Prof. Dr. R. Loogen, M. Dieterle, M. Grebe
Fachbereich Mathematik und Informatik
Hans-Meerwein-Straße
D-35032 Marburg

2. November 2011

Philipps Universität Marburg

Übungen zu „Parallele und Verteilte Algorithmen“, Winter 2011/12

Nr. 3, Abgabe der Aufgaben: 9. November 2011 vor der Vorlesung

---

Aufgaben

3.1 Kostenoptimalität der parallelen Präfixsummenberechnung — 6 Punkte

(a) Zeigen Sie, dass Sie durch Anwendung des Satzes von Brent für den parallelen Algorithmus zur Präfixsummenberechnung keine Kostenoptimalität nachweisen können, wenn Sie die Anzahl der benutzten Prozessoren auf $\frac{n}{logn}$ reduzieren. / 2

(b) Alternativ wird folgende Variante zur Präfixsummenberechnung von n Werten auf $p < n - 1$ Prozessoren vorgeschlagen:

i. Aufteilen der n Werte in p Teilbereiche mit nicht mehr als $\lceil \frac{n}{p} \rceil$ Elementen.

ii. Alle p Prozessoren benutzen das optimale sequentielle Verfahren um Präfixsummen auf ihren Teilbereichen zu berechnen.

iii. *Die ersten* $p-1$ *Prozessoren* berechnen die Präfixsummen der Gesamtsummen der Teilbereiche mit dem bekannten parallelen Verfahren.

iv. *Die letzten* $p - 1$ *Prozessoren* addieren die Gesamtsumme der niedrigeren Blöcke auf die Elemente ihres Teilbereichs.

Weisen Sie die Kostenoptimalität des hier beschriebenen Verfahrens für $p = \frac{n}{logn}$ nach. / 4

3.2 All-Gather mit Shuffle-Exchange Verbindungen — 6 Punkte

All-Gather ist eine verteilte Operation, bei der jeder Prozess die Werte aller Prozesse sammelt. Sie kann als Spezialfall der Reduktion gesehen werden, die in der VL für das Shuffle-Exchange Netzwerk definiert wurde, wobei die Reduktionsoperation + der Konkatenation von Listen +++ entspricht. Allerdings ist darauf zu achten, dass die Teillisten in der Reihenfolge ihrer Verteilung über die Prozesse zusammengesetzt werden (+++ ist nicht kommutativ). Implementieren Sie All-Gather mit Scala Actors. Folgen Sie dabei dem Algorithmus der VL und nutzen Sie nur Shuffle-Exchange Verbindungen.