

Testen Sie Ihre Parallelisierungs-Kenntnisse

Persönliche Daten:

Name:

Firma:

E-Mail Adresse:

Erzählen Sie uns etwas über sich und Ihr Interesse für Parallelisierung:

Was kann man über Multicore-Prozessoren sagen?

Parallelisierung verbessert die Leistung

Die Verwendung von Multicore-Prozessoren erhöht den Stromverbrauch

Mehrere Kerne verbessern das Multitasking

Die Cache-Kohärenz vereinfacht die Programmierung

Ist zu erwarten, dass sequentielle (nicht parallele) Programme auf Multicore-Prozessoren immer schneller werden?

Ja, die Single-Core-Leistung verbessert sich immer noch stetig

Generell „Nein“ - es ist sinnvoller, die Leistung durch Parallelität zu verbessern

Nein, die Single-Core-Leistung verschlechtert sich, je mehr Kerne einem Prozessor hinzugefügt werden

Angenommen, Sie haben ein Programm parallelisiert und es geschafft, seine Laufzeit von 10 s auf 4 s zu reduzieren. Was ist die daraus resultierende Beschleunigung?

- 4
- 2,5
- 2
- 0,4

Wenn ein Programm zu 90% parallel und zu 10% sequentiell ist, mit welcher Beschleunigung können Sie theoretisch im besten Fall rechnen?

- Unbegrenzt
- 90
- 10
- 9

Angenommen, Sie haben viel Zeit damit verbracht, ein Programm für die Ausführung auf zwei Kernen zu parallelisieren. Ist es möglich, den gleichen Aufwand zu vermeiden, wenn das Programm an vier, acht oder mehreren Kernen angepasst wird?

Ja, indem man versucht, das Programm in so viele unabhängige Aufgaben wie möglich zu zerlegen

Nein, ein Programm muss für eine bestimmte Anzahl von Kernen parallelisiert werden

Wann würden Sie ein paralleles Programm als skalierbar betrachten?

Wenn es mit mehr Prozessoren oder Kernen schneller läuft

Wenn Sie mehr Prozessoren oder Kerne verwenden, können Sie die Problemgröße erhöhen, und das Programm wird nicht langsamer

Wenn die Beschleunigung mit N Prozessoren oder Kernen genau N ist

Was sind mögliche Gründe für eine nicht perfekte Beschleunigung paralleler Programme?

Kommunikation und Synchronisation zwischen den Prozessen / Threads

Viel mehr unabhängige Aufgaben als es Prozessoren / Kerne gibt

Eine ungleichmäßige Aufgabenverteilung

Eine große sequentielle Aufgabe, die nicht parallelisiert werden kann

Was ist eine Race Condition?

Eine Race Condition entsteht, wenn mehrere Prozesse / Threads gegeneinander antreten, um zuerst fertig zu werden

Eine Race Condition ist ein schwerwiegender Fehler, bei dem das Timing oder die Reihenfolge der Ereignisse die Richtigkeit eines Programms beeinflusst

Kann Parallelisierung eine Race Condition begünstigen?

Ja, parallele Programme sind vom Aufbau her nicht deterministisch

Ja, bei falscher oder unzureichender Synchronisation

Nein, parallele Programme sind vom Aufbau her deterministisch

Beschreiben Sie, wie Sie verhindern können, dass mehrere Prozesse / Threads konkurrierende Änderungen an einer gemeinsam genutzten Ressource vornehmen.