

Energieeffiziente Ausführung von qualitätsbewussten Algorithmen für mobile Simulationen

Dominik Schreiber

Universität Stuttgart

mail@dominikschreiber.de

6. Oktober 2016

Gliederung

Einführung

Grundlagen

Systemmodell u. Anforderungen

Voruntersuchung

Entwurf

Auswertung

Motivation

- ▶ Mobilgeräte allgegenwärtig
- ▶ Anspruchsvolle Berechnungen rund um die Uhr
- ▶ Anwendungsgebiete wie Augmented Reality, Internet of Things, Wearables, ...
- ▶ „Explosion“ der Features von Smartphones gegenüber langsamer Evolution von Akkus [Ke12]



Bildquelle: [Wk]

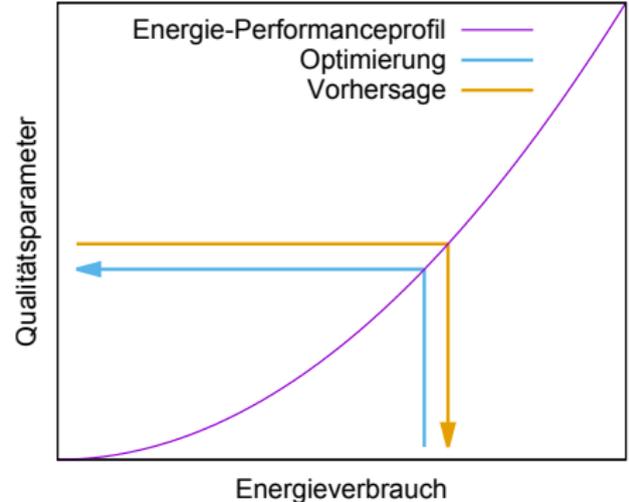
⇒ Laufzeit der Geräte leidet

Motivation

- ▶ Möglicher Lösungsansatz: Dynamische Anpassung der Berechnungsqualität und damit der verbrauchten Energie
- ▶ Verwendung von *qualitätsbewussten Algorithmen*, die gegebene Ressourcenschranken einhalten
- ▶ „Nur so gut berechnen, wie es die Situation zulässt“

Qualitätsbewusste Algorithmen

- ▶ *Qualitätsbewusste Algorithmen*
 - ▶ approximieren Ergebnis
 - ▶ Einhaltung vorgegebener Schranken auf Kosten der Ergebnisgüte [Z96]
- ▶ *Performance-Profile*
 - ▶ Abbildung Eingabeparameter ↔ Ressourcenverbrauch



Verwandte Arbeiten

Bestehende Ansätze zu energiebewusster Berechnung auf Mobilgeräten

- ▶ Analyse des Softwareartefakts auf Grundlage zusätzlicher Informationen
 - ▶ CPU-/System-Profile [HB12] [HP13]
 - ▶ Energiemessungen [BM12]
 - ▶ Trainingsdaten [KM13]
- ▶ Modifikation des Artefakts [HP13] [HB12] [PP16] [KM13]
- ▶ Verallgemeinerung auf kompletten Akkuzyklus anstatt auf einzelne Berechnungen [BM12]
- ▶ Komplexe verteilte Anwendungen für vielfältige Zwecke [PP16]

Systemmodell (1)

Mobilgerät

- ▶ nimmt zeitabhängige Leistung $P(t)$ auf
- ▶ Kapazität des Akku als Energieschranke
- ▶ Weitere Schranken situationsbedingt ansetzbar

Betrachtete Anwendung

- ▶ Simulationsberechnung
- ▶ rein lokal – kein Netzwerkzugriff während Berechnung
- ▶ keine weiteren Anwendungen zur Ausführungszeit aktiv

Systemmodell (2)

Anwender

- ▶ Frequenz und Intensität der Nutzung der Anwendung nichtdeterministisch

Berechnung

- ▶ erhält zu Beginn einen *Qualitätsparameter* $p \in \mathbb{N}$
- ▶ Laufzeit und Energieverbrauch steigen abhängig von p monoton an \rightarrow höhere Qualität erfordert mehr Ressourcen

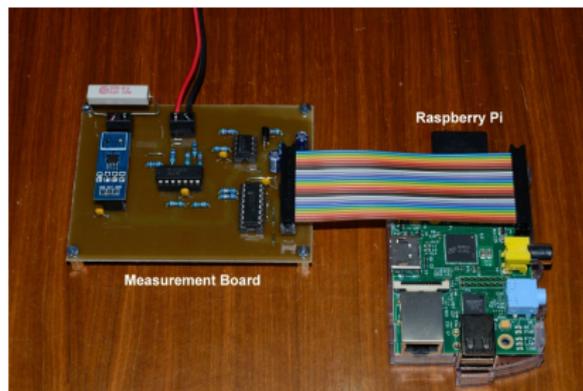
Anforderungen

Verfahren zur Aufstellung eines Vorhersagemodells für den Energieverbrauch einer Simulationsberechnung

- ▶ *Vorhersage*: Eingabe der Berechnung \mapsto Energieverbrauch
- ▶ *Optimierung*: Energieschranke \mapsto optimale Eingabe
- ▶ So wenig Energiemessungen wie möglich
- ▶ Vorhersagen und Optimierungen möglichst leichtgewichtig

Voruntersuchung: Aufbau

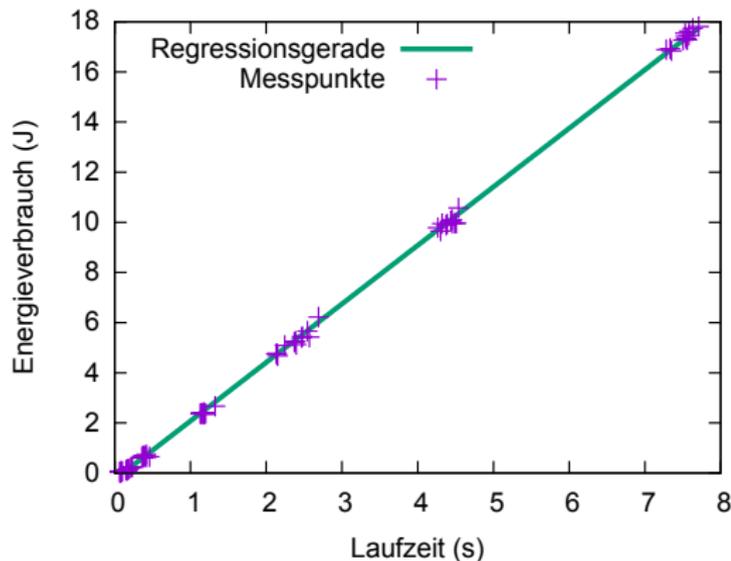
- ▶ Ausführung einer *Diffusion-Advection*-Berechnung auf einem Android-Smartphone (Samsung Galaxy S4 Note)
- ▶ Feingranulare Messung der Spannung und Stromstärke
- ▶ Daten werden gefiltert, digitalisiert und zu Raspberry Pi übertragen [DD16]



Bildquelle: [DD16]

Voruntersuchung: Ergebnisse

- ▶ Mehrere Messungen mit verschiedenen Qualitätsparametern
- ▶ Betrachtete Berechnung: Proportionaler Zusammenhang zwischen Laufzeit und Energie



Entwurf eines initialen Modells

Grundidee:

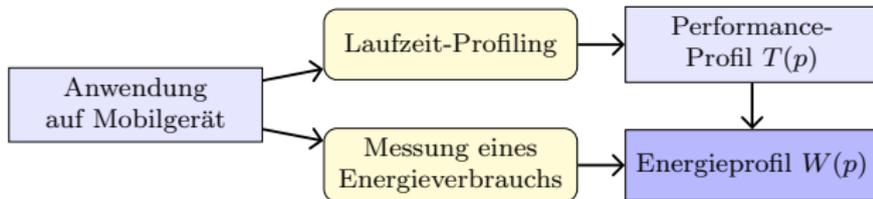
- ▶ Aufstellung des Laufzeit-Performance-Profils $T(p)$ durch einfache Laufzeitmessungen
- ▶ Einzelne Energiemessungen für einen bestimmten Qualitätsparameter
- ▶ Proportionalität von Laufzeit und Energieverbrauch
⇒ Berechnung des Energie-Performance-Profils $W(p)$ aus $T(p)$ und den Energiemessungen

Initiales Modell (1)



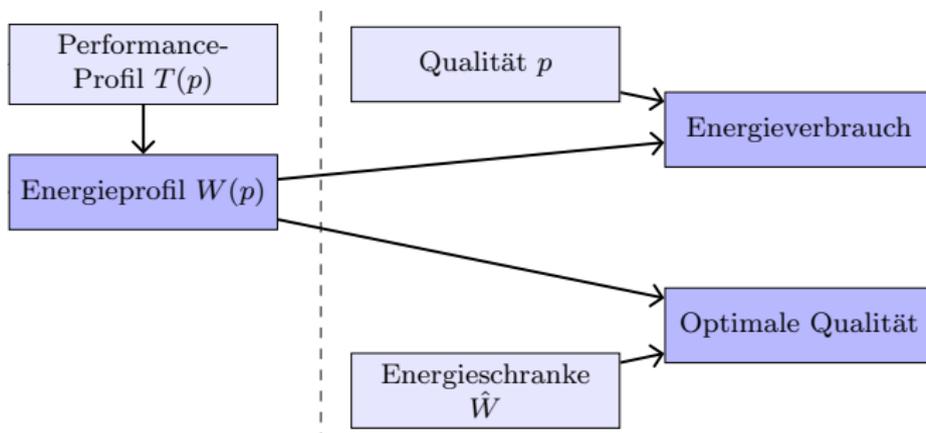
- ▶ Mehrere Laufzeitmessungen mit verschiedenen Qualitätsparametern führen zu Datenpunkten (p_i, t_i) (Streuung: mehrfach messen, Quantile verwenden)
- ▶ Errechnung von $T(p)$ durch polynomielle Regression

Initiales Modell (2)



- ▶ Proportionalität von Laufzeit und Energie: $W(p) = \alpha T(p)$ für einen Koeffizienten α
- ▶ Einzelne Energiemessung (p_i, w) ergibt zusammen mit $T(p_i)$:
 $\alpha = \frac{w}{T(p_i)}$ (Durchschnittsleistung der Berechnung)

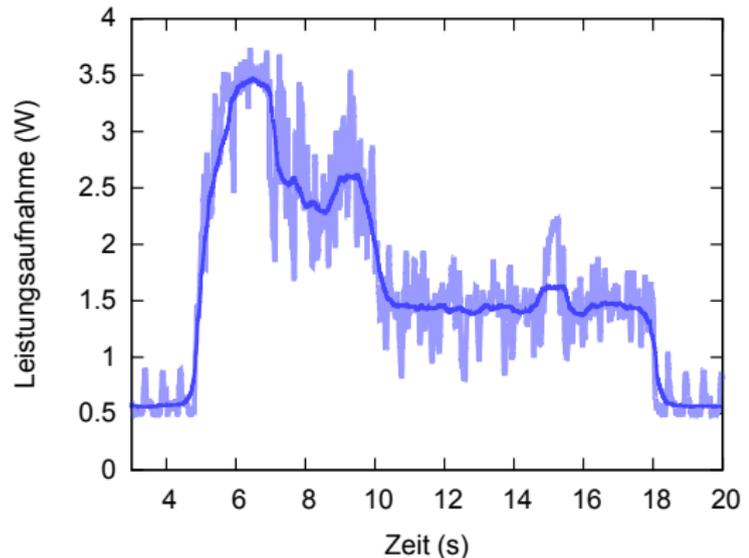
Initiales Modell (3)



- ▶ Vorhersage des Verbrauchs bei Qualität p durch Auswertung $W(p)$
- ▶ Optimaler Parameter für Energieschranke \hat{W} durch Lösung von $W(p) - \hat{W} = 0$

Initiales Modell

- ▶ Sinnvoll für konzeptionell einfache Simulationsberechnungen wie *Diffusion-Advection*
- ▶ Ungenau, falls keine einheitliche Leistungsaufnahme über die gesamte Berechnung hinweg angenommen werden kann

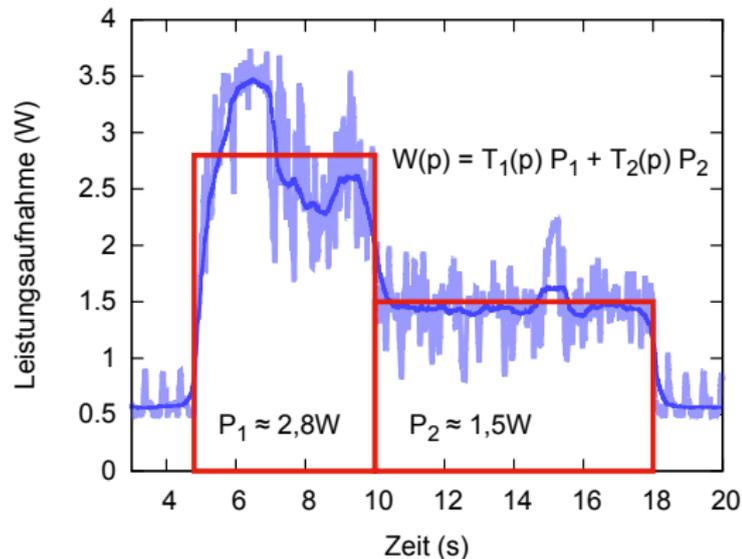


Modellerweiterung

- ▶ Jetzt: Berechnung mit mehreren Phasen, für die *jeweils* eine konstante Leistungsaufnahme angenommen wird
- ▶ Bekannte Phasen i mit Leistungen P_i und Laufzeit-Performance-Profil $T_i(p)$:

Vorhersagemodell

$$W(p) = \sum_{i=1}^n T_i(p) P_i$$



Erweitertes Modell: Ermittlung der Phasen

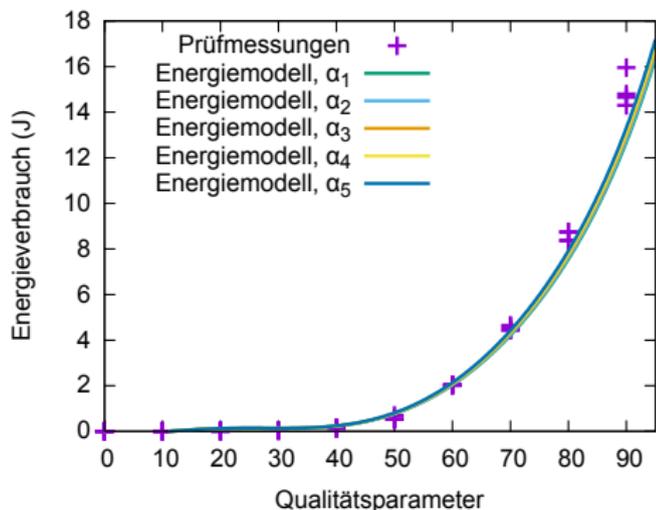
- ▶ *Indikatoren*: Stellen im Code, an denen sich das Leistungsniveau der Berechnung vermutlich ändert
 - ▶ Beginn oder Ende von I/O oder paralleler Berechnung
 - ▶ Neuer semantischer Abschnitt mit anderen Systemaufrufen und/oder abweichender (Speicher-)Effizienz
- ▶ Markierung der Indikatoren im Code mit Zeitnahmen
⇒ Laufzeitmessungen für alle Phasen zugleich
- ▶ Berechnung der jeweiligen durchschnittl. Leistungsaufnahmen auf Grundlage von Energiemessungen und Zeitstempeln
⇒ Identifikation tatsächlicher Phasen über unterschiedliche Leistungswerte

Erweitertes Modell: Zusammenfassung

- ▶ Aufsatz auf initiales Modell
- ▶ Verwendung (Vorhersage und Optimierung) analog
- ▶ Anzahl der Energiemessungen kann weiterhin gering gehalten werden (einzelne Messungen für die Leistungsaufnahmen der vermuteten Phasen)

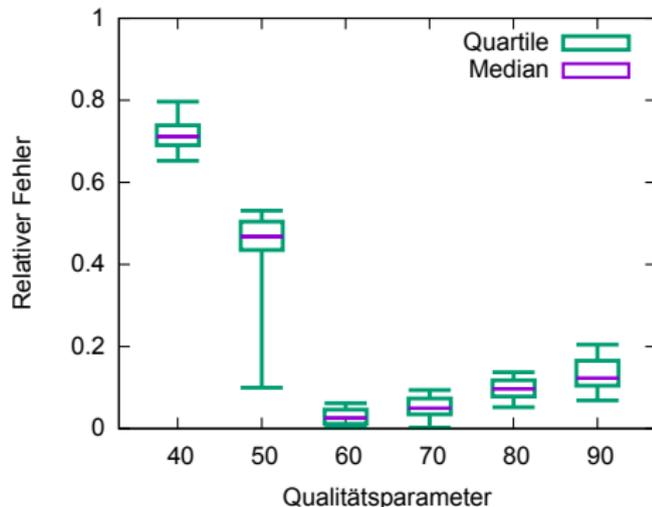
Evaluation: Initiales Modell (1)

- ▶ Auswertung anhand automatisierter Berechnungen des *Diffusion-Advection*-Problems
- ▶ Gegenüberstellung der Modell-Vorhersagen mit separaten Prüfmessungen



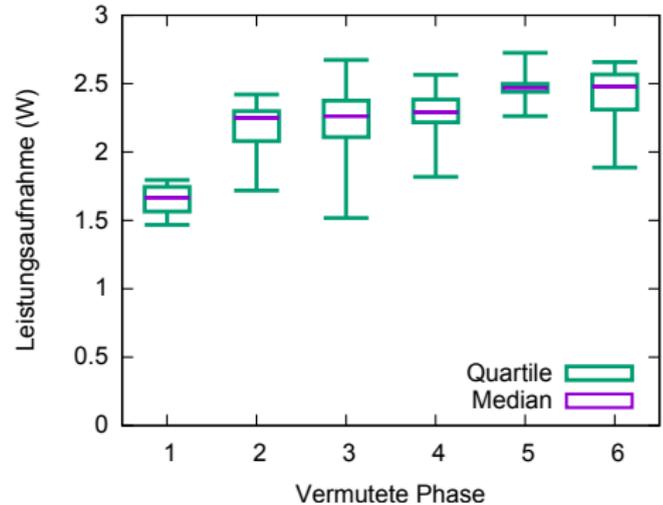
Evaluation: Initiales Modell (2)

- ▶ genaue Vorhersage am Punkt der Energiemessungen ($p = 60$)
- ▶ deutliche Abweichungen für kleine Qualität ($0 \leq p \leq 50$): keine Proportionalität von Laufzeit und Energie in diesem Bereich
- ▶ Abweichungen für sehr hohe Qualität ($p \geq 80$) wegen Methodik der Automatisierung



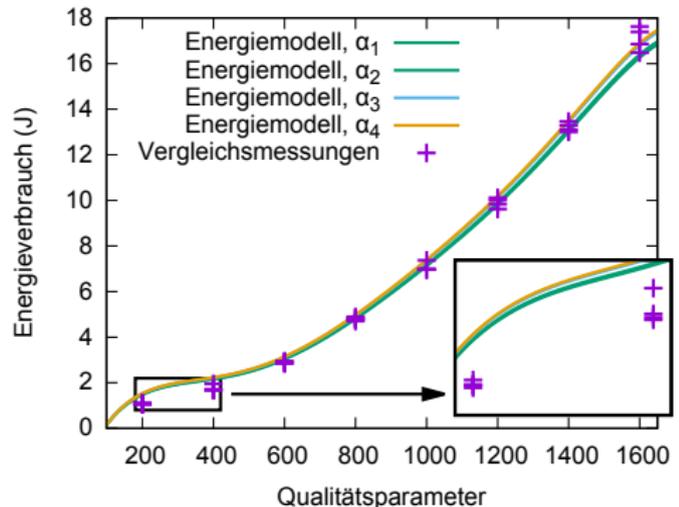
Evaluation: Erweitertes Modell (1)

- ▶ Ausführungen der *Canny-Kantenerkennung* auf dynamisch skaliertem Bild
- ▶ Verschiedene Einlese-, Berechnungs- und Schreibphasen
- ▶ Durchschnittliche Leistungsaufnahmen der vermuteten Phasen \Rightarrow Konsolidierung der beiden Phasen $\{1\}$ und $\{2, 3, 4, 5, 6\}$



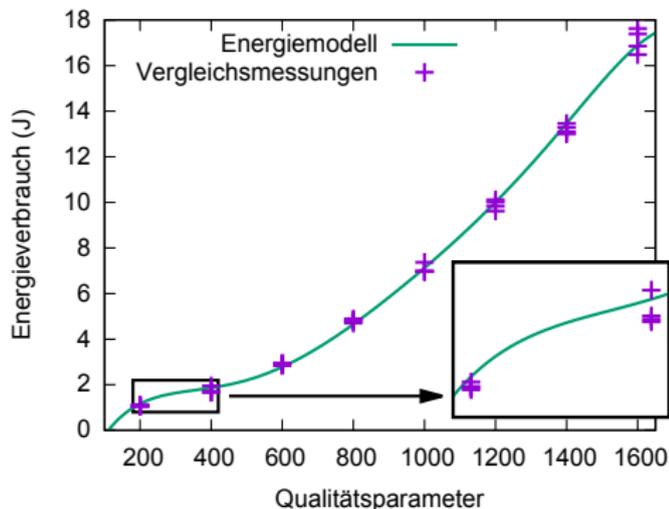
Evaluation: Erweitertes Modell (2)

- ▶ Berechnung des initialen Modells: auffällige Abweichungen im Bereich $200 \leq p \leq 400$
- ▶ Verschiedene Phasen der Berechnung machen sich bemerkbar



Evaluation: Erweitertes Modell (3)

- ▶ Erweitertes Modell:
Berücksichtigung zweier Phasen führt zu exakter Vorhersage im gesamten Definitionsbereich
- ▶ Modellerweiterung sinnvoll und notwendig, sofern relevante Leistungsunterschiede in einer Berechnung bestehen



Fazit

- ▶ Proportionaler Zusammenhang von Laufzeit und Energie bei lokalen Simulationsberechnungen gewinnbringend nutzbar
- ▶ Qualitätsbewusste Algorithmen als vielversprechender Weg für ressourcenschonende Anwendungen

Ausblick auf zukünftige Arbeiten

- ▶ Einbettung in Framework für qualitätsbewusste Anwendungen
- ▶ Energieverbrauch bei Multicore-Architekturen
- ▶ Optimierung durch mehrere Qualitätsparameter

Vielen Dank
für Ihre Aufmerksamkeit

Quellenverzeichnis I



Shlomo Zilberstein (1996)

Using anytime algorithms in intelligent systems

AI magazine 17(3), 73.



Frank Dürr und Christoph Dibak (2016)

rpi-powermeter

<https://github.com/duerrfk/rpi-powermeter>



Wikimedia Commons

Mediated Reality on iPhone

[https://upload.wikimedia.org/wikipedia/commons/7/7b/](https://upload.wikimedia.org/wikipedia/commons/7/7b/MediatedReality_on_iPhone2009_07_13_21_33_39.jpg)

[MediatedReality_on_iPhone2009_07_13_21_33_39.jpg](https://upload.wikimedia.org/wikipedia/commons/7/7b/MediatedReality_on_iPhone2009_07_13_21_33_39.jpg)

Quellenverzeichnis II



Noam Kedem (2012)

Six things to know about smartphone batteries

<http://www.cnet.com/news/six-things-to-know-about-smartphone-batteries/>



James Bornholdt, Todd Mytkowicz, Kathryn McKinley (2012)

The model is not enough: Understanding energy consumption in mobile devices

Power (watts), 1.2 (2012), S.3



Yongin Kwon, Sangmin Lee et al. (2013)

Mantis: automatic performance prediction for smartphone applications

Proceedings of the 2013 USENIX conference on Annual Technical Conference, S.297-308

Quellenverzeichnis III



Parul Pandey und Dario Pompili (2016)

MobiDiC: Exploiting the untapped potential of mobile distributed computing via approximation

2016 IEEE International Conference on Pervasive Computing and Communications (PerCom), S.1-9



Shuai Hao, Ding Li et al. (2012)

Estimating Android applications' CPU energy usage via bytecode profiling

Proceedings of the First International Workshop on Green and Sustainable Software, 2012, S.1-7



Shuai Hao, Ding Li et al. (2013)

Estimating Android applications' CPU energy usage using program analysis

2013 35th International Conference on Software Engineering (ICSE), S.92-101