LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

MNM TEAM

A Symphony of Metrics: Assessing the Advantages of eBPF over conventional Benchmarking Tools

Abschlussvortrag zur Bachelorarbeit

Daniel Schneider

dan.schneider@campus.lmu.de

MNM-Team, Betreuer: Maximilian Höb

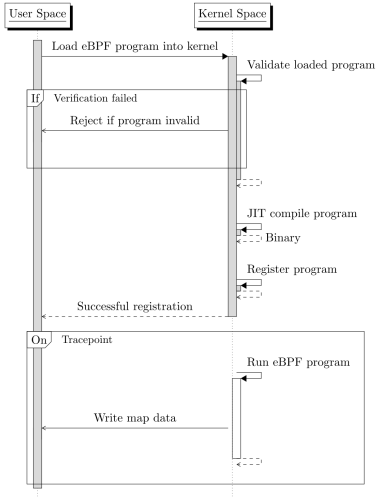
February 6, 2024

Motivation

- Benchmarking ist wichtig um Ressourcennutzung zu optimieren
- eBPF (extended Berkley Packet Filter) ist besonders interessant für Benchmarking
- **Welche Vorteile bringt eBPF im Vergleich zu konventionellen Lösungen?**

eBPF

- Wird im Linux Kernel ausgeführt
- Event-basierte Ausführung
- Normalerweise in C geschrieben
- Wird vom Kernel verifiziert und kompiliert
- Nebenläufiges user-space Programm vorausgesetzt
- Viele Limitiertungen: Komplexität, Speicherzugriff...



Methodik

Wie vergleicht man eBPF zu konventionellen Lösungen?

- Korrektheit
- Modularität
- Einfachheit der Verwendung
- Skalierbarkeit

Methodik

Wie vergleicht man eBPF zu konventionellen Lösungen?

- Korrektheit
- Modularität
- Einfachheit der Verwendung
- Skalierbarkeit

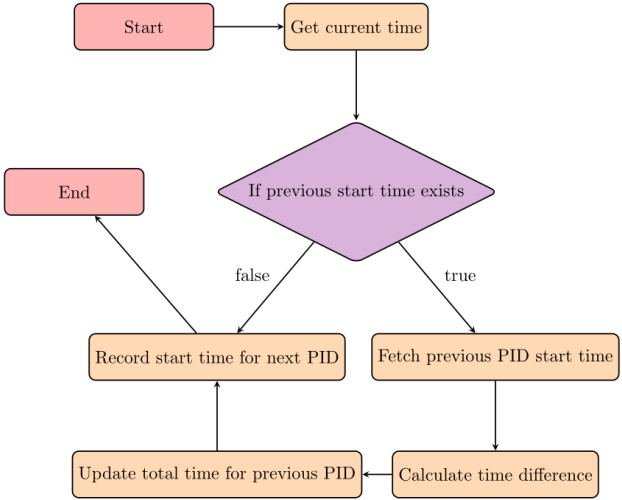
=> **Referenzimplementierung**

Metriken

- CPU Auslastung
- Speicher Nutzung
- Disk I/O Operations

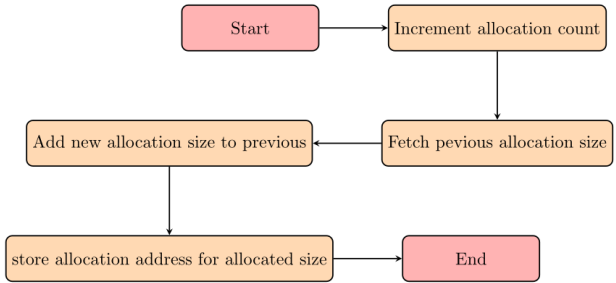
CPU Probe

- Tracepoint 'sched_switch'
- CPU timings pro PID



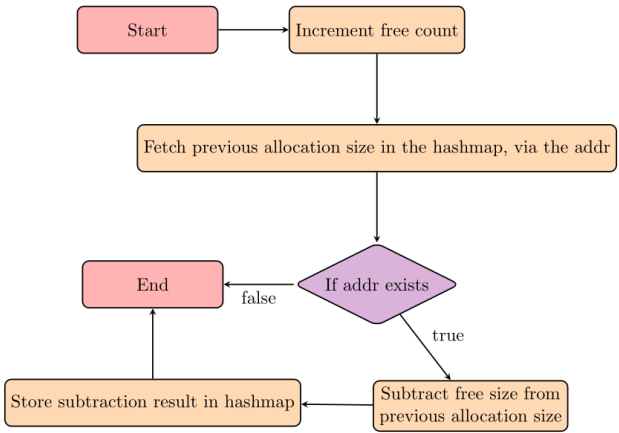
Memory Probe

- Zwei Kprobes
  - 'kmalloc'
  - 'kfree'
- Vorherige information über Speichernutzung aller PIDs nötig



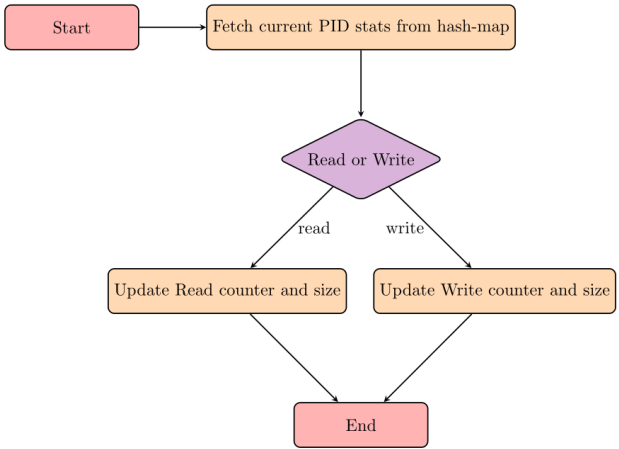
Memory Probe

- Zwei Kprobes
  - 'kmalloc'
  - 'kfree'
- Vorherige information über Speichernutzung aller PIDs nötig



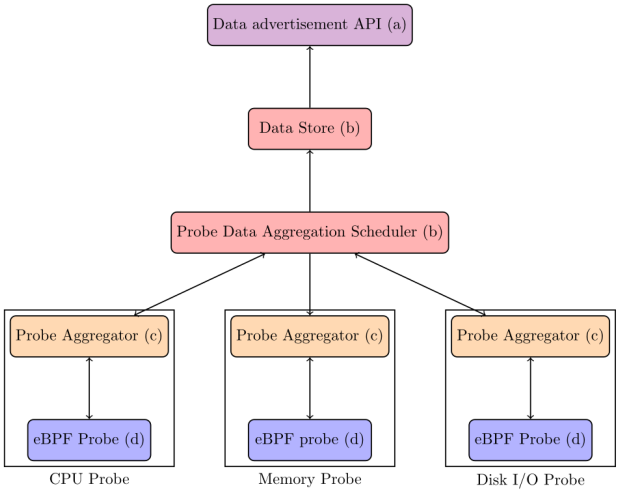
Disk I/O Probe

- Tracepoint 'block_rq_complete'
- Unterscheidung zwischen
  - Read
  - Write



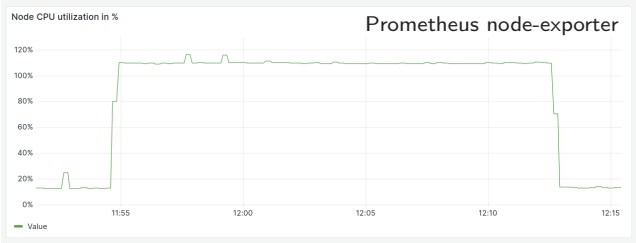
Gesamtaufbau

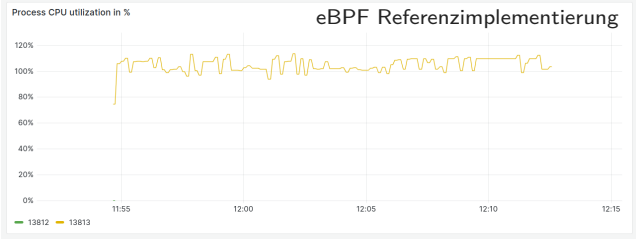
- Prometheus Data API
- Scheduler für Aggregationsintervalle
- Dynamische Registrierung von Probes
- Probes bestehen aus
  - User-space Teil
  - eBPF Teil



Korrektheit CPU Probe

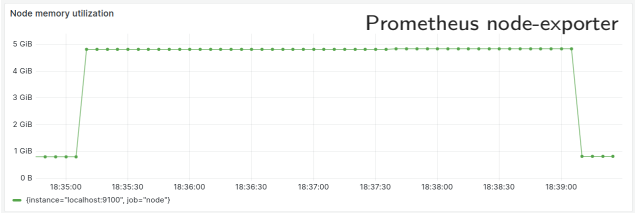
- stress -c 1
- Erhöhte Werte bei Konventionellem System auf Messstrategie zurückzuführen
- eBPF Werte auf PID Ebene
- eBPF hohe Volatilität





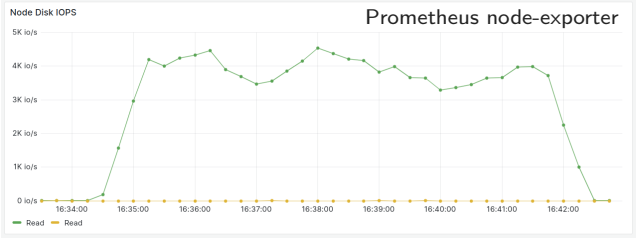
Korrektheit Memory Probe

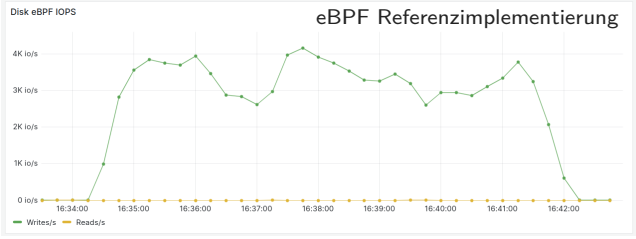
- stress -m 1 –vm-bytes 4G –vm-keep
- Erhöhte Werte bei Konventionellem System auf Messstrategie zurückzuführen
- eBPF Werte auf PID Ebene



Korrektheit Disk I/O Probe

- stress –hdd 10 –io 1
- Erhöhte Werte bei Konventionellem System auf Messstrategie zurückzuführen
- eBPF Werte auf PID Ebene
- Hohe positive Korrelation





Modularität

- eBPF hoch modular/erweiterbar
- Konventionelle Systeme oft kaum bis garnicht modular/erweiterbar
- Spezialisierte Implementierungen einfach möglich
- Erhöhter Entwicklungsaufwand

Einfachheit

- Hohe Komplexität bei der eBPF-Programmentwicklung
- Tiefes Verständnis von Linux nötig
- Nicht ohne weitere Hilfsmittel nutzbar
  - User-space Programm
  - eBPF Compiler Toolchain
  - Richtige Kernel Konfiguration
  - Ausreichende Nutzerrechte

Skalierbarkeit

- Implementierungsabhängig
- Theoretisch hoch skalierbar
- Verbunden mit erhöhtem Entwicklungsaufwand
- Spezifische konventionelle Systeme auch hoch skalierbar

Conclusion

- eBPF breitflächig und korrekt einsetzbar
- Erhöhter Entwicklungsaufwand
- Weitere Forschung:
  - Wie verhält sich eBPF bei containerisierten workloads
  - Mehr vergleichende Assessments nötig