

Middleware - Cloud Computing – Übung

Tobias Distler, Klaus Stengel, Timo Hönig, Johannes Behl

Friedrich-Alexander-Universität Erlangen-Nürnberg
Lehrstuhl Informatik 4 (Verteilte Systeme und Betriebssysteme)

www4.informatik.uni-erlangen.de

Wintersemester 2011/12



Evaluation

Prüfung

Studentische Arbeiten am Lehrstuhl für Informatik 4



- Fokus der Arbeiten des Lehrstuhls für Informatik 4 lässt sich am besten mit dem Begriff *Systems Software Engineering* beschreiben
 - Betriebssysteme
 - Verteilte Systeme
- Forschungsgebiete von Rüdigers Arbeitsgruppe
 - Cloud Computing
 - Johannes und Klaus arbeiten an TClouds
 - <http://www.tclouds-project.eu>
 - Byzantinische Fehlertoleranz (BFT), ressourceneffiziente BFT-Systeme
 - Tobias
 - <http://www4.cs.fau.de/Research/REFIT/>
 - Energiegewahre Programmierung und energiegewahre Systeme
 - Timo
 - <http://www4.cs.fau.de/~thoenig/>





1981



2011



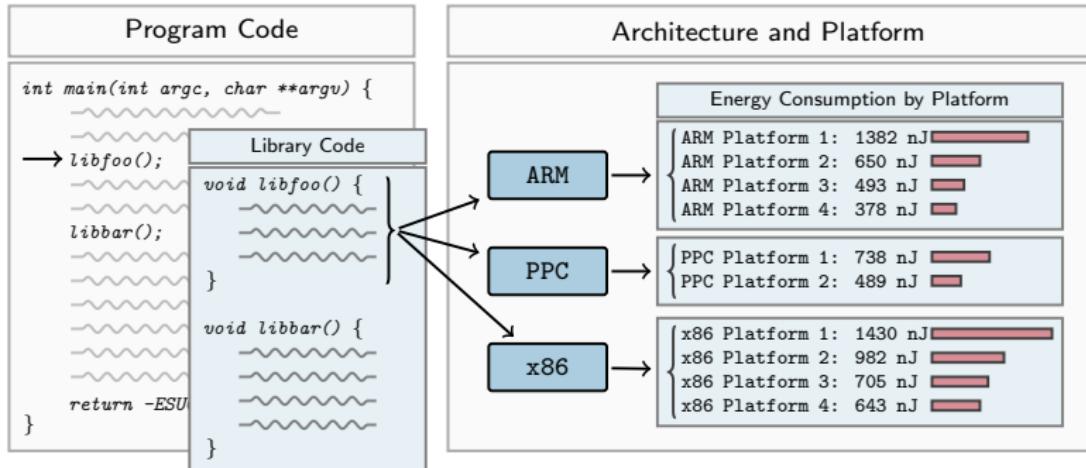
- Fortschritt der letzten zwei Jahrzehnte
 - Netzwerk: Faktor 3,3 Millionen
→ 300 Bit/s vs. 1 GBit/s
 - Speicherkapazität: Faktor 1,4 Millionen
→ 360 kByte vs. 500 GByte
 - Arbeitsspeicher: Faktor 0,5 Millionen
→ 4 kByte vs. 2 GByte
- Energiesparmechanismen
 - Dynamische Systemaspekte: aktive Beeinflussung des Systemverhaltens
 - Statische Systemaspekte: Compiler-Optimierungen, Speicherzugriffsmuster
- Energiegewahre Programmierung
 - Rückwärtsgerichteter Optimierungsprozess
 - Aufwendige Analysemethoden



- Fortschritt der letzten zwei Jahrzehnte
 - Netzwerk: Faktor 3,3 Millionen
→ 300 Bit/s vs. 1 GBit/s
 - Speicherkapazität: Faktor 1,4 Millionen
→ 360 kByte vs. 500 GByte
 - Arbeitsspeicher: Faktor 0,5 Millionen
→ 4 kByte vs. 2 GByte
 - Batterilaufzeit → 1 h vs. 10 h (Faktor 10)
- Energiesparmechanismen
 - Dynamische Systemaspekte: aktive Beeinflussung des Systemverhaltens
 - Statische Systemaspekte: Compiler-Optimierungen, Speicherzugriffsmuster
- Energiegewahre Programmierung
 - Rückwärtsgerichteter Optimierungsprozess
 - Aufwendige Analysemethoden



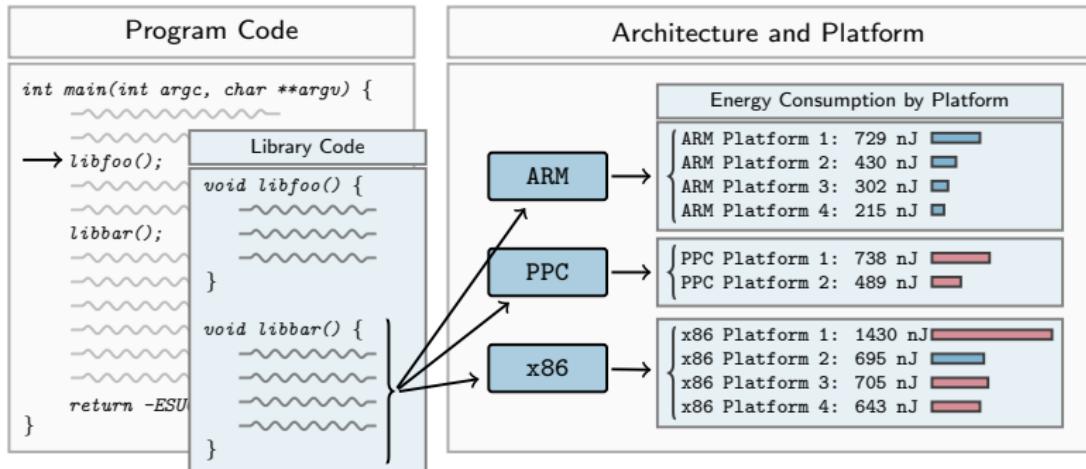
Energiegewahre Programmierung mit SEEP



- Vorwärtsgerichteter Prozess um Programmierer dabei zu unterstützen, energiegewahre Programme zu entwerfen
- Automatisierte Energieabschätzung für heterogene Plattformen



Energiegewahre Programmierung mit SEEP



- Vorwärtsgerichteter Prozess um Programmierer dabei zu unterstützen, energiegewahre Programme zu entwerfen
- Automatisierte Energieabschätzung für heterogene Plattformen



- Aktuelle Forschungsarbeiten
 - SEEP-Ansatz auf unterschiedliche Systemklassen skalieren
 - Anwendungsspektrum erweitern
 - Studien- und Diplomarbeiten
 - Bachelor- und Masterarbeiten
- System-Kontext
 - Energy-Aware High Performance Computing (HPC)
 - Sensornetzwerke



T. Hönig, C. Eibel, R. Kapitza, and W. Schröder-Preikschat

SEEP: Exploiting Symbolic Execution for Energy-Aware Programming

ACM Operating Systems Review (Best Papers from HotPower '11 Workshop on Power-Aware Computing and Systems) 45(3):58–62, 2012.

