

# **Overview of compiler activities at Institute for System Programming of RAS**

**[www.ispras.ru](http://www.ispras.ru)**

**Gelato compiler focus group, October 2003**

## *Background*

We were engaged to the development and implementation of system software for the “Elektronika SS-BIS” supercomputer (similar to Cray):

Programming System including Assembly Language, Linker and Loader, as well as PL/1, Fortran 77, Pascal, and ANSI C optimizing and parallelizing compilers were designed.

## *Background*

Design and implementation of the SQL compiler and request optimizer for free SQL server supported by Free Software Foundation. Development from scratch of the Protel and Protel 2 optimizing compilers for HPUX against contract with Nortel Networks (Protel and Protel 2 are Nortel Networks private languages used for implementation of switch operating system and other system software).

## *Current works*

- Extracting information from legacy software
- Tools for formal specification languages
- Cross development tools for embedded systems
- Detection of security vulnerabilities
- Research of new methods of static and profile based program optimizations

## *Supporting tools*

Cocktail Toolbox – Front-end and Abstract Syntax Tree

IRE – Optimisation phases

IRE is a collection of program optimization algorithms (control-flow optimization, data-flow optimizations, profile-based optimizations).

All algorithms operate over the common medium-level intermediate representation.

The IRE provides graphical interface to all its components.

## *Extracting information from legacy software*

The following works are performed under contract with ***KLOCwork Solutions Corporation***:

- Compilation of C, C++, Java to the knowledge base of code facts
- Architecture recovery and refactoring
- Defect detection
- Metrics calculation

## *Tools for formal specification languages*

Under contract with **Telelogic** the following works are carried out:

- ASN.1 Compiler development
  - Generation of SDL packages and TTCN declarations from ASN.1 specification as well as generation of encoding and decoding procedures
- Implementation of front end and back end tools that work with SDL and UML 2.0 languages
- MSC to SDL synthesizer
- Maintenance of Telelogic TTCN support component called TTCN Suite

## Cross development tools

Under contract with **VIA Technologies** a complete cross development SW tools chain for a proprietary DSP processor is developed:

- Fast Cycle True Simulator (*10 MCPS on PIII-1000Mhz*)
- Macro Assembler, Disassembler
- Visual Linker
- Optimizing DSP-C Compiler
- Source Level Debugger and Various Profilers
- HW Emulator
- Integrated Development Environment (IDE)

## *Detection of security vulnerabilities*

Under contract with **Nortel Networks** a tool for automatic detection of security vulnerabilities in C programs is being implemented.

A security vulnerability in a program is an error, which results in violation of security restrictions, if this error is exploited.

*Example: by feeding a carefully crafter URL to certain versions of MS IIS (web server) it is possible to gain administrative privileges on the machine, which runs IIS.*

## *Detection of security vulnerabilities*

Various classes of security vulnerabilities known: buffer overflow, format string, tainted input, etc.

A data-flow algorithm detecting buffer overflow and format string errors is designed. This algorithm is based on use of advanced aliasing techniques and interprocedural flow-sensitive context-insensitive passes implemented in IRE.

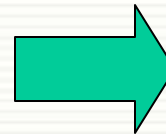
# *New methods of static and profile based program optimizations*

This research project is supported by a grant provided by **Intel**

## **Questions for research:**

How to automatically separate into threads non-loop segments of a program?

How locality considerations may be used for better thread separation?



## **Problem statement:**

Develop a new profile-based algorithm for thread separation on SMP machines

Implement this algorithm and build it into IRE currently developed

# *New methods of static and profile based program optimizations*

## **An algorithm proposed:**

Build data dependence graph of a program (DDG)  
Find its nodes' execution time from program profile  
Perform thread partitioning with full modeling of cache behavior and locality events using known execution times

## **Research results:**

Evaluation of the algorithm on sample function showed a speedup of 1.5 to 2.0 on 4-processor Itanium machine using ICC 8.0  
Evaluation and comparison with other approach on generated DDGs showed promising results

## *Our proposal*

There is strong need for open-source C++ compilers with strong optimization for Intel Itanium platform.

We propose to implement the following optimization phases for C++ compiler:

- profile-based method/function inlining;
- profile-based function (method) specialization;
- automatic thread separation;
- interprocedural analysis framework

## *C++ compiler optimization*

The project is ready to start. It is based on the current version of the GNU Compiler Collection (GCC).

The project will use the current version of IRE which contains prototype implementations of:

- profile-based function specialization
- automatic thread partitioning and locality improvement