

Britta Nestler, Alexander Vondrous, Fakultät IWI, Hochschule Karlsruhe

## Klausur zu High Performance Computing

12. Juli 2013

Es sind keine Hilfsmittel zugelassen.

Bearbeitungszeit: 90 Minuten

---

### Aufgabe 1: Nebenläufigkeit (2P)

Moors Law beschreibt, dass sich die Transistoren pro Flächeneinheit etwa alle 2 Jahre verdoppeln. Dieses Gesetz ist bis heute gültig und wird auch in den nächsten Jahren gültig bleiben. Auch wenn Moors Law bestand hat, begann vor wenigen Jahren ein Paradigmenwechsel bei der Prozessorentwicklung.

- a) Erläutern Sie in wenigen Worten welcher Paradigmenwechsel bei der Entwicklung von Prozessoren stattgefunden hat um bei gleichbleibendem Takt eine höhere Rechenleistung zu erzielen. (0,5P)
- b) Welche Konsequenz hat dieser Paradigmenwechsel für die Entwicklung leistungsfähiger Software? (0,5P)
- c) Nennen und Beschreiben Sie kurz 2 Designkriterien für parallele Software, die bei der Entwicklung Nebenläufiger Software berücksichtigt werden sollen. (1P)

### Aufgabe 2: Parallele Leistungsmaße Teil 1 von 2 (3P)

Der neue Zentralrechner der Terranischen Liga auf Korhal IV soll zur Kryptanalyse von Protoss Nachrichten genutzt werden. Die Wissenschaftler auf Korhal testen eine zentrale Komponente des Softwarenervensystems um eine hohe Effizienz zu erhalten. Analysieren Sie als Leiter der Abteilung für Hochleistungsrechnen die Messergebnisse.

- a) Berechnen Sie Speedup und Effizienz der Testläufe. Tragen Sie Ihre Ergebnisse in die Tabelle ein. (2P)
- b) Welche Konfiguration würden Sie für den Produktiveinsatz vorschlagen? Begründen Sie Ihre Antwort kurz. (1P)

CPUs	Laufzeit	Speedup	Effizienz
1	4 800	-	-
6	960		
110	48		
603	8		
1600	6		

**Aufgabe 3: Parallele Leistungsmaße Teil 2 von 2 (2P)**

Eine weitere Komponente des Softwarenervensystems wurde getestet. Weil die Problemstellung von Natur aus sehr groß ist, soll jeweils der relative Speedup zur vorherigen Messung berechnet werden.

- a) Berechnen Sie den relativen Speedup basierend auf der jeweils vorherigen Messung für die untere Tabelle und tragen Sie Ihre Ergebnisse in die Tabelle ein. Bestimmen Sie auch den idealen relativen Speedup zur jeweils vorherigen Messung. (1P)
- b) Würden Sie die Effizienz des Programms als gut oder als schlecht einschätzen? Begründen Sie in wenigen Worten Ihre Antwort. (1P)

CPUs	Laufzeit	relativer Speedup	Idealer relativer Speedup
60	540	-	-
210	180		
840	60		

**Aufgabe 4: Amdahl und Gustafson (4P)**

Für die Sprungvorhersage der Kreuzer im Weltall besitzt jedes Schiff eine handelsüblichen JUMP-42 Chip, der mit 21 Prozessoren ( $P = 21$ ) Rechnet. Trotz seiner 21 Recheneinheiten erzielt der JUMP-42 Chip lediglich einen Speedup von 15 ( $S = 15$ ). Bestimmen Sie für das zentrale Rechenzentrum der Konföderation  $t_s$  und  $t_p$  nach Amdahl und nach Gustafson.

- a) Bestimmen Sie  $t_s$  und  $t_p$  nach Amdahl. (1,5P)
- b) Berechnen Sie welcher Speedup nach Amdahl höchstens erreicht werden kann? Lassen Sie dafür die Anzahl der Prozessoren gegen unendlich streben. (1P)
- c) Bestimmen Sie  $t_s$  und  $t_p$  nach Gustafson. (1,5P)

**Aufgabe 5: OpenMP Teil 1 von 2 (1,5P)**

Als einer von Kerrigans Zerebraten haben Sie die Aufgabe ein infiziertes Schlachtschiff der Liga für eine spezielle Operation zu manipulieren. Leider haben die terranischen Entwickler den Code für die geheime psionische Waffe nur fehlerhaft implementiert. Es liegt an Ihnen die Fehler im Parallelem Modul zu beseitigen.

Finden Sie den Fehler im Programm, der dazu führt, dass die Summe nicht korrekt berechnet wird und beschreiben Sie ggf. im Pseudocode Ihre Korrektur. (1,5P)

```
[...]
int    i,tid;
double sum = 0.0;
double part;

#pragma omp parallel shared (part, tid) private(sum)
{
    part = 0.0;
    tid = omp_get_thread_num();

    for (i=0; i < 42; i++) {
        part += calcFoo(tid, i);
    }
    #pragma omp atomic
    sum += part;
}

printf("sum: %lf\n", sum);
[...]
```

**Aufgabe 6: OpenMP Teil 2 von 2 (2P)**

Die verschachtelte Parallelisierung mit OpenMP macht den terranischen Entwicklern Schwierigkeiten, weil der gleiche Code auf unterschiedlichen Maschinen unterschiedliche Ergebnisse erzeugt. Erstellen Sie die möglichen Ausgaben des folgenden Programmes mit und ohne nested parallelization.

- Welche Ausgabe wird bei eingeschalteter Schachtelung (nesting) erscheinen? (1P)
- Auf einigen Rechnern ist nesting ausgeschaltet. Welche Ausgaben wird sich bei ausgeschaltetem Nestling zeigen? Schreiben Sie die Ausgabe in verkürzter Form auf. (1P)

```
[...]
#pragma omp parallel num_threads(3)
{
    printf("LVL 1   ID: %ld\n", omp_get_thread_num());
    #pragma omp parallel num_threads(3)
    {
        printf("LVL 2.1   ID: %ld\n", omp_get_thread_num());
    }
    #pragma omp parallel num_threads(3)
    {
        printf("LVL 2.2   ID: %ld\n", omp_get_thread_num());
    }
}
[...]
```

### Aufgabe 7: MPI Teil 1 von 2 (2P)

Weil die terranische Liga Kolonien am Rande des Koprulu Sektors besitzt, haben Ingenieure der Randzone typischerweise MPI zur parallelen Datenverarbeitung genutzt. Leider konnten Sie Ihr Projekt nicht testen, weil deren Bewusstsein von einer psionischen Masse unbekannter Herkunft entführt wurde. Weil Sie mit den Köchen aus der Kantine eine Zwist angefangen haben, müssen Sie das tote Projekt vervollständigen. Ihnen fallen bestimmte Stellen im Code auf, die nicht funktioniert haben können. Finden Sie die Fehler.

```
[...]
int result[1];
int data[1];

if (rank != 1) {
    data = getData();
    MPI_Allreduce( data, result, 7, MPI_INT, MPI_SUM, MPI_COMM_WORLD );
}
[...]
```

- Finden und beheben Sie die 2 Fehler, die sich in den obigen Quelltext befinden. (1P)
- Wozu dient die `MPI_Allreduce(...)` Funktion? Ergänzen Sie Ihre Antwort durch 2 Beispiele. (1P)

### Aufgabe 8: MPI Teil 2 von 2 (4P)

Auch nach der erfolgreichen Wiederbelebung des alten Projektes hat Sie Jim Raynor nach wiederholten Auseinandersetzungen mit Kachinsky zur Korrektur der Raumstabilisatoren der Hyperion verdonnert. Korrigieren Sie die Fehler in der Ansteuerung der Stabilisatoren.

```
[...]
long localmax;

if (rank != 1) {
    for (int i = 0; i < num_workers-1; i=i+2) {
        MPI_Irecv(&localmax, 1, MPI_LONG, i, 123, MPI_COMM_WORLD, &request);
    }
    MPI_Wait(&request, &status);
    globalmax = findMaxValue(localmax, globalmax);
    doImportantCalculation(globalmax);
} else {
    MPI_Isend(&localmax, 1, MPI_LONG, 1, 123, MPI_COMM_WORLD, &request);
    someFooBarLongCalc();
    MPI_Wait(&request, &status);
}
[...]
```

- Welche Funktionalität sollte laut der Variablenbenennung implementiert werden? (0,5P)
- Beheben Sie die Fehler im Code, so dass der Code das macht, wozu er gedacht ist. (Für Aufgabenteil c) bitte umblättern) (2P)

- c) Mit welcher MPI Funktion kann diese Funktionalität wesentlich übersichtlicher gestaltet werden. Implementieren Sie im Pseudocode die verbesserte Variante. (1,5P)

### Aufgabe 9: Performance Modellierung (4P)

Ein Programm zur Bestimmung der Weglänge von Streckenabschnitten soll mit dem Performance Modell nach Van der Velde untersucht werden. Eine Matrix enthält alle Vektoren, deren Länge bestimmt und aufsummiert werden soll. Gehen Sie davon aus, dass die Problemgröße glatt durch die Anzahl Prozessoren geteilt wird. Die Wurzelberechnung `sqrt(...)` wird als eine Operation angesehen.

**Hinweise:** Verwenden Sie die Notation der folgenden Tabelle

$N$	Problemgröße
$p$	Anzahl Prozesse
$t_a$	Rechenzeit je Operation
$t_k$	Kommunikationszeit
$T_s$	Sequentielle Laufzeit
$T_p$	Partielle Laufzeit

#### Sequentiell

```
len = 0.0;
for (i = 0; i < N; i++) {
    len += sqrt(matrix[i][X] * matrix[i][X] +
               matrix[i][Y] * matrix[i][Y] +
               matrix[i][Z] * matrix[i][Z]);
}
```

#### Parallel

```
len = 0.0;
// each parallel task has its own matrix of size N/p
for (i = 0; i < N/p; i++) {
    len += sqrt(matrix[i][X] * matrix[i][X] +
               matrix[i][Y] * matrix[i][Y] +
               matrix[i][Z] * matrix[i][Z]);
}

if (myID == 0) {
    for (i = 1; i < p; i++) {
        len += receive from i
    }
} else {
    send len to 0
}
```

- a) Erstellen Sie das Performance Modell nach Van der Velde für das sequentielle Programm. (1P)
- b) Erstellen Sie das Performance Modell nach Van der Velde für das parallele Programm. (2P)
- d) Nennen Sie 2 sinnvolle Anwendungen für Performance Modelle. (1P)

**Aufgabe 10: OpenCL (1,5P)**

Der Supercomputer ATLAS des Supertransporters Nagglfar basiert auf alten aber dennoch effizienten Grafikprozessoren. Um nach dem Absturz auf Tarsonis die Rechenleistung für das Terraformingvorhaben zu nutzen, müssen sehr große Datenlisten (`sectorarray`) verarbeitet werden. Weil es nicht effizient ist mehr Threads als die Hardware besitzt zu verwenden, müssen Sie einen OpenCL Kernel schreiben, der eine Liste der Länge `arraylength` mit einer beschränkten Anzahl OpenCL threads (`threads`) verarbeiten kann. Jedes Element der Liste muss von der bereits implementierten Funktion

`calcSektor(sector)` aufgerufen werden. Sie müssen davon ausgehen, dass `arraylength > threads` ist. Nutzen Sie dafür das folgende Grundgerüst.

**Hinweis:** Bedenken Sie, dass alle Threads Zugriff auf die globale Liste haben.

```
// OpenCL kernel function for list computation
__kernel void calcTerraforming(__global sector* sectorarray,
                               int arraylength,
                               int threads) {

    // get thread id
    int threadID = get_global_id(0);

    // Call calcSektor(..) for all sectors here

}
```

### Aufgabe 11: Kommunikationsnetze Teil 1 von 2 (2,5P)

Der Supercomputer ATLAS wurde auf Tarsonis wieder reaktiviert. Er enthüllt ein psionisches Netzwerk der Xel'Naga auf dem Zerg Planeten Kaldir. Als Wissenschaftler haben Sie die Aufgabe dieses Netzwerk für die Liga zu untersuchen. Bestimmen Sie Kennzahlen und tragen Sie diese in die unten angegebene Tabelle ein.

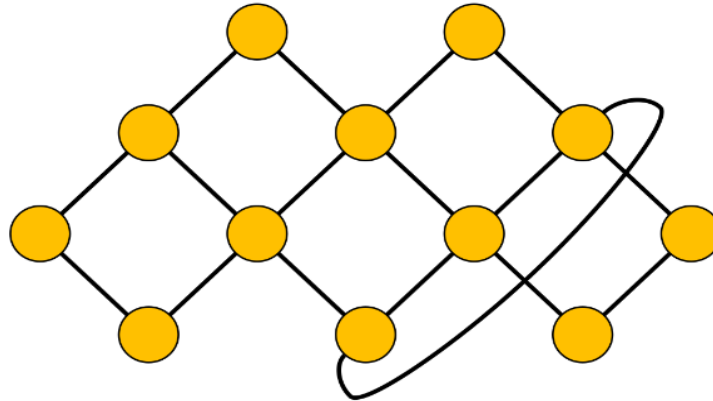


Abbildung 1: Psionisches Xel'Naga Netzwerk

Durchmesser	
Grad (max/min)	/
Bisektionsbandbreite	
Konnektivität	

### Aufgabe 12: Kommunikationsnetze Teil 2 von 2 (1,5P)

Durch den Technologietransfer kann die Liga diese psionischen Kräfte zur Kontrolle von Zerg Einheiten einsetzen. Leider muss dafür ein bestimmtes Netzwerk errichtet werden.

- Erstellen Sie ein Netzwerk, dass einen Durchmesser von 3 besitzt. (0,5P)
- Erstellen Sie ein Netzwerk, dass einen Durchmesser von 3 und den Grad 4 besitzt. (0,5P)
- Erstellen Sie ein Netzwerk, dass einen Durchmesser von 3, den Grad 4 und eine Konnektivität von 2 besitzt. (0,5P)

**Hinweis:** Es kann passieren, dass der Entwurf aus Teilaufgabe b) für die Lösung von Teilaufgabe c) ungeeignet ist, so dass evtl. ein anderen Ansatz gewählt werden muss.

**Viel Erfolg!**

The only way you get good at Starcraft is if you're self critical. - Day9