

Zwischenbericht zur Arbeit im DFG-Projekt
"Partitionierungsalgorithmen für
Modelldatenstrukturen zur parallelen
compilergesteuerten Logiksimulation"

Report Nr. 2 (1999)

W.G. Spruth, K. Hering, R. Haupt, U. Petri

Institut für Informatik
Universität Leipzig, Augustusplatz 10/11
04109 Leipzig

e-mail: {spruth,hering,haupt,petri}@informatik.uni-leipzig.de

1 Personalia

Ende Januar 1999 beendete Herr Udo Petri aufgrund eines Stellenangebotes seine Tätigkeit im Rahmen des Projektes. Im April 1999 wurde diese Stelle von Herrn Dr. Klaus Hering übernommen. Auf der zweiten Mitarbeiterstelle wurde die Arbeit durch Herrn Dr. Reiner Haupt kontinuierlich fortgeführt. Die bewilligten Stellen für die studentische Hilfskräfte wurden entsprechend inhaltlicher, organisatorischer und administrativer Gegebenheiten über unterschiedliche Zeiträume durch die Herren Fabian Schmidt, Robert Reilein, Hilmar Hennings, Thomas Siedschlag, Jork Löser, Daniel Lucke, Hendrik Schulze, Christian Schmidt-Gütter, Andrei Voinikonis und Jens Markwardt besetzt.

2 Projektziele

Unser Forschungsvorhaben hat die *Entwicklung, Analyse* und *Implementierung* von Partitionierungsalgorithmen für industrierelevante Modelle kompletter Prozessorstrukturen mit mehreren Millionen Elementen auf der *Gate-Ebene* in Vorbereitung der parallelen zyklusbasierten Logiksimulation zum Ziel. Dabei verkörpern die von uns auf Basis der sequentiellen Simulatoren *TEXSIM* und *MVLSIM* entwickelten Simulatoren *parallelTEXSIM*, *parallelMVLSIM* und der im Berichtszeitraum realisierte *dlbSIM* das unmittelbare Anwendungsfeld. Aufbauend auf dem im bisherigen Projektverlauf entstandenen formalen Modell der parallelen Simulation, der hierarchischen Partitionierungsstrategie sowie den in diesem Rahmen entwickelten Partitionierungsalgorithmen konzentrierten wir uns in der laufenden Arbeit auf die Erweiterung des Parametrisierungskonzepts für Partitionierungsalgorithmen und die Beschleunigung der Partitionierung durch deren Parallelisierung. Im folgenden seien grundlegende Ziele hervorgehoben, die wir uns für die momentan laufende zweite Verlängerungsphase unseres Projekts gesetzt haben:

- Algorithmenentwicklung und umfangreiche experimentelle Untersuchungen zur Validierung unserer theoretischen Arbeit zur *parametrisierten Partitionsbewertung* auf Basis des Modells der *parallelen Zyklussimulation (PCS)*.
- *Parallelisierung* geeigneter Partitionierungsalgorithmen

- Weiterentwicklung der Partitionierungsumgebung *parallelMAP*
- Entwicklung und Untersuchung von *dynamischen Lastbalancierungsstrategien* sowie Implementierung des Simulators *dlbSIM* zum lastabhängigen Partitionsmanagement
- Erweiterung des Applikationsfeldes unserer Partitionierungsstrategien auf die *parallele Logiksynthese* sowie *komplexe Simulationen in der Meteorologie*.

Bezüglich dieser Ziele werden im folgenden Abschnitt die Resultate unserer Arbeit im Berichtszeitraum dargestellt.

3 Resultate

3.1 Parametrisierte Partitionierungsalgorithmen

Auf der Grundlage unserer theoretischen Vorarbeiten zur parametrisierten Partitionsbewertung [Her96] wurden experimentelle Untersuchungen zur Abhängigkeit der Partitionsgröße von zur Verfügung stehenden Parametern, die die Prozessorleistung, Memory-Eigenschaften und Kommunikationsspezifika beschreiben, anhand des Vergleiches der bei *IBM* Böblingen installierten *IBM SP* Parallelrechner und unseres auf Basis verschiedener *IBM* Workstations vorhandenen Clusters durchgeführt.

Durch Einbeziehung der parametrisierten Bewertungsfunktion haben wir bestehende Partitionierungsalgorithmen (*MOCC*, *nBCC*, *Evolutionäre Algorithmen* [Sch98]) vervollkommenet sowie neue Algorithmen (*Iterative Improvement Algorithms* [Sie98]) entwickelt und implementiert, um eine gezielte Suche nach Partitionen, die für entsprechende parallele Simulationen auf einer ganz bestimmten Zielarchitektur bestimmt sind, zu ermöglichen. Eine differenzierte Berücksichtigung elementarer Hardwaremodell-Komponenten innerhalb der Bewertungsfunktion hat zu keiner signifikanten Verbesserung der Abschätzung der prognostizierten Laufzeit entsprechender paralleler Simulationen geführt.

Nach Nutzbarmachung der *Evolutionären Algorithmen* (EA) für die 2. Stufe unserer *hierarchischen Partitionierungsstrategie* sind für diese Algorithmen eine Reihe von Erweiterungen vorgenommen worden, die der Anpassung der

EA an unser spezielles Partitionierungsproblem dienen. Neben der Integration des Konzepts der Superposition [HHPV97] in die Evolutionsstrategie wurden die EA durch Einführung eines *Local Search Operators* [HHS98] zu einer hybriden Heuristik erweitert, die insbesondere bei einer sehr komplexen mit vielen lokalen Minima behafteten Fitneßfunktion (Bewertungsfunktion) Erfolg zeigt. In [Sch98] wurde eine *dynamische Fitneßfunktion* vorgeschlagen, die neben der konventionellen Partitions-Bewertung einen zusätzlichen generationsabhängigen Term beinhaltet, der inhomogene Partitionen insbesondere zu Beginn der Evolution bevorzugt.

Alternativ zu den EA als kollektive Optimierungsverfahren wurden die *Iterative Improvement Algorithms* auf der Grundlage der Algorithmen von FIDUCCIA-MATTHEYSES in Form von 4 konkreten Realisierungen (*ADAPTED, MIXED, MIN-CUT, TABU SEARCH*) [Sie98] in unser Partitionierungskonzept integriert. Dazu wurden neben der Anpassung bestehender Datenstrukturen (Überlappungs- und Kommunikations-Hypergraph \rightarrow Kombiniertes Hypergraph) neue Datenstrukturen entwickelt (Gewinnstrukturen).

Es wurde eine Erweiterung des Partitionsbegriffs hinsichtlich der Mehrfachverteilung von Partitionsgrundelementen (Cones) untersucht. Das Ziel, durch Mehrfachverteilung von Cones Kommunikationseinsparungen herbeizuführen, welche die damit verbundenen Lasterhöhungen überkompensieren, kann erreicht werden. Allerdings rechtfertigt der geringfügige Gewinn für relevante Modelle den damit verbundenen zusätzlichen Partitionierungsaufwand einschließlich weitreichender Änderungen am Gesamtkonzept der hierarchischen Partitionierungsstrategie nicht [Sie98].

3.2 Parallele Partitionierungsalgorithmen

Im konkreten Anwendungsfall bildet bei uns eine Modellpartitionierung in der Regel die Grundlage für eine ganze Reihe zeitintensiver Simulationen (bezüglich ein- und desselben Hardware-Modells). Aus diesem Grund ziehen wir im Vergleich zu einigen anderen Partitionierungsprojekten zugunsten der Partitionsgröße durchaus auch relativ zeitaufwendige Partitionierungen in Betracht. Hinsichtlich der Minimierung dieses Zeitaufwandes wird die Modellpartitionierung selbst zu einem interessanten Gegenstand der Parallelisierung. Wie bei der parallelen Simulation gehen wir dabei von lose gekoppelten Architekturen als Zielhardware aus.

Im Fall des Einsatzes evolutionärer Algorithmen nehmen diese einen entschei-

denden Zeitanteil am Modellpartitionierungsprozeß ein. Aufbauend auf einem *Multiple Subpopulation Approach* haben wir eine parallele Variante evolutionärer Partitionierungsalgorithmen (PEPA) [Sch98] mit dynamischer Fitneßfunktion [SHH99a] und einer als *Lazy Communication* bezeichneten asynchronen Kommunikationsstrategie entwickelt. Über Parameter sind verschiedene Kommunikationsstrukturen und Migrationsstrategien [VHHS99] wählbar. Experimentelle Resultate sind in [SHH99b] zusammengefaßt. Durch den Einsatz paralleler evolutionärer Algorithmen in der finalen Partitionierungsphase konnten gegenüber dem sequentiellen Fall qualitativ bessere Partitionen in signifikant kürzerer Zeit erhalten werden. Von einer Parallelisierung des MOCC-Algorithmus und der im Berichtszeitraum entwickelten iterativen Verfahren wurde aufgrund ihres relativ geringen Zeitanteils an relevanten hierarchischen Partitionierungsprozessen abgesehen. Neben der Realisierung von Partitionierungsalgorithmen im engeren Sinn im Rahmen unserer hierarchischen Strategie (Zerlegung modellbezogener Mengen von Cones bzw. Supercones) sind zur Erzeugung des partitionsspezifischen Inputs für parallele Simulationen Prozesse zur Bildung von Simulationsmodellen, Signalschnitt- und Crossreference-Listen erforderlich. Das damit insgesamt gegebene Parallelitätspotential wird im Rahmen einer parallelen Partitionierungsumgebung genutzt.

3.3 Parallele Partitionierungsumgebung

Die Partitionierungsumgebung *parallelMAP* [Hen99] ermöglicht eine einfache Integration sequentieller und paralleler Partitionierungsalgorithmen. Die sowohl auf Unterstützung der Algorithmenentwicklung als auch auf den unmittelbaren Einsatz im industriellen Design-Prozeß ausgerichtete Umgebung ist im Laufe ihrer Entwicklung konzeptionell wesentlich erweitert worden. Sie verkörpert eine Realisierung der universell als Rahmen für die Entwicklung von *Problem Solving Environments* einsetzbaren Client/Server-Architektur *DRIVE (Distributed Runtime Environment)* [HHH99]. Ein besonderer Vorteil von *DRIVE* besteht in der Möglichkeit, parallele Programme ausgehend von sequentiellen und MPI-basierten parallelen Modulen im Rahmen einer einfachen *Skriptsprache* zu konstruieren. Die Skriptausführung wird durch Monitoring und halbautomatische Ressourcenzuweisung unterstützt. Prozeßmigration wird nicht in Betracht gezogen. *DRIVE* ist multiuserfähig und bietet ein ansprechendes *Graphical User Interface*. Die Funktionalität der Umgebung ist auf der Basis von *Shared Libraries* und dynamischem

Linken zur Laufzeit in einfacher Weise modifizierbar. Die Anwendungsspezifik einer konkreten *DRIVE* - Realisierung wird durch die einbezogenen Datenbanken und Bibliotheken gegeben. Im Fall von *parallelMAP* ist die *Design Automation Database DA_DB* (IBM) zur Behandlung struktureller Hardwaremodelle eingebunden. Momentan sind in *parallelMAP* insbesondere Bibliotheken in Bezug auf Vorpartitionierung, Hypergrapherzeugung, Startpartitionierung, parallele evolutionäre Partitionierungsalgorithmen, iterative Partitionierungsalgorithmen und Partitionsauswertung enthalten.

Innerhalb des mit diesem Bericht verbundenen Fortsetzungsantrags setzen wir uns als einen Zielschwerpunkt die Entwicklung und Untersuchung von Scheduling-Algorithmen zur automatischen Skripterstellung für *parallelMAP* ausgehend von Task-Graphen.

3.4 Dynamisch balancierte Simulation

Das unserer Arbeit zugrundeliegende Modell *PCS* [Her96] der parallelen Zyklussimulation ist nicht mit einem zur Laufzeit veränderlichen Applikationsgraphen verknüpft. Unter der Annahme separater Simulationsläufe auf einem Parallelrechner ohne gleichzeitige weitere Applikationen ist ein statischer Modellpartitionierungsansatz gerechtfertigt. Ausgehend von künftigen Anwendungsszenarien unter Fremdlasteinflüssen haben wir einen Ansatz zur dynamischen Lastbalancierung für die parallele Zyklussimulation entwickelt, der im Rahmen des Simulators *dlbSIM* [Lös98] realisiert wurde. *dlbSIM* baut auf dem bereits im vergangenen Berichtszeitraum realisierten Simulator *parallelMVLSIM* auf. In der Konzipierungs- und Realisierungsphase der Lastbalancierung wurden von K. HERING und J. LÖSER im Oberseminar der Arbeitsgruppe von B. MONIEN an der Universität Paderborn zwei Vorträge gehalten, deren Diskussion wertvolle Anregungen für unsere weitere Arbeit erbracht hat.

Unser Ansatz geht von einer redundanten Verteilung von Lastelementen in Form von Partitionskomponenten eines Hardwaremodells auf kooperierende Simulatorinstanzen aus. Diese Verteilung bildet einen Rahmen für zur Laufzeit mögliche Lastverschiebungen, welche selbst nicht mit einer tatsächlichen Verschiebung von Modellteilen verbunden sind. Die von beteiligten Simulatorinstanzen gewonnenen Lastinformationen werden zentral parallel zur laufenden Simulation von einem Backtracking-Algorithmus auf der Basis imaginärer Verschiebungen von Modellteilen ausgewertet. Liefert der Algorithmus einen Vorschlag zur Lastverschiebung, so wird dieser an Synchro-

nisationspunkten der parallelen Simulation über ein dynamisches Partitionsmanagement wirksam.

Die Entwicklung und Untersuchung spezieller Partitionierungsalgorithmen in Zusammenhang mit redundanten Verteilungen von Modellteilen für Simulationen mit *dlbSIM* stellen einen zweiten Zielschwerpunkt unseres Fortsetzungsantrages dar.

3.5 Erweiterung des Applikationsfelds

Wie für die von uns betrachtete Logiksimulation ist auch für die Logiksynthese eine Parallelisierung nach dem *Replicated Worker* - Prinzip auf der Basis einer cone-bezogenen Modellpartitionierung vorteilhaft. Auf dem 3. Workshop *Parallel Logic Simulation and Synthesis* im Dezember 1997 in Böblingen haben wir mit unseren Praxispartnern von IBM die Möglichkeit der Übertragung von Partitionierungsstrategien aus dem Kontext der parallelen Logiksimulation diskutiert. In einer momentan laufenden Arbeit wird ausgehend von dem durch R. REILEIN entwickelten Algorithmus *SLUCC* (*Source Latch Unifying Cone Clustering*) [Rei98] ein synthesespezifisches Partitionierungsverfahren entwickelt.

Desweiteren stehen wir in engem Kontakt zu dem als Gruppe von 3 DFG-Anträgen laufenden Projekt *Parallelisierung der numerischen Modelle ausgewählter meteorologischer und luftchemischer Prozesse für lose gekoppelte Rechnerarchitekturen*, in dem das Institut für Informatik (Frau Prof. G. RÜNGER, Prof. W.G. SPRUTH) und das Meteorologische Institut (Prof. G. TETZLAFF) der Universität Leipzig sowie das Institut für Troposphärenforschung e.V. Leipzig (Prof. E. RENNER) kooperieren. In die Arbeit dieses Projektes können wir insbesondere unsere Erfahrung bei der Entwicklung von Kostenmodellen für parallele Prozeßsysteme einbringen. Auf dem 1. Projektworkshop im April 1998 wurde von K. HERING ein Vortrag zur Entwicklung paralleler Programme gehalten.

4 Vorträge und Veröffentlichungen

Im Berichtszeitraum haben wir folgende projektrelevante Vorträge gehalten:

1. K. Hering: Parameterized Partition Valuation for Parallel Logic Simulation. IASTED International Conference on Parallel and Distributed Computing and Networks (PDCN'97), Singapore, 12. August 1997

2. R. Haupt: Hierarchical Model Partitioning for Parallel VLSI-Simulation Using Evolutionary Algorithms Improved by Superpositions of Partitions. 5th European Congress on Intelligent Techniques and Soft Computing (EUFIT'97), Aachen, 11. September 1997
3. K. Hering: Lastaspekte bei der parallelen compilergesteuerten Logiksimulation. Oberseminar AG Monien, Universität Paderborn, 18. November 1997
4. K. Hering: Partitioning for Simulation. 3rd Workshop on Parallel Logic Simulation and Synthesis, IBM Böblingen, 9. Dezember 1997
5. K. Hering: Dynamic Load Balancing. 3rd Workshop on Parallel Logic Simulation and Synthesis, IBM Böblingen, 9. Dezember 1997
6. R. Reilein: Model Build. 3rd Workshop on Parallel Logic Simulation and Synthesis, IBM Böblingen, 9. Dezember 1997
7. J. Löser: Time Measurement. 3rd Workshop on Parallel Logic Simulation and Synthesis, IBM Böblingen, 9. Dezember 1997
8. K. Hering: Aspekte der Entwicklung paralleler Programme. 1. Workshop zum DFG-Projekt *Parallelisierung der numerischen Modelle ausgewählter meteorologischer und luftchemischer Prozesse für lose gekoppelte Rechnerarchitekturen*, LIM Institut für Meteorologie der Universität Leipzig, 16. April 1998
9. K. Hering: Parallele compilergesteuerte Logiksimulation. 1. Halle-Leipziger Seminar zur Parallelverarbeitung (HLP), Institut für Informatik der Universität Leipzig, 8. Juni 1998
10. J. Löser: Dynamische Lastbalancierung bei der parallelen Logiksimulation. 1. Halle-Leipziger Seminar zur Parallelverarbeitung (HLP), Institut für Informatik der Universität Leipzig, 8. Juni 1998
11. H. Schulze: Parallele Evolutionäre Algorithmen zur Modellpartitionierung. 1. Halle-Leipziger Seminar zur Parallelverarbeitung (HLP), Institut für Informatik der Universität Leipzig, 8. Juni 1998

12. H. Hennings: ParallelMAP-Eine Client-Server Architektur zur Modellanalyse und -partitionierung. 1. Halle-Leipziger Seminar zur Parallelverarbeitung (HLP), Institut für Informatik der Universität Leipzig, 8. Juni 1998
13. R. Haupt: Integration of a Local Search Operator into Evolutionary Algorithms for VLSI-Model Partitioning. 6th European Congress on Intelligent Techniques and Soft Computing (EUFIT'98), Aachen, 8. September 1998
14. K. Hering: Cycle-Based Simulation on Loosely-Coupled Systems. 11th Annual IEEE International ASIC Conference (ASIC'98), Rochester (NY), 16. September 1998
15. K. Hering: The Leipzig Simulation Project. VLSI Design Systems, IBM Austin (TX), 21. September 1998
16. J. Löser: Dynamische Lastbalancierung bei der parallelen Logiksimulation. Oberseminar AG Monien, Universität Paderborn, 1. September 1998
17. K. Hering, H. Hennings: Parallele Modellpartitionierung und Logiksimulation. 4. Kolloquium des DFG-Schwerpunktes Effiziente Algorithmen für diskrete Probleme und ihre Anwendungen, Köln, 26. November 1998 (Vortrag und Softwaredemonstration)
18. R. Haupt: Parallel Evolutionary Algorithms with SOM-like Migration and their Application to Real World Data Sets. 4th International Conference on Artificial Neural Networks and Genetic Algorithms (ICANN-GA'99), Portoroz, Slowenien, 7. April 1999

Mit dem Poster

H. Hennings, W.G. Spruth, K. Hering, U. Petri, R. Haupt:
parallelMAP - Eine parallele Laufzeitumgebung zum automatischen Verteilen und Verwalten von Applikationen auf Workstationclustern und Parallelrechnern

haben wir uns neben dem 4. Kolloquium unseres Schwerpunktprogramms auch am 2. Halle-Leipziger Seminar zur Parallelverarbeitung (HLP) in Halle (11. Januar 1999) beteiligt.

Veröffentlichungen erfolgten in den Proceedings nachstehender Konferenzen: PDCN'97 [HHP97], EUFIT'97 [HHPV97], EUFIT'98 [HHS98], ASIC'98 [DHS98] und ICANNGA'99 [VHHS99]

Beiträge zu PARCO'99 [SHH99b], EUFIT'99 [SHH99a] und PDCS'99 [HHH99] sind angenommen worden.

5 Kooperation mit Praxispartnern

Wir arbeiten unmittelbar mit der *IBM Deutschland Entwicklung GmbH*, Abteilung *Processor Development 1* (P. KILLES, H.-W. ANDERSON) in Böblingen und der *IBM Corporation*, Abteilung *VLSI Design Systems* (W. ROESNER, J. LONG) in Austin (TX) zusammen. Aus der Arbeit dieser Einrichtungen ging die Aufgabenstellung zur Parallelisierung des Simulators *TEXSIM* und seines Nachfolgers *MVLSIM* hervor. *IBM* stellte uns den Code der sequentiellen Simulatoren, die *Design Automation Database DA_DB* und damit verbundene Tools, Upgrades für das Betriebssystem *AIX* und das *AIX Parallel Environment* sowie Modelle komplexer Prozessorstrukturen zur Verfügung. Im Berichtszeitraum konnten wir auch weiterhin einen *IBM 595 POWERServer* als Leihmaschine in Leipzig nutzen. Aufgrund einer Umstrukturierung und Aufgabenverlagerung innerhalb von *IBM* waren Testläufe auf *IBM SP* Parallelrechnern in Böblingen nicht in dem ursprünglich geplanten Umfang durchführbar.

Vom 9. bis 11. Dezember 1997 fand in Böblingen der im Rahmen unserer Kooperation 3. Workshop unter dem Thema *Parallel Logic Simulation and Synthesis* statt. Die Schwerpunkte des Vortrags- und Diskussionsprogramms lassen sich wie folgt zusammenfassen:

- Stand und Ausblick in Bezug auf die Entwicklung von Modellpartitionierungsalgorithmen für die parallele Logiksimulation
- Vervollkommnung der Partitionierungsumgebung *parallelMAP*
- Integration dynamischer Lastbalancierungsalgorithmen in *parallel-MVLSIM*

- Übertragung von Modellpartitionierungsstrategien auf die parallele Logiksimulation auf das Gebiet der parallelen Logiksynthese

Im Rahmen eines Arbeitsaufenthalts von K. HERING bei *IBM* in Austin (TX) im September 1998 wurden Stand und zukünftige Möglichkeiten der Kooperation diskutiert. Angesichts der weiteren Entwicklung des Simulators *MVLSIM* in Austin wurde für einen deutschen Studenten ein sechsmonatiges Praktikum in Texas mit dem Ziel der Übertragung und Anpassung der Simulatoren *parallelMVLSIM* und *dlbSIM* vereinbart. Momentan bereitet sich darauf in Leipzig der Student J. MARKWARDT vor.

Im Berichtszeitraum gab es eine Reihe von Studentenaufenthalten bei *IBM* in Böblingen. Als Beispiele genannt seien die Aufenthalte von H. SCHULZE (Experimente in Bezug auf parallele evolutionäre Algorithmen), J. LÖSER (Untersuchungen zur dynamischen Lastbalancierung), H. HENNINGS (Implementierung von *parallelMAP*) und D. LUCKE (Entwicklung von Modellpartitionierungsalgorithmen zur parallelen Logiksynthese).

6 Bezug zur Lehre

Im Kontext unserer Projektarbeit ermöglichen wir Informatik-Studenten im Hauptstudium die Mitgestaltung eines offenen Problemseminars "Theoretische und praktische Aspekte der Modellpartitionierung im Umfeld der Logiksimulation". Für direkt an unserem Projekt beteiligte Studenten streben wir an, Arbeitslinien ausgehend von Berufs-Praktika über eine Tätigkeit als studentische Hilfskraft bis hin zur Diplomarbeit aufzubauen. In diesem Zusammenhang möchten wir aus unserer Erfahrung heraus erneut betonen, daß die projektbezogene Kombination der Potentiale von *Universität*, *DFG* und *Industrie* eine sehr fruchtbare Wirkung auf Forschung und Lehre hat. Betrachtet man den gesamten Förderzeitraum, waren bisher insgesamt 12 Studenten in unser Projekt eingebunden. In gemeinsamer Betreuung mit *IBM* wurden 5 Berufspraktika absolviert. Es entstanden 7 Diplomarbeiten, davon die folgenden im Berichtszeitraum:

- Robert Reilein: Modellpartitionierung zur parallelen Logiksimulation [Rei98]
- Jork Löser: Dynamische Lastbalancierung bei der parallelen Logiksimulation [Lös98]

- Hendrik Schulze: Entwicklung, Untersuchung und Implementierung von Algorithmen für die Modellpartitionierungskomponente *parallelMAP* [Sch98]
- Thomas Siedschlag: Iterative Modellpartitionierungsverfahren für die parallele Logiksimulation [Sie98]
- Hilmar Hennings: Entwurf und Implementierung der Komponente *parallelMAP* zur Modellanalyse und -partitionierung im Kontext von *parallelTEXSIM* [Hen99]

Ein weiteres Diplomthema befindet sich momentan in Bearbeitung:

- Daniel Lucke: Modellpartitionierung zur parallelen Logiksynthese

Literatur

- [DHS98] D. Döhler, K. Hering, and W.G. Spruth. Cycle-Based Simulation on Loosely-Coupled Systems. In *Proc. of the 11th Annual IEEE International ASIC Conference (ASIC'98)*, pages 301–305, Rochester, New York, 1998.
- [Hen99] H. Hennings. *Entwurf und Implementierung der Komponente parallelMAP zur Modellanalyse und -partitionierung im Kontext von parallelTEXSIM*. Diplomarbeit, Universität Leipzig, Fakultät für Mathematik und Informatik, 1999.
- [Her96] K. Hering. Parallel cycle simulation. Technical Report 13(96), Department of Computer Science, University of Leipzig, 1996.
- [HHH99] K. Hering, H. Hennings, and R. Haupt. DRIVE: A Distributed Run-Time Environment Supporting Combination of Sequential and Parallel Program Modules. 1999. Accepted at 11th IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS'99).
- [HHP97] K. Hering, R. Haupt, and U. Petri. Parameterized partition valuation for parallel logic simulation. In *Proc. of the Conference on Parallel and Distributed Computing and Networks (PDCN'97)*, pages 144–150. IASTED/Acta Press, Anaheim, 1997.

- [HHPV97] R. Haupt, K. Hering, U. Petri, and T. Villmann. Hierarchical Model Partitioning for Parallel VLSI-Simulation Using Evolutionary Algorithms Improved by Superpositions of Partitions. In *Proc. of the 5th European Congress on Intelligent Techniques and Soft Computing (EUFIT'97), Volume 1*, pages 804 – 808. Verlag Mainz, 1997.
- [HHS98] R. Haupt, K. Hering, and Th. Siedschlag. Integration of a Local Search Operator into Evolutionary Algorithms for VLSI-Model Partitioning. In *Proc. of the 6th European Congress on Intelligent Techniques and Soft Computing (EUFIT'98), Volume 1*, pages 377 – 381. Verlag Mainz, 1998.
- [Lös98] J. Löser. *Dynamische Lastbalacierung bei der parallelen Logiksimulation*. Diplomarbeit, Universität Leipzig, Fakultät für Mathematik und Informatik, 1998.
- [Rei98] R. Reilein. *Modellpartitionierung zur parallelen Logiksimulation*. Diplomarbeit, Universität Leipzig, Fakultät für Mathematik und Informatik, 1998.
- [Sch98] H. Schulze. *Entwicklung, Untersuchung und Implementierung von Algorithmen für die Modellpartitionierungskomponente parallelMAP*. Diplomarbeit, Universität Leipzig, Fakultät für Mathematik und Informatik, 1998.
- [SHH99a] H. Schulze, R. Haupt, and K. Hering. Dynamic Fitness Function for Parallel Evolutionary Partitioning Algorithms in the Context of Parallel Logic Simulation. 1999. Accepted at 7th European Congress on Intelligent Techniques and Soft Computing (EUFIT'99).
- [SHH99b] H. Schulze, R. Haupt, and K. Hering. Experiments in Parallel Evolutionary Partitioning. 1999. Accepted at Parallel Computing (PARCO'99).
- [Sie98] T. Siedschlag. *Iterative Modellpartitionierungsverfahren für die parallele Logiksimulation*. Diplomarbeit, Universität Leipzig, Fakultät für Mathematik und Informatik, 1998.

- [VHHS99] T. Villmann, R. Haupt, K. Hering, and H. Schulze. Parallel Evolutionary Algorithms with SOM-Like Migration and their Application to Real World Data Sets. In *5th International Conference on Artificial Neural Networks and Genetic Algorithms (ICANN-NGA '99)*, pages 274 – 279. Springer-Verlag, Wien, New York, 1999.