Ein Betriebssystemansatz zum Energiesparen ohne Performanzverlust

Abstract

Jan Richling[1], Jan Schönherr[1], Gero Mühl[1], Matthias Werner[2]

[1] Fachgebiet Kommunikations- und Betriebssysteme, TU Berlin,
{richling|schnhrr|gmuehl}@cs.tu-berlin.de
[2] Professur für Betriebssysteme, TU Chemnitz, mwerner@cs.tu-chemnitz.de

Durch die konsequente Umsetzung der Multikernstrategie erhöhen die Prosessorhersteller die Anzahl der Kerne pro Prozessor fortlaufend, wodurch auch die erreichbare Ausführungsparallelität immer weiter zunimmt. Auf der anderen Seite ist die Parallelität, die existierende Anwendungen sinnvoll nutzen können, in vielen Fällen beschränkt. Deshalb muss damit gerechnet werden, dass Systeme mit sehr vielen Kernen in vielen Anwendungsfällen zumindest zeitweise mit einer Parallelität betrieben werden, die deutlich geringer als die Anzahl der Kerne ist. Neben dem Trend zu Mehrkernprozessoren ist in den letzten Jahren das Bewusstsein für den Energieverbrauch von Computersystemen in den Vordergrund getreten. Eine Reihe von Technologien wurde vorgestellt, um insbesondere den Energieverbrauch von Mehrkernprozessoren zu senken. Das teilweise oder komplette Abschalten sowie das Heruntertakten einzelner Kerne werden von modernen Prozessoren aller Hersteller unterstützt, so dass dem Betriebssystem eine Reihe von Möglichkeiten geboten wird, den Energieverbrauch im Teillastbetrieb zu senken.

In aktuellen Betriebssystemen existiert jedoch an genau dieser Stelle ein Problem: Infolge des „Teile und Herrsche“-Ansatzes, der die Funktionalitäten des Schedulers (welche Task wird wann wo ausgeführt) und des Energie-Governors (welcher Kern wird wann mit welcher Frequenz betrieben) streng trennt, werden die nicht-funktionalen Abhängigkeiten zwischen diesen beiden Funktionalitäten komplett ignoriert. Der Scheduler benutzt die verfügbaren Kerne fair, während der Governor per Polling die Auslastung der einzelnen Kerne ermittelt und dann Entscheidungen hinsichtlich der Nutzung der von der Hardware gebotenen Energiesparmechanismen trifft. Diese getrennte Vorgehensweise degradiert jedoch signifikant die Performanz des Systems [1], da der Scheduler ohne Kenntnis der Entscheidungen des Governors zumeist Tasks Kernen zuweist, die sich gerade in einem Energiesparmodus befinden, während auf der anderen Seite die Systemsicht des Governors infolge des Pollings dem tatsächlichen Systemzustand hinterherläuft.

Im vorliegenden Beitrag wird die in [1] vorgeschlagene Lösung aufgegriffen, die Trennung der beiden auf nicht-funktionaler Ebene zusammenhängenden Funktionalitäten aufzuheben und damit Scheduling und feingranulares CPU-Energiemanagement in einer einzelnen Komponente zu vereinen. Es wird der erste Schritt auf diesem Weg vorgestellt, nämlich der Wegfall des Pollings seitens

des Governors und der direkte Aufruf von Umschaltungen der Energiesparmodi aus dem Betriebssystemscheduler heraus. Auf diese Weise muss Wissen hinsichtlich des Systemzustandes, das im Scheduler ohnehin vorhanden ist, nicht per Polling neu gewonnen werden, sondern kann direkt verwendet werden, um innerhalb des Schedulers das Energiemanagement der einzelnen Kerne so zu benutzen, wie es die aktuellen Schedulingentscheidungen erfordern. Der Beitrag diskutiert diesen Ansatz und evaluiert ihn anhand einer Reihe von Messungen unter Verwendung des Linux-Kernels auf verschiedenen Mehrkernsystemen mit einem und auch mehreren Sockeln. Abgeschlossen wird dieser Beitrag mit einer Betrachtung von Erweiterungen, die unter anderem auch den umgekehrten Weg berücksichtigen, nämlich die Anpassung von Scheduling-Entscheidungen an den aktuellen Zustand des Energiemanagements.

Literatur

1. Jan Richling, Jan H. Schönherr, Gero Mühl, and Matthias Werner. Towards energy-aware multi-core scheduling. *Praxis der Informationsverarbeitung und Kommunikation (PIK)*, 32(2):88–96, April 2009.