

## SREcon: Performance Checklists for SREs 2016

04 May 2016

When Netflix is down, minutes matter, and there's little time for traditional performance engineering. At [SREcon16 Santa Clara](#) I gave the closing address on performance checklists for SREs. Checklists are vital for this kind of work, and are often implemented at Netflix as custom dashboards of selected metrics.

This was my first talk about my SRE work at Netflix, where I've joined the on-call rotation for the Core incident response team. I began by summarizing the difference between performance engineering (where I spend most of my time, and is the team I'm on), and SRE incident response for performance issues.

The video is on [youtube](#) and [usenix.org](#):

## SREcon16 - Performance Checklists for SREs



And the slides are on [slideshare](#):



I summarized a dozen checklists in talk, as well as methodologies to derive them. They are roughly sorted in intended order of use: starting with cloud-wide dashboards and ending with Linux specific checklists.

The first two checklists are our Performance and Reliability Engineering (PRE) Triage Checklist, a shared document, and then predash, a custom dashboard. These are Netflix specific, and show how we begin this type of analysis. I thought for a moment that they were too specific to Netflix, but wanted to include them anyway for completeness.

I've reproduced the Linux checklists below, which should be implemented as GUI dashboards. Check the presentation for eight other checklists.

## 6. Linux Perf Analysis in 60s

1. **uptime** → load averages
2. **dmesg -T | tail** → kernel errors
3. **vmstat 1** → overall stats by time
4. **mpstat -P ALL 1** → CPU balance
5. **pidstat 1** → process usage
6. **iostat -xz 1** → disk I/O

7. **free -m** → memory usage
8. **sar -n DEV 1** → network I/O
9. **sar -n TCP,ETCP 1** → TCP stats
10. **top** → check overview

These are explained in the post [Linux Performance Analysis in 60 seconds](#) from the Netflix tech blog.

## 7. Linux Disk Checklist

1. **iostat -xz 1** → any disk I/O? if not, stop looking
2. **vmstat 1** → is this swapping? or, high sys time?
3. **df -h** → are file systems nearly full?
4. **ext4slower 10** → (zfs\*, xfs\*, etc.) slow file system I/O?
5. **bioslower 10** → if so, check disks
6. **ext4dist 1** → check distribution and rate
7. **biolatency 1** → if interesting, check disks
8. **cat /sys/devices/.../ioerr\_cnt** → (if available) errors
9. **smartctl -l error /dev/sda1** → (if available) errors

Another short checklist. Won't solve everything. ext4slower/dist, bioslower/latency, are from [bcc/BPF tools](#).

## 8. Linux Network Checklist

1. **sar -n DEV,EDEV 1** → at interface limits? or use nicstat
2. **sar -n TCP,ETCP 1** → active/passive load, retransmit rate
3. **cat /etc/resolv.conf** → it's always DNS
4. **mpstat -P ALL 1** → high kernel time? single hot CPU?
5. **tcpretrans** → what are the retransmits? state?
6. **tcpconnect** → connecting to anything unexpected?
7. **tcpaccept** → unexpected workload?
8. **netstat -rnv** → any inefficient routes?
9. **check firewall config** → anything blocking/throttling?
10. **netstat -s** → play 252 metric pickup

tcp\*, are from bcc/BPF tools.

## 9. Linux CPU Checklist

1. **uptime** → load averages
2. **vmstat 1** → system-wide utilization, run q length
3. **mpstat -P ALL 1** → CPU balance
4. **pidstat 1** → per-process CPU
5. **CPU flame graph** → CPU profiling
6. **CPU subsecond offset heat map** → look for gaps
7. **perf stat -a -- sleep 10** → IPC, LLC hit ratio

htop can do 1-4. I'm tempted to add execsnoop for short-lived processes (it's in [perf-tools](#) or bcc/BPF tools).

For more about SRE at Netflix, see my colleague Jonah Horowitz's talk [Netflix: 190 Countries and 5 CORE SREs](#). We're also hiring SREs (keep an eye on Netflix [jobs](#)). For other talks about SRE (Site Reliability Engineering), see the SREcon16 [program](#).

This was my first SREcon and I found it very useful and informative, particularly to see what SRE really means to different companies. Thanks to USENIX and the organizers for a great conference!

*You can comment here, but I can't guarantee your comment will remain here forever: I might switch comment systems at some point (eg, if Disqus add advertisements).*

---

Copyright 2017 Brendan Gregg.  
[About this blog](#)

[TSA Method](#)  
[Off-CPU Analysis](#)  
[Active Bench.](#)  
[Flame Graphs](#)  
[Heat Maps](#)  
[Frequency Trails](#)  
[Colony Graphs](#)  
[perf Examples](#)  
[eBPF Tools](#)  
[DTrace Tools](#)  
[DTraceToolkit](#)  
[DtkshDemos](#)  
[Guessing Game](#)  
[Specials](#)  
[Books](#)  
[Other Sites](#)