

## **Deterministisches Multithreading: Aspekte eines noch nicht vollständig gelösten Problems**

Franz J. Hauck  
Institut für Verteilte Systeme  
Universität Ulm

Abstract für einen Beitrag zum Fachgruppentreffen Betriebssysteme im März 2015 in Chemnitz

Multithreading führt zu nicht wiederholbarem Verhalten von Softwaresystemen, da die Umschaltzeitpunkte von Threads von vielen nicht vorhersagbaren Bedingungen abhängen, z.B. Interrupts, Bus-Arbitrierungen, Adressumsetzungen, TLB-Misses, Cache-Misses, Page-Faults, Timer-Auflösungen, Anzahl der CPUs oder Cores, um nur einige zu nennen. Aus den Forschungsarbeiten heraus, die sich mit Debugging und Fehlertoleranz beschäftigen, gibt es eine Reihe von Überlegungen, das Scheduling mehrerer Threads deterministisch zu gestalten, so dass sich eine Anwendung trotz Multithreading auf jedem System gleichartig verhält. Für ein vollständig deterministisches Softwaresystem sind neben dem Multithreading noch weitere Quellen von Nichtdeterminismus auszuschalten, die im Vortrag jedoch nicht betrachtet werden sollen.

Nicht wenige in der Praxis eingesetzte Verfahren sorgen für eine serielle deterministische Ausführung. Dies läuft jedoch der zunehmenden Parallelisierung auf der Ebene moderner Hardware entgegen, da von Multi-Core-Systemen kaum profitiert werden kann.

Im Vortrag werden bekannte Verfahren und aktuelle Ergebnisse aus unseren Forschungsarbeiten vorgestellt, die nebenläufige Ausführung und gleichzeitig ein deterministisches Multithreading ermöglichen. Interessant ist, dass alle bisher bekannten Algorithmen unterschiedliche optimale Einsatzbedingungen aufweisen, so dass es von der Anwendung, dem Lastverhalten der Klienten als auch von der eingesetzten Hardware abhängt, welche konkrete Konfiguration zum deterministischen Scheduling den höchsten Durchsatz bietet.

Am Schluss des Vortrags soll kurz das gerade angelaufene und in Kooperation mit Hans Reiser von der Universität Passau durchgeführte DFG Projekt OptSCORE angesprochen werden, in dem es unter anderem darum geht, die Konfigurationsparameter für einen deterministischen Scheduler dynamisch an die aktuellen Bedingungen anzupassen, so dass Anwendungsentwickler von der Auswahl einer optimalen Konfiguration entlastet werden.