

Algorithm Engineering Lab Assignment 1

Brian Zahoransky (brian.zahoransky@uni-jena.de)

November 21, 2020

1. Describe how parallelism differs from concurrency.

Parallelität ist eine Weiterführung des Prinzips der Concurrency. In beiden Fällen werden mehrere Aufgaben zur gleichen Zeit bearbeitet. Dabei entsteht bei Concurrency lediglich der Eindruck von Parallelität, da Teilaufgaben von unterschiedlichen Hauptaufgaben abwechselnd bearbeitet werden. Bei echter Parallelität hingegen werden die Teilaufgaben tatsächlich zu gleichen Zeit abgearbeitet, z.B. durch mehrere Prozessorkerne.

2. What is fork-join parallelism?

Es existiert ein Hauptthread. Zu gegebenem Anlass startet dieser Nebenthreads zur Bearbeitung einer Aufgabe. Man spricht dann davon, dass der Thread gegabelt (engl. fork) wird. Nach Bearbeitung der Aufgabe werden die Threads wieder zusammengeführt (engl. join). Es gibt wieder nur einen Hauptthread. Diese Abfolge kann sich in einem Programm beliebig oft wiederholen. Zudem können auch Threads innerhalb eines solchen Zyklus nach dem gleichen Prinzip, weitere Nebenthreads starten und anschließend zusammenführen.

3. About which „new“ (2005) CPU technologies does Sutter report?

Hyperthreading: Eine Kern besitzt mehrere Register. Somit kann direkt mit einem weiteren Thread weitergearbeitet werden, falls in einem Thread Daten oder Instruktionen aus einer höheren Speicherhierarchie geladen werden müssen. Multicore: Echte Parallelität. Jeder Kern kann an einer Aufgabe (Thread) arbeiten. Caches: Zur Überbrückung der Latenz des Arbeitsspeichers. Daten die benötigt werden, können in den Cache vorgeladen werden, damit der Prozessor nicht hunderte Zyklen warten muss, bis die Daten vom Arbeitsspeicher geholt wurden.

4. What do you think, can we have a free lunch again if we exploit them in our applications?

Dies lässt sich derzeit nur für einen kurzen Zeithorizont (wenn überhaupt für 5-10 Jahre) absehen. Für diesen kurzen Zeitraum würde ich sagen ja, es gibt noch einen "free lunch". Dieser Annahme liegen mehrere Beobachtungen zu Grunde. Zum einen wachsen momentan noch die Cachegrößen recht regelmäßig. Dies könnte zwar jeder Zeit zum Stillstand kommen, da schnelle Caches eine Frage des Platzes sind und auch schon das Limit der Clock-Speed für die Hersteller nicht ganz unüberraschend kam (zumindest dem Artikel zufolge). Doch selbst, falls die Caches widererwartend plötzlich aufhören würden zu wachsen, so kann man davon ausgehen, dass die Anzahl der Kerne pro Prozessor noch weiterhin steigt. Das heißt, falls man ein Programm so schreibt, das es für eine wachsende Anzahl von Prozessoren ausgelegt ist, wird man noch ein paar Jahre von einem "free lunch" sprechen können.