

IDPA: Programmiersprachen benchmarken

Nick Zbinden und Matthias Gasser

21. Februar 2011

1 Vorwort

Dieses Projekt ist aus dem Willen heraus entstanden eine Arbeit zu machen die Applikations- und Systemtechnikenkenntnis nutzt und verbindet. Nach verschiedenen Ideen einigten wir uns ...

2 Abstract

3 Einleitung

Wir haben uns ein sehr kompliziertes Thema vorgenommen. Programmiersprachen und Compiler sind, zumindest für die Informatik, ein sehr altes Thema.

Hochsprachen wie wir Sie in dieser Arbeit verwenden werden seit mehr als fünfzig Jahren immer wieder weiter entwickelt und verbessert. Seit der Anfangszeit herrscht der Konflikt zwischen hoher Abstraktion und Geschwindigkeit. Umso höher die Abstraktion ist die eine Programmiersprache bietet umso schwieriger ist die Programmiersprache auf der darunterliegenden effizient auszuführen.

Wir haben uns die Aufgabe gestellt drei Programmiersprachen unter dem Aspekt der Geschwindigkeit anzuschauen und zu vergleichen. Ein definitives Ergebnis ist in diesem Bereich fast unmöglich da die Anwendungsgebiete einer Programmiersprache zu verschieden sind und jeder Anwendungsfall andere Requirements hat. Auch die unter der Programmiersprache liegenden Layers (Betriebssysteme und Hardware) sind von entscheidender Bedeutung. Um korrekte Aussagen zu machen müssen diese Layers entweder herausgerechnet werden oder identisch sein.

Um komplizierte Algorithmen in drei verschiedenen Sprachen zu programmieren braucht man viel Erfahrung in diesen Sprachen. Um den Aufwand nicht zu hoch werden zu lassen haben wir uns auf die Implementierung in einer Programmiersprache spezialisiert und vergleichen diese mit reference Implementationen die wir nur Erklären und messen.

4 Material und Methoden

4.1 Allgemeines zu den Versuchen

4.2 Verwendete Programmiersprachen

4.3 Das Testsystem

4.4 Aufnahme der Daten

4.5 Datenauswertungsmethoden

5 Ergebnisse

5.1 Allgemeines zu den Sprachen

Um es einfacher und fair zu machen haben wir uns Entscheiden Programmiersprachen zu nehmen die auf der JVM lafen

5.2 Java

5.3 Scala

5.4 Clojure

Clojure ist eine dynamische Programmiersprache

6 Diskussion

6.1 Vergleich der Resultate

6.2 Schlussfolgerung

7 Abkürzungsverzeichnis

VM - Virtuall Maschine JVM - Java Virtuell Maschine IL - Interniediet Language

8 Literaturverzeichnis

9 Glossar

Virtuell Maschine: Eine Virtuell Maschine ist eine Programm welches simuliert eine echte Maschine zu sein. Sie bekommt als input irgend eine Sprache und fuert diese auf der darunter liegenden Sprache aus.

Java Virtuell Maschiene: Eine implementation einer VM die dafuer ausgelegt ist Java Bytecode auszufueren. Bytecode

Java Bytecode: The Java Bytecode wurde als Zielsprache fuer

10 Anhang