Certified Complexity (CerCo)
An Overview of the Past, Present and Future of CerCo

Claudio Sacerdoti Coen

May 16, 2013

SEVENTH FRAMEWORK PROGRAMME

Outline

Outline

Problem Statement

Functional properties are checked on high level source code
Non functional properties are checked on object code

1. The two analyses are NOT orthogonal
   Misleading results, duplicated work
2. Complex optimizations (e.g. parallelization) ⇒
   user contribution to control flow analysis very hard
   Loss of precision, wrong results, simple optimizations only
3. Performance improvement is hard
4. Parametric cost analysis very hard
   Loss of precision in compositionality
5. Cost analysis not available in early development
   Tackling the problems later is more costly
6. Fully automatic (+) but boolean answers only (-)
   No incremental refinement and tradeoffs

SEVENTH FRAMEWORK PROGRAMME

Problem Statement

Moreover

The most important thread of theoretical computer science is raising the level of abstraction, from the machine to the human one

State of the art:

Analysis of non functional properties either is bound to the machine level or it abstracts away too much by talking only of complexity classes

As a result, real time embedded systems still coded in a constrained way and at low level to ensure non functional guarantees.

CerCo's vision and approach

Vision:

Reconcile functional and non-functional analysis

Perform both on the source code where more techniques can be employed

Approach:

Induce a precise (thus parametric and non uniform) cost model on the source code

Compilers must be modified to transfer cost models from the object to the source code

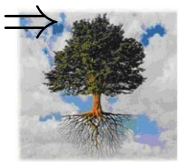
Relevance of the Problem Statement

Is the problem still valid? Yes
Do you still believe in its relevance? Definitely yes

Idealized vision of ICT from the theoretical CS community:

Most programs will be (partially)
specified and certified

1. Moving beyond constrained domains ⇒
   parametric and interactive analysis a necessity
2. A lot of recent work (e.g. POPL papers) on static prediction of non functional properties on idealized cost models ⇒
   realistic, non uniform cost models a necessity



Relevance of the Problem Statement

Is the problem still valid?
Do you still believe in its relevance?

Software engineering perspective:

System development should be made cheaper and faster and deliver safer and most performant products

In favour of its relevance:

1. Resource analysis in early development phases is crucial
2. Massive parallelization for multicores $\Rightarrow$ disruption of the control flow (no interactivity, harder code improvements)

Relevance of the Problem Statement

Is the problem still valid?
Do you still believe in its relevance?

Software engineering perspective:

System development should be made cheaper and faster and deliver safer and most performant products

Against its relevance:

1. Diminishing returns: for embedded systems code the major source of imprecision is becoming the hardware model
2. Static analysis of current hardware is getting hopeless (e.g. multicores, unavailable precise hardware models)

Relevance of the Labelling Approach

Is the approach still valid? Yes
Do you still believe in its relevance? From our experience, yes

Idealized vision of ICT from the theoretical CS community:

Most programs will be (partially)
specified and certified

1. The approach provides a precise cost model that replaces the idealized ones
2. It complements, leverages and enhances all other existing approaches (but requires parametricity of the costs)

Relevance of the Approach

Is the approach still valid?
Do you still believe in its relevance?

Software engineering perspective:

System development should be made cheaper and faster and deliver safer and most performant products

In favour:

1. Supports early analysis and progressive refinement

Against:

1. Ad-hoc compiler and IDE, non fully automatic
2. Does not leverage existing SE techniques and tools
3. No major improvement expected for constrained code

Alternative Approach

Is the approach still valid? Do you still believe in its relevance?

Reverse CerCo:

The compiler transfers functional properties
from the source to the object code

Non functional analysis is performed on the object code reusing the functional information

+ Leverages more existing WCET techniques (not tools)
+ Less sensitive to modern hardware issues
- Does not support early analysis and progressive refinement
- Ad-hoc compiler and IDE, non fully automatic

Conclusions

Is the long term vision described in the proposal still valid?
Yes, but there are major challenges

1. Driving vision shared across the formal methods community (cfr. EMBounded)
2. The approach does not target the most relevant issue of current cost analysis, namely the intrinsic unpredictability of modern complex architectures, and is affected by it
3. Unresolved rising challenge faced by all working in cost analysis (cfr. PROARTIS, PARMERASA, T-CREST, REMS, TACLe)

Outline

Contributions

How far the concrete project achiev. contribute to the l.t. vision?

Sub-problems posed by the CerCo approach:

1. Compute manageable parametric cost models on the object c.
   No contribution: we deliberately focused on 2 and 3 as 1 requires expertise and resources that were not part of the consortium
2. Transfer the cost models in a certified way without introducing approximations
   Major contribution: a technique, a proof methodology, evidence that it scales to more complex scenarios, a certified proof
3. Exploit the cost models on the source level
   Initial contribution: integration of the compiler in Frama-C, evidence that cost invariant inference is feasible and that generic automatic techniques can handle the proof obligations

Self assessment

How successful is the CERCO project in your own view?
Fully successful in achieving the original goals and we achieved more than what was planned in dealing with optimizations that alter the control flow.

We proposed from scratch a novel approach to time analysis.

Our goals for the project:

- To be able to transfer the cost models in a certified way without introducing approximations, possibly restricting the compilation/optimization techniques
- To understand if the cost models obtained are simple enough to be exploitable

Self assessment

How successful is the CERCO project in your own view?

We did not have as an initial goal for the project (and no expertise in the consortium) for:

- being able to compute manageable parametric cost models on the object code for advanced hardware architectures

The proposed goals were already challenging alone and critical for the long term vision.

Self assessment

How successful is the CERCO project in your own view?

We believe to have been moderately successful in identifying extensions of the basic approach that have the potential of contributing solutions to the forthcoming problems.

We have been less successful in convincing the *engineering* community of the potentiality of our approach

- targeting the non critical issues (multicores, complex hardware, lack of clock precise models)
- no focus on early development
- no early evidence of success in performing as state of the art
- certification not appealing (and workplan focused on that)

SEVENTH FRAMEWORK PROGRAMME

Next steps

What are the next steps still needed to move forward?

1. Involvement of experts of WCET/clock precise hw models
2. Inference of compositional, manageable dependent cost models for modern hardware
   - Introduce approximations, to reduce the size of the model
   - May require combination of CerCo + reverse CerCo
3. Improve time analysis on the source code; introduce tradeoff between precision and termination/speed of the analysis
4. Focus on time analysis for early development phases
5. Applications of labelling to other non functional properties (e.g. information flow security)
6. Integration with IDEs used in development
7. Tracking/integration of proposals to perform static analysis for multicores and systems-on-chip

SEVENTH FRAMEWORK PROGRAMME

Effect on future ICT

How future ICT will be affected by CERCO results?

Early development phases (medium term)

CerCo applies to early development phases and supports progressive refinement.

General purpose code (long-medium term)

Major ICT failures (4 in Italy alone in last 2 years) for non real time software that did not scale to production.

Non functional analysis harder for general purpose code (e.g. dynamic data structures).
Non parametric WCET techniques insufficient, CerCo adaptable.

Effect on future ICT

How future ICT will be affected by CERCO results?

High level languages everywhere (very long term)

The history of computer science is a journey from lower to higher levels of abstractions.

The industrial world is well behind the academic community, but slowly and selectively catching up.

We do not see a real advancement in ICT unless it becomes possible to easily reason on non functional properties for generic code.

CerCo will allow such an advancement iff future hardware will still allow for static analysis at all.

Exploitation of results

Any concrete idea for exploitation of the project results?

In the medium term:

CerCo's prototype as a tool for time analysis
in early development phases

- Reduced dependency on hardware models
- Emphasis on tools and techniques for invariant generation and proving on the source level
- Priority: graceful degradation of precision to always grant answers in a reasonable time
- The topic of one work package of TACLe

SEVENTH FRAMEWORK PROGRAMME

CerCo as a Collaborative Project

Which has been the advantage (if any) to run CERCO research in the framework of a "Collaborative Project"?

Honest answer:

1. The consortium put together in a focused collaboration expertises that were not found in any single site
2. The manpower required by the project corresponded to a level of funding that is no longer provided by national or bi-lateral projects

Which funding scheme (if not FET-Open) is more appropriate for basic research that requires a great implementation effort to deliver a unified end product?

SEVENTH FRAMEWORK PROGRAMME