

Modellierung und Meßunterstützung mit dem Werkzeug PEPP

Franz Hartleb, Peter Dauphin, Rainer Klar, Andreas Quick, Markus Siegle

Institut für Mathematische Maschinen und Datenverarbeitung VII
Universität Erlangen-Nürnberg, Martensstraße 3, 91058 Erlangen
{dauphin,klar,siegle}@informatik.uni-erlangen.de

Kurzfassung: PEPP erlaubt die ablauforientierte Modellierung paralleler Programme mit stochastischen Graphmodellen. Exakte und approximative Auswerteverfahren ermöglichen Laufzeitvorhersagen für Graphmodelle von unterschiedlichem Typus und Komplexität. Darüberhinaus bietet PEPP Funktionen zur Unterstützung der modellgesteuerten Messung und die Möglichkeit, Modelle mit beliebigen gemessenen Verteilungsfunktionen zu parametrieren.

Modelling and Monitoring Support with the Tool PEPP: PEPP is a tool for modelling parallel programs using stochastic graph models. It provides exact and approximate analysis methods for runtime prediction of different graph model types. PEPP also supports model-driven monitoring and allows to use arbitrary measured distributions for model parametrization.

1 Einleitung

Dieses Papier beschreibt das Modellierungswerkzeug PEPP (*Performance Evaluation of Parallel Programs*) und die ihm zugrunde liegenden Analyseverfahren. PEPP entstand aus unserem Interesse an der Leistungsbewertung paralleler Programme. Dafür werden ablauforientierte Verfahren eingesetzt, die die Ablaufdynamik derart repräsentieren, daß die Modelle *Leistungsvorhersagen* für den kompletten Programmablauf auf einer gegebenen Rechnerkonfiguration erlauben und eine *modellgesteuerte Messung* unterstützen.

Zusammen mit der rasanten Entwicklung moderner Hardware und Software hat sich das Interesse des Leistungsbewerthers von der Betrachtung der reinen Maschinenleistung immer mehr hin zur Bewertung von Rechensystemen unter komplexer, strukturierter Last verlagert [7]. Je differenzierter die Ablaufstruktur der Programme modelliert wird, desto stärker steht bei der Modellierung die Beschreibung der Last und ihrer Ablaufdynamik im Vordergrund. Bei parallelen Algorithmen ist die Dauer paralleler Rechenphasen meist nicht gleichlang. Zudem gibt es neben dem Verwaltungsaufwand für die Abwicklung der Parallelisierung auch das Problem funktionaler Abhängigkeiten zwischen den sequentiellen und parallelen Programmabschnitten. Hält man sich diese Tatsachen vor Augen, so wird deutlich, warum Graphmodelle [9, 10], die einer solchen Beschreibung entgegenkommen, betrachtet werden.

Zur Darstellung eines Programmablaufs als Graphmodell wird der Algorithmus in sequentiell und parallel ablauffähige Einheiten, sogenannte *Teilaufgaben* zerlegt, zwischen denen *Reihenfolgebeziehungen* angegeben werden, die die funktionalen Abhängigkeiten der Teilaufgaben widerspiegeln. Studien zum Zweck der Leistungsbewertung setzen voraus, daß neben dem funktionalen Verhalten auch zeitliche Aspekte durch das Graphmodell beschrieben werden. Den einzelnen Teilaufgaben werden daher *Ausführungszeiten* bzw. *Ausführungszeitverteilungen* in Form kontinuierlicher Zufallsvariablen zugeordnet. Dies ermöglicht die Berechnung der Gesamtausführungszeit.

Sei $V = \{v_1, \dots, v_n\}$ die Menge der den Teilaufgaben entsprechenden *Knoten* v_i , und sei $E \subseteq V \times V$ die Menge der *gerichteten Kanten*, die die Reihenfolgebeziehungen ausdrücken. Der *gerichtete Graph* (directed graph, digraph) G ist dann definiert als $G = (V, E)$. Ist der gerichtete Graph G *endlich* und *azyklisch*, so heißt er *Präzedenzgraph*. Ein Präzedenzgraph, bei welchem jedem Knoten v_i eine stochastisch verteilte Laufzeit in Form einer kontinuierlichen Zufallsvariablen X_i zugeordnet ist, heißt *Stochastisches Graphmodell*.

Wir wollen uns hier auf Präzedenzgraphen beschränken und zunächst den scheinbaren Nachteil akzeptieren, daß das in Programmen wichtige Konstrukt der Schleife im Präzedenzgraphen nicht vorkommt. Wir beschreiben in Abschnitt 2 dann eine Möglichkeit, mit diesem Problem umzugehen, indem zyklische Knoten eingeführt werden.

2 Der Modellwerkzeugkasten PEPP — Übersicht

In Abb. 1 ist die graphische Bedienoberfläche von PEPP [2] mit dem Hauptmenü und einem Ausschnitt aus einem Graphmodell dargestellt. Die Oberfläche umfaßt einen graphischen Modell-editor, dient zur Auswahl der Analyseverfahren (s. Abschnitt 3) und hat Schnittstellenfunktion zu anderen Tools für die Integration von Messung und Modellierung (s. Abschnitt 4).