Fakultät Informatik Institut für Software- und Multimediatechnik, Lehrstuhl Softwaretechnologie

The journey has just begun

RACR - MQUAT

René Schöne

Dresden, June 22, 2015

Agenda

1 The past

2 The present

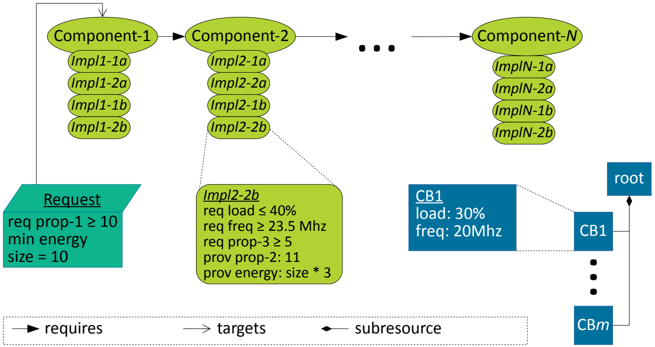
3 The future

Where we started

- MQuAT concept [GKP+14]
  - Self-adaptive system optimizing for multiple qualities
  - Component-based design for both hardware and software
  - Quality contracts capturing requirements and guarantees of components
- THEATRE [GWC+10] as a Java-based implementation of MQuAT
  - Knowledge represented using EMF-(Meta)Models
  - Optimization problem solved by transformation to ILP
  - Designed for distributed operation (see HAECubie) using Master-Slave-Pattern [Sah96]

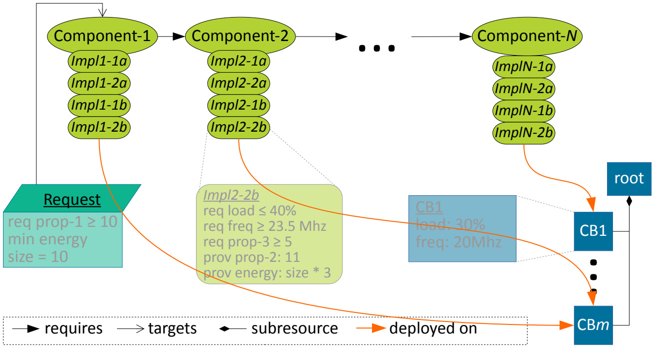
The optimization problem

Mapping *n* components on *m* resources



The optimization problem solved

Mapping *n* components on *m* resources



What was the problem

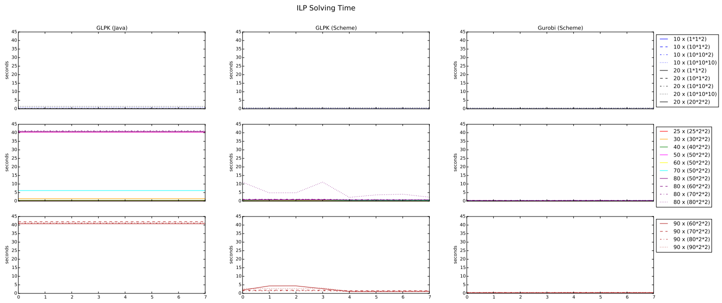
- ILP-Solution branded “unusable” for bigger systems
  - current measurements disprove this, see next slides
  - ILP still generated from scratch for each request
  - measurements on Cubieboards still to be done for new approach
- EMF-Model (element)s somewhat ambiguous or superfluous
  - component requirement possible on both, component- and mode-level
  - structure and variant model contain similar information
  - general approach of structural model not easy to use (especially for ILP-Generation)

Measurement on solving times

- Measurement on solving times
  - Context: System Generator to generate ILP for increasing size of systems
  - 23 different sizes of systems, 40sec timeout
  - Format of ILP adopted for glpk via own Python script
- Result for Java-based solution
  - Timeouts: glpk 10/23, lpsolve 8/23
- Result for RACR-based solution (enhanced ILP)
  - All but one systems solved within 5sec (outlier 12sec)

Measurement on solving times (2)

- Even better performance using Gurobi (commercial solver)



Agenda

1 The past

2 The present

3 The future

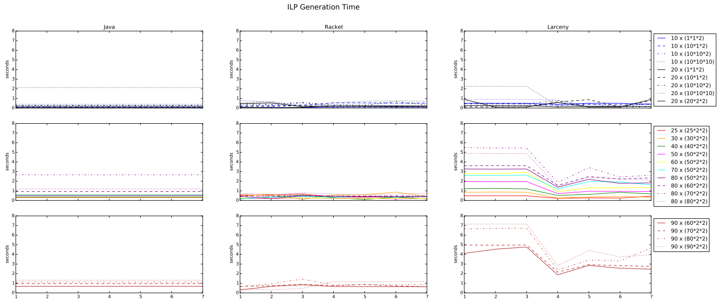
How we want to achieve scalability

- Use RACR [Bür12]
  - **R**eference **A**ttribute Grammer **C**ontrolled **R**ewriting
  - Specify knowledge as an ASG[1] whose structure is defined by a RAG[2]
  - RAG is a combination of structural and variant model, avoiding duplicate information
  - Analyses run on ASG now run inherently **incremental** and are defined **declarative**

[1]Abstract Syntax Graph, i.e. an Abstract Syntax Tree with references
[2]Reference Attribute Grammar

Current state, compared to existing work

- Not quite there yet:



TECHNISCHE UNIVERSITÄT DRESDEN

General Facts

- `https://bitbucket.org/rschoene/racr-mquat`
- IPython Notebook[3] used for measurement plots
- Main language: Scheme
  - implementations used: Racket[4], Larceny[5]
- Measurement and test scripts: Python

[3]http://nbviewer.ipython.org/urls/bitbucket.org/rschoene/racr-mquat/raw/master/ilp-measurement.ipynb
[4]http://racket-lang.org/
[5]http://www.larcenists.org/

Code Facts

Using cloc[6]

Language	files	blank	comment	code
Scheme	15	224	316	2287
Python	8	81	44	509
Bourne Again Shell	1	0	0	2
SUM:	24	305	360	2798

[6] http://cloc.sourceforge.net/

Current pitfalls

- Different input formats accepted by lp_solve and glpk
  - Transformation (mostly syntactical) needed
  - Still, different solution computed (GLPK occasionally ignore binary variables, value e.g. 0.348485)
- Slow running Larceny
  - Unexpected as Larceny compiles to machine code
- Caching not fully exploited
  - Some constraints still unnecessarily recomputed

Agenda

1 The past

2 The present

3 The future

Where we should go next

- Do not transform to ILP
  - Implement an heuristic similar to RACRtune demo[7] of Daniel Langner and Johannes Mey
- Apply static analysis where appropriate, e.g.
  - Abstract Interpretation [CC77, Ros90] to estimate energy consumption [JM06, RMM03]
  - Describe decisions [Dan15]
  - Find configurations, which can never be used
  - Unify constraints (in contracts) of modes
- Extend AG
  - Describe multiple systems and their interaction, e.g. [WSG+13]
  - Include behavior model for more fine grained description

[7]Shown at HAEC review and OUTPUT'15

References I

Christoff Bürger.
Racr: A scheme library for reference attribute grammar controlled rewriting.
2012.

Patrick Cousot and Radhia Cousot.
Abstract interpretation.
In *Proceedings of the 4th ACM SIGACT-SIGPLAN symposium on Principles of programming languages - POPL '77*, pages 238–252, New York, New York, USA, January 1977. ACM Press.

Antonia Danylenko.
Decision Algebra: A General Approach to Learning and Using Classifiers.
PhD thesis, Linnaeus University, Växjö, Sweden, 2015.

References II

Sebastian Götz, Thomas Kühn, Christian Piechnick, Georg Püschel, and Uwe Aßmann.
A models@ run. time approach for multi-objective self-optimizing software.
In *Adaptive and Intelligent Systems*, pages 100–109. Springer, 2014.

Sebastian Götz, Claas Wilke, Sebastian Cech, Johannes Waltsgott, Ronny Fritzsche, Jan Reimann, and Matthias Schmidt.
THEATRE Resource Manager Interface Specification v. 1.0.
Techn. Univ., Fakultät Informatik, 2010.

R. Jayaseelan and T. Mitra.
Estimating the Worst-Case Energy Consumption of Embedded Software.
In *12th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'06)*, pages 81–90. IEEE, 2006.

References III

Cosmin Rusu, Rami Melhem, and Daniel Mossé.
Maximizing rewards for real-time applications with energy constraints.
ACM Transactions on Embedded Computing Systems, 2(4):537–559, November 2003.

Mads Rosendahl.
Abstract interpretation using attribute grammars.
pages 143–156, September 1990.

Sartaj Sahni.
Scheduling master-slave multiprocessor systems.
IEEE transactions on Computers, (10):1195–1199, 1996.

References IV

Danny Weyns, Bradley Schmerl, Vincenzo Grassi, Sam Malek, Raffaela Mirandola, Christian Prehofer, Jochen Wuttke, Jesper Andersson, Holger Giese, and Karl M. Göschka.

On patterns for decentralized control in self-adaptive systems.

In *Software Engineering for Self-Adaptive Systems II*, volume 7475 LNCS, pages 76–107. Springer, 2013.