**Improving Energy-Bound Application Effectiveness through Redundancy and Approximation**

**Liu Liu**, Timothy Yong, Jonathan Risinger, Sibren Isaacman, Abhishek Bhattacharjee, and Ulrich Kremer

Department of Computer Science
1 Rutgers University / 2 Loyola University

---

**Target Applications**

1. Composed of sets of collaborating services (producer-consumer, resource sharing)
2. Each individual service may provide discrete levels of quality vs. energy (resource) tradeoffs.
3. Nothing is free! Want best overall application outcome within a given energy budget.

**RSDG Glider**

Multiple sensors and dependencies between services

**RSDG NavApp**

Navigation application on Android that balances multiple services to meet the energy requirement.

---

**Design Foundation**

3 types of redundancy:
- approximation
- implementation
- replication

**RSDG service selection problem**

- Services with mission values are considered (user) critical
- For each critical service, select a single service level and implementation; the same for each service the selected service depends on (transitive dependent set)
- Sum of mission values of critical services is maximal under given energy constraints; energy is minimized for this maximal mission value

**System Overview**

**Application Analysis**

Energy Cost vs. Overhead vs. Benefit

- Solution is non-intuitive.
- Given sufficient energy, RSDG’s solution saved ~50.4% Joules compared to the heuristic.
- Saved more than 45% energy by introducing less than 2.5% overhead.

**Conclusion**

A new framework that allows the programmer to easily describe an adaptive system that balances the services configurations across the entire system to produce highest productivity with lowest cost within user-specific budget.

---

**Related Work:**

- Dynamic Knobs for Responsive Power-Aware Computing [ASPLOS-11]:
  - Improving performance and power savings by dynamically adjusting the parameters to "trade off the accuracy of computation in return for reductions in resources."
  - "problem: accuracy could not be a measurement for quality in most of the time. (Users’ preferences/priority) got ignored"

- JouleGuard: Energy Guarantees for Approximate Applications [SOSP-15]:
  - Maximizing accuracy within energy budget by "dividing the problem into two subproblems, configuring hardware system to be energy efficient and dynamically adjusting program to manage performance."
  - "problem: ignored the dependencies.(smaller search space)"

- CareDroid: Adaptation Framework for Android Context-Aware Applications [MobiCom-15]:
  - Improving efficiency of context-aware applications by "monitoring the context of the physical environment and intercepts calls to sensitive methods."
  - "problem: not being able to handle dependencies other than physical context.(single layer RSDG)"

**Future Work:**

- more application examples
- better tools that support easier construction for RSDG
- better tools for users to explore the tradeoff spaces
- better execution model
- optimization on 0-1 problem’s constraints
- reducing the RSDG overhead

---

**Application Examples**

1. 10 x 10 meters
2. 5 - 100 soda cans
3. Randomly generated locations
4. traveling cost: 50J/m

**Solution for energy = 40**

- Mission value: 11
- Example mission values: 7

**Solution for energy = 12**

- Mission value: 5
- Example mission values: 5

**Productivity Profile**

Optimal solution balances service qualities across the entire application: Solution seems sometimes non-intuitive.

Solution is NP-complete: Proof: Reduction from 3SAT. Use 0-1 integer programming formulation and gurobi, cplex, or lp_solve

Simulations can provide feedback during tradeoff space exploration via the Productivity Profile: