

2. Übungsblatt

Ausgabe: 14. Oktober 2019 · Besprechung: 21. Oktober 2019

Einleitung

Die meisten modernen Prozessoren verfügen über so genannte “Performance Counters”, mit denen verschiedene Cache- oder CPU-Effekte direkt in Hardware analysiert werden können. Es handelt sich dabei um Zähler, mit denen bestimmte Ereignisse, z. B. “All requests that missed L2”, gezählt werden können.

Zum Zugriff auf die Counter gibt es verschiedene Werkzeuge und Bibliotheken, z. B.

- Intel Performance Counter Monitor¹

Diese Bibliothek wurde von Intel entwickelt und steht als Open Source-Software zur Verfügung. Vorteil der Bibliothek ist, dass auch sehr moderne Features der Intel-Prozessoren unterstützt werden. Allerdings sind u. U. weit reichende Benutzerrechte notwendig, um die Counter auszulesen; Intel PCM sollte unter Linux, Windows und macOS funktionieren (zumindest letzteres ist allerdings wegen des notwendigen Kernelmoduls problematisch).

- PAPI²

PAPI ist eine Open Source-Entwicklung. Die Bibliothek ist flexibler was die Hardware-Plattformen anbelangt, gleichzeitig aber auch eingeschränkt auf einige Unix-Derivate und Linux. Vorteil der PAPI-Bibliothek ist, dass sie auf Kernel-Interfaces aufsetzt, die auch mit “normalen” Benutzerrechten zugreifbar sein sollten. Die Bibliothek sollte damit z. B. auch auf den Poolrechnern funktionieren (was ich allerdings nicht ausprobiert habe).

- perf³

perf ist aus der Linux-Kernelentwicklung entstanden. Es ist keine Bibliothek, sondern ein eigenständiges Tool.

¹<http://software.intel.com/en-us/articles/intel-performance-counter-monitor-a-better-way-to-measure-cpu-utilization>

²<http://icl.cs.utk.edu/papi/>

³<https://perf.wiki.kernel.org/>

- Intel VTune⁴

Hier handelt es sich um sehr bekanntes Werkzeug aus dem Hause Intel. VTune ist sehr mächtig und funktioniert als eigenständiges Tool (wie perf), jedoch mit grafischer Oberfläche.

- Apple Instruments

Mit seiner XCode-Umgebung liefert Apple ein Werkzeug, das eine Funktionalität ähnlich von VTune bietet; ebenfalls ein eigenständiges Tool.

Aufgabe — Diskussion

Wie man in der obigen Liste sehen kann, kann man zur Arbeit mit den Performance Counters entweder “Bibliotheken” oder aber eigenständige “Werkzeuge” verwenden. Das macht konzeptionell durchaus einen Unterschied.

- Diskutieren Sie die beiden Ansätze.
- Für welchen Einsatz könnte die eine Variante sinnvoll sein, für welchen die andere?

Aufgabe — Implementierung

Auf dem ersten Aufgabenblatt haben wir die Auswirkungen von Caches indirekt beobachtet und dabei insbesondere durch Laufzeitmessungen (und Kenntnis unseres Codes) auf Cache Misses zurück geschlossen. Mit den oben aufgezeigten Wegen kann man Cache Misses auch direkt zählen.

- Nutzen Sie eines (oder mehrere?) der obigen Werkzeuge, um für Ihren Cache-Benchmark aus dem ersten Aufgabenblatt die Cache Misses sichtbar zu machen.

Hinweise

Zur Auswertung stellt jede CPU eine Vielzahl unterschiedlicher Ereignistypen zur Verfügung. Sie finden eine Aufstellung davon z. B. in den Handbüchern von Intel:

<http://www.intel.com/products/processor/manuals/>

Beachten Sie, dass die CPU nur in der Lage ist, eine limitierte Anzahl von Ereignistypen *gleichzeitig* zu überwachen (z. B. 4).

⁴<https://software.intel.com/en-us/vtune>