

CyPhOS – Betriebssystem für hardware-basierte Cache-Verwaltungsmechanismen

Hendrik Borghorst
Fakultät für Informatik
Technische Universität Dortmund
Dortmund, Deutschland
hendrik.borghorst@udo.edu

Olaf Spinczyk
Institut für Informatik
Universität Osnabrück
Osnabrück, Deutschland
olaf.spinczyk@uos.de

Moderne Mehrkernprozessoren stellen eine hohe Rechenleistung für einen geringen Preis bereit. Diese Rechenleistung wird unter anderem durch geteilte Ressourcen ermöglicht, wodurch es allerdings zu sehr hohen Schwankungen der Programmlaufzeiten kommen kann, da mehrere Prozessorkerne um die Ressourcen konkurrieren. Der Bus zum Hauptspeicher, der Hauptspeicher-Controller sowie ein geteilter Cache sind Hauptursachen für diese Schwankungen. Gegenseitige Verdrängung von Daten aus dem Cache kann dazu führen, dass Prozessoren sich im Cache gegenseitig im permanenten Wechsel die Daten verdrängen, wodurch es zu erheblich mehr Speicherzugriffen kommt. Eine Möglichkeit dies zu verhindern ist die Verwendung von Cache-Verwaltungsstrategien [1]. Ein Beispiel dafür ist das sogenannte *Cache locking* bei dem der Inhalt des Caches eingefroren wird. Eine weitere Technik ist die Partitionierung des Caches, das heißt der Cache wird in Partitionen eingeteilt, in denen jeder Prozessorkern exklusiv arbeiten kann.

Diverse Hardware-Architekturen bieten diese Funktionen als Hardware-Feature an, sodass es möglich ist die Kontrolle über den Cache-Inhalt an das Betriebssystem zu übergeben. Das Forschungsbetriebssystem **CyPhOS** [2] ist speziell für die Verwendung dieser Hardware-Funktionen entworfen worden. Durch ein komponentenbasiertes Design ist jeder Treiber und jede Anwendung in einen kleinen Speicherbereich begrenzt, sodass sich diese Komponenten hervorragend für ein schnelles Ein- bzw. Auslagern in den Cache eignen.

Die Hardware-Funktion *lockdown by way* auf der ARM-Architektur erlaubt das Partitionieren des Caches auf einer Granularität pro Prozessorkern. Eine ähnliche Funktion wird auf der X86-Architektur durch die *Intel Cache Allocation Technology* bereitgestellt. Durch diese Funktionen, kann CyPhOS vor der Aktivierung jeder Komponente, die gesamte Komponente aus dem Hauptspeicher in einen dedizierten Teil des Caches laden und so eine Ausführung ohne Hauptspeicherzugriffe ermöglichen.

Die Evaluation dieses Verfahren hat gezeigt, dass sich die relative Standardabweichung für synthetische Speicherlasttests von ca. 25 % auf ca. 1 % reduzieren lässt. Die Differenz der durchschnittlichen Laufzeit zur maximal gemessenen Laufzeit konnte für diese Tests um bis zu 98 % reduziert werden. Diese Ergebnisse sind exemplarisch für die NXP i.MX6 Plattform

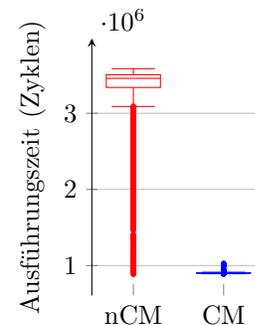


Abbildung 1: Ausführungszeiten von synthetischen Speicherlasttests ohne (nCM) und mit (CM) Cache-Verwaltungstechnik auf der NXP i.MX6 Plattform

durch Abbildung 1 dargestellt. Für die Evaluation wurden zwei ARM-basierte und eine Intel Xeon-basierte Plattform verwendet.

Weitere Benchmarks mittels der TACLe-Benchmarksuite [3] haben gezeigt, dass auch für reale Anwendungsszenarien die Schwankungen der Laufzeit erheblich reduziert werden konnten.

LITERATUR

- [1] A. Wolfe, “Software-based cache partitioning for real-time applications”, in *Proceedings of Workshop on Responsive Computer Systems*, 1993.
- [2] H. Borghorst und O. Spinczyk, “CyPhOS – A Component-based Cache-Aware Multi-Core Operating System”, in *Proceedings of the 32th International Conference on Architecture of Computing Systems (ARCS '19)*, to appear, 2019.
- [3] H. Falk, S. Altmeyer, P. Hellinckx, B. Lisper, W. Puffitsch, C. Rochange, M. Schoeberl, R. B. Sørensen, P. Wägemann und S. Wegener, “TACLeBench: A Benchmark Collection to Support Worst-Case Execution Time Research”, in *16th International Workshop on Worst-Case Execution Time Analysis (WCET 2016)*, M. Schoeberl, Hrsg., Ser. OpenAccess Series in Informatics (OA-SICs), Bd. 55, Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2016, 2:1–2:10.