

# Parallelisierung der Dosisberechnung des Bestrahlungsplanungsprogramms TRiP98

Jan Trautmann<sup>1,2</sup>, Michael Krämer<sup>2</sup>, Christoph Bert<sup>2</sup>, Wolfgang Heenes<sup>1</sup>

1: Technische Universität Darmstadt  
FG Rechnerarchitektur  
Hochschulstr. 10, 64289 Darmstadt

2: GSI Helmholtzzentrum für Schwerionenforschung GmbH Darmstadt  
Planckstraße 1, 64291 Darmstadt  
Email: j.trautmann@gsi.de

**Abstract:** Die Dosisberechnung mit dem Bestrahlungsplanungsprogramm TRiP98 ist ein zeitaufwendiger Vorgang. Durch die immer höhere Verbreitung von Multicore-Prozessoren erscheint es daher lohnenswert, eine Parallelisierung des Programmcodes zumindest für ausgewählte Funktionen von TRiP98, z. B. die Dosisberechnung, durchzuführen. Es wird ein Überblick über TRiP98 und die im Programm enthaltene Dosisberechnung gegeben und die unterschiedlichen Algorithmen, welche zur Berechnung der Dosisverteilung zur Verfügung stehen, werden kurz vorgestellt. Probleme, welche bei der Umsetzung der Parallelisierung mit OpenMP aufgetreten sind, werden beschrieben und die Lösungsansätze erläutert. Abschließend werden die Implementierungen durch Tests an verschiedenen Patientendatensätzen verglichen. Durch Parallelisierung konnte, in Abhängigkeit von dem jeweiligen Patientendatensatz und dem verwendeten Algorithmus, ein Speedup von bis zu 20,81 ermittelt werden. Weiter wird auf entdeckte Besonderheiten bei den Speedupfaktoren eingegangen.

## 1 Einleitung

Das Bestrahlungsplanungsprogramm TRiP98 wurde im Rahmen des am GSI Helmholtzzentrum für Schwerionenforschung GmbH (GSI) durchgeführten Pilotprojekts für die Bestrahlungsplanung in der Krebstherapie mit Schwerionenstrahlen eingesetzt. Es kam hier bei über 400 Patienten erfolgreich zum Einsatz [JKKD01]. TRiP98 ist eine Eigenentwicklung der GSI. Seit Abschluss des Pilotprojektes dient es allein zu Forschungszwecken und unterliegt somit einer stetigen Weiterentwicklung.

Durch neue Erweiterungen von TRiP98 kommt es zu einer deutlichen Erhöhung des Berechnungsaufwandes. Dies dient maßgeblich als Motivation, eine Parallelisierung der Dosisberechnung umzusetzen. Zudem kann es generell abhängig von Patient und verwendetem Algorithmus die Berechnung der Dosisverteilung von wenigen Sekunden bis hin zu mehr als zwei Tagen betragen. Dieser gravierende Unterschied wird durch die verschiedene Genauigkeit der Algorithmen hervorgerufen und macht eine Parallelisierung auch so bereits interessant.

## 2 Funktion, Struktur und Analyse von TRiP98

Zur Implementierung der Parallelisierung der Dosisberechnung wird OpenMP verwendet. TRiP98 ist in der Programmiersprache C programmiert und wird an der GSI in einer IBM POWER Umgebung mit dem UNIX-Betriebssystem AIX betrieben. Das aktuell leistungsfähigste vorhandene System ist hier ein IBM BladeCenter PS701 Server [IBM] mit einem 8-Kern-Prozessor der POWER7 Baureihe. Durch Verwendung der eingebauten 4-way simultaneous multithreading (SMT) Technologie erhöht sich die Anzahl der verfügbaren Threads von 8 auf 32.

Ein wichtiges Ziel, welches es bei der Parallelisierung zu beachten gilt, ist es, den Code möglichst wenig verändern zu müssen. Er soll weiter einheitlich und somit leicht zu pflegen sein, da es verschiedene Versionen von TRiP98 für unterschiedliche Forschungsvorhaben gibt. Außerdem spielt die Validität eine zentrale Rolle, welche so einfacher gewahrt werden kann.

In dem hier beschriebenen ersten Schritt der Umsetzung einer Parallelisierung wird die Dosisberechnung für die 3D-Bestrahlungsplanung untersucht. Diese beinhaltet verschiedene Algorithmen zur Bestimmung der Dosisverteilung. Sie lassen sich in zwei Kategorien unterteilen:

- Absorbierte Dosis [KJH<sup>+</sup>00]
- Biologische Dosis [KS00]

Generell besteht die Dosisberechnung im Wesentlichen aus drei Schleifen, welche die z-, y- und x-Richtung der CT-Aufnahme ablaufen. Im Falle der Behandlung von Patienten wird eine Bestrahlung meist aus mehreren Richtungen durchgeführt, um das umliegende Gewebe zu schonen. Für jede dieser Einstrahlrichtungen existiert ein Bestrahlungsfeld. Im Inneren der Schleifen wird die Dosis für das jeweilige Voxel ermittelt.

```
1 for (Jedes Voxel in z-Richtung) {
2   for (Jedes Voxel in y-Richtung) {
3     for (Jedes Voxel in x-Richtung) {
4       for (Jedes Feld){
5         Berechnung_der_absorbierten_Dosis()
6         Aufsummieren der absorbierten Dosis
7       }
8       if (biologische Dosis) {
9         Berechnen_der_biologischen_Dosis()
10      }
11      Speichern der Dosis an entsprechender Position in der
12      3-dimensionalen Dosisstruktur
13    }
  }
```

Ein Profiling des Codes wurde nicht durchgeführt. Aus dem jahrelangen Einsatz von TRiP98 ist hinreichend bekannt, dass es sich bei der Dosisberechnung um eine der zeitintensivsten und häufigsten genutzten Funktionen des Programms handelt. Zugleich sind durch

die Konstanz der Datensatzgrößen keine Überraschungen im Ablauf der Berechnung zu erwarten.

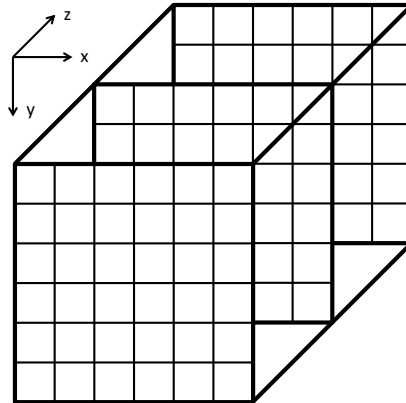


Abbildung 1: Veranschaulichung der Verteilung der Daten bei einer CT-Aufnahme

Von den drei Schleifen der Dosisberechnung wird die äußerste parallelisiert, bei der es sich um den Durchgang in z-Richtung handelt. Hierdurch soll der Aufwand für das Erstellen und Schließen der Threads möglichst gering gehalten werden. Es ist auf Grund der Abmessungen der CT-Aufnahme immer klar, wie viele parallel zu verarbeitende Threads es geben wird. Die typische Auflösung einer CT-Aufnahme in z-Richtung beträgt etwa 100 Schichten und ist somit in dieser Richtung deutlich dünner besetzt als in x- und y-Richtung. Dies ist beispielhaft in Abbildung 1 dargestellt.

Bei der so durch zu führenden Dosisberechnung muss zunächst stets eine absorbierte Dosis, welche die physikalischen Aspekte berücksichtigt, ermittelt werden. Hier gibt es drei Algorithmen, die jeweils alternative verwendet werden können:

- classic (cl)
- allpoints (ap)
- multiplescatter (ms)

Die Genauigkeit der Algorithmen und somit auch die Berechnungszeit steigen in der hier aufgeführten Reihenfolge an.

Die biologische Dosisberechnung ist optional durchführbar. Sie benötigt jedoch immer eine absorbierte Dosis als Ausgangsbasis und basiert auf dem Local-Effect-Modell (LEM)

[SKKWK97]. Für die biologische Dosis stehen zwei alternative Algorithmen zur Verfügung:

- classic (cl)
- lowdose (ld)

Hier ist nun der cl Algorithmus berechnungsintensiver. Auf Grund der sehr hohen Laufzeiten des Algorithmus cl wurde der Algorithmus ld entwickelt, der durch Verwendung von Approximationen eine drastische Verkürzung der Berechnungsdauer erreicht [KS06]. Die Implementierung der Parallelisierung der Dosisberechnung ließ sich größtenteils problemlos umsetzen. Einen erheblichen Teil der Bearbeitung diente der Ermittlung, welche Variablen von allen Threads gleichzeitig benutzt werden können (Spezifizierung als „shared“ innerhalb von OpenMP) und welche für jeden Thread vorliegen müssen (private). Von größerem Aufwand war lediglich der Algorithmus classic bei der biologischen Dosisberechnung. Hier werden Pseudozufallszahlen zur Berechnung verwendet. Jede dieser Zahlen wird auf Basis von Informationen der vorangegangenen Zahl ermittelt. Somit liegt hier eine Datenabhängigkeit vor. Diese wurde aufgelöst, indem jedem Thread seine eigenen Informationen bezüglich der Berechnung der Zufallszahlen zur Verfügung gestellt wurden. Hierdurch kommt es zu einer leichten Veränderung des Ergebnisses. Um für die anschließenden Geschwindigkeitsmessungen eine einheitliche Ausgangsbasis mit identischen Ergebnissen in der Dosisverteilung zu erhalten, wurde diese Änderung auch in die für die Berechnungen verwendete Vergleichsversion ohne OpenMP übernommen.

### 3 Ergebnisse

Um die Effizienz der Parallelisierung zu ermitteln, wurden Laufzeitmessungen der Dosisberechnung durchgeführt (siehe Tabelle 1). Hierfür wurden vier Patientendatensätze aus dem Pilotprojekt an der GSI verwendet. Die Berechnungen wurden mit verschiedenen Anzahlen an Threads durchgeführt. Gemessen wurden alle drei verschiedenen Algorithmen der absorbierten Dosisberechnung sowie die beiden Algorithmen der biologischen Dosisberechnung. Im Falle der biologischen Dosisberechnung liegt eine absorbierte Dosis nach dem cl Algorithmus zu Grunde. Dieser Algorithmus wurde hier gewählt, um den zeitlichen Einfluss der absorbierten Dosisberechnung möglichst gering zu halten. Im Fall der Vergleichsberechnung mit der normalen TRiP98 Version ergaben sich Laufzeiten zwischen 8,1 Sekunden (Patient 135, absorbierte Dosis cl) und 6,8 Stunden (Patient 305, biologische Dosis cl). Die schnellsten mit der parallelisierten Version ermittelten Ergebnisse für diese beiden Fälle betragen dagegen 1,2 Sekunden und 49,5 Minuten. Die in der Einleitung angesprochenen Berechnungszeiten von über zwei Tagen ergeben sich nur bei der Kombination einer aufwendigeren Berechnung der absorbierten Dosis (Algorithmus ap oder ms) in Verbindung mit der biologischen Dosisberechnung nach dem cl Algorithmus.

Für eine bessere Veranschaulichung der gemessenen Zeiten wurden diese in Speedup-Faktoren umgerechnet. Die Berechnungszeit der normalen Version von TRiP98 wurde

hierbei als Referenz verwendet. Je Patient und Algorithmus sind die maximal erreichten Speedup-Faktoren in Tabelle 1 aufgeführt.

Patient	Bestrahlte Voxel	Absorb. Dosis <b>cl</b> 32 Thr.	Absorb. Dosis <b>ap</b> 32 Thr.	Absorb. Dosis <b>ms</b> 32 Thr.	Bio. Dosis <b>cl</b> 24 Thr.	Bio. Dosis <b>ld</b> 32 Thr.
135	14560	9,76	13,04	16,76	11,75	9,27
239	69983	10,44	12,06	19,03	12,24	10,69
290	47818	8,95	12,43	20,81	11,77	8,97
305	49529	11,12	12,43	17,86	12,27	11,67

Tabelle 1: Bestrahlte Voxel (je Patient) und maximale Speedup-Faktoren (je Patient und Algorithmus)

Auf Grund der großen Unterschiede in den Patientendaten ergeben sich Unterschiede in den Speedup-Faktoren bei gleichen Algorithmen. Hierbei spielt es neben der Voxelanzahl selbst vor allem auch eine Rolle, wie gleichmäßig diese über die verschiedenen Schichten verteilt sind. Die Abarbeitungszeiten der einzelnen Threads können dadurch stark variieren. Eine Besonderheit ergab sich bei der Berechnung der biologischen Dosis mit dem **cl** Algorithmus. Auf Grund der Menge der hier benötigten Daten hatte die Ausführung von vier Threads pro Prozessorkern im Vergleich zu drei Threads nachteilige Auswirkungen.

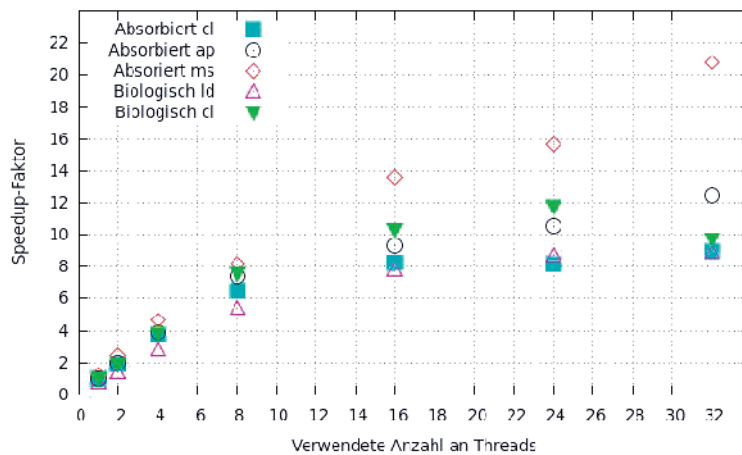


Abbildung 2: Speedup-Faktoren aller Messungen bei Patient 290

In Abbildung 2 wird die Leistungssteigerung in Abhängigkeit der verwendeten Anzahl an Threads mittels der Daten von Patient 290 dargestellt. Für die anderen Patienten ergibt sich keine nennenswerte Abweichung im Verhalten der Algorithmen. Gut erkennbar ist die Tatsache, dass die Leistungssteigerung im Bereich bis zu acht Threads, also so lange jedem Thread ein eigener Prozessorkern zur Verfügung steht, sehr gut verläuft. Erst wenn

sich durch die SMT Technologie bedingt mehrere Threads einen Kern teilen müssen, sinkt der Leistungszuwachs deutlich und ist wie bereits erläutert im Falle des cl Algorithmus bei der biologischen Dosisberechnung sogar rückläufig.

Auch das Ziel, den Code möglichst wenig verändern zu müssen, konnte bei der Umsetzung erreicht werden. Lediglich die Berechnung der Zufallszahlen für den cl Algorithmus bei der biologischen Dosisberechnung benötigt, wie bereits geschildert, eine alternative Funktion. Alle weiteren Eingriffe zur Variablendefinition und zur Ausführung der Parallelisierung konnten innerhalb der Funktion der Dosisberechnung selbst implementiert werden. Durch die so geringe Anzahl an Positionen im Code, an denen Veränderungen nötig waren, bleibt der Code gut wartbar.

## 4 Zusammenfassung und Ausblick

Wie die Implementierung gezeigt hat, ist eine Parallelisierung der Dosisberechnung gut durchführbar. Als weitere Anwendung der hier gewonnenen Erkenntnisse ist eine Umsetzung einer Parallelisierung bei anderen berechnungsintensiven Aufgaben innerhalb von TRiP98 denkbar. Hier wäre die Dosisoptimierung in Betracht zu ziehen, da sie zu großen Teilen die gleichen Funktionen wie die Dosisberechnung verwendet. Somit könnten hier viele Erkenntnisse, insbesondere auch über die Verwendung der Variablen als private oder shared, genutzt werden.

## Literatur

- [IBM] IBM. Informationen zur IBM Blade Server 700 Reihe. <http://www-03.ibm.com/systems/de/bladecenter/hardware/servers/ps700series/index.html>. Stand: 03.03.11, 17:48 Uhr.
- [JKKD01] O. Jaekel, M. Kraemer, C. P. Karger und J. Debus. Treatment planning for heavy ion radiotherapy: clinical implementation and application. *Phys. Med. Biol.*, 46(4):1101–1116, 2001.
- [KJH<sup>+</sup>00] M. Kraemer, O. Jaekel, T. Haberer, G. Kraft, D. Schardt und U. Weber. Treatment planning for heavy-ion radiotherapy: physical beam model and dose optimization. *Phys. Med. Biol.*, 45(11):3299–3317, 2000.
- [KS00] M. Kraemer und M. Scholz. Treatment planning for heavy-ion radiotherapy: calculation and optimization of biologically effective dose. *Phys. Med. Biol.*, 45(11):3319–3330, 2000.
- [KS06] M. Kraemer und M. Scholz. Rapid calculation of biological effects in ion radiotherapy. *Phys. Med. Biol.*, 51(8):1959–1970, 2006.
- [SKKWK97] M. Scholz, A. M. Kellerer, W. Kraft-Weyrather und G. Kraft. Computation of cell survival in heavy ion beams for therapy. The model and its approximation. *Radiat Environ. Biophys.*, 36(1):59–66, 1997.