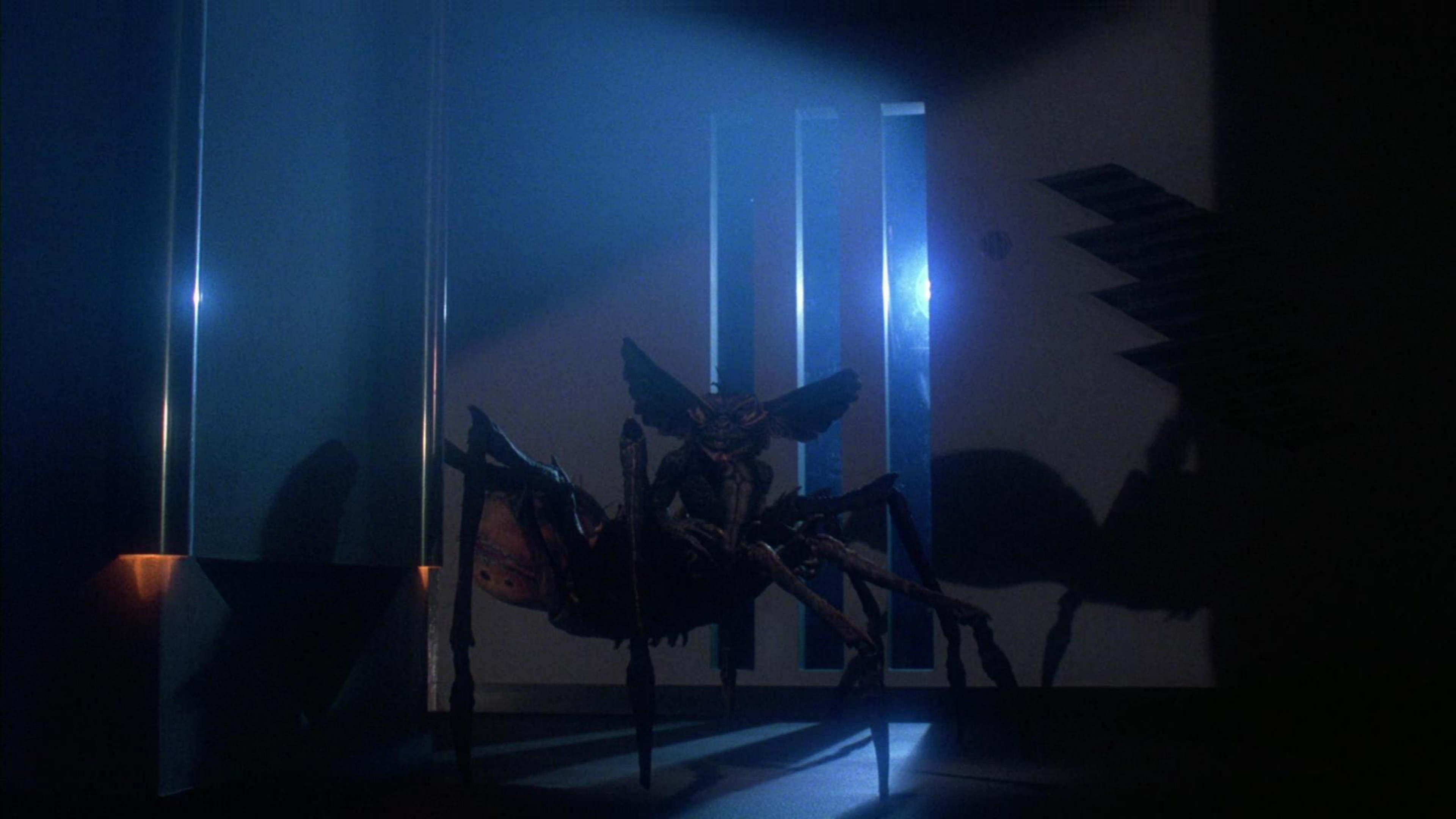
Gremlins Exposed

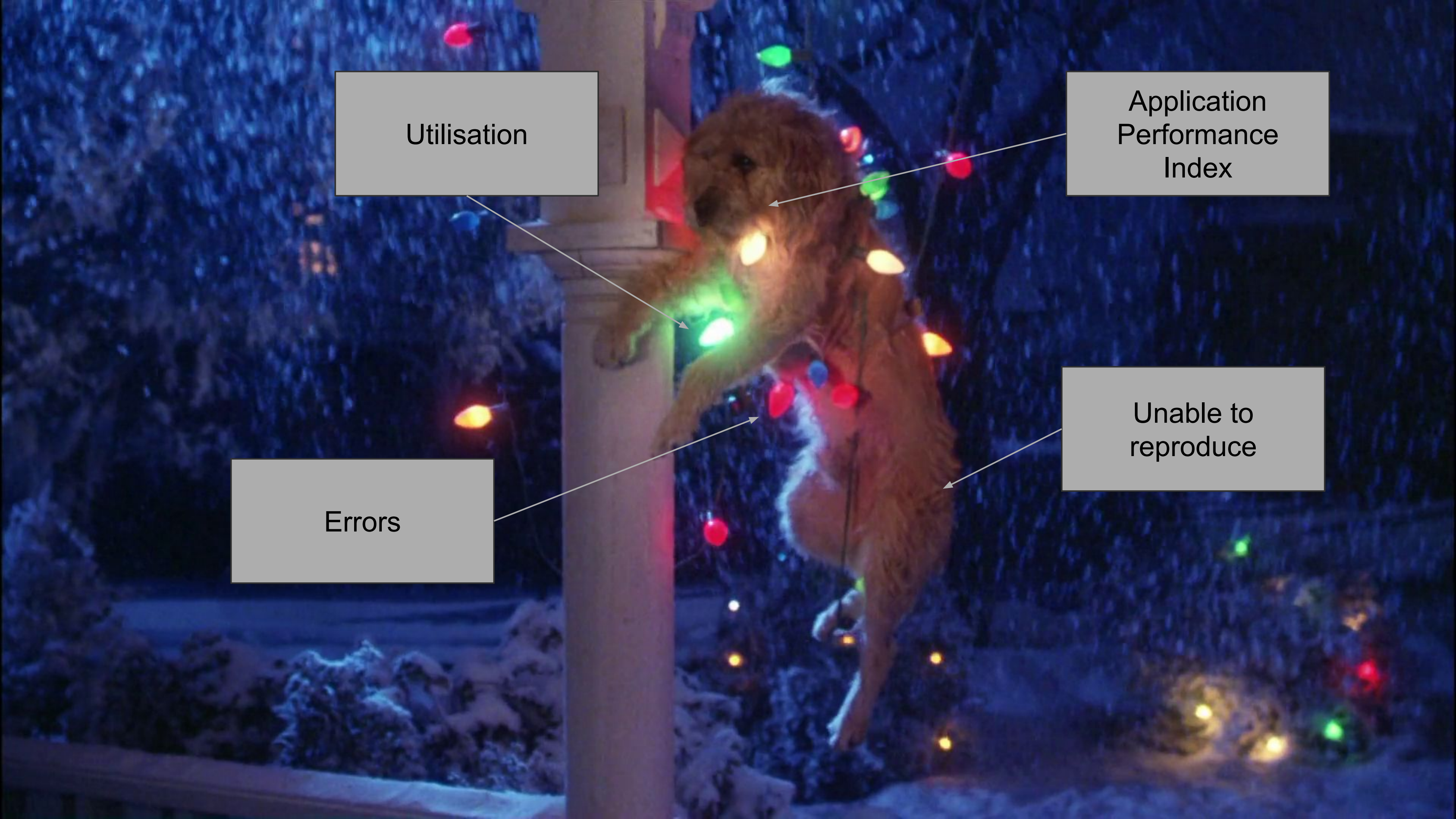
SHINING A LIGHT ON MISCHIEVOUS SYSTEMS

Thomas Cuthbert <tom@tcuthbert.id.au>

twcuthbert

keybase://tcuthbert

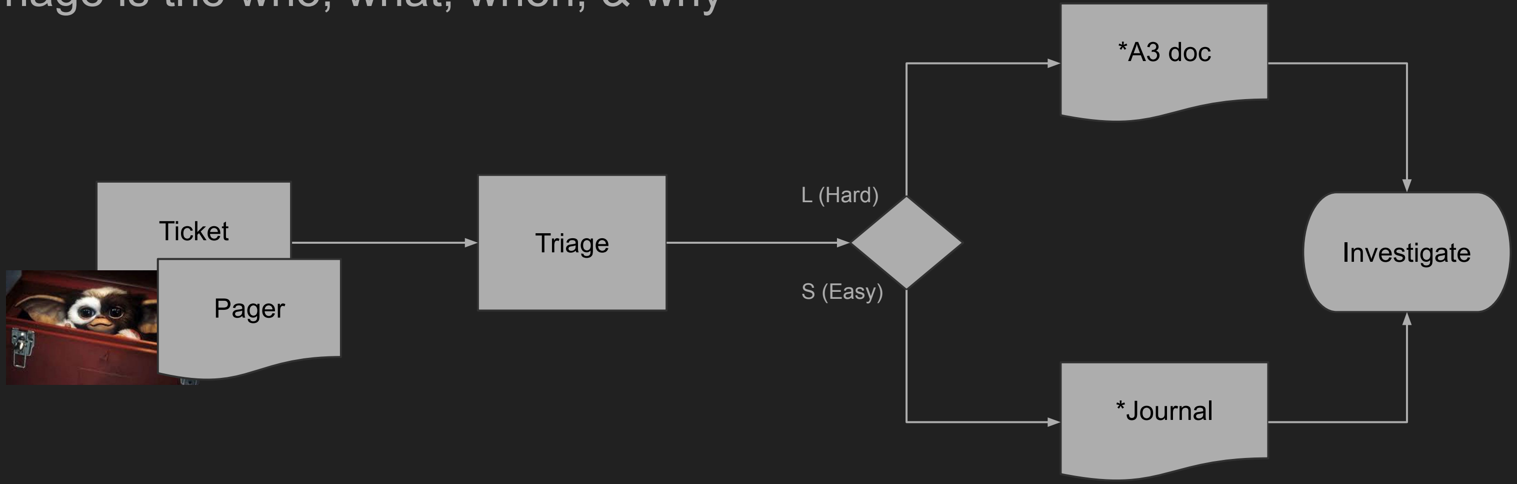



Utilisation
Application Performance Index
Unable to reproduce
Errors

Understanding the Problem

Structured wisdom is the key to effective problem solving

- Triage is the who, what, when, & why

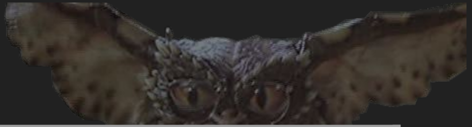


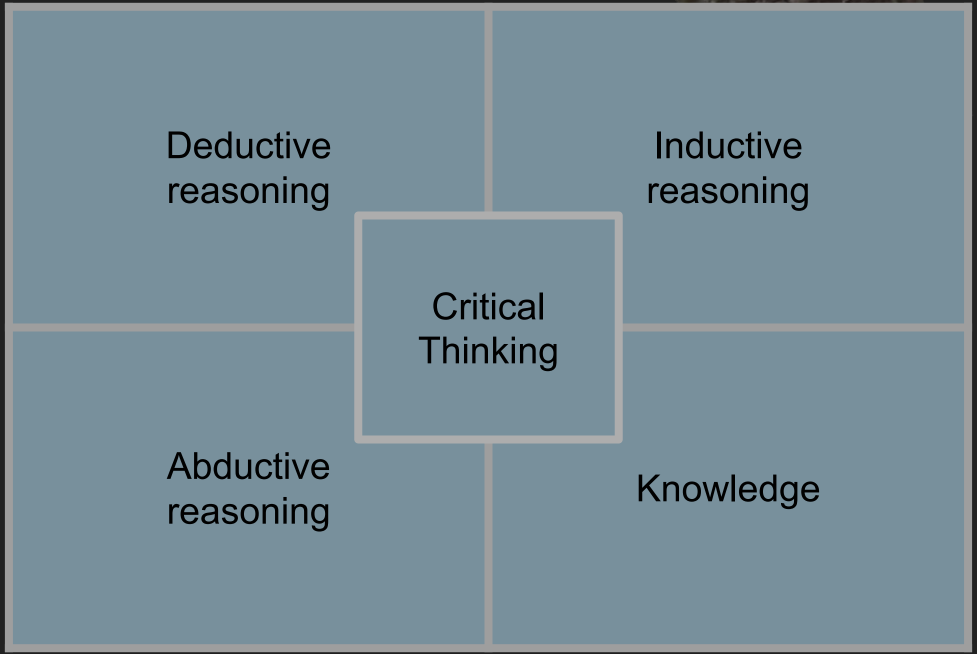
- Translation of human and non-human information
- Don't overload your brain; structure your notes

* A3 example template
* Zettelkasten

Gremlins are unreasonable

Critical thinking is the cornerstone of any investigation

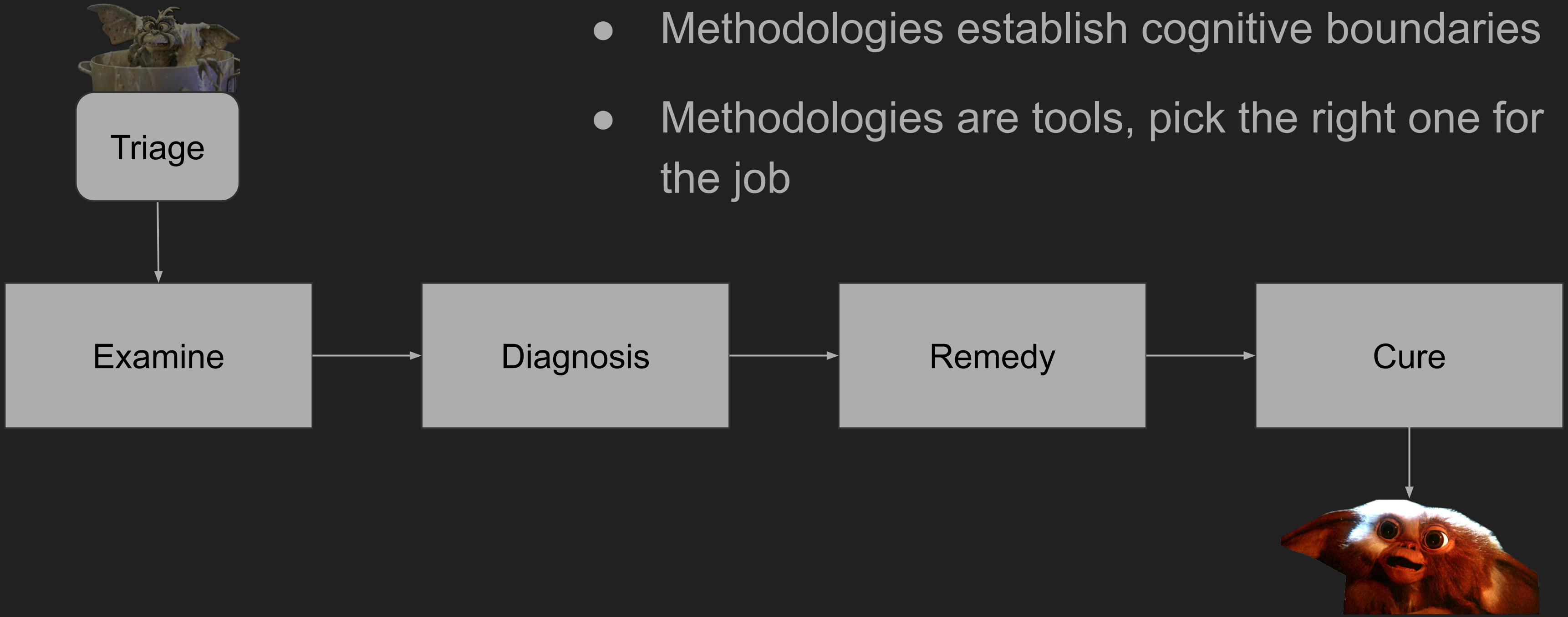




Methodologies

The more you know the less you understand

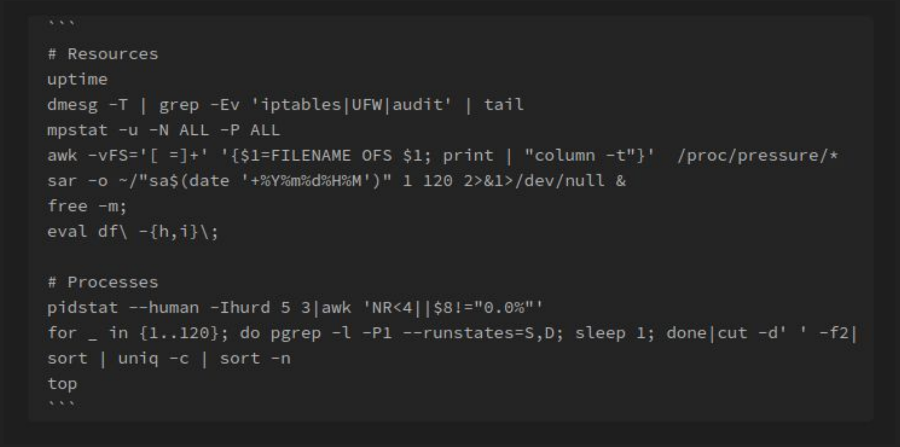
- Methodologies stimulate critical thinking
- Methodologies establish cognitive boundaries
- Methodologies are tools, pick the right one for the job



Act II
Gremlin Response

First few minutes

1. Is it really broken?
2. *Analyse Grafana dashboard
3. Run through a *checklist of commands & oneliners



~~~
```
# Resources
uptime
dmesg -T | grep -Ev 'iptables|UFW|audit' | tail
mpstat -u -N ALL -P ALL
awk -vFS='[ =]+' '{$1=FILENAME OFS $1; print | "column -t"}'  /proc/pressure/*
sar -o ~/"sa$(date '+%Y%m%d%H%M')" 1 120 2>&1>/dev/null &
free -m;
eval df\ -{h,i}\;

# Processes
pidstat --human -Ihurd 5 3|awk 'NR<4||$8!="0.0%"'
for _ in {1..120}; do pgrep -l -P1 --runstates=S,D; sleep 1; done|cut -d' ' -f2|
sort | uniq -c | sort -n
top
```
~~~

3. Start asking *questions
   - ✔ Have I seen this before?
   - × Anything in the logs?

Your first few minutes process should be organic & match your own growth!

* Check the last slide for links to analysis hints

Method	Technique	*Goal*	Tools	Indicators
Examine	**Top-down**	*Reproducing:*	curl, dig, apt-get, dd, cat, time	error codes, messages, durations
Diagnosis	**Divide-and-conquer**	*Filtering:*	ping, sort, uniq, find, grep, jq, netcat	last-mod, size/length, RTT, timestamp, IP, path
Examine	**Trace the path**	*Locality:*	traceroute, mtr, curl --write-out, strace	TTLs, ASN, TTFB, packet markings, MTUs
Examine	**Comparing differences**	*What changed:*	diff, git bisect, htop, free, sysstat, email, bash history, syslog	history, context, metrics
Remedy	**Component swapping**	*Short-circuiting:*	add/remove resources	reducing urgency/criticality
Diagnosis	**Bottom-up**	*Last resort:*	tcpdump, perf/bpftrace, pdb, git blame	knowledge acquisition, working from first principles

```
./haproxy-simple.sh
```

Act III

Shining Light on Invisible Gremlins

Invisible Gremlins

- Interpreting data is confusing: why is the cache fine with CPU[1] at 95%, yet falls over when 50% idle?
- Bursts[2] in latency, contention, and errors are invisible to tools that use averages[3]
- Correlate bottom-up measurements with top-down observations, then map it to hypothesis & prediction[4]

[1] SquidProfiling
[2] See Queueing Theory & QoS
[3] Utilisation, Saturation, Errors
[4] See Scientific method for more details

Method	Technique	Scenario	Example Test
Diagnosis	**Top-down**	Is data cached effectively?	/usr/bin/time -v cat /slow/$datafile > /dev/null
Diagnosis	**Bottom-up**		perf stat -p $pid -e major-faults -e 'vmscan:*' --repeat 60 --table sleep 1
Diagnosis	**Divide-and-conquer**	I want to know if the application is dropping packets	perf top -e skb:kfree_skb -ns comm
Remedy	**Component swapping**	When I add another server it will no longer be the top process dropping packets	juju add-unit $application
Diagnosis	**Compare the differences**	top/vmstat doesn't show any performance cliffs yet application writes are slow Bursting I/O is masked by averaging. Measure I/O counts and latency instead	bcc.biosnoop -Q bcc.biolatency bcc.biotop

The most efficient use of your time is to not spend it

All Gremlins need to feel better is a KISS

- Gremlins multiply exponentially so time is of the essence!
- Keep it simple with the easier techniques during the first few minutes
- Short-circuiting reduces urgency
- Communicate workarounds in the wiki, MOTD, deployment specs

Reflection

→ Critical thinking and reasoning is the key to effective troubleshooting

→ Methodologies & techniques are performance tools for our brains

→ Metrics can be misleading, let measurements & observations influence how you find them

→ Take shortcuts

→ Don't be afraid to communicate & ask for guidance

THANKS FOR LISTENING

Thomas Cuthbert <tom@tcuthbert.id.au>

twcuthbert

keybase://tcuthbert

Grimoires for Gremlins

My curated tomes for the dark art of catching Gremlins

Topic	Literature
Methodology	Problem solving methods, YBecause, Asking questions, A3 Problem Solving, Reasoning explained for Sysadmins
Technique	Algorithms to Live By, Systems Performance, SRE Handbook, Debugging/writing code & Why did they design it this way?, Queueing theory
Analysis	Data Mangling, Shell Tricks, How Linux networking works, Prometheus Tips, PromQL, tshark for network analysis, USE, perf, Nightmare mode BPF (bpftrace), EZ mode BPF (bcc), My oneliner collection, *How computers work, Mapping statistical analysis theory to PromQL
Triage & Reflection	Indexing your mail (notmuch), Personal Knowledge Management (zettelkasten), Managing Your Time, How to Win Friends & Influence People

*Addison-Wesley Professional Computing bundle