Bitte bearbeiten Sie die folgenden Aufgaben in Zweier- oder Dreiergruppen. Wenn Sie Feedback zu Ihren Lösungen haben möchten, notieren Sie Ihre Lösungen in einem Textdokument und schicken Sie es mir per E-Mail (bitte keine Mehrfachabgabe identischer Lösungen).

18. Parallelisierung

Folgendes Programm (Pseudocode) soll durch Parallelisierung beschleunigt werden:

```
program seqentiell {
  int records[1000];          // Platz für 1000 Records
  int sums[100];              // Platz für 100 Zwischenwerte
  int sum = 0;                // Summe
  load_from_disc (records);   // füllt die 1000 Records mit Werten von der
                               // Platte, benötigt 100 ZE
  for (i in 0..100) {
    sums[i] = 0;
    for (j in 0..10) {
      sums[i] = sums[i] + records[i*10 + j];
                              // jede Addition benötigt 5 ZE
    }
  sum = sum + sums[i];        // auch hier: 5 ZE pro Addition
  }
  write_to_disk (sums);       // schreibt die Ergebnisse auf die Platte,
                              // benötigt 10 ZE
  write_to_disk (sum);        // schreibt sum auf Platte, benötigt 1 ZE
}
```

a) Welche Berechnung erledigt das Programm? Wie können Sie es offensichtlich parallelisieren?

b) Wie viele Zeiteinheiten benötigt das sequentielle Programm, wenn Sie die nötigen Zeiten für die Verwaltung der Schleifen ignorieren?

c) Wie viele Zeiteinheiten benötigt das parallele Programm, wenn Ihnen 10, 100 oder 1000 Prozessoren zur Verfügung stehen und Sie die nötige Zeit für das Erzeugen von Threads ignorieren? Welchem Speed-up entspricht das?

d) Verwenden Sie Amdahls Gesetz, um den Grenzwert für den Speed-up zu bestimmen; berechnen Sie dazu zunächst den Wert α (den Anteil des nicht-parallelisierbaren Codes).

19. Asynchronous RPC

a) Überlegen Sie sich zwei sinnvolle Beispiele für den Einsatz von Asynchronous Remote Procedure Calls – einmal eine Variante, bei der ein Rückgabewert geliefert wird, und einmal eine Variante, die keinen Rückgabewert braucht. Die Beispiele sollen so gestaltet sein, dass der Einsatz von Asynchronous RPC einen echten (Zeit-) Vorteil gegenüber klassischem RPC bringt.

b) Wie könnte das Programm von Aufgabe 17 aussehen, wenn Ihnen ein RPC-Server zur Verfügung steht, der eine asynchrone RPC-Prozedur `add100` exportiert? Dieser Funktion übergeben Sie ein Array von 100 Integers und erhalten die Summe zurück. Der Server ist so ausgelegt, dass er bis zu fünf Anfragen echt parallel bearbeiten kann. Der Rückgabewert von `add100()` soll eine Request-ID sein, die Sie verwenden können, um mit `get_result(id)` das Ergebnis der Berechnung abzufragen.

20. Cloud vs. Grid

Nennen Sie zwei wesentliche Kriterien, in denen sich Cloud und Grid unterscheiden, und erklären Sie, warum diese für die Popularität der Clouds relevant sind.

21. Cloud

Lesen Sie Kapitel 1–3 (Seiten 1–6) des Artikels „Above the Clouds: A Berkeley View of Cloud Computing“ (im Campus-System verfügbar, alternativer Download unter: http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf) und beantworten Sie die folgenden Verständnisfragen:

a) Was ist der Unterschied zwischen „Public Clouds“ und „Private Clouds“?

b) Was sind laut dem Artikel sinnvolle Gründe für ein Unternehmen, um sich zu entscheiden, selbst ein Cloud-Anbieter zu werden?

c) Warum stehen die Rechenzentren von Cloud-Anbietern teilweise in Gegenden, die keine klassischen Standorte für IT-Unternehmen und teils weit vom Firmensitz des jeweiligen Unternehmens entfernt sind?