Effiziente Scheduling-Algorithmen für datenparallele Anwendungen der funktionellen medizinischen Bildgebung auf NOWs

F. Munz[1,2], T. Ludwig[2], A. Bode[2], S. Ziegler[1] und M. Schwaiger[1]

[1] Nuklearmedizinische Klinik und Poliklinik des Klinikums rechts der Isar
[2] Lehrstuhl für Rechnertechnik und Rechnerorganisation Technische Universität München (TUM)

email: Munz@Informatik.TU-Muenchen.DE

Zusammenfassung. Ziel dieser Arbeit ist die Evaluierung effizienter Scheduling-Algorithmen für den praktischen Einsatz in der medizinischen Bildverarbeitung, am Beispiel der Spektralanalyse dynamischer Aufnahmen der Positronen-Emissions-Tomographie (PET). Die Spektralanalyse, die im Gegensatz zu klassischen Kompartmentmodellen zur bildpunktweisen Berechnung physiologischer Parameter ohne a priori Modellannahmen eingesetzt wird, stellt ein wichtiges Werkzeug zur Analyse dynamischer PET-Aufnahmen dar. Die parallele Berechnung parametrischer Bilder ermöglicht den Einsatz modernster Analysemethoden auch für klinische Datensätze, beispielsweise im Rahmen der Epilepsiechirurgie. Die Spektralanalyse repräsentiert durch die hohe inhärente Parallelität eine charakteristische Anwendung der funktionellen Bildgebung. Durch die verteilte Berechnung der physiologischen Parameter auf einem Network of Workstations (NOW), lässt sich die Gesamtlaufzeit der Anwendung auf ca. 3 Minuten reduzieren. Wir betrachten fünf Scheduling-Strategien, die auf den Methoden theoretischer Arbeiten über parallelisierende Compiler für speichergekoppelte Parallelrechner aufbauen.
Schlüsselwörter:: Positronen-Emissions-Tomographie, Cluster Computing, kinetische Analyse, parallele Algorithmen.

1 Einleitung

Diese Arbeit setzt sich mit der Frage auseinander, in welchem Maße geeignete Zuteilungsstrategien der Einzelaufträge die Gesamtlaufzeit der parallelen kinetischen Analyse dynamischer PET-Daten reduzieren können. Wir betrachten die Verfahren im Zusammenhang mit der Spektralanalyse eines dynamischen Diprenorphindatensatzes. Die theoretischen Grundlagen und das methodische Vorgehen sind jedoch nicht von der Anwendung an sich abhängig, sondern können allgemein bei datenparallelen Algorithmen angewendet werden, wie sie in der Bildverarbeitung häufig anzutreffen sind. Die Frage, wie eine Zuteilung der Datensegmente auszusehen hat, die eine möglichst kurze Gesamtlaufzeit der Anwendung garantiert, läßt sich nicht allgemein beantworten.

Eine Vielzahl von Faktoren, angefangen von den Eigenschaften der Verbindungsstruktur zwischen den einzelnen Knoten eines Clusters bis hin zum Erwartungswert der Berechnungsdauer eines Auftrages spielen hier eine Rolle. Wir betrachten zunächst eine Reihe von Zuordnungsstrategien ausgehend von deren theoretischen Eigenschaften und analysieren dann deren Einfluß auf die Implementierung der parallelen Spektralanalyse bei unterschiedlichen Rahmenbedingungen.

2 Theoretische Überlegungen

Die theoretischen Grundlagen von Zuteilungsstrategien für nachrichtengekoppelte Systeme lassen sich aus Arbeiten zum Scheduling paralleler Schleifen auf Multiprozessoren mit gemeinsamem Speicher übertragen. Da voneinander unabhängige Schleifeniterationen, vor allem bei naturwissenschaftlichen Anwendungen, häufig auftreten, stellt sich beispielsweise für parallelisierende Übersetzer die Frage, wie bei minimalen Schedulingoverhead die Last auf die einzelnen Prozessoren optimal verteilt werden kann. Während man bei symmetrischen Multiprozessoren oft von einer Warteschlange mit den zu bearbeitenden Aufträgen und einer Schrankenoperation zur Synchronisation am Schleifenende ausgeht, stellt sich die Situation bei nachrichtengekoppelten Knoten eines Cluster anders dar: Hier erfolgt die Synchronisation implizit durch das Versenden von Nachrichten. So wird beispielsweise die Schrankenoperation am Schleifenende durch das blockierende Warten auf die ausstehenden Nachrichten ersetzt. Die prinzipielle Aufgabe ist allerdings die selbe. Die Last der Prozessoren soll möglichst gleichmäßig verteilt sein, aber die Verteilung der Datensegmente ist mit einer nicht unerheblichen Zusatzlast verbunden.

Zwei Verteilungsstrategien sind naheliegend: Gehen wir von der Situation aus, daß die entfernt laufenden Prozesse zur Spektralanalyse bereits gestartet und mit den Parametern initialisiert sind, dann können n Zeit-Aktivitätskurven auf triviale Weise p Prozessoren in n/p Blöcken zugeordnet werden. Wir bezeichnen die Strategie als statische Zuteilung (SC). Die eingesetzte Methode zur Rauschunterdrückung hat zur Folge, daß sich die Berechnungszeiten für die einzelnen Aufträge stark unterscheiden können. Die statische Zuteilungsstrategie SC führt zu Aufträgen mit geringeren Schwankungen bei der Bearbeitungszeit durch das Zusammenfassen der einzelnen Zeit-Aktivitätskurven. Da jedem Knoten nur ein Auftrag zugeordnet wird, kommt es zu hohen Gesamtlaufzeiten, falls die Bearbeitungszeiten für die Aufträge nicht identisch sind. Im ungünstigsten Fall warten $p - 1$ Prozessoren auf die Fertigstellung eines Auftrags. Andererseits ist es auch möglich, n Zeit-Aktivitätskurven als einzelne Aufträge den Prozessoren zuzuordnen. Wir sprechen dabei von Selbstzuteilung (SS). Die Selbstzuteilung führt zu einer gleichmäßigen Verteilung der Aufträge, die jedoch durch n Zuordnungsoperationen erkauft wird.

Kruskal und Weiss leiten in ihrer Arbeit mit Methoden der Statistik die Abschätzung einer oberen Grenze für die Berechnungsdauer von n Einzelaufträgen auf p Prozessoren zu jeweils gr Blöcken her. Wir sprechen dabei von einer Zuordnungsstrategie mit fester Auftragsgröße. Der Erwartungswert der Zeit T für die Berechnung der Aufträge kann bei p Prozessoren, denen insgesamt n Aufträge mit einer mittleren Ausführungszeit μ und einer Standardabweichung σ zugeteilt werden, nach oben wie folgt abgeschätzt werden [KW85]:

$$E(T) \leq \frac{n}{p}\mu + \frac{n \cdot h}{p \cdot gr} + \sigma\sqrt{2gr \log n} \tag{1}$$

Dabei bezeichnet gr die Größe der einzelnen Aufträge und h die Kosten, um einen einzelnen Auftrag einem Prozessor zuzuordnen. Der erste Summand beschreibt die Zeit, die p Prozessoren bei perfekter Lastbalancierung brauchen, um n Aufträge zu bearbeiten. Der zweite Summand beschreibt die Kosten der Auftragszuordnung und der dritte Summand die Ineffizienz, die durch die unterschiedlichen Berechnungszeiten der Aufträge bedingt ist. Der optimale Wert für gr läßt sich durch Differenzieren von Gleichung 1 ermitteln und ist unabhängig von der mittleren Ausführungszeit μ. Allerdings ist die Bestimmung von σ und h in der Praxis oft problematisch.

Als Alternative zur Bestimmung eines bestmöglichen Wertes für die Blockgröße gr haben Polychronopoulos und Kuck die Methode der geführten Selbstverteilung (GSS) entwickelt. Die Strategie der GSS liegt in der abnehmenden Größe der Aufträge. Dabei sorgt eine Reihe relativ großer Aufträge am Anfang für eine geringe Zusatzlast bei der Zuordnung durch wenige Zuordnungsoperationen, und die kleiner werdenden Aufträge zum Schluß hin dienen zum Ausgleich von Lastunterschieden, die durch die größeren Aufträge am Anfang entstanden sind. Die Blockgröße G_i eines Auftrages ist bestimmt durch die verbleibende Anzahl von Einzelaufträgen R_i dividiert durch die Anzahl der Prozessoren p:

$$R_0 = n \,; R_{i+1} = R_i - G_i \,; G_i = \left\lceil \frac{R_i}{p} \right\rceil = \left\lceil \left(1 - \frac{1}{p}\right)^i \frac{n}{p} \right\rceil \tag{2}$$

Polychronopoulos und Kuck zeigen, daß bei Einzelaufträgen mit konstanter Ausführungsdauer und unterschiedlichen Startzeiten der Prozessoren die Prozessoren innerhalb der Zeitspanne eines Einzelauftrages die Berechnungen beenden. Für Aufträge mit unterschiedlicher Ausführungsdauer ist die GSS-Strategie der SS-Strategie überlegen. Die von Flynn und Hummel vorgeschlagene Strategie der Faktorisierung basiert auf der Erkenntnis, daß bei der GSS-Strategie ungefähr zwei Drittel der verbleibenden Einzelaufträge während der nächsten p Zuordnungsoperationen verteilt werden, was im manchen Fällen bereits zu einem zu großen Lastungleichgewicht führen kann. Bei der Faktorisierung erhalten alle Prozessoren in einer Runde gleichgroße Aufträge, demnach ist auch die Gesamtbelastung der Prozessoren identisch mit n/p. Die Größe der Aufträge ergibt sich aus einem festen Verhältnis mit der Zahl der verbleibenden Einzelaufträge [FSF92].

$$F_j = \left\lceil \frac{R_j}{xp} \right\rceil = \left\lceil \left(1 - \frac{1}{x}\right)^j \frac{n}{xp} \right\rceil ; F_j = \left\lceil \left(\frac{1}{2}\right)^{j+1} \frac{n}{p} \right\rceil (mit\ x = 2) \tag{3}$$

Sowohl bei der Selbstzuteilung als auch bei der Faktorisierung besteht theoretisch die Möglichkeit, ab einer bestimmten Auftragsgröße die Aufträge nicht mehr weiter zu verkleinern, sondern die Strategie auf die Zuteilung von Aufträgen mit konstanter Auftragsgröße zu ändern.

3 Einfluß von Scheduling-Strategien auf die parallele Spektralanalyse

Bei der Beurteilung der Scheduling-Strategien unterscheiden wir vier Fälle. Als Testdatensatz dient der dynamische Datensatz eines Probanden. Wir vergleichen die statische Aufteilung, die Aufteilung mit fester Auftragsgröße, die geführte Selbstzuteilung und die Faktorisierung auf einem unbelasteten Sun Ultra 20 Cluster, mit denselben Strategien auf einem Cluster mit zusätzlicher Last und interaktiver Nutzung. Messungen mit der Selbstzuteilung von Einzelaufträgen führten zu extrem hohen Laufzeiten und Kosten und sind deswegen in den Abbildungen nicht dargestellt. Bei der Zuteilungsstrategie mit fester Auftragsgröße wurden jeweils 128 Zeit-Aktivitätskurven je Auftrag zusammengefaßt, was einer Zeile in den parametrischen Bildern entspricht.

4 Ergebnisse

Betrachtet man die Kosten für die Anwendung auf dem dedizierten Cluster ohne zusätzliche Last, dann schneidet die statische Zuteilung am schlechtesten von allen Strategien ab. Sowohl die Laufzeit als auch die Kosten der statischen Zuteilung fallen deutlich höher als die der anderen Verfahren aus. Die anderen drei Zuteilungsstrategien liegen enger beieinander. Die Zuteilung von Aufträgen mit einer festen Größe ist die Strategie mit den geringesten Kosten, gefolgt von der Faktorisierung und der geführten Selbstzuteilung. Die Kosten für die beste Strategie der Messung, der festen Aufteilung der Aufträge, betragen bei zehn Knoten 82% der Kosten der schlechtesten Strategie, der statischen Aufteilung.

Vergleicht man die beste Strategie, die feste Aufteilung der Aufträge mit der zweitbesten Strategie, der Faktorisierung, dann zeigt sich, daß der größte Unterschied bei zehn Knoten liegt. Die Kosten der festen Aufteilung der Aufträge betragen dann 95% der Kosten der Faktorisierung.

Abbildung 1 zeigt die Situation bei zusätzlicher interaktiver Benutzung des Cluster. Auf allen Knoten liefen zeitweise Prozesse von anderen Anwendern, auf dem fünften und dem achten Knoten war jeweils ein ständig rechenbereiter Prozeß eines anderen Anwenders plaziert. Neben den höheren Gesamtkosten, die bei allen vier Strategien bemerkbar sind, zeigt sich als auffälligster Unterschied zur Situation ohne Last, daß die feste Aufteilung der Datensegmente nur noch die drittbeste Lösung darstellt. Mit zusätzlicher Last ist die Faktorisierung, gefolgt von der geführten Selbstzuteilung die beste Strategie. Während die feste Aufteilung die größten Schwankungen bei den Kosten zeigt, nehmen bei der Faktorisierung die Kosten gleichmäßiger zu. Das bessere Abschneiden der Faktorisierung gegenüber der Zuteilung von Aufträgen mit einer festen Größe kommt daher, daß durch die zusätzliche Last die Zuteilung der Daten zu den einzelnen Knoten im Verhältnis zur lastfreien Situation aufwendiger wird und die Zuteilungsoperationen bei der Gesamtlaufzeit mehr ins Gewicht fallen. Die Faktorisierung schneidet deswegen besser ab, da durch die Zuteilung großer Segmente bei den ersten Zuteilungsoperationen der Faktorisierung insgesamt weniger Zuteilungsoperationen notwendig sind. Bei zusätzlicher Last überwiegt dieser Effekt die etwas höhere Lastungleichheit im Vergleich zur Zuteilung von Aufträgen mit fester Größe.

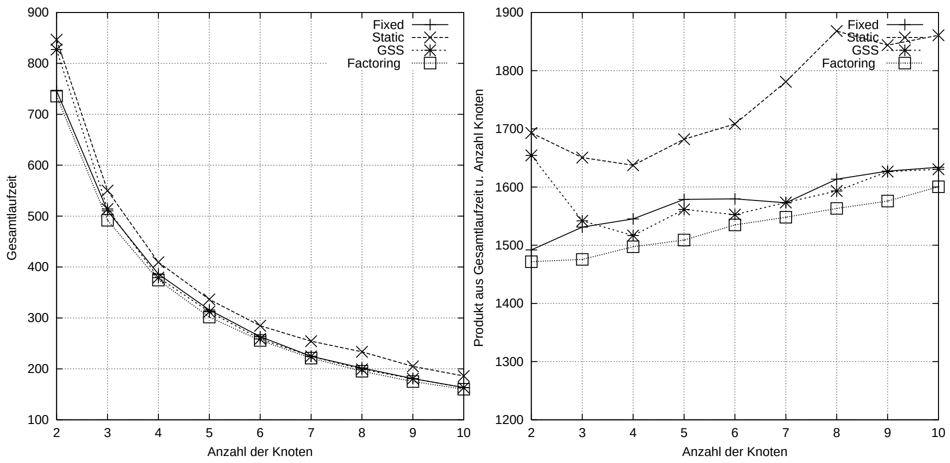

Abb. 1. Darstellung der Laufzeit (in Sekunden) der parallelen Spektralanalyse bei gleichzeitiger interaktiver Nutzung des Cluster (links). Die rechte Abbildung zeigt das Produkt aus Laufzeit und Anzahl der Prozessoren für die jeweilige Zuteilungsstrategie.

Die Faktorisierung erweist sich als robuste Strategie der Datenzuteilung, die im Vergleich zur Methode von Kruskal und Weiss ohne die exakte Ermittlung der Kosten für eine Datenzuteilungsoperation und der Standardabweichung der Bearbeitungsszeiten auskommt. Die Methode stellt damit eine Alternative zur Bestimmung der optimalen Nachrichtengröße durch iteratives Leistungsdebugging oder einer analytischen Leistungsmodellierung dar.

Literatur

[FSF92] S. Flynn, E. Schonberg, and L. Flynn. Factoring: A method for scheduling parallel loops. *Communictions of the ACM*, 35(8):90–101, 1992.

[KW85] C. Kruskal and A. Weiss. Allocating independent subtasks on parallel processors. *IEEE Transactions of Software Engineering*, SE-11(10), October 1985.