Kurzfassung

Grafikkarten (GPUs) werden als hochparallele Coprozessoren zunehmend für allgemeine Berechnungen (GPGPU) auch im Bereich des Hochleistungsrechnens (HPC) genutzt. Die Programmierung ist z.B. mit Hilfe von CUDA für GPUs der Firma NVIDIA möglich. Eine Neuentwicklung oder Portierung von vorhandenem Code wird mit dem Ziel einer schnelleren Laufzeit der Applikation durchgeführt. Daher sind Werkzeuge zur Verbesserung des Laufzeitverhaltens für die Entwicklung besonders wichtig. Vorhandene Werkzeuge beschränken sich allerdings auf die Analyse kompletter Funktionen, welche auf der GPU ausgeführt werden (sogenannte Kernel). Eine feingranularere Erhebung und Darstellung von Performancedaten wie Laufzeit oder Zählerwerte könnte das Finden und Beheben von Flaschenhälsen erheblich vereinfachen. Dabei sind insbesondere Daten per Codezeile oder für Threadgruppen interessant. Es wird ein Tool vorgestellt, welches in Zusammenarbeit mit NVIDIA entwickelt wurde, um Performanceinformationen durch Veränderung des Binärcodes (patchen) zu sammeln und darzustellen. Neben den technischen Details des binären Patchens, werden verschiedene Tracing- und Profilingpatches diskutiert.

Abstract

Graphic cards (GPUs) are increasingly being used as highly parallel co-processors for general purpose computation (GPGPU) in the field of high performance computing (HPC). The programming of applications is possible using, for example, CUDA for GPUs made by NVIDIA. Developing new GPU-specific applications or porting existing applications to GPUs usually has the goal of optimizing application runtime. Hence, tools that help the developer improve performance of an application are particularly important. Current performance analysis tools are limited to analyzing only complete functions that are executed on the GPUs (so-called kernels). More granular data acquisition and the illustration of performance data, such as duration or counter values, could help to find and remove bottlenecks more efficiently than before. Information at the level of the source line or thread group would be most interesting for developers. A tool which was developed in cooperation with NVIDIA to collect such granular performance data using binary code patching will be presented. Besides showing the principles behind binary patching, several tracing and profiling patches used in the tool will be discussed.