Project Results

MANY
More efficient reuse of software code

The practice of reusing existing software coupled with the arrival of many-core semiconductor architectures poses the problem of how to rewrite software applications to exploit the increased parallel processing available. The ITEA 2 MANY project responded with an improved programming environment for embedded systems to facilitate the faster development of applications for a variety of hardware platforms. By focusing on the legacy code of the software already developed and the possibility of automatically reusing code, potentially huge savings are created and tools provided that solve the issues and enable better utilisation of the technology. MANY not only solves some dead ends by providing better operating software but also optimises the quality of the output and, thus, performance.

KEY APPLICATIONS
There are three main applications of MANY: a software source that enables parallelised software that runs faster (optimised) and uses the higher-performance that multi-core hardware provides for speed and power; interactive support tools by providing extensive analysis and pinpoint with mark-up, leaving the decision to developer where to make the actual changes as well as helping them to understand and learn; the capability to change the actual code during execution rather than pre-runtime in order to provide a virtualisation platform on which to run the application. This effectively serves as the layer between the actual hardware and the application, and so enables all concepts to be combined.

The major beneficiaries of the tools are software developers since the tools solve two basic issues that are considered almost impossible to do manually: to run legacy software (single core software) on new advanced high-performance hardware (multi-core hardware) and to parse and transform source-to-source legacy code (single core software) into optimum high-performance application (multi-core software, or parallel software code). The benefit of reusing historical investments on software and automatically transform it to high-performing applications, is the nil or at least the negligible cost.

BREAKTHROUGH
This parsing of legacy code represents a major breakthrough; it is a complex technology with great expectations and huge potential at low investment. Estimates of the project results show important and significant savings in development costs while some of the consortium partners have begun to generate income. No other similar technologies have yet matured or exist and having no tool is tantamount to entering a ‘black hole’ of a never-ending development cost.

High-performance, low-power computing is particularly important in the embedded systems market in which video recognition, streaming media and complex algorithms are typical applications in the telecom and radio communication domain as well as increasingly in the automotive domain. Some examples of benefits for the final customers of high performance products can include faster mobile phones or devices, longer lifetime on handheld devices, potential for increased performance. Advanced and complex algorithms make it possible to take full advantage of the hardware performance.
ADDED VALUE OF THE RESULTS
There are two distinct areas in which the project achieved results: development tools and code analysis and transformation. The development tools derive from the need of high performing software applications and the associated software support whereby sequential software migration enables parallel programming to harness multi-core architectures and address multi-core architecture requirements. Standards ensure the portability of application and performance while runtime mechanisms facilitate program efficiency. The added value of MANY in code analysis and transformation is evident in finding hotspots (static analysis, hotspot region isolation), dynamic dependence analysis (less target specific, cross-platform analysis), automatic parallelisation (broader coverage of codes and target architectures, extended parallelisation and optimisation capability) and last, but not least, unique integration of the tools.

TANGIBLE BENEFITS
Project coordinator and industrial partner, Alten, provided software algorithms (scheduling and communication), tools and industrial validation, resulting in a partly open source outcome and transferable skills that benefit both partners and customers within the NG automotive and energy aware ICT. Another industrial partner, TCS, which contributed a radio software application, gained significant acceleration and power-saving using MANY methodology and tools. Collaboration with South Korean partners has generated mutual and beneficial outcomes, benefiting the technology as well as opening up business opportunities. One example is the SME Vector Fabrics in the commercialisation of the Pareon tool, which smoothens multi-core software optimisation and is being used in a few dozen locations, and where negotiations are ongoing with among others a major Korean company in the mobile domain. SevenCore’s hypervisor tool, which provides a dynamic guarantee of system reliability in reliable and resilient embedded systems (including automotive, defence and medical systems as well as unmanned control vehicles), has been selected by Hyundai Heavy Industry for its new HP robot controller. Finally, there is a positive impact on the academic environment in the incorporation of improved techniques for scheduling time-division multiplexing that are now also part of the Quantitative Evaluation of Embedded Systems course in the Embedded Systems Master’s curriculum at all three Dutch universities of technology. The ultimate gain, however, will be a stronger European industry that will see not only accelerated time to market but also a reduction in the need for upfront investments.

Major project outcomes

DISSEMINATION

EXPLOITATION (SO FAR)
- PAREON - Analysis tool for parallelisation
- CodeletFinder - Identifying code for parallelisation
- PyLINX - Low level HP communication package
- PoCC - Code parallelisation tool
- STEP - Code transformation tool
- LLWM extension for Code parallelisation tool
- Virtualization platform with online parallelisation
- Monitoring for the virtualisation platform

STANDARDISATION
- The work is aligned with a variety of standards and defacto standards, such as MCAPI, MPI, OpenMP, POSIX, OPENCL, OpenACC, MTAPI, Linux

SPIN-OFFS
- Commercial negotiation is ongoing with several customers