.1 Tool Coverage**

4.1.1 Static Performance Tools

4.1.2 Crisis Tools

### **4.2 Tool Types**

4.2.1 Fixed Counters

4.2.2 Profiling

4.2.3 Tracing

4.2.4 Monitoring

### **4.3 Observability Sources**

4.3.1 /proc

4.3.2 /sys

4.3.3 Delay Accounting

4.3.4 netlink

4.3.5 Tracepoints

4.3.6 kprobes

4.3.7 uprobes

4.3.8 USDT

4.3.9 Hardware Counters (PMCs)

4.3.10 Other Observability Sources

# **Table of Contents**

## **4.4 sar**

4.4.1 sar(1) Coverage

4.4.2 sar(1) Monitoring

4.4.3 sar(1) Live

4.4.4 sar(1) Documentation

## **4.5 Tracing Tools**

## **4.6 Observing Observability**

## **4.7 Exercises**

## **4.8 References**

# **5 Applications**

## **5.1 Application Basics**

5.1.1 Objectives

5.1.2 Optimize the Common Case

5.1.3 Observability

5.1.4 Big O Notation

## **5.2 Application Performance Techniques**

5.2.1 Selecting an I/O Size

5.2.2 Caching

5.2.3 Buffering

5.2.4 Polling

5.2.5 Concurrency and Parallelism

5.2.6 Non-Blocking I/O

5.2.7 Processor Binding

5.2.8 Performance Mantras

## **5.3 Programming Languages**

5.3.1 Compiled Languages

5.3.2 Interpreted Languages

5.3.3 Virtual Machines

# **Table of Contents**

5.3.4 Garbage Collection

## **5.4 Methodology**

5.4.1 CPU Profiling

5.4.2 Off-CPU Analysis

5.4.3 Syscall Analysis

5.4.4 USE Method

5.4.5 Thread State Analysis

5.4.6 Lock Analysis

5.4.7 Static Performance Tuning

5.4.8 Distributed Tracing

## **5.5 Observability Tools**

5.5.1 perf

5.5.2 profile

5.5.3 offcputime

5.5.4 strace

5.5.5 execsnoop

5.5.6 syscount

5.5.7 bpftrace

## **5.6 Gotchas**

5.6.1 Missing Symbols

5.6.2 Missing Stacks

## **5.7 Exercises**

## **5.8 References**

# **6 CPUs**

## **6.1 Terminology**

## **6.2 Models**

6.2.1 CPU Architecture

6.2.2 CPU Memory Caches

6.2.3 CPU Run Queues

# **Table of Contents**

## **6.3 Concepts**

- 6.3.1 Clock Rate
- 6.3.2 Instructions
- 6.3.3 Instruction Pipeline
- 6.3.4 Instruction Width
- 6.3.5 Instruction Size
- 6.3.6 SMT
- 6.3.7 IPC, CPI
- 6.3.8 Utilization
- 6.3.9 User Time/Kernel Time
- 6.3.10 Saturation
- 6.3.11 Preemption
- 6.3.12 Priority Inversion
- 6.3.13 Multiprocess, Multithreading
- 6.3.14 Word Size
- 6.3.15 Compiler Optimization

## **6.4 Architecture**

- 6.4.1 Hardware
- 6.4.2 Software

## **6.5 Methodology**

- 6.5.1 Tools Method
- 6.5.2 USE Method
- 6.5.3 Workload Characterization
- 6.5.4 Profiling
- 6.5.5 Cycle Analysis
- 6.5.6 Performance Monitoring
- 6.5.7 Static Performance Tuning
- 6.5.8 Priority Tuning
- 6.5.9 Resource Controls

# **Table of Contents**

6.5.10 CPU Binding

6.5.11 Micro-Benchmarking

## **6.6 Observability Tools**

6.6.1 uptime

6.6.2 vmstat

6.6.3 mpstat

6.6.4 sar

6.6.5 ps

6.6.6 top

6.6.7 pidstat

6.6.8 time, ptime

6.6.9 turbostat

6.6.10 showboost

6.6.11 pmcarch

6.6.12 tlstat

6.6.13 perf

6.6.14 profile

6.6.15 cpudist

6.6.16 runqlat

6.6.17 runqlen

6.6.18 softirqs

6.6.19 hardirqs

6.6.20 bpfttrace

6.6.21 Other Tools

## **6.7 Visualizations**

6.7.1 Utilization Heat Map

6.7.2 Subsecond-Offset Heat Map

6.7.3 Flame Graphs

6.7.4 FlameScope

# **Table of Contents**

## **6.8 Experimentation**

6.8.1 Ad Hoc

6.8.2 SysBench

## **6.9 Tuning**

6.9.1 Compiler Options

6.9.2 Scheduling Priority and Class

6.9.3 Scheduler Options

6.9.4 Scaling Governors

6.9.5 Power States

6.9.6 CPU Binding

6.9.7 Exclusive CPU Sets

6.9.8 Resource Controls

6.9.9 Security Boot Options

6.9.10 Processor Options (BIOS Tuning)

## **6.10 Exercises**

## **6.11 References**

# **7 Memory**

## **7.1 Terminology**

## **7.2 Concepts**

7.2.1 Virtual Memory

7.2.2 Paging

7.2.3 Demand Paging

7.2.4 Overcommit

7.2.5 Process Swapping

7.2.6 File System Cache Usage

7.2.7 Utilization and Saturation

7.2.8 Allocators

7.2.9 Shared Memory

7.2.10 Working Set Size

# **Table of Contents**

7.2.11 Word Size

## **7.3 Architecture**

7.3.1 Hardware

7.3.2 Software

7.3.3 Process Virtual Address Space

## **7.4 Methodology**

7.4.1 Tools Method

7.4.2 USE Method

7.4.3 Characterizing Usage

7.4.4 Cycle Analysis

7.4.5 Performance Monitoring

7.4.6 Leak Detection

7.4.7 Static Performance Tuning

7.4.8 Resource Controls

7.4.9 Micro-Benchmarking

7.4.10 Memory Shrinking

## **7.5 Observability Tools**

7.5.1 vmstat

7.5.2 PSI

7.5.3 swapon

7.5.4 sar

7.5.5 slabtop

7.5.6 numastat

7.5.7 ps

7.5.8 top

7.5.9 pmap

7.5.10 perf

7.5.11 drsnoop

7.5.12 wss

# **Table of Contents**

7.5.13 bpftrace

7.5.14 Other Tools

## **7.6 Tuning**

7.6.1 Tunable Parameters

7.6.2 Multiple Page Sizes

7.6.3 Allocators

7.6.4 NUMA Binding

7.6.5 Resource Controls

## **7.7 Exercises**

## **7.8 References**

# **8 File Systems**

## **8.1 Terminology**

## **8.2 Models**

8.2.1 File System Interfaces

8.2.2 File System Cache

8.2.3 Second-Level Cache

## **8.3 Concepts**

8.3.1 File System Latency

8.3.2 Caching

8.3.3 Random vs. Sequential I/O

8.3.4 Prefetch

8.3.5 Read-Ahead

8.3.6 Write-Back Caching

8.3.7 Synchronous Writes

8.3.8 Raw and Direct I/O

8.3.9 Non-Blocking I/O

8.3.10 Memory-Mapped Files

8.3.11 Metadata

8.3.12 Logical vs. Physical I/O

# **Table of Contents**

8.3.13 Operations Are Not Equal

8.3.14 Special File Systems

8.3.15 Access Timestamps

8.3.16 Capacity

## **8.4 Architecture**

8.4.1 File System I/O Stack

8.4.2 VFS

8.4.3 File System Caches

8.4.4 File System Features

8.4.5 File System Types

8.4.6 Volumes and Pools

## **8.5 Methodology**

8.5.1 Disk Analysis

8.5.2 Latency Analysis

8.5.3 Workload Characterization

8.5.4 Performance Monitoring

8.5.5 Static Performance Tuning

8.5.6 Cache Tuning

8.5.7 Workload Separation

8.5.8 Micro-Benchmarking

## **8.6 Observability Tools**

8.6.1 mount

8.6.2 free

8.6.3 top

8.6.4 vmstat

8.6.5 sar

8.6.6 slabtop

8.6.7 strace

8.6.8 fatrace

# **Table of Contents**

8.6.9 LatencyTOP

8.6.10 opensnoop

8.6.11 filetop

8.6.12 cachestat

8.6.13 ext4dist (xfs, zfs, btrfs, nfs)

8.6.14 ext4slower (xfs, zfs, btrfs, nfs)

8.6.15 bpftrace

8.6.17 Other Tools

8.6.18 Visualizations

## **8.7 Experimentation**

8.7.1 Ad Hoc

8.7.2 Micro-Benchmark Tools

8.7.3 Cache Flushing

## **8.8 Tuning**

8.8.1 Application Calls

8.8.2 ext4

8.8.3 ZFS

## **8.9 Exercises**

## **8.10 References**

# **9 Disks**

## **9.1 Terminology**

## **9.2 Models**

9.2.1 Simple Disk

9.2.2 Caching Disk

9.2.3 Controller

## **9.3 Concepts**

9.3.1 Measuring Time

9.3.2 Time Scales

9.3.3 Caching

# **Table of Contents**

- 9.3.4 Random vs. Sequential I/O
- 9.3.5 Read/Write Ratio
- 9.3.6 I/O Size
- 9.3.7 IOPS Are Not Equal
- 9.3.8 Non-Data-Transfer Disk Commands
- 9.3.9 Utilization
- 9.3.10 Saturation
- 9.3.11 I/O Wait
- 9.3.12 Synchronous vs. Asynchronous
- 9.3.13 Disk vs. Application I/O

## **9.4 Architecture**

- 9.4.1 Disk Types
- 9.4.2 Interfaces
- 9.4.3 Storage Types
- 9.4.4 Operating System Disk I/O Stack

## **9.5 Methodology**

- 9.5.1 Tools Method
- 9.5.2 USE Method
- 9.5.3 Performance Monitoring
- 9.5.4 Workload Characterization
- 9.5.5 Latency Analysis
- 9.5.6 Static Performance Tuning
- 9.5.7 Cache Tuning
- 9.5.8 Resource Controls
- 9.5.9 Micro-Benchmarking
- 9.5.10 Scaling

## **9.6 Observability Tools**

- 9.6.1 iostat
- 9.6.2 sar

# **Table of Contents**

9.6.3 PSI

9.6.4 pidstat

9.6.5 perf

9.6.6 biolatency

9.6.7 biosnoop

9.6.8 iotop, biotop

9.6.9 biostacks

9.6.10 blktrace

9.6.11 bpftrace

9.6.12 MegaCli

9.6.13 smartctl

9.6.14 SCSI Logging

9.6.15 Other Tools

## **9.7 Visualizations**

9.7.1 Line Graphs

9.7.2 Latency Scatter Plots

9.7.3 Latency Heat Maps

9.7.4 Offset Heat Maps

9.7.5 Utilization Heat Maps

## **9.8 Experimentation**

9.8.1 Ad Hoc

9.8.2 Custom Load Generators

9.8.3 Micro-Benchmark Tools

9.8.4 Random Read Example

9.8.5 ioping

9.8.6 fio

9.8.7 blkreplay

## **9.9 Tuning**

9.9.1 Operating System Tunables

# **Table of Contents**

9.9.2 Disk Device Tunables

9.9.3 Disk Controller Tunables

9.10 Exercises

9.11 References

## **10 Network**

10.1 Terminology

10.2 Models

10.2.1 Network Interface

10.2.2 Controller

10.2.3 Protocol Stack

10.3 Concepts

10.3.1 Networks and Routing

10.3.2 Protocols

10.3.3 Encapsulation

10.3.4 Packet Size

10.3.5 Latency

10.3.6 Buffering

10.3.7 Connection Backlog

10.3.8 Interface Negotiation

10.3.9 Congestion Avoidance

10.3.10 Utilization

10.3.11 Local Connections

10.4 Architecture

10.4.1 Protocols

10.4.2 Hardware

10.4.3 Software

10.5 Methodology

10.5.1 Tools Method

10.5.2 USE Method

# **Table of Contents**

10.5.3 Workload Characterization

10.5.4 Latency Analysis

10.5.5 Performance Monitoring

10.5.6 Packet Sniffing

10.5.7 TCP Analysis

10.5.8 Static Performance Tuning

10.5.9 Resource Controls

10.5.10 Micro-Benchmarking

## **10.6 Observability Tools**

10.6.1 ss

10.6.2 ip

10.6.3 ifconfig

10.6.4 nstat

10.6.5 netstat

10.6.6 sar

10.6.7 nicstat

10.6.8 ethtool

10.6.9 tcplife

10.6.10 tcptop

10.6.11 tcpretrans

10.6.12 bpftrace

10.6.13 tcpdump

10.6.14 Wireshark

10.6.15 Other Tools

## **10.7 Experimentation**

10.7.1 ping

10.7.2 traceroute

10.7.3 pathchar

10.7.4 iperf

# **Table of Contents**

10.7.5 netperf

10.7.6 tc

10.7.7 Other Tools

## **10.8 Tuning**

10.8.1 System-Wide

10.8.2 Socket Options

10.8.3 Configuration

## **10.9 Exercises**

## **10.10 References**

# **11 Cloud Computing**

## **11.1 Background**

11.1.1 Instance Types

11.1.2 Scalable Architecture

11.1.3 Capacity Planning

11.1.4 Storage

11.1.5 Multitenancy

11.1.6 Orchestration (Kubernetes)

## **11.2 Hardware Virtualization**

11.2.1 Implementation

11.2.2 Overhead

11.2.3 Resource Controls

11.2.4 Observability

## **11.3 OS Virtualization**

11.3.1 Implementation

11.3.2 Overhead

11.3.3 Resource Controls

11.3.4 Observability

## **11.4 Lightweight Virtualization**

# **Table of Contents**

11.4.1 Implementation

11.4.2 Overhead

11.4.3 Resource Controls

11.4.4 Observability

11.5 Other Types

11.6 Comparisons

11.7 Exercises

11.8 References

## **12 Benchmarking**

12.1 Background

12.1.1 Reasons

12.1.2 Effective Benchmarking

12.1.3 Benchmarking Failures

12.2 Benchmarking Types

12.2.1 Micro-Benchmarking

12.2.2 Simulation

12.2.3 Replay

12.2.4 Industry Standards

12.3 Methodology

12.3.1 Passive Benchmarking

12.3.2 Active Benchmarking

12.3.3 CPU Profiling

12.3.4 USE Method

12.3.5 Workload Characterization

12.3.6 Custom Benchmarks

12.3.7 Ramping Load

12.3.8 Sanity Check

12.3.9 Statistical Analysis

12.3.10 Benchmarking Checklist

# **Table of Contents**

12.4 Benchmark Questions

12.5 Exercises

12.6 References

## **13 perf**

13.1 Subcommands Overview

13.2 One-Liners

13.3 perf Events

13.4 Hardware Events

13.4.1 Frequency Sampling

13.5 Software Events

13.6 Tracepoint Events

13.7 Probe Events

13.7.1 kprobes

13.7.2 uprobes

13.7.3 USDT

13.8 perf stat

13.8.1 Options

13.8.2 Interval Statistics

13.8.3 Per-CPU Balance

13.8.4 Event Filters

13.8.5 Shadow Statistics

13.9 perf record

13.9.1 Options

13.9.2 CPU Profiling

13.9.3 Stack Walking

13.10 perf report

13.10.1 TUI

13.10.2 STDIO

# **Table of Contents**

## 13.11 perf script

### 13.11.1 Flame Graphs

### 13.11.2 Trace Scripts

## 13.12 perf trace

### 13.12.1 Kernel Versions

## 13.13 Other Commands

## 13.14 perf Documentation

## 13.15 References

# 14 Ftrace

## 14.1 Capabilities Overview

## 14.2 tracefs (/sys)

### 14.2.1 tracefs Contents

## 14.3 Ftrace Function Profiler

## 14.4 Ftrace Function Tracing

### 14.4.1 Using trace

### 14.4.2 Using trace\_pipe

### 14.4.3 Options

## 14.5 Tracepoints

### 14.5.1 Filter

### 14.5.2 Trigger

## 14.6 kprobes

### 14.6.1 Event Tracing

### 14.6.2 Arguments

### 14.6.3 Return Values

### 14.6.4 Filters and Triggers

### 14.6.5 kprobe Profiling

## 14.7 uprobes

### 14.7.1 Event Tracing

### 14.7.2 Arguments and Return Values

# **Table of Contents**

- 14.7.3 Filters and Triggers
- 14.7.4 uprobe Profiling
- 14.8 Ftrace function\_graph
  - 14.8.1 Graph Tracing
  - 14.8.2 Options
- 14.9 Ftrace hwlat
- 14.10 Ftrace Hist Triggers
  - 14.10.1 Single Keys
  - 14.10.2 Fields
  - 14.10.3 Modifiers
  - 14.10.4 PID Filters
  - 14.10.5 Multiple Keys
  - 14.10.6 Stack Trace Keys
  - 14.10.7 Synthetic Events
- 14.11 trace-cmd
  - 14.11.1 Subcommands Overview
  - 14.11.2 trace-cmd One-Liners
  - 14.11.3 trace-cmd vs. perf(1)
  - 14.11.4 trace-cmd function\_graph
  - 14.11.5 KernelShark
  - 14.11.6 trace-cmd Documentation
- 14.12 perf ftrace
- 14.13 perf-tools
  - 14.13.1 Tool Coverage
  - 14.13.2 Single-Purpose Tools
  - 14.13.3 Multi-Purpose Tools
  - 14.13.4 perf-tools One-Liners
  - 14.13.5 Example
  - 14.13.6 perf-tools vs. BCC/BPF

# **Table of Contents**

14.13.7 Documentation

14.14 Ftrace Documentation

14.15 References

## **15 BPF**

15.1 BCC

15.1.1 Installation

15.1.2 Tool Coverage

15.1.3 Single-Purpose Tools

15.1.4 Multi-Purpose Tools

15.1.5 One-Liners

15.1.6 Multi-Tool Example

15.1.7 BCC vs. bpftrace

15.1.8 Documentation

15.2 bpftrace

15.2.1 Installation

15.2.2 Tools

15.2.3 One-Liners

15.2.4 Programming

15.2.5 Reference

15.2.6 Documentation

15.3 References

## **16 Case Study**

16.1 An Unexplained Win

16.1.1 Problem Statement

16.1.2 Analysis Strategy

16.1.3 Statistics

16.1.4 Configuration

16.1.5 PMCs

# **Table of Contents**

16.1.6 Software Events

16.1.7 Tracing

16.1.8 Conclusion

16.2 Additional Information

16.3 References

Appendix A: USE Method: Linux

Appendix B: sar Summary

Appendix C: bpftrace One-Liners

Appendix D: Solutions to Selected Exercises

Appendix E: Systems Performance Whos Who

Glossary

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

# **Table of Contents**

R

S

T

U

V

W

X

Z

Index