

Zusammenfassung zum Vortrag Performanceanalyse

- top oder htop:
 - Überblick wer für die Systemlast verantwortlich ist, eher auf Systemen mit vielen verschiedenen Diensten (wie Servern) nützlich.
- vmstat -w 1
 - -w für wider output, 1 als Intervall der Aktualisierung der Werte
 - Überblick durch welche Last das System belastet wird, z.B. viele Kontext-Switches oder wenig verfügbarer Arbeitsspeicher. Man kann auch die Aufteilung der CPU-Zeit in user und systemspace sehen.
- iostat -x 1
 - -x für extended report, 1 als Intervall der Aktualisierung
 - Zusammenfassung der Disk-IO. Man kann z.B. sehen, wenn die Festplatte durch viele kleine Zugriffe ausgelastet ist.
- sar -n DEV 1
 - -n für network, DEV für device Statistiken, 1 als Intervall der Aktualisierung
 - Überblick über die Netzwerkaktivität. Man kann z.B. erkennen, ob das System durch viele kleine ankommende Nachrichten belastet wird oder das System zu viele Nachrichten verschicken möchte.
- Man kann auch noch mit sar -n NFS 1 die Aktivität des NFS-client analysieren.
- sar bietet viele weitere Möglichkeiten (z.B. sar -m TEMP 1 für Systemtemperatur oder -y für Aktivität der seriellen Schnittstelle)
- Vorgehen: zunächst alle wichtigen Hardware-Ressourcen auf Auslastung prüfen, diese gegen die erwartete Auslastung abgleichen, um eine Vorstellung davon zu bekommen, welche Teile einer Anwendung optimiert werden sollten. Im Vortrag daher nur: wie finde ich - schnell und ohne großen overhead für das Programm - eine Idee, wo ich z.B. mit Tracing-Tools gezielt tiefer in die Analyse einsteigen kann.
- Weitere Tools in der Übersicht: http://www.brendangregg.com/Perf/linux_perf_tools_full.png
- Weiterer Faktor, den man betrachten kann: Hardwarebedingte Fehler: z.B. bei der Netzwerkübertragung: sar -n EDEV 1