PHILIPPS-UNIVERSITÄT MARBURG
Fachbereich Mathematik und Informatik
Prof. Dr. R. Loogen

D-35032 Marburg
Hans Meerwein Straße
Lahnberge

29. April 2002

Übungen zu „Parallele und verteilte Algorithmen“, Sommer 2002

Nr. 3, Abgabe: 7. Mai in der Vorlesung

Die Abgabe ist in Gruppen bis zu zwei Personen erlaubt. Programme sind schriftlich (etwa als Ausdruck) **und** per email an `pinf3.mathematik.uni-marburg.de` abzugeben.

---

7. PVM Broadcast-Analyse [10 P.]

   Schreiben Sie ein PVM-Programm, das einen Broadcast für eine vorgegebene Anzahl von Prozessen möglichst effizient durch Punkt-zu-Punkt-Nachrichten realisiert. Vergleichen Sie die Laufzeit mit der direkt in PVM implementierten Broadcastfunktion `pvm_mcast()`.

   Untersuchen Sie, ob es Laufzeitunterschiede zwischen den PVM-Routinen `pvm_mcast()` zum Broadcast von Nachrichten an alle Prozesse und `pvm_bcast()` zum Broadcast von Nachrichten innerhalb von Prozessgruppen gibt.

   Hinweis: Modifizieren Sie das im Verzeichnis `PVM_ROOT/examples` vorhandene Programm `timing.c` in geeigneter Weise.

8. Pipeline-Broadcast im Hypercube [4 P.]

   Das Standardverfahren für einen Broadcast im Hypercube der Dimension k benötigt k Schritte, bei denen sukzessive $2^0, 2^1, \ldots, 2^{k-1}$ Kommunikationen gleichzeitig stattfinden. Die in der Vorlesung vorgestellte Optimierung von Johnsson und Ho zerlegt die Nachricht in k Pakete, die pro Schritt in die k Dimensionen des Hypercubes verteilt werden. Dabei finden sukzessive $k*2^0, k*2^1, \ldots, k*2^{k-1}$ Kommunikationen gleichzeitig statt.

   Das Standardverfahren kann alternativ dadurch optimiert werden, dass große Nachrichten fließbandartig verbreitet werden, d.h. die Nachricht wird in Pakete zerlegt, die direkt nacheinander verschickt werden: Befindet sich Paket i in Phase j des Broadcasts, so befindet sich das Paket $i+1$ bereits in Phase $j-1$.

   Analysieren und vergleichen Sie den Aufwand dieser drei Broadcast-Verfahren unter der vereinfachenden Annahme, dass die Übertragungszeit einer Nachricht linear in der Nachrichtengröße ist.

9. Untere Schranken für paralleles Sortieren [6 P.]

   Begründen Sie die Korrektheit der im folgenden angegebenen unteren Schranken für das Sortieren von n Elementen auf verschiedenen Netzwerken mit jeweils n Knoten. Vor und nach dem Sortiervorgang sollen die zu sortierenden Elemente gleichmäßig verteilt sein, d.h. ein Element pro Prozessor.

   (a) $\Omega(n)$ auf einem eindimensionalen Gitter

   (b) $\Omega(\sqrt{n})$ auf einem zweidimensionalen Gitter

   (c) $\Omega(\log n)$ auf einem Shuffle-Exchange-Netzwerk