

## Übungen zu „Parallele Programmierung“, WS 2008/09

Nr. 10, Abgabe der Aufgaben: MITTWOCH 04. Februar 2009 per E-Mail

### 10.1 HPF-Direktiven

5 Punkte

Versehen Sie die Fortran-Codefragmente mit geeigneten `distribute` und `align` Direktiven.

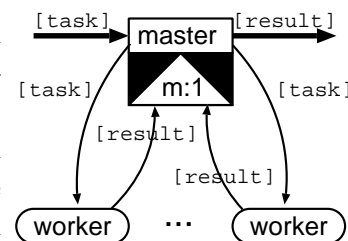
<pre>real, dimension (N,N)    :: A, A1 integer, dimension (N,N) :: IND real G</pre>	<pre>real, dimension (N,N) :: A real, dimension (N)   :: X, Y</pre>
<p>(a)</p> <pre>A = A1 A = G A(1,1:N) = A1(2,1:N) A(I,J) = A1(IND(I,J),J)</pre>	<p>(b)</p> <pre>do I = 1, N   do J = 1, N     Y(I) = Y(I) + A(I,J) * X(J)   end do end do</pre>

<pre>real, dimension (0:2*N) :: A real, dimension (0:N)   :: B</pre>	<pre>real, dimension (1:2*N-1) :: A1</pre>
<p>(c)</p> <pre>forall (I=0:N) B(I) = A(2*I) forall (I=1:2*N-1) A1(I) = A(I)</pre>	

### 10.2 Master Worker Skelett in C und MPI

7 Punkte

Schreiben Sie in C und MPI ein Skelett, durch das sich eine Menge an Aufgaben *dynamisch* von mehreren Prozessen abarbeiten lässt. Die Aufgabenverteilung soll durch einen zentralen *Master*-Prozess erfolgen, der  $p - 1$  *Worker*-Prozesse bedient. Initial erhält jeder *Worker*  $k$  Aufgaben, deren Ergebnisse nach Bearbeitung wieder zum Master zurückgeschickt werden. Für jedes empfangene Ergebnis versendet der Master eine weitere Aufgabe an den entsprechenden Worker, bis keine Aufgaben mehr vorhanden sind.



- (a) Programmieren sie ein entsprechendes Skelett in C und MPI. Als Spielbeispiel soll die Aufgabenmenge aus  $n$  zufälligen Integerwerten bestehen. Ein *Worker*, der als Aufgabe einen Wert  $m$  erhält, soll  $m * x$  leere Schleifendurchläufe abarbeiten und als Ergebnis einen *Dummy*-Wert zurückschicken. Die Werte  $n$ ,  $x$  und  $k$  sollen Parameter sein.
- (b) Diskutieren Sie, wie Ihre Version verändert werden müsste, damit sowohl die von den *Workern* abzuarbeitende Funktion, als auch der Typ der Aufgaben und Ergebnisse parametrisierbar sind.