ARTIST 2
Network of Excellence

IST-004527 ARTIST2:
Embedded Systems Design

Activity Progress Report for Year 1

JPIA-Platform:
Compilers Platform

Cluster:
Compilers and Timing Analysis

Activity Leader:
Rainer Leupers

The objective is to provide world-class code-synthesis and compiler tools for the generation of efficient machine code. Goals of the cluster include the integration of existing compiler-generation approaches allowing compilers for new architectures to be built quickly, efficiently and reliably.

One goal of the compilers sub-cluster is to achieve a tighter integration of European R&D activities by building on a carefully chosen industrial re-targetable compiler development platform that ensures interoperability.

The CoSy compiler platform provided by ACE is a state-of-the-art software system on which the common activities will build. This will reinforce Europe's leading position in the area of compilers for embedded processors.

Also, the partners aim at making existing advanced optimization algorithms available to designers of embedded systems. Finally, advanced high-level source-to-source transformations will be made available to users.

Table of Contents

1. Introduction

1.1 Activity Leader

Team Leader: Rainer Leupers (RWTH Aachen)
Areas of his team's expertise: compilers.

1.2 Policy Objective

The objective is to provide world-class code-synthesis and compiler tools for the generation of efficient machine code. Goals of the cluster include the integration of existing compiler-generation approaches allowing compilers for new architectures to be built quickly, efficiently and reliably.

One goal of the compilers sub-cluster is to achieve a tighter integration of European R&D activities by building on a carefully chosen industrial re-targetable compiler development platform that ensures interoperability.

The CoSy compiler platform provided by ACE is a state-of-the-art software system on which the common activities will build. This will reinforce Europe's leading position in the area of compilers for embedded processors.

Also, the partners aim at making existing advanced optimization algorithms available to designers of embedded systems. Finally, advanced high-level source-to-source transformations will be made available to users.

1.3 Industrial Sectors

Mainly: embedded software for low power and real-time applications (consumer, automotive, telecom).

Specifically: audio processing, video processing and data streaming applications in the TV, Set Top box, DVD player and recorder, mobile, base stations, printer and disk drive markets.

2. Overview of the Activity

2.1 Artist Participants and roles

Team Leader: Reinhard Wilhelm (Saarland Univ.)
Areas of his team's expertise: co-leader, timing analysis.

Team Leader: Rainer Leupers (RWTH Aachen)
Areas of his team's expertise: co-leader, compilers.

Team Leader: Christian Bertin (STMicroelectronics)
Areas of his team's expertise: driver applications.

Team Leader: Peter Marwedel (Dortmund Univ.)
Areas of his team's expertise: low power compilation.

2.2 Affiliated partners and Roles

Team Leader: ans van Someren(ACE)
Areas of his team's expertise: compilation techniques.

Team Leader: Christian Ferdinand (AbsInt)
Areas of his team's expertise: Program-Analysis Tool.

Team Leader: Andreas Krall (Tu Vienna)
Areas of his team's expertise: Program-Analysis Tool.

2.3 Starting date, and expected ending date

September 1st, 2004 until the European compiler platform has been constructed.

2.4 Baseline

Members of this activity have comprehensive experiences in different areas of compilers for embedded systems. Contacts and cooperation are already partially in place. The CoSy compiler platform provided by ACE is a state-of-the-art software system on which the common activities will build.

2.5 Technical Description

ACE will provide the CoSy platform as well as training and support. Other core and affiliated partners will focus on partially retargeting their existing compiler technology towards that common platform in order to ensure interoperability of existing and newly developed techniques. Existing advanced optimization algorithms will be made available to the partners.

3. Activity Progress Report

[Reviewer comment: The document must be revised to incorporate the outcomes shown in the presentation, and to better separate the foreground outcomes from the background outcomes.]

Overview and classification of outcomes

Broadly speaking, many of the results described below can be classified as **background outcomes**. This is due to the low funding level in the ARTIST2 NoE, which does not allow to cover significant research staff costs. For instance, the *Aachen-ACE cooperation* on the LISATek Compiler Designer has been largely funded by a third party, the *STM-ACE cooperation* has been existing before and goes significantly beyond the ARTIST2 framework, and a large fraction of manpower invested by the *university partners* are generally funded by other sources. Hence, the compilers platform activity is actually to be seen as a pure integration activity.

However, there are major **foreground outcomes** as well. First, ARTIST2 enabled (and continues to do so) *regular cluster meetings*, during which the leading European R&D teams in that area convened and exchanged information about their current projects and research interests. The significant fraction of industrial participants in the cluster ensures that universities are supplied with valuable information about real-life problems, while industry receives an outlook on emerging technologies. Another key foreground result is the agreement on the *CoSy compiler platform* as the primary platform (see section 4), based on intensive support by partner ACE, which also led to the use of CoSy for teaching activities (e.g. ALARI). The required initial broad review of existing platform options would most likely not have happened without ARTIST2. As an indirect foreground result, this common platform enabled further new sub-projects in the tightly related "*architecture aware compilation*" activity in the same cluster. Finally, ARTIST2 funding enabled several *bilateral and mini-cluster meetings* during which planning for concrete cooperations were carried out that finally led (or may lead on the future) to background outcomes. An example is the planning done on the CoSy-PAG integration between partners Absint and ACE, in which case more external funding still needs to be acquired, though, to achieve another significant background outcome.

3.1 Work achieved in the first 6 months

In the first 6 months, the partners evaluated different options for compiler platforms, e.g. gcc CoSy and SUIF. CoSy has been chosen as a primary platform. Furthermore, integration goals have been defined, and mini-clusters focusing on specific topics have been formed. Details are described in section 4.1 and 4.2. Preliminary results achieved at the mini-cluster level are described in section 3.2.

3.2 Work achieved in months 6-12

3.2.1 Cooperation Aachen-ACE:

RWTH Aachen and ACE have intensified their existing cooperation w.r.t. coupling the LISATek tool suite and the CoSy platform. This has led to different extensions of the Coware LISATek Compiler Designer tool.

3.2.2 Cooperation Absint-TU Vienna:

Improving the productivity of developing embedded software requires that high-level abstractions can be used and reused. It is paramount for the next decade that these abstractions can be optimized such that they can be used in embedded systems without a significant performance impact. One of the major problems in optimizing high-level abstractions is aliasing, introduced by operations on references and pointers in programs. Our effort focuses on building and integrating existing infrastructures such that we can offer platforms for evaluating the impact of aliasing algorithms on today's languages performance that are used for embedded software.

Our goal for Year 1 was the integration of the Program Analysis Generator (PAG) of AbsInt in several platforms to share the same analysis in different infrastructures and leverage existing optimizations for evaluation. We created a tool, the PAG Interface Generator (PIG), to automate the PAG integration. The two different infrastructures which served as applications for the PAG integration by using PIG were ROSE and OCE/xDSPcore. ROSE is a source-to-source infrastructure that supports C++ (and Fortran in near future). The OCE/xDSPcore is the ATAIR open compiler infrastructure with a backend for digital signal processors.

The Input to PIG is a rule specification and PAG's syn-file. The PAG syn-file is processed to extract all information about the structure of the AST and the PIG rules are used to match with the extracted information. As output all required AST interface code is generated. Additionally we had to present the control flow information such that PAG can operate on the respective data of an infrastructure. This completed the integration of PAG.

Our goal for Y1 was the creation of a tool, PIG, to automate most aspects of a PAG integration and prove its usefulness by using it for integrating PAG in ROSE and OCE/xDSPcore. We have achieved both goals such that we can demonstrate the result by having a constant propagation analysis (as test) running in both environments.

Future work includes adding analysis specifications for different aliasing algorithms (using PAG) and creating components for post-computing the gathered information to prepare the data such that subsequent transformations can utilize the new program information.

3.2.3 Cooperation STM-ACE:

Interprocedural framework for retargetable compilers

Interprocedural analysis and optimization is appropriate for embedded applications, because several important system parameters are fixed as constants at integration time. It enables global optimization between compilation units and the time overhead required is paid back by the ROM and flash memory savings.

In partnership with ACE, ST has worked to define an interprocedural framework to be added to the CoSy and FlexCC retargetable compiler technologies. It implies:

- to specify how the compiler internal representation has to be extended to support this new capability,
- to define and implement algorithms to apply for extracting the necessary knowledge about interprocedural dependencies,
- to define and implement the compilation cycle.

Interprocedural analysis is mainly an enabler. It can be validated and brings actual benefits only if some derived optimizations based on it are implemented as well. This has been achieved thanks to different means:

- extension of the scope of existing optimizations like cross-module inlining, interprocedural common sub-expression elimination...
- implementation of new optimizations based on IPA: for instance, ST focusses on a data memory placement algorithm for DSP processors with several memory banks. This is especially efficient in critical loop nests found in vocoders. ST has also enhanced the alias analysis so it fully takes interprocedural aspects into account.

3.3 Difficulties Encountered

No particular difficulties have been encountered.

3.4 Recommendations

It is recommended to involve partner ACE more tightly, probably as a core partners, due to their basic role in the compiler cluster.

3.5 Milestones

Development of a SIMD code optiization engine for the CoSy compiler platform during the Aachen/ACE cooperation.

Agreements for special CoSy and Compiler Designer licenses have been agreed with ACE and CoWare. These licenses have been used for teaching a compilers course at ALARI, Lugano.

3.6 Main Funding

Main sources of funding are <indicate where the rest of the money for funding the non-integration part is coming from>

Large national project proposal to DFG, AVACS.

Several partners participate in a STREP proposal, PRESS.

ASTEC support by VINNOVA.

INRIA; CNRS, university funding.

3.7 Indicators for Integration

It is expected that this activity will lead to a world-leading compiler platform prototype whose capabilities includes many existing and newly developed techniques which are so far largely separated due to heterogeneous compiler platforms in use by the different partners. High-level transformations will be available to the partners.

3.8 Evolution

It is expected that joint activities will continue for selected areas beyond the ARTIST duration in order to implement refinements, extensions, and transfer of ARTIST results. It is anticipated

that further refinements and transfer of new techniques will be carried out between partners also beyond the ARTIST funding period.

Concerning the Aachen/ACE cooperation, the development of new code optmization engines is envisioned, e.g. for utilization of conditional instructions in the compiler backend.

4. Detailed Technical View

4.1 Brief State of the Art

A large number of compiler platforms are available in industry and academia, e.g. GCC (GNU), LANCE (Dortmund/Aachen), OCE (Atair/Mentor), SUIF (Stanford), ROSE (LLNL) and CoSy (ACE). Compiler platforms are usually conceived as software systems that allow for quick development of compilers for new target machines and that permit efficient research by means of an open, easily extensible infrastructure.

A key problem, though, is the fact that there is still no “one-size-fits-all” platform. Each of the available platforms has its specific strengths and weaknesses w.r.t. openness, IP rights issues, code quality etc. Furthermore, different platforms serve different research requirements, e.g. some are more suitable for backend modifications while others are better for source level transformations. Therefore, it is expected that the heterogeneous platform landscape will continue to exist in the future.

Nevertheless, the members of the ARTIST2 compiler cluster have decided to largely focus on one specific compiler platform, i.e. the CoSy system by ACE. ACE is offering a special ARTIST2 research license which eliminations IP rights issues, it shows a good degree of openness, and it comes with the quality and support of an industrial software tool. Therefore, CoSy has been identified as the best compromise for the compiler cluster, even though some partners are still using (and will most likely continue to do so) their own choice of platforms. A good degree of integration is still ensured, though, by the option to integrate techniques stemming from different platforms at a later point of time (possibly beyond the ARTIST2 time frame) once sufficient resources are available for the required implementation work.

Sharing common platforms between the ARTIST2 partners has the great benefit of being able to leverage each other´s specialized knowledge and technologies. Due to the heterogeneity of existing tools, a significant part of the compilers platform effort is spent for defining and implementing appropriate interfaces. Examples are the interfaces defined for the Aachen/Dortmund cooperation on compiler backend support for SIMD instructions and the LISATek/MPARM interface by Aachen and Bologna that bridges a gap between the compiler and execution platforms cluster. It is expected that once stable interfaces have been designed for all intra- and inter-cluster cooperations, more focus can be on production of measurable results, e.g. in terms of an improved code quality for specific generated compilers.

4.2 Industrial Needs and Experience

In embedded systems industry there is currently an increasing demand for compiler platforms that enable fast compiler retargeting for application specific processors. In particular this holds for large semiconductor and system houses like STM, Infineon, or Nokia, all of which have recently adopted a compiler based methodology for processor architecture exploration. At the same time, there is a demand for very high code quality, w.r.t. code size, performance, and/or power consumption. The concept of a common compiler platform greatly helps to integrate the expertise from different partners, instead of re-inventing and re-implementing the same techniques over and over again. An example is the already existing cooperation between STM and ACE based on the CoSy platform. This cooperation is being continued within the ARTIST2 compiler platform activitiy, and it allows further partners to directly participate in industry-relevant research.

4.3 Ongoing Work in the Partner Institutions

Aachen University coordinates the compiler platform integration activities and performs R&D work on retargetable compilation and code optimization. In particular, Aachen has a tight cooperation with ACE on extensions of the CoSy platform and has already transferred results to industry. Furthermore, Aachen cooperates with Bologna University beyond the compiler platform cluster on the coupling with a multiprocessor simulation platform.

Improved code quality for embedded application is the main goal of the work at Dortmund University. Due to the wideing gap between processor and memory speeds, emphasis has been on improving the efficiency of memory accesses, in terms of average and worst case execution time and in terms of the energy consumption. Algorithms have been extended to now also cover multi-threaded scenarios. In order to improve the applicability of the optimizations, a stand-alone tool was designed which can be added to existing tool chains. In cooperation with Bologna University, this tool was integrated with a multiprocessor simulation platform. In a new (initially unconceived) cooperation with AbsInt, integration of worst-case timing analysis tools and compilers was started. Source-to-source optimizations were explored in cooperation with IMEC. The cooperation with Aachen University aims at the exploitation of SIMD instructions, especially in retargetable compilers. Work on the dissemination of results is described in other contexts.

Absint: The CoSy-PAG integration has been evaluated and planned. It will be continued when appropriate funding sources are available.

TU Vienna works on analysis and optimization of high-level abstractions such that they can be used in programming embedded systems in future. The languages include C++ and C. In this effort TU Vienna has a close collaboration with AbsInt and the Lawrence Livermore National Laboratory (LLNL) to investigate the impact of optimizations on various different platforms. The necessary program analysis is specified as a high-level specification using the Program Analysis Generator (PAG) from AbsInt. TU Vienna integrated PAG into ROSE, a C/C++ source-to-source infrastructure developed at LLNL. The integration of PAG into ROSE was automated by developing the PAG Interface Generator (PIG). The flexibility of PIG was demonstrated by also integrating PAG into OCE. The existing high-level optimizations in ROSE can now be refined with additional analysis developed with PAG. ROSE has a C/C++ backend which permits further processing with CoSy.

4.4 Interaction, Building Excellence Between Partners

Interaction between the compiler cluster partners takes place via regular global meetings every few months as well as numerous bilateral “min-cluster” meetings. Furthermore, there is an extensive exchange of software components. For instance, ACE is making the CoSy platform available free of charge to interested parties, and specific tools and interfaces are being exchanged for common R&D work. As a result, ARTIST2 has significantly contributed to the partners´ awareness of each others achievements and technologies. With the progress of interfaces being defined and implemented and the increasing shift of focus towards measurable results it is strongly expected that the ARTIST2 network will be able to boost the European excellence in compiler for embedded systems.

4.5 Spreading Excellence

As the cooperation within the compiler cluster, based on common platforms, gives a lot of opportunities for exploring new avenues in compilers for embedded processors, it is expected that a number of joint publications will soon result, some of which are already in progress. Furthermore, there are common teaching activities. Members of the University of Dortmund

and RWTH Aachen taught at ALARI in Lugano, Switzerland. Special arrangements were made with CoWare and ACE to provide group licenses for design software used during hands-on sessions. Students at ALARI are going for a Master's degree in embedded system design. The program is organized in cooperation with industrial sponsors. The members also taught at EPFL, Lausanne. EPFL runs a continuing education program aiming at advanced PhD students and industrial participants. In both cases, research knowledge was transferred. P. Marwedel, head of the group at Dortmund, is also chair of the steering committee of the SCOPES workshop. SCOPES focusses on compilers for embedded systems. In 2005, the SCOPES workshop is being held in Dallas. In addition, an ARTIST workshop was held during DATE (Design, Automation and Test in Europe) 2005 in Munich. Furthermore, dissemination also includes the publication of the text book "embedded system design" by P. Marwedel. This book is being adopted by a growing number of Universities around the world and a cheaper paperback version will be published in 2005. The group at Dortmund is also transfering research results via the local technology transfer centre ICD. ICD works on a contract basis for industrial customers.