Automating AI out of a Job

Adam Leskis · Kraken Tech

How should we learn stuff now?

AI doing all the things means minimal learning

We can learn faster, we have the technology

Quick prototypes we never would have attempted before

GUIDE
use-tool

→

CREATE PRESSURE
use-practice

→

EXPLORE
Doom Perf

Building to learn < Building to teach

Focus on assessing the learning outcomes

Let's code our way out!

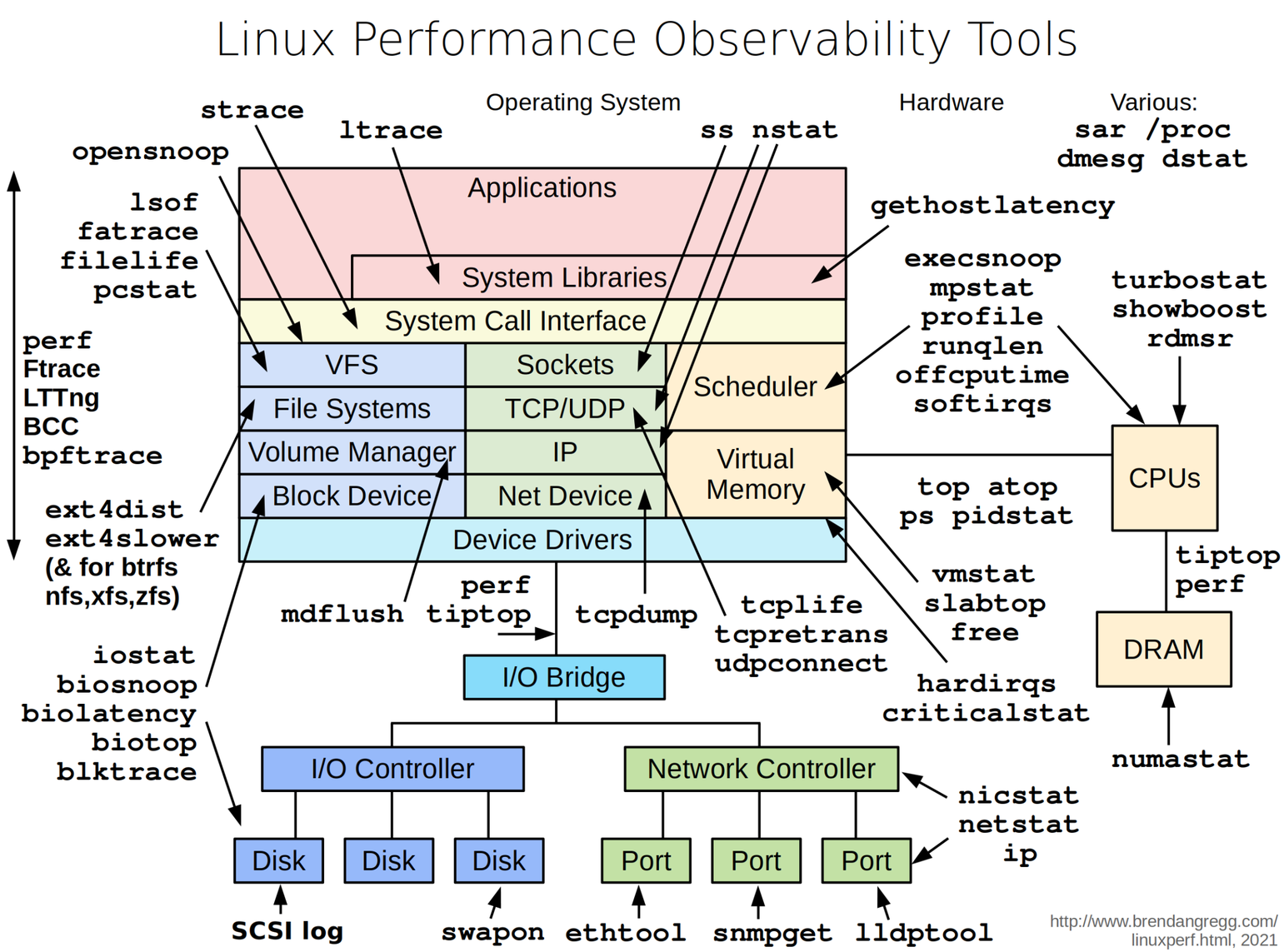
Not for prod

These tools have not *solved* perf

No time for the learning design part (but that's also super cool/relevant)


What is a perf?
I need more perf knowledge...but how?

All of this


Linux Performance Observability Tools
Operating System
Hardware
Various:
sar /proc
dmesg dstat
strace
ltrace
ss nstat
opensnoop
lsof
fatrace
filelife
pcstat
perf
Ftrace
LTTng
BCC
bpftrace
ext4dist
ext4slower
(& for btrfs
nfs,xfs,zfs)
iostat
biosnoop
biolatency
biotop
blktrace
Applications
System Libraries
System Call Interface
VFS
Sockets
File Systems
TCP/UDP
Volume Manager
IP
Block Device
Net Device
Device Drivers
Scheduler
Virtual
Memory
I/O Bridge
I/O Controller
Network Controller
Disk
Disk
Disk
Port
Port
Port
SCSI log
swapon
ethtool
snmpget
lldptool
gethostlatency
execsnoop
mpstat
profile
runqlen
offcputime
softirqs
turbostat
showboost
rdmsr
CPUs
top atop
ps pidstat
vmstat
slabtop
free
hardirqs
criticalstat
tiptop
perf
DRAM
numastat
mdflush
perf
tiptop
tcpdump
tcplife
tcpretrans
udpconnect
nicstat
netstat
ip
http://www.brendangregg.com/
linuxperf.html, 2021

The short version of what to know

Linux Performance Analysis in 60,000 Milliseconds · brendangregg.com/Articles/Netflix_Linux_Perf_Analysis_60s.pdf

```
01 uptime
02 dmesg | tail
03 vmstat 1
04 mpstat -P ALL 1
05 pidstat 1
06 iostat -xz 1
07 free -m
08 sar -n DEV 1
09 sar -n TCP,ETCP 1
10 top
```

$ use-tool guide

We need a mech suit!

https://github.com/lpmi-13/use-tool

User-defined Playground

USE Practice Playground

by Adam Leskis

on Linux Networking Observability

Practice Linux USE-method investigations against randomized host-level scenarios. It already has the investigation and load generating tools installed, so you can begin with that, or just run a free-form investigation.

Startup configuration

Use-Practice: What if we could investigate USE issues on demand?

https://github.com/lpmi-13/use-practice

$ use-practice run

Real pressure in a disposable Linux VM

> NB: not for prod

CPU | MEMORY | DISK | NETWORK

SCENARIO → **LINUX VM** → **REAL SIGNALS**



$ use-practice run random

Less focus on facts, more on contextualized application

DISPOSABLE LINUX VM

CHOOSE
blind or targeted

→

START
culprit + decoys

→

INVESTIGATE
real host signals

→

REVEAL
check and reset

User-defined Playground

USE Practice Playground

by Adam Leskis

on Linux Networking Observability

Practice Linux USE-method investigations against randomized host-level scenarios. It already has the investigation and load generating tools installed, so you can begin with that, or just run a free-form investigation.

Startup configuration

What if you could walk through telemetry?

Doom Perf · a world driven by system signals

https://github.com/lpmi-13/doom-perf

Perf knowledge is its own reward

USE is merely the beginning! There's so much more to know
  (flamegraphs, etc)

User-defined Playground

Doom Perf Lab Playground

by Adam Leskis

on Linux Observability

Explore live host CPU/Memory/Disk/Network telemetry as an interactive Doom diagnostics lab. All the elements in the game react to system signals (eg, the cpu cores glow red when there's high utilization).

Startup configuration

Adam Leskis

github.com/lpmi-13
dev.to/lpmi13
leskis@gmail.com
who.has.the.most.subdomains.and.why.is.it.adamleskis.com



TRY THE LAB

Doom Perf on iximiuz Labs

Scan or click to open ↗

EXPLORE THE CODE

github.com/lpmi-13

Adam's open-source projects ↗