Seminar

Analyse von Programmausführungszeiten

Prof. Dr. Reinhard von Hanxleden

Vortrag zu Thema
Analysen von WCET-Analysen
von Lars Kühl
im WS 2002 / 2003
am 13. Januar 2003

Übersicht

- Einleitung
- Charakterisierung von Analysen
- Statische Analyse
- Genetische Algorithmen / Evolutionäres Testen
- Vergleich zwischen der Statischen Analyse und Evolutionärem Testen
- Zusammenfassung
- Literaturverzeichnis

Einleitung

- Bisher hatten wir nur Teile zur Berechnung der WCET eines Systems betrachtet (Cache, Pipelinesund die Pfadanalyse)
- Hier werden zwei Verfahren zur Bestimmung der WCET verglichen.
- Vergleich der Verfahren an ausgewählten Algorithmen
- Beurteilung der verschieden Verfahren

Low-Level Analyse I: Caches (Ingo Schiller)
Low-Level Analyse II: Pipelining (Hendrik Janz)
Pfad Analyse (Oliver Otte)

WCET

- Die Berechnung der Worst Case Execution Time ist ein NP-hartes Problem (man kann es auf das Halteproblem zurückführen)

NP : Es existiert ein nichtdeterministischer Algorithmus, der das Problem in polynomieller Zeit löst.

NP hart : Man kann das Problem auf ein anderes NP -vollständiges Problem zurückführen. Für das Halteproblem wurde dies in Informatik IV gezeigt.

Charakterisierung von Analysen

- Für die White Box Analyse benötigen wir ein Umfangreiches Wissen über die Hardware (Cache , Pipelines) und die Implementierung der Software (Flow Analyse)
- Für die Black Box Analyse benötigen wir nur das fertige zu Analysierende System und genaues Wissen über die Eingabe in das System

Korrektheit von Programmen

Es gibt zwei unterschiedliche Arten der Korrektheit

- Die logische Korrektheit des Programmes. Hier für gibt es bereits viele Analysearten.
- Und die temporale Korrektheit, für diese Überprüfung gibt es erst wenige Methoden. Da die temporale Korrektheit von der WCET abhängt .

Statische Analyse

- Die Statische Analyse (SA) ist eine Methode zur Berechnung der WCET.
- Die SA wird hauptsächlich in der Entwicklungsphase eingesetzt um die WCET bzw. die BCET zu bestimmen.
- In die SA gehen die bereits bekannten Verfahren der Low Level Analyse und die Pfadanalyse ein

Wenn man die Statische Analyse zur Betrachtung des temporalen Verhaltens eines Programmes benutzt, kann man in der Regel neben der WCET auch die BCET ohne großen zusätzlichen Aufwand berechnen.

Verschiedene Ansätze der Statische Analyse

- Quellcode orientiert, hier wird die WCET über „trees of timing nodes“ und pipelineing bestimmt
- Hardware Simmulatoren, die um den Ablauf mit einer unbekannten Eingabe zu simmulieren
- Constraint based, d.h die WCET wird mit Hilfe der ILP berechnet.
- Data-Flow based, wobei das cache Verhalten getrennt von der Pipeline Simulation betrachtet wird

Das Tool, welches ich im folgenden betrachte basiert auf dem Data Flow.

Die Methode, die Quellcode betrachtet wird hauptsächlich auf RISC Prozessoren mit First level split caches angewandt

Die Constraints based Methode versucht die Architektur zu modellieren, und man berechnet die WCET dann mit der ILP aus.

Ein SA Tool

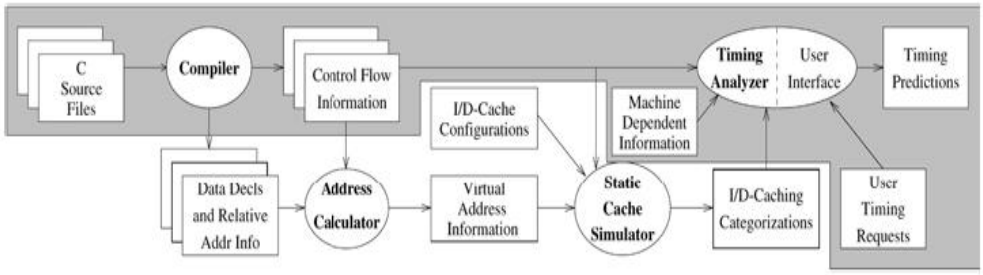


Figure 1. Framework for timing predictions.

Das Bild stammt aus der Quelle II Seite 246

Nur der grau Unterlegte Teil wird im folgenden betrachtet.
Die in den anderen Quellen benutzten SA Tools sind ähnlich aufgebaut, so dass sich eine separate Betrachtung nicht lohnt.
Das hier betrachte Tool bekommt einen C Sourcecode der mit dem VPPC/VPO Compiler übersetzt wird. Der Einsatz des gcc soll auch bald möglich sein.
Der Compiler wurde dahingehend modifiziert, das er den Control Flow ausgibt.
Der Timing Analyser übernimmst die Pfadanalyse und ermittelt die WCET und BCET .

Warum Evolutionäres Testen (ET)

Wir wollen die WCET über einen Blackbox Test erhalten. D.h. wir kennen nur die Eingabedaten.

Um die WCET zu bestimmen müsste man eigentlich alle Möglichkeiten testen. Dies ist aber bei komplexen Systemen sehr aufwendig ($\mathcal{O}(2^n)$).

Mit Hilfe des ET versucht man, Maximalwerte zu finden,ohne alle Möglichkeiten zu überprüfen.

ET wird mit genetischen Algorithmen realisiert

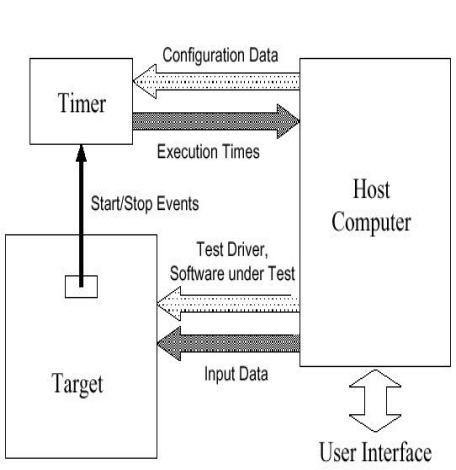
Struktur der Genetische Algorithmen

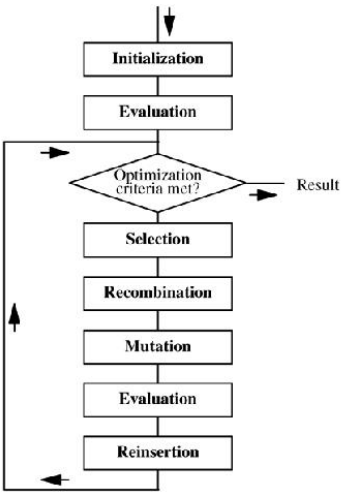
Genetische Algorithmen bestehen aus

- Genen, den Eingabedaten (Integer Strukturen)
- Chromosomen, eine Zusammenfassung von mehreren Genen zu einer Einheit z.B ein Chromosom ist eine Matrix bei der Matrizenmultiplikation.
- Individium, ein kompletter Eingabedatensatz
- Der Population, eine große Anzahl von Individuen

Eine genauere Erklärung der Genetischen Algorithmen findet man in der Quelle I
Hier wurde nahezu der gesammte Aufbau und die Begriffe aus der Biologie übernommen.
Die Größe der Population hängt von der Komplexität des Problemes ab.

Arbeitsweise der GA bzw. ET





Vortragender: Lars Kühl Datum: 13. Januar 2003 Folie 12

Die Bilder stammen aus der Quelle I Seite 2 und aus der Quelle II Seite 251

Zum linken Bild
Hier ist der Testaufbau dargestellt
Die hellen Pfeile stellen den Datenfluss bei der Initialisierung , die dunklen Pfeile den Datenfluss während der Ausführung dar .

Die Ausführungszeiten werden über einen extra Timer ermittelt, da die Ausführungszeit Messung nicht das Zielsystem zusätzlich belasten soll.
Der Host führt den gesamten Test aus, d.h. auf ihm „lebt“ die Population

Zum rechten Bild
Hier ist der generelle Ablauf des ET dargestellt. Die einzelnen Phasen werden im folgenden beschrieben.

Schritte beim ET

- Initialization, die erste Population wird durch Zufallswerte erstellt
- Evaluation
- Optimization Criteria
- Selection
- Recombination
- Mutation
- Reinsertion

Evaluation / Fitness

- In der Evaluation Phase wird der Fitnesswert der einzelnen Individuen berechnet. Also indem man das RTS mit der Population testet.
- Wie der Fitnesswert berechnet und bewertet wird, hängt von der Problemstellung ab. Für die WCET sollte die Laufzeit möglichst groß sein
- Der Fitnesswert entscheidet in einer späteren Phase über das verbleiben eines Individuums in der Population

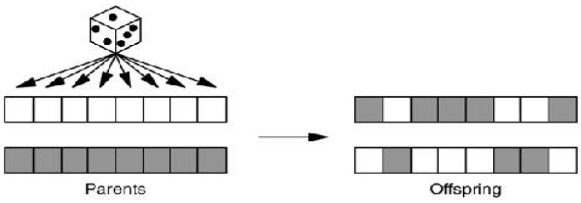
Bei den Versuchsaufbau wird der Fitnesswert vom Timer geliefert.

Optimization Criteria

- Hier wird überprüft, ob es Fitnesswerte gibt, die gegen eine bestimmte Bedingung erfüllen.
- Bei RTS gegen die maximale Laufzeit. D.h die WCET ist beim RTS zu groß.
- Ein weiteres Kriterium ist die Anzahl der Generationen, die erstellt werden,wenn die Abbruchbedingung nicht erfüllt wird. Dies hängt aber von der Anzahl und Komplexität der Eingabedaten ab.(Hier wir noch geforscht)

Selection / Recombination / Mutation

- Es werden zwei Individuen anhand ihrer Fitnesswerte ausgesucht, die Gene gemischt und ein neues Individuum erstellt



- Bei den so erzeugten „Kinder“ wird jetzt mit einer vorher festgelegten Wahrscheinlichkeit einzelne Bits in den Genen verändert.

Reinsertion

- Von den Kindern wird nun ebenfalls der Fitnesswert bestimmt.
- Da die Anzahl der Individuen in der Population begrenzt ist, werden nun die Individuen mit dem schlechtesten Fitnesswerte aus der Population entfernt (Entspricht Darwin).

Charles Darwin geboren am 12.2.1809 in Shrewsbury, England, gestorben am 19.04. 1882. Darwin entwickelte in seinem Buch On the Origin of Species by Means of Natural Selection seine Theorie über die Evolution.

RTS bei denen man SA und ET verglichen hat

Zunächst drei RTS mit unterschiedlicher Komplexität

- Ein Grafikalgorithmus, der überprüft, ob eine Line von einem Rechteck überdeckt wird
- Ein Eisenbahnkontrollsystem, welches Diskrepanzen in einem redundanten System entdeckt
- Ein Sicherheitssystem, welches Bilder analysiert

und noch die Matrixmultiplikation und Bubblesort

Die Versuche wurden auf einem Sparc IPX mit Solaris 2.3 und 40 Mhz durchgeführt.(Keine Caches)
Der Grapfikalgorithmus erhält als Eingabe drei Koordinaten und die Höhe und Breite des Rechteckes.
Das Eisenbahnkontrollsysem benötigt 512 Eingabeparameter.
Das Sicherheitssystem benötigt 843 Eingabeparameter.
Bei der Matrixmultiplikation wurden 50x50 Matrizen verwendet.
Bubblesort wurde auf ein Feld der Länge 500 angesetzt.

Ergebnis des Vergleichs

Table 3. Execution times (cycles) and % difference for industrial test programs.

Program	Graphics		Railroad		Defense	
Method	Best	Worst	Best	Worst	Best	Worst
SA	309	2602	389	23,466	848	71,350
ET	457	2176	508	22,626	9095	35,226
Difference	32%	16%	24%	4%	N/A	N/A

Table 4. Execution times (cycles) and % difference from actual time for standard benchmarks.

Program	Matrix				Sort			
Method	Best		Worst		Best		Worst	
SA	8,411,378	18%	15,357,471	16%	16,003	24%	12,268,014	3%
Actual	10,315,619		13,190,619		20,998		11,872,718	
ET	12,050,569	17%	13,007,019	1%	20,998	0%	11,826,117	1%

Die maximale Anzahl der Generationen wurde bei den verschieden Test unterschiedlich angesetzt.
Grafik 50 Generationen und 50 Individuen
Eisenbahn 100 Generationen und 100 Individuen
Sicherheit 300 Generationen und 100 Individuen
Matrixmultiplikation 800 Generationen und 100 Individuen
Bubblesort 500 Generationen und 300 Individuen

Beim Sicherheitsprogramm wurde festgestellt, das mehre gleiche Befehle unterschiedlich Zeit beim Pipelining benötigten. Dies trat bei den Befehlen zur Multiplikation und Division auf.

Statische Analyse

Pro

- Die berechnete WCET ist für jeden Fall korrekt und es kann zu keiner schlimmeren Ausführungszeit kommen

Contra

- Man muss über sehr genaue Kenntnisse der Hard und Software verfügen, um die WCET zu berechnen
- Fehler bei der Spezifikation der HW/ SW führen zu einer falschen WCET

Evolutionäres Testen

Pro

- Man braucht nur das zu testende System und Kentnisse über die Eingabedaten

Contra

- Die WCET vom ET kann völlig falsch sein, falls Pipelines im System vorkommt
- Die getestet WCET ist kleiner gleich der richtigen WCET

Folgerungen aus dem Test

- Man sollte auf jeden Fall beide Methoden bei der Systementwicklung benutzen .
- SA wird während des Entwicklungsstadiums benutzt
- ET wird in der Testphase benutzt, um Fehler bei der SA zu entdecken (z.B. falsche Cachekonfiguration etc.)
- Beide Verfahren zusammen sollten die Fehlerwahrscheinlichkeit stark reduzieren

Zusammenfassung

- Die Statische Analyse basiert auf der theoretischen Berechnung der WCET über die bekannten Arten und nähert sich der WCET von oben.
- Die SA hat den Nachteil der Empfindlichkeit auf Fehler in der HW SW Beschreibung
- Das Evolutionäre Testen basiert auf dem randomisierten Testen und der natürlichen Auslese(Darwin). Es nährt sich der WCET von unten her. Ein Problem ist die geeignete Abbruchbedingung.

Zusammenfassung

- Es folgt also EV <= WCET <= SA.
- Man sollte also beide Verfahren bei der Entwicklung eines (Real Time) Systems verwenden um eine gewisse Sicherheit bezüglich der WCET zu haben.

Literaturverzeichnis

I. **Peter P. Puschner, Roman Nossal, Testing the Results of Static Worst-Case Execution-Time Analysis. Proceedings of the 19th IEEE Real-Time Systems Symposium, Madrid, Spain, 2-4 December, 1998, pp. 134-143.**

II. **Joachim Wegener, Frank Mueller, A Comparison of Static Analysis and Evolutionary Testing for the Verification of Timing Constraints, Real-Time Systems, Kluwer, November 2001, Volume 21, Number 3, pp. 241-268**

III. Jakob Engblom, Andreas Ermedahl, Friedhelm Stappert, Validating a Worst-Case Execution Time Analysis Method for an Embedded Processor, Uppsala University, Dept. of Information Technology, Technical Report 2001-030, December 2001

IV. Jakob Engblom, Andreas Ermedahl, Friedhelm Stappert, Structured Testing of Worst-Case Execution Time Analysis Methods, Real-Time Systems Symposium, Orlando Florida, Nov. 2000

V. Jakob Engblom, Andreas Ermedahl, Friedhelm Stappert, Comparing Different Worst-Case Execution Time Analysis Method, Real-Time Systems Symposium, Orlando Florida, Nov. 2000