Energy-Conscious Deployment Optimization for Component-Based Cyber-Foraging Systems

Master’s Thesis – Final Presentation
Supervisor: Grace Lewis (SEI), Christian Stier (KIT)
Reviewer: Prof. Dr. Ralf H. Reussner (1st), Jun.-Prof. Dr.-Ing. Anne Koziolek (2nd)

Sebastian Dieter Krach | 2015-05-22
Cyber-Foraging

- Circumvent restrictions of resource-constraint environments
- Offloading of application components to server infrastructure in one-hop proximity (Cloudlet)
  - Increase performance (e.g. decrease response time)
  - Leverage hardware features not available on mobile platforms
  - Increase battery run-time (through decreasing energy consumption)
Motivation: Speech-Recognition

- Idea: Reduce data transmission by local preprocessing
- Under different network conditions different deployment scenarios optimize the energy consumption
- → No simple energy-optimal partitioning scheme

- Foundations
- Approach
- Profiling
- Optimization
- Evaluation
- Related Work
- Conclusion
Motivation

Problem
• No simple energy-optimal partitioning (w/r/t to energy consumption of the mobile device)

Idea
• Derive energy consumption predictions using model-based simulation
• Automatic architecture optimization

Benefit
• Determine deployment scheme that maximizes battery run-time for a scenario with static conditions at design time

Action
• Leverage existing architecture optimization approach based on model-driven performance analysis; Extend with energy-awareness capabilities w/r/t energy consumption of the mobile device

Source: [Kumar2010]
FOUNDATIONS
Foundations

- Palladio Simulator / PCM
  - Domain-specific language to describe application component-based architectures
  - Simulative prediction of non-functional characteristics

- PerOpteryx: Automated design optimization
  - Initial system candidate (PCM instance)
  - Optimization objectives (quality criteria) + constraints
  - Result: Set of Pareto-optimal candidates
Prediction of Power Consumption

- **System-metric-based power models**: Describe relation between measurable quantitative values characterizing the system state and power consumption
  
e.g. system power consumption linear function of CPU utilization

- **PowerConsumptionAnalyzer** approach leverages system-metric-based power models [Stier2014]
  
  - Extension to Palladio Bench
  
  - Model-based description of power consumption at resource-level granularity
  
  - Predict power consumption by calculating power models based on simulatively determined metrics

![Graph showing linear relationship between CPU utilization and power consumption](image)
THE APPROACH
Overview Approach

Application Architecture Description

Generate Candidates

Analyze Candidates

Rank Candidates

Architecture Optimization

Power Model Specifications

Automated Device Profiling

Palladio Simulation

Network Demand Completions

Generate System Metric Predictions

Calculate Power Cons. Predictions

Calculate Energy Cons.

Aggregate result for candidate

Power Model Parameters

PCA

unchanged adapted new

Motivation Foundations Approach Profiling Optimization Evaluation Related Work Conclusion
PROFILING
Automated Device Power Profiling

- Fully automated process to cope with heterogeneous hardware landscape and large number of devices

- **Approach:**
  - Induce specific system metric levels on real device using micro-benchmarks and observe power consumption behavior
  - Derive power model parameters using regression analysis

---

Experiment Description

- Specifies:
  - System-metric targets
  - Workload (micro-benchmarks)

Experiment Execution & Monitoring

Power traces

Power Model Specification

Model Parameter Extraction

Power model parameters ready for PCA execution

Extension of PCA models with expression language
OPTIMIZATION

Application Architecture Description

Architecture Optimization

Generate Candidates

Analyze Candidates

Rank Candidates

Generate System Metric Predictions

Network Demand Completions

Calculate Power Cons. Predictions

Calculate Energy Cons.

Aggregate result for candidate

Power Model Specifications

Automated Device Profiling

Power Model Parameters

Palladio Simulation

PCA

OPTIMIZATION

Sebastian Dieter Krach - Energy-Conscious Deployment Optimization for Component-Based Cyber-Foraging Systems

Software Design and Quality Group
Institute for Program Structures and Data Organization
Energy-Conscious Architecture Evaluation

- Quality dimension to optimize: energy consumption
- Result of calculating cumulative energy consumption for usage scenario simulation:

  ![Graph showing cumulative energy consumption over time]

  Predicted trace of cumulated amount of energy consumed up to point in time

- Two possibilities to rank architecture candidates w/r/t energy consumption
  - Absolute energy consumption at end of simulation
  - Average consumption per service call (chosen for single user scenario)
EVALUATION
Evaluation – Power Models – CPU

- Resulting power model for Galaxy Nexus mobile phone (2x 1.2GHz)
- Experiment conducted for CPU Utilization 0% - 100% (in steps of 20%)
- Underlying regression function:

\[
\text{Power} \sim \text{Min}_{\text{Power}} + \text{Util}^d \times (\text{Max}_{\text{Power}} - \text{Min}_{\text{Power}})
\]

- Evaluation
  - Residual Sum of Squares: 0.001
  - Standard Deviation: 0.015W
Evaluation – Power Models – WiFi

\[
Power \sim c_1 \cdot e^{\frac{1}{\text{Size}}} + c_2 \cdot e^{\frac{1}{\text{Rate}}} + c_3 \cdot \text{Size}^{c_4} \cdot \text{Rate}^{c_5}
\]

Significant for low values
- Correction factor for rapid increase

Approximation of metric based model presented by [Zhang2010] and [Mittal2012] which differentiates between two different power states (low and high) with linear increase for high state.

Assumptions due to approximation:
- No distinction of sending and receiving
- Reduce amount of input variables to packet rate and packet size
- Model can be expressed as continuous function

Evaluation
- Residual Sum of Squares: 0.071
- Standard Deviation: 0.045W
Speech is a cyber-foraging speech recognition application for the Android platform leveraging CMU Sphinx on the cloudlet.

CMU Sphinx 4 is a component-based framework for speech recognition developed entirely in Java.

Major parts: FrontEnd, Decoder, Linguist (each consisting of multiple components)
- Frontend: 11 Stage pipeline filtering input, extracting relevant information
- Linguist: Handling of recognition relevant (language-) models
- Decoder: Matching of extracted information against models

Due to memory constraints Frontend components only viable candidates for local execution

→ 11 different deployment alternatives (last stage of pipeline also depends on memory-intensive models)
Absolute Consumption Candidates

Evaluation of the 11 candidates for the different transmission speeds unlimited (approx. 2.2MByte/s), 128KByte/s and 64KByte/s.

64KByte/s Scenario

2.2MByte/s Scenario

Major sources of inaccuracy

- Significant underestimation of WiFi consumption for higher data rates (approx. 25%)
- Significant prediction errors w.r.t response time particularly for high data rates
Relative Candidate Ordering

Evaluation of the 11 candidates for the different transmission speeds unlimited (approx. 2.2MByte/s), 128KByte/s and 64KByte/s.

- Worst two candidates correctly identified for all scenarios
- (Almost) optimal candidate identified correctly
  - Two scenarios correctly (unlimited and 128KByte/s)
  - 64KByte/s scenario: best and 2\textsuperscript{nd} best alternative reversed
    - Measured difference: 17.9\(\mu\)Wh (absolute), 1.44\% (relative)
    - Simulated difference: 1.42\(\mu\)Wh (absolute), 0.14\% (relative)
Automated Optimization using PerOpteryx

- PerOpteryx proved useful to automate and collect information on architectural candidates.
- Evolutionary design space exploring requires evaluating a large number of candidates.
  - 10 components with choice of allocation $\rightarrow$ 1024 deployment candidates.
- Results of automated optimization:
  - Execution Time: approx. 2h (did not finish).
  - Evaluated 11 generations á 10 candidates, 65 distinct candidates.
  - 2 candidates identified with response time < 120s.
  - Not effective without further extension.
Related Work

- Prediction of energy consumption for mobile devices
  - Benchmarking & Profiling using metric-based energy models [Zhang2010], [Jung2012], [Mittal2012]
  - Energy consumption prediction using Palladio [Willnecker2014], particularly w/r/t cyber-foraging [Rosenthal2014]

- Energy-conscious application offloading
  - Fine-grained mobile application profiling [Saarinen2011]
  - Dynamic offloading frameworks
e.g. MAUI [Cuervo2010], CloudClone [Chun2011]
Energy consumption analysis for deployment candidates based on model-driven architecture performance analysis viable

- Automated device profiling approach determines power model parameters leveraging micro-benchmark execution and regression analyses
- Energy consumption determined based on specified CPU demand and completed network demand
- Deployment scheme presenting best energy consumption identified or comparable alternative with little relative error

Automated optimization of component allocation for energy-consumption not effective with current approach

- Large number of candidates w/ small number of viable alternatives
- Time consuming evaluation of unpractical candidates
Future Work

- Improve WiFi consumption models
  - Take additional metrics into account (e.g. channel usage)
  - Distinguish sending and receiving
  - Realize more elaborate power model

- Increase automated optimization efficiency
  - Decrease design space (leveraging annotations of architectural patterns)
  - Generate alternative deployment candidates more systematically