C++ AMP
Programmer Massive Parallelism GPUs

Auf der C++-Konferenz GoingNative 2012 gab Herb Sutter, Microsofts verantwortlicher Softwarearchitekt für native Programmiersprachen, in seiner Keynote " C++11, VC++11 and Beyond " bekannt, dass sein Arbeitgeber, wie zur Vorstellung der Bibliothek im Juni des letzten Jahres versprochen, die AMP-Spezifikation (Accelerated Massive Parallelism) als freie Implementierung veröffentlicht hat.

C++ Accelerated Massive Parallelism (C++ AMP) beschleunigt die Ausführung von C++-Code, indem die Vorteile von parallel arbeitender Hardware, wie beispielsweise der Grafikprozessor (englisch Graphics Processing Unit – GPU), genutzt werden. Entwickler können künftig jede Art von Rechnerressource im System ansprechen und Anweisungen auf CPUs, GPUs und diskreten Grafikprozessoren verteilen. Neu kompiliert nutzen Programme alle Rechnerressourcen im System für parallele Datenanweisungen aus.

C++ AMP gestattet es Programme mit mehrdimensionalen Datenalgorithmen zu erstellen, so dass die Ausführung auf heterogenen Hardwaresystemen durch die Parallelisierung beschleunigt wird. Das C++ AMP-Modell beinhaltet mehrdimensionale Vektoren, Methoden zur Speichertransferierung, Indizierung und eine Bibliothek für mathematische Operationen. Mit den C++ AMP Spracherweiterungen lässt sich kontrollieren, wie Daten von der CPU zur GPU und wieder zurück transferiert werden, so dass die Leistung gesteigert wird. Stufenweise soll AMP später auch auf virtuelle Maschinen auf anderen Rechnern und Cloud-Architekturen erweitert werden.

Microsoft beabsichtigt die Bibliothek als offene Spezifikation zur Verfügung zu stellen, und wird sie dem Spezifikationsorganisation ANSI zur Standardisierung in C++ vorschlagen. Microsofts Entwicklungsumgebung für C++-Entwickler Visual Studio soll AMP ab der nächsten Version (2012) unterstützen, kündigte Sutter an. Und für C# und .NET sind ebenfalls AMP-Erweiterungen vorgesehen. Damit folgt C++ AMP dem Trend zur parallelen Programmierung, wie CUDA , eine von Nvidia entwickelte Technik, die es Programmierern erlaubt, Programmteile zu entwickeln, die durch den Grafikprozessor (GPU) auf der Grafikkarte abgearbeitet werden.

Auf der Konferenz lud Sutter auch Entwickler ein, die die Implementierungsreihenfolge der noch fehlenden C++11-Features beeinflussen möchten, an der MSC++11-Survey teilzunehmen. Als Erscheinungsdatum eines "halbwegs vollständig konformen" ("reasonably fully conforming") C

++11-Compilers
(nicht notwendigerweise VC++) nannte er das Jahr 2013. Das wären dann lediglich zwei Jahre nach der Verabschiedung des Standards; C++98 brauchte deutlich länger, um eine ähnliche Position zu erreichen.

Auf der Fusion-Konferenz führte Sutter auf einem APU-System von AMD eine mit C++ AMP programmierte 3-D-Visualisierung vor, bei der Tausende frei schwebender Lichtpunkte sich gegenseitig anstoßen und interagieren. Die Menge der berechneten Objekte lässt sich offenbar millionenfach steigern. Durch An- und Abschalten von GPU und diskreter Grafikkarte machte die Anwendung deutlich, dass die auf hunderte von GFlops steigende Arbeitslast nahtlos auf die zusätzlichen Rechenkerne verteilt wird und die Frame-Rate der Bilder dabei sogar noch steigt.