



Research Report Series

Lehrstuhl für Rechner-technik und
Rechnerorganisation (LRR-TUM)
Technische Universität München

<http://www.bode.in.tum.de/>

Editor: Prof. Dr. A. Bode

Vol. 29

**Konzepte zur Optimierung
der Skalierbarkeit von parallelen
Fahrzeugkollisionsberechnungen und
ihre industrielle Realisierbarkeit**

Josef Weidendorfer

SHAKER

V E R L A G

Aachen 2003

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Weidendorfer, Josef:

Konzepte zur Optimierung der Skalierbarkeit von parallelen
Fahrzeugkollisionsberechnungen und ihre industrielle
Realisierbarkeit / Josef Weidendorfer.

Aachen : Shaker, 2003

(Research Report Series/Lehrstuhl für Rechnertechnik und Rechner-
organisation (LRR-TUM), Technische Universität München ; Vol. 29)

Zugl.: München, Techn. Univ., Diss., 2003

ISBN3-8322-1421-6

Copyright Shaker Verlag 2003

Alle Rechte, auch das des auszugsweisen Nachdruckes, der auszugsweisen
oder vollständigen Wiedergabe, der Speicherung in Datenverarbeitungs-
anlagen und der Übersetzung, vorbehalten.

Printed in Germany.

ISBN 3-8322-1421-6

ISSN 1432-0169

Shaker Verlag GmbH • Postfach 101818 • 52018 Aachen

Telefon: 02407 / 95 96 - 0 • Telefax: 02407 / 95 96 - 9

Internet: www.shaker.de • eMail: info@shaker.de

Kurzfassung

Parallele Anwendungen im Bereich des Hochleistungsrechnens erfordern neben der Implementierung geeigneter numerischer Algorithmen die effiziente Nutzung der in einem Parallelrechner bereitgestellten Kommunikationsstruktur und die Integration dynamischer Lastbalancierung, falls verwendete Algorithmen unvorhersehbares und damit statisch schlecht aufteilbares Lastaufkommen besitzen. Die Verflechtung dieser teilweise plattformabhängigen Programmteile, die bei Benutzung üblicher Programmiermodelle notwendigerweise entsteht, führt zu unflexibler Software, die nur schwer Weiterentwicklung, Wartung und Anpassung an neue Plattformen erlaubt.

Zur Lösung dieses Problems stellt die vorliegende Arbeit ein Programmiermodell für die Klasse der feldbasierten iterativen Anwendungen vor, das es erlaubt, die eigentlichen Berechnungen des Programms von der notwendigen Kommunikation und Synchronisation zwischen den parallel ablaufenden Prozessen zu trennen. Dadurch wird eine unabhängig arbeitende Lastbalancierung ermöglicht. Obwohl das Modell dem Programmierer gemeinsamen Speicher zur Verfügung stellt, kann es mit *Message Passing* implementiert werden. Grundlage dabei ist, daß die Spezifikation a priori bekannter Abhängigkeiten von Anweisungen, deren Zugriffsverhalten auf gemeinsame Daten explizit anzugeben ist, die automatisierte Berechnung der zwischen den Anweisungen zur Konsistenzhaltung durchzuführenden Kommunikation erlaubt. Vermieden wird dadurch der Zeitverlust, der bei einer üblichen Realisierung von verteilt gemeinsamem Speicher entsteht, in der erst auf Anforderung ein Abgleich zur Konsistenzsicherung ausgeführt wird.

Die Praxistauglichkeit des Programmiermodells wird durch den erfolgreichen Einsatz in einer Industrieanwendung, einer Software zur Simulation von Fahrzeugkollisionen gezeigt. Diese Simulationen werden von der Automobilindustrie neben physikalischen Kollisionsversuchen heutzutage deshalb verstärkt eingesetzt, da bei gleicher Entwicklungszeit weitaus mehr Konstruktionsalternativen erprobt werden können und damit die Sicherheit von Fahrzeuginsassen bei Verkehrsunfällen gesteigert werden kann. Der hohe Rechenleistungsbedarf macht allerdings Parallelisierung zwingend erforderlich. Ein diese Arbeit begleitendes Projekt untersuchte die Skalierungsprobleme, die in der bei der BMW AG benutzten Simulationssoftware PAM-CRASHTM auftreten. Der Einsatz des oben beschriebenen Programmiermodells zur Einführung von Lastbalancierung kann diese Probleme reduzieren, wie Verbesserungen in der Laufzeit praxisrelevanter Simulationsläufe belegen.