

Inhalt

| | |
|---|-----------|
| Einführung | 1 |
| 1. Parallele Rechnerarchitekturen | 3 |
| 1.1. Vektorrechner | 3 |
| 1.2. Multiprozessoren | 4 |
| 1.2.1. Speicherkopplung mit global gemeinsamem Speicher (Beispiele Alliant FX/8, Encore Multimax, Sequent Balance) | 5 |
| 1.2.2. Speicherkopplung mit global verteiltem Speicher (Beispiel DIRMU 25) | 7 |
| 1.2.3. Nachrichtenkopplung (Beispiel SUPRENUM) | 9 |
| 1.3. Kommunikationsmechanismen in Multiprozessoren | 12 |
| 1.3.1. Programmiermodell bei nachrichtengekoppelten Systemen | 13 |
| 1.3.2. Kommunikation in nachrichtengekoppelten Systemen | 13 |
| 1.3.3. Programmiermodell bei speichergekoppelten Systemen | 14 |
| 1.3.4. Kommunikation in speichergekoppelten Systemen | 14 |
| 2. Sequentielle Lösung von großen, linearen Gleichungssystemen mit dünn besetzter Koeffizientenmatrix | 17 |
| 2.1. Definitionen | 17 |
| 2.2. Direkte Verfahren | 19 |
| 2.2.1. Datenstrukturen | 19 |
| 2.2.2. Pivotisierung | 21 |
| 2.3. Iterative Verfahren | 22 |
| 2.3.1. Eingitterverfahren | 23 |
| 2.3.2. Mehrgitterverfahren | 24 |
| 2.4. Semi-iterative Verfahren | 26 |
| 2.4.1. Methode der konjugierten Gradienten | 26 |
| 2.4.2. Vorkonditionierung | 27 |
| 3. Parallelisierung der Algorithmen | 31 |
| 3.1. Datenabhängigkeitsanalyse | 32 |
| 3.2. Vektorisierung | 34 |
| 3.2.1. Direkte Verfahren | 34 |
| 3.2.2. Iterative Verfahren | 35 |
| 3.2.3. Semi-iterative Verfahren | 36 |

| | |
|--|-----------|
| 3.3. Nebenläufigkeit | 37 |
| 3.3.1. Aufteilung des Lösungsgebiets | 38 |
| 3.3.2. Direkte Verfahren | 38 |
| 3.3.3. Iterative Verfahren | 40 |
| 3.3.4. Semi-iterative Verfahren | 41 |
| 3.4. Kommunikationstopologien | 42 |
| 3.4.1. Prozessorkonfigurationen für DIRMU 25 | 43 |
| 3.4.2. Kommunikationstopologien für SUPRENUM | 43 |
| 4. Implementierung der Algorithmen auf unterschiedlichen Multiprozessoren | 49 |
| 4.1. Geschwindigkeitsgewinn und Effizienz | 49 |
| 4.2. Effizienzverluste bei parallelen Algorithmen | 53 |
| 4.3. Implementierung auf Multiprozessoren mit global gemeinsamem Speicher | 55 |
| 4.3.1. FORCE - eine parallele Programmiersprache | 55 |
| 4.3.2. Ergebnisse für Multiprozessoren mit global gemeinsamem Speicher | 58 |
| 4.3.2.1. Direkte Verfahren | 58 |
| 4.3.2.2. Semi-iterative Verfahren | 58 |
| 4.3.2.3. Iterative Verfahren | 61 |
| 4.4. Implementierung auf Multiprozessoren mit verteiltem Speicher | 62 |
| 4.4.1. Modula-2 auf DIRMU 25 | 63 |
| 4.4.2. Implementierungen auf DIRMU 25 | 63 |
| 4.4.2.1. Direkte Verfahren | 64 |
| 4.4.2.2. Iterative Verfahren | 65 |
| 4.5. Implementierung auf SUPRENUM | 67 |
| 4.5.1. Concurrent Modula-2 für SUPRENUM | 67 |
| 4.5.2. MIMD Fortran für SUPRENUM | 68 |
| 4.5.3. SUPRENUM Simulationssystem | 69 |
| 4.5.4. Ergebnisse für SUPRENUM | 70 |
| 4.5.4.1. Direkte Verfahren | 70 |
| 4.5.4.2. Semi-iterative Verfahren | 73 |
| 4.5.4.3. Iterative Verfahren | 77 |
| 4.6. Zusammenfassung der Ergebnisse | 80 |
| 5. Bewertung der Ergebnisse | 81 |
| 5.1. Algorithmische Konsequenzen | 81 |
| 5.1.1. Entwicklung von neuen, parallelen und vektorisierbaren Algorithmen | 81 |
| 5.1.2. Formulierung von Algorithmen für parallele Rechnerstrukturen | 83 |

| | |
|---|----------------|
| 5.2. Konsequenzen für die Rechnerarchitektur | 83 |
| 5.2.1. Reduktion der Kosten für lokale Kommunikation | 84 |
| 5.2.2. Hardwareunterstützung bei globalen Kommunikationsvorgängen | 88 |
| 5.2.3. Entwicklung eines Benchmark-Konzepts für Multiprozessoren | 89 |
| 5.3. Parallele Entwicklungsumgebungen | 89 |
| 5.3.1. Simulationssysteme | 90 |
| 5.3.2. Parallele Debugger | 90 |
| 5.3.3. Diagnoseunterstützung | 91 |
| 6. Eine parallele, architekturunabhängige Programmierumgebung | 93 |
| 6.1 Existierende Ansätze für parallele Programmiersprachen | 93 |
| 6.1.1 VSM - Virtual Shared Memory | 93 |
| 6.1.2 Virtuelle Kanäle | 94 |
| 6.1.3 Hilfsmittel für den INTEL Hypercube | 95 |
| 6.1.4 Autoparallelisierer für Algorithmen auf SUPRENUM | 96 |
| 6.2. Die parallele Programmierumgebung PPRC | 97 |
| 6.3. Feldkonfiguration: PPRC Programm für ein iteratives Verfahren | 103 |
| 6.4. Baumkonfiguration: PPRC Programm für das globale Skalarprodukt | 107 |
| 6.5. PPRC im Unterschied zu den existierenden Ansätzen | 113 |
| 6.6. Implementierung von PPRC auf Multiprozessoren | 113 |
| 6.6.1 Implementierung der Konfigurationsmanager | 114 |
| 6.6.2 Implementierung der Basiskommunikationsroutinen (give_info, get_info) | 115 |
| 6.6.3 Implementierung von komplexen Kommunikationsroutinen | 116 |
| 6.7. PICL und PPRC | 116 |
| 6.8. PPRC: Zusammenfassung und Ausblick | 117 |
| 7. Ausblick | 119 |
| 7.1. Massiv parallele Systeme | 119 |
| 7.1.1 Connection Machine | 119 |
| 7.1.2 MEMSY | 121 |
| 7.2. The Roads to El Dorado | 123 |
| 8. Literaturverzeichnis | 125 |
| Index | 135 |