SEVENTH FRAMEWORK PROGRAMME

HiPEAC info 24

COMPILATION ARCHITECTURE

appears quarterly | October 2010

Network of Excellence on High Performance and Embedded Architecture and Compilation

Welcome to Computing System Week in Barcelona, Spain, October 19-22

www.HiPEAC.net

HiPEAC'11 Conference, 24-26 January 2011, Heraklion, Greece
http://www.hipeac.net/hipeac2011

Message from the HiPEAC Coordinator

Koen De Bosschere

Dear friends,
I hope all of you have enjoyed relaxing summer holidays and that you are ready to tackle the challenges of the coming year. This summer, the international news was overshadowed by big items like the massive oil spill in the Mexican Gulf, and the huge flooding in Pakistan. However, on August 21, I was struck by a small footnote in the news: Earth Overshoot Day, i.e. the day where the Earth's population had consumed all resources that earth produces in 2010. This year it was three days earlier than in 2009, and 40 days earlier than in 2000. Clearly, our impact on planet Earth is still increasing. I was wondering about the impact of computing on the planet. On the one hand it definitely helps to preserve the environment, save energy, make the planet a safer and better place to live. On the other hand it clearly consumes energy, but it also seems to indirectly stimulate the accelerated consumption of natural resources by making consumer goods and services more affordable.

In June, HiPEAC underwent its second review. The reviewers concluded that we have made good progress on the recommendations of the first review, and they asked us to continue our efforts. They also suggest we start a more structural collaboration with the different Computing Systems projects in FP7 and to promote the HiPEAC roadmap more widely. To that end, we have created a small roadmap movie that you can watch at http://www.hipeac.net/roadmap.

In July, many of us enjoyed the yearly ACACES summer school in La Mola, Barcelona. Like the previous years, this flagship event of our community received a high appreciation score by the participants. I would like to thank all the instructors for their excellent performance, and I would also like to thank our colleagues from UPC and BSC for all the logistic support to make the event successful.

In October we organize our fall Computing Systems Week in Barcelona. Like in 2008, it is co-located with the Barcelona Multi-core Workshop. I hope that we will all take the opportunity to network and to start working on high quality project proposals for the 2011 Call in Computing Systems. This is the last opportunity before the deadline of the 2011 Call.

Next, there is the HiPEAC Conference, currently being prepared by our Greek colleagues in beautiful Heraklion, Crete. The conference runs for three days in January 2011, and it is preceded by a rich set of workshops and tutorials, several of which are directly linked to the HiPEAC research clusters. The whole event is expected to be a major networking event for our community.

Finally, I would like to thank Wouter De Raeve who handled the administrative and financial management of the HiPEAC project since its beginning. He decided to leave the network on August 1 for a new position within Ghent University. The HiPEAC project management has since then been taken over by Jeroen Ongenae.

Take care
Koen De Bosschere

HiPEAC Management Staff Change

As of August 1st, the HiPEAC project manager Wouter De Raeve has left the network. Wouter managed all the administrative and financial aspects of HiPEAC. He also helped with the logistics for several HiPEAC activities, such as the cluster meetings, the conference, and the summer school. Wouter has moved on to another job in the central administration of Ghent University. The HiPEAC community thanks Wouter for his excellent work and wishes him success in his new position.

Wouter has been replaced by Jeroen Ongenae. Jeroen has a Master degree in History, and he works in the same department for several years. He was already familiar with HiPEAC, so the transition should go smoothly. If you have questions, do not hesitate to contact him at jeroen@hipeac.net.

Wouter De Raeve

Jeroen Ongenae

Panos Tsarchopoulos
Panagiotis.Tsarchopoulos@ec.europa.eu

Message from the Project Officer

Over the last few years, the European Commission has consistently increased the amount of funding going to research in computing architectures and tools with special emphasis on multicore computing. Tracing back to the recent past, the European Commission has funded the HiPEAC Network of Excellence which commenced in 2004. Networks of Excellence are an instrument to overcome the fragmentation of the European research landscape in a given area; their purpose is to reach a durable restructuring/shaping and integration of efforts and institutions.

In the following years, numerous European collaborative research projects in Computing have started. It is important to note here that collaborative research projects are the major route of funding in the European research landscape in a way that is quite unique worldwide. In European collaborative research projects, international consortia consisting of universities, companies and research centers, are working together to advance the state of the art in a given area.

In 2006, the European Commission launched the Future and Emerging Technologies initiative in Advanced Computing Architectures as well as a number of projects covering Embedded Computing. In 2008, a new set of projects were launched to address the challenges of the multi/many core transition - in embedded, mobile and general-purpose computing - under the research headings "Computing Systems" and "Embedded Systems"; these projects were complemented by a second wave of projects that have started in 2010 under the same research headings together with a new Future and Emerging Technologies initiative on "Concurrent Tera-device Computing".

In Call 7, which is open now, there are two relevant research objectives, one under the heading "Computing Systems" with 45 million euros funding and the other under the heading "Exascale Computing" with 25 million euros of funding; deadline for proposal submission is January 2011.

It is now crucial that the research community delivers high-impact proposals that will reinforce European excellence. We need to speed ahead on a higher gear.

Panos Tsarchopoulos

HiPEAC News

Mateo Valero Awarded Honorary Doctoral Degree by the Universidad Veracruzana, Mexico

The Universidad Veracruzana (UV), located in the Mexican state of Veracruz, conferred the honorary doctoral degree to Professor Mateo Valero, director of the Barcelona Supercomputing Center (BSC). The president of the Universitat Politècnica de Catalunya (UPC), Antoni Giró, delivered the oration for Professor Valero.

At the ceremony, held on May 28, UV also awarded honorary degrees to the Spanish stage director Manuel Montoro Tuells and expert on education and democracy, Gilberto Guevara Niebla, from Mexico.

The Universidad Veracruzana honored the significant contributions Mateo Valero has made to Computer Architecture, citing him as among the field's top contributors of the past 25 years.

In his acceptance speech, Valero thanked the University for bestowing the honor and recalled his ties with Mexico dating back to his childhood and numerous trips to cities around the country. He also emphasized the role of research as one of the main drivers of the country's economy. "A country must produce wealth in a sustainable manner if it is to offer a solid education and good healthcare to the entire population. The research being conducted also provides answers to the complexities facing modern societies. In short, we need to focus on and nurture our research."

UPC president Antoni Giró, full professor at the Department of Physics and Nuclear Engineering, delivered the encomium honoring the director of the BSC. In his address, Giró praised

the success achieved by Valero during his 36-year career, which has been characterized by "dedication and excellence, dedication and generosity, dedication and imagination, dedication and enthusiasm, dedication and presence, dedication and integrity, dedication and effort, effort and dedication."

For UV, each honorary degree awarded during the ceremony recognizes an invaluable contribution to society. Therefore, the honorees join the senate of the UV.

PUMPS Summer School

The PUMPS 2010 Summer School, "Programming and tUning Massively Parallel Systems (PUMPS)," was offered in July 5-9, 2010 at the Universitat Politecnica de Catalunya (UPC) campus. The event was co-sponsored by the Barcelona Supercomputing Center, UPC, the University of Illinois, HiPEAC NOE and NVIDIA, with distinguished faculty members Dr. David B. Kirk of NVIDIA and Prof. Wen-mei Hwu of the University of Illinois. PUMPS attracted more than 250 applications, but could only host 100 attendees from throughout the EU and beyond, from beginners to advanced programmers and faculty. The content was based on CUDA, OpenCL, OpenMP, and StarSs programming languages and numerical methods including FFT, Graph, Tiling, Grid, Montecarlo, FDTD, Sparse matrices, etc. The school featured lectures on cutting-edge new techniques and hands-on laboratory experience in developing applications for many-core processors with massively parallel computing resources such as the GPU accelerators, highlighting the new Fermi architecture. Applications case studies focused on Molecular Dynamics, MRI Reconstruction, RTM Stencil, and Dense Linear Systems. The social activities, the presentation of the GPUComputing.net community as well a showcase of high-end equipments by SuperMicro, AZKEN, and Bull, promoted the participants' interaction and the seed for future collaborations. The PUMPS "NVIDIA Best Achievement Award" Contest was launched as a continuation for the attendees to optimize four proposed challenges using what they learned at the summer school.

More information at:
http://bcw.ac.upc.edu

Computer Architecture Performance Evaluation Methods

by Lieven Eeckhout

Morgan & Claypool Publishers
Synthesis Lectures on Computer Architecture
Ed. Mark D. Hill

Performance evaluation is at the foundation of computer architecture research and development. Contemporary microprocessors are so complex that architects cannot design systems based on intuition and simple models only. Adequate performance evaluation methods are absolutely crucial to steer the research and development process in the right direction. However, rigorous performance evaluation is non-trivial as there are multiple aspects to performance evaluation, such as picking workloads, selecting an appropriate modeling or simulation approach, running the model and interpreting the results using meaningful metrics. Each of these aspects is equally important and a performance evaluation method that lacks rigor in any of these crucial aspects may lead to inaccurate performance data and may drive research and development in a wrong direction. The goal of this book is to present an overview of the current state-of-the-art in computer architecture performance evaluation, with a special emphasis on methods for exploring processor architectures. The book focuses on fundamental concepts and ideas for obtaining accurate performance data. The book covers various topics in performance evaluation, ranging from performance metrics, to workload selection, to various modeling approaches including mechanistic and empirical modeling. And because simulation is by far the most prevalent modeling technique, more than half the book's content is devoted to simulation. The book provides an overview of the simulation techniques in the computer designer's toolbox, followed by various simulation acceleration techniques including sampled simulation, statistical simulation, parallel simulation and hardware-accelerated simulation.

HiPEAC Goes Across the Pond to DAC

DAC (Design Automation Conference) is the world's leading technical conference and tradeshow of the electronic design and design automation, truly "the show of the year". This year's edition was held in Anaheim in the week of June 13-18, which as usual attracted thousands of attendees from academia and industry (official count: 5920 participants). The venue, Anaheim Convention Center, is a large modern facility to house both the trade-show in the ground floor and the comprehensive technical conference parts in the floors above. The great exhibition features ingredients from nearly all major players in the industry. The conference boasts with a significant number of excellent technical papers. Vibrant panels and user track sessions on-site, live broadcasting of FIFA World Cup games, freebies and lotteries everyday, southern California weather and many others more contributed to another great success of DAC.

Following the successful booth shows at the past two DATE events, HiPEAC has decided to go across the pond this time to get increased international exposure and expand its impact. The booth preparations were carried out by Ghent University and RWTH Aachen University – the booth was decorated with shining posters, latest newsletters/ HiPEAC leaflets, a running presentation on a LCD stand-point, and – for the first time of HiPEAC booth history – giveaways. We prepared boomerangs with HiPEAC logo and URL, which turned out a great success and provided for good traffic at DAC's trade-show with its very commercial nature. With the careful booth location selection, we were close to the main road of the hall and direct neighbor of the gigantic Synopsys camp, attracting hundreds of visitors during the exhibition days. We are proud to say that, as the only present European project setting flag in DAC, the booth has delivered key messages of the mission and functionalities of the HiPEAC network. It was well received and we had many new visitors, especially from the US, everyday learning the relative new concept of NoE (Network of Excellence), the research clusters of HiPEAC and expressing interests in joining the HiPEAC events.

The HiPEAC booth in DAC stepped up beyond its European border to extend its visibility and influence to a larger community. It is expected that through follow-up communications HiPEAC will make new friends and see many new faces in the upcoming events. It was a wonderful experience running the booth, greeting HiPEAC friends and talking to the visitors in a great summer!

Weihua Sheng and Felix Engel introducing HiPEAC to DAC attendees

Tom Crick Takes Over HiPEAC Newsletter Proofreading Role

Igor Böhm, a PhD student at Edinburgh University, who has been the HiPEAC newsletter proofreader for over a year, has recently left this role due to other commitments. Dr Tom Crick has volunteered to replace Igor. Tom is a recent member of HiPEAC and is a Lecturer in Computing at the University of Wales Institute, Cardiff (UWIC), having previously been a PhD research student and post-doctoral researcher at the University of Bath.
The HiPEAC community thanks Igor for the reliable service he performed and welcomes Tom to the newsletter production team!

Igor Böhm

Tom Crick

Habilitation Thesis Defence of Sid Touati on Backend Code Optimization


Sid-Ahmed-Ali Touati

This habilitation thesis was defended in June 30th, 2010 at the University of Versailles Saint-Quentin en Yvelines (France). We start our document by a global view on the phase ordering problem in optimising compilation. Nowadays, hundreds of compilation passes and code optimisation methods exist, but nobody knows exactly how to combine and order them efficiently. Consequently, a best effort strategy consists in doing an iterative compilation by successively executing the program to decide about the passes and optimisation parameters to apply. We prove that iterative compilation does not fundamentally simplify the problem, and using static performance models remains a reasonable choice.

A well-known phase ordering dilemma between register allocation and instruction scheduling has been debated for long time in the literature. We show how to efficiently decouple register constraints from instruction scheduling by introducing the notion of register saturation (RS). RS is the maximal register need of all the possible schedules of a data dependence graph. We provide formal methods for its efficient computation, that allows us to detect obsolete register constraints. Consequently, they can be neglected from the instruction scheduling process.

In order to guarantee the absence of spilling before instruction scheduling, we introduce the SIRA framework. It is a graph theoretical approach that bound the maximal register need for any subsequent software pipelining, while saving instruction level parallelism. SIRA model periodic register constraints in the context of multiple register types, buffers and rotating register files. We provide an efficient heuristic that show satisfactory results as a standalone tool, as well as an integrated compilation pass inside a real compiler.

SIRA defines a formal relationship between the number of allocated registers, the instruction level parallelism and the loop unrolling factor. We use this relationship to write an optimal algorithm that minimises the unrolling factor while saving instruction level parallelism and guaranteeing the absence of spilling. As far as we know, this is the first result in the literature proving that code size compaction and code performance are not antagonistic optimisation objectives.

The interaction between memory hierarchy and instruction level parallelism is of crucial issue if we want to hide or to tolerate load latencies. Firstly, we practically demonstrate that superscalar out-of-order processors have a performance bug in their memory disambiguation mechanism. We show that a load/store vectorisation solves this problem for regular codes. For irregular codes, we study the combination of low-level data pre-loading and prefetching, designed for embedded VLIW processors.

Finally, with the introduction of multicore processors, we observe that program execution times may be very variable in practice. In order to improve the reproducibility of the experimental results, we design the Speedup-Test, which is a rigorous statistical protocol. We rely on well known statistical tests (Shapiro-Wilk's test, Fisher's F-test, Student's t-test, Kolmogorov-Smirnov's test, Wilcoxon-Mann-Whitney's test) to evaluate if an observed speedup of the average or the median execution time is significant.

Barcelona Multicore Workshop 2010

The Barcelona Supercomputing Center, HiPEAC and Microsoft Research are proud to organize the Second Barcelona Multicore Workshop (BMW), which will take place in Barcelona on 21-22 October 2010. Building on the success of the highly productive 2008 Barcelona Multicore Workshop, BMW2010 strives to examine critically the developments in the intervening two years. The Second Barcelona Multicore Workshop (BMW) is part of the upcoming HiPEAC Barcelona Computing Systems Week.
Our goal is to bring together prominent researchers active to cross-fertilize ideas across the hardware and software communities and discuss the cutting edge in research and products: how to maximize the effectiveness of many-cores. BMW2010 will consist of a two-day workshop with invited talks, panel sessions and time for discussion.
More updated information at:
http://www.bscmsrc.eu/media/events/barcelona-multicore-workshop-2010

Collaboration on the Adaptation of Numerical Libraries to Graphics Processors for the Windows OS

Prof. Gregorio Quintana, from the Department of Computer Science and Engineering of the Universidad Jaume I (UJI) will lead a group of researchers working during the next year in a contract working with Microsoft with the aim of adapting numerical libraries for graphics processors under the Windows operating system.

The High Performance Computing & Architectures (HPCA) research group of the UJI, to which 14 researchers belong, has been working for some years in the development of numerical libraries of programs for supercomputers, and lately, in multicore processors and graphics processors.

A couple of years ago, the HPCA group devoted a specific research line to the study of graphics processors and the research of new functionalities. The work developed in this area, with the creation of optimized software for the matrix calculus in this kind of processors, made the group worthy of the Professor Partnership Award from NVIDIA, the leading worldwide manufacturer of graphic cards for computers.

This award and the collaboration they maintain with The University of Texas at Austin have given the group the possibility to work directly with Microsoft. The company was interested in accessing the technology of graphic processors developed by the group from Castellón, which, at that point, was one of the few groups developing such technology.

The programs developed by the HPCA group are used in millions of computers in applications and software packages such as Matlab, Labview, Octave and Lapack. These programs are currently the fastest solvers for complex calculus operations as linear-least squares problems, numerical rank computations or eigenvalue computations. This software is used in numerous scientific and engineering applications, such as in analysis of building structures, computational physics, chemistry simulations, with the use of a graphics processor yields a considerable reduction of the time-to-solution in this area.

The members of the HPCA group are currently working on energy saving techniques in order to control the power consumption of large-scale computer installations. The group has also collaborated with organizations such as NASA to adapt numerical libraries to space vehicles; with Chicago Hospital where they used the group's programs for magnetic resonance to diagnose arterial blockages; with Boeing to model airplanes; and with researchers from companies including Intel, NVIDIA and ClearSpeed.

Handbook of Signal Processing Systems

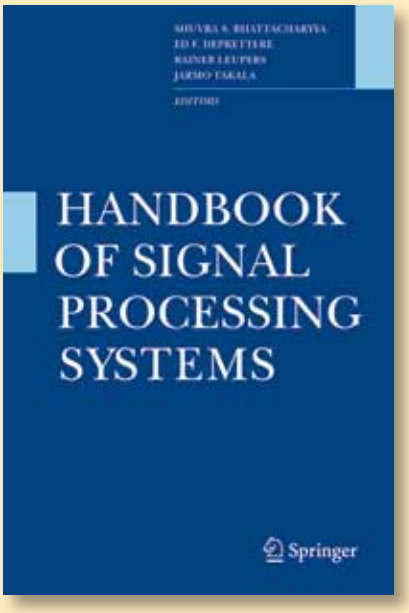
(ISBN: 978-1-4419-6344-4)

By Shuvra S. Bhattacharyya, Ed F. Deprettere, Rainer Leupers, and Jarmo Takala

Topics covered:

- Applications;
- Architectures;
- Programming and Simulation Tools;
- Design Methods

The Handbook of Signal Processing Systems provides a standalone, complete reference to signal processing systems organized in four parts. The first part motivates representative applications that drive and apply state-of-the-art methods for design and implementation of signal processing systems; the second part discusses architectures for implementing these applications; the third part focuses on compilers and simulation tools; and the fourth part describes models of computation and their associated design tools and methodologies. Each part has 6 - 12 chapters from leading experts.



Lieven Eeckhout Awarded an ERC Starting Independent Researcher Grant

Contemporary microprocessors seek to improve performance through thread-level parallelism by co-executing multiple threads on a single microprocessor chip. Projections suggest that future processors will feature multiple tens to hundreds of threads, hence called many-thread processors. Many-thread processors, however, lead to non-dependable performance: co-executing threads affect each other's performance in unpredictable ways because of resource sharing across threads. Failure to deliver dependable performance leads to missed deadlines, priority inversion, unbalanced parallel execution, etc., which will severely impact the usage model and the performance growth path for many important future and emerging application domains (e.g., media, medical, datacenter).

This project on Dependable Performance on Many-Thread Processors envisions that performance introspection using a cycle accounting architecture that tracks per-thread performance, will be the breakthrough to delivering dependable performance in future many-thread processors. To this end, it will develop a hardware cycle accounting architecture that estimates single-thread progress during many-thread execution. The ability to track per-thread progress enables system software to deliver dependable performance by assigning hardware resources to threads depending on their relative progress. Through this cooperative hardware-software approach, this project addresses a fundamental problem in multi-threaded and multi/many-core processing. The project consists of funding for five years for three PhD students and one post-doctoral researcher.

Lieven Eeckhout is an Associate Professor at Ghent University, Belgium. His main research interests include computer architecture and the hardware/software interface in general, and performance modeling and analysis, simulation methodology and workload characterization in particular. He received two IEEE Micro Top Picks Awards and recently wrote a synthesis lecture on "Computer Architecture Performance Evaluation Methods". He has successfully graduated six PhD students and current supervises three post-doctoral researchers and eight PhD students. He also participates in the ExaScience Lab, part of Intel Labs Europe, focusing on architectural simulation techniques for exascale systems.

ERC Starting Grants

The European Research Council (ERC), through its ERC Starting Independent Researcher Grants, aims at supporting "up-and-coming research leaders who are about to establish or consolidate a proper research team and to start conducting independent research in Europe. The scheme targets promising researchers who have the proven potential of becoming independent research leaders. It will support the creation of excellent new research teams and will strengthen others that have been recently created." ERC grants are considered the most competitive research grants in Europe.

ComputerSciencePodcast.com A Computer Science Podcast

I sometimes pretend to exercise, though to be honest I don't pretend very hard. But when I am panting and wheezing on the treadmill or cursing the hills on my bike that didn't look so steep in the car, I like to listen to science podcasts. It takes my mind off the pain. Well, no it doesn't really, but you know what I mean.

I have lots of these sciencey podcasts; you can find them all easily enough and some of them are pretty good. But I tell you, after listening to a few of them I find myself thinking that if I have to hear another article about global warming I shall set fire to an oil well. Don't get me wrong, I believe in climate change and all that, but damn it, I'm a geek! Why do they never, ever say anything about computers? Is our work so dull or difficult to explain that no one will go near us? I refused to believe it!

So, I looked for a computer science podcast. Just one would have done. I looked pretty hard and asked everyone at the school about it, too. Nothing! Oh, there were a few people who'd put their undergrad lectures online and some people doing engineering and gadget podcasts, but for scientists? Nothing.

Foiled in my quest and being characteristically lazy you would expect me to call it a day, and I would have, too. Except I have this weird mental glitch that sometimes takes over and makes me do things I ordinarily wouldn't. This glitch (which some might call `procrastination from writing up his PhD') decided that if no one else was going to do it, then I would just have to suck it up and start doing it my self - I would make my own computer science podcast! I spoke to a few friends, and after the inevitable questions of "You're actually serious, aren't you?" had been dealt with I was amazed that they wanted to come on board, too. In fact, nobody thought this was a barmy idea at all. Before I knew it we were looking very much like we were going to be recording our first episode for a computer scientist podcast!

And it's happened. We got together and recorded news items, some paper reviews and I even managed to wangle an interview with Stephen Cook - a Turing Award recipient in our first show! It was pretty nerve wracking in the beginning, but fortunately for us (and you when you listen to it) my next door neighbour is a free lance sound engineer who has made our stuttering disappear and, though I say it myself, by the time he'd finished with us we come across pretty professionally!

Since then, we've recorded another show and will be doing a third soon. It looks like this podcast, with its monthly shows, will take on a life of its own and will keep getting better and better. You'll hear about all sorts of computer science in the show and you should find something you like. There's always the quiz and the computer science joke of the month to entertain you. We really hope you like it, and when you do listen to it, drop us a note to let us know what you think about it, even if it's just to say you can't stand the sound of my voice. Enjoy listening!

Hugh Leather

You can download the podcast or subscribe to the RSS feed on www.computersciencepodcast.com. There's also a Facebook group ("CompuCast") where you can get all the latest information about the podcast. You can also post any suggestions, comments or feedback there, or email mail@computersciencepodcast.com. We're always on the look out for good stories, too.

HiPEAC Students

Towards Joint Research between Augsburg and Sibiu

My name is Ralf Jahr and I am a PhD student of Prof. Theo Ungerer at the University of Augsburg, Germany. The focus of my work is on software optimizations for the Grid Alu Processor (GAP). The GAP is a novel approach to speed up the execution of sequential single-threaded code. Therefore, it combines a superscalar front-end with a novel configuration unit and a coarse-grained reconfigurable grid of functional units (FUs). No special software is required to prepare a program for the execution on the GAP because placement and routing are both performed in hardware. Because we do not want to loose the advantage of executing legacy programs without having access to their source code we decided to implement target-specific code optimizations in a post-link optimizer. This optimizer is called GAPtimize and offers for example static speculation, function inlining and branch predication.

In August 2010 I visited Horia Calborean for three weeks. He is a PhD student of Prof. Lucian Vintan and works for Advanced Computer Architecture & Processing Systems (ACAPS) at Lucian Blaga University of Sibiu in Romania. Horia Calborean is developing a tool, called FADSE, which performs automatic design space exploration on multiobjective problems. He focuses on multi- and many-core processor simulators and his objective is to find a near Pareto-optimal set of configurations. FADSE is available for free.

During my visit we worked on bringing together FADSE and GAP/GAPtimize. To be able to do so we had to expose the parameters of GAP and GAPtimize to FADSE. We had to choose multiple objectives that would measure the quality of the configurations generated by FADSE. In this process, we created a model to approximate the hardware complexity of a configuration of the GAP. To be able to cope with the needed computation power and to decrease the time needed to complete the high number of simulations required by DSE algorithms we extended FADSE to run in a distributed manner on multiple computers in a network. In addition, to be able to manage network errors and errors in the used software components, we increase the robustness of FADSE on multiple levels.

In the next weeks and months we will continue our work. The goals we try to achieve are various. First, we want to show that the GAP is a scalable processor architecture in terms of hardware complexity. Second, we want to elaborate the different challenges posed to FADSE by using it for the DSE of hardware or code-optimization parameters. Furthermore, we want to show that it is able to cope with them. We also

want to be able to use it for future research to find good parameters for novel code-optimizations or hardware-software co-design approaches. Third, it shall be possible to use FADSE as platform to evaluate various DSE algorithms with the different simulators linked to and reference problems available in FADSE. Of course we will publish our current and future results in one or more papers.

I would like to thank the former HiPEAC Cluster on Adaptive Compilation for generously supporting my trip to Romania. It was a great opportunity to establish a collaboration that is very likely to be successful. Due to the outstanding and admirable hospitality of my hosts Prof. Lucian Vintan and Horia Calborean, Ciprian Radu, and Arpad Gellert, I was also able to learn a lot about Romania and enjoyed a very pleasant stay there.

Ralf Jahr

HiPEAC Start-ups

Ateji, Language Tools for Software Productivity

Ateji is a Paris-based software start-up who recently introduced Ateji PX, an extension of the Java language with syntactic constructs for parallel programming. With a background and unique expertise in programming language design and technology, Ateji aims at bringing novel and innovative software solutions to the world of HPC, including multi-core, grid, cloud and GPU programming.

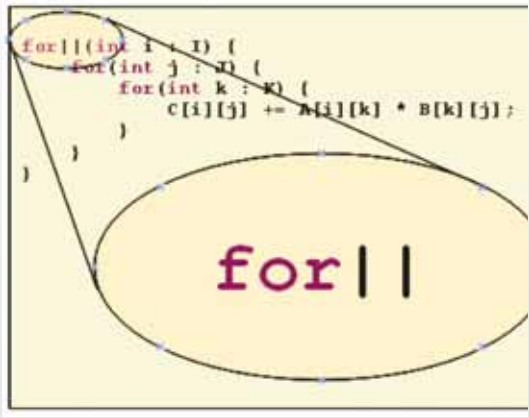
The Ateji PX language is built upon on the mathematical foundations of the pi-calculus. This makes parallel programming simple and intuitive, close to the way we "think" parallel, in addition to being analyzable by the compiler. A handful of cleverly designed constructs make it possible to express within a single language a wide range of patterns, including data-, task-, recursive- and speculative parallelism, on shared-memory or distributed architectures, and paradigms such as data flow, stream programming, MapReduce and the Actor model (all these examples are available in the Ateji PX distribution).

Ateji PX is a plugable language extension compatible at the source level with Java - the most popular language nowadays - with support in the state-of-the-art Eclipse IDE. Unlike libraries, preprocessors or brand new languages, Ateji's language extension approach brings expressive power and efficiency while leveraging existing source code, development processes and developer's training. A major goal in the design of Ateji PX is making parallel programming accessible to a much larger audience of application developers.

Efficiency (program speed) is achieved for much lower development cost and time. First, using the Java language with a recent JIT (just-in-time) compiler provides performance similar to C/C++, with code generated and optimized on-the-fly for a specific hardware configuration, while avoiding memory allocation bugs and multi-target compilation nightmare associated with low-level languages. For this reason, Java has been adopted for the handling of huge data volumes in scientific data centers such as ESA and CERN.

Second, while the Java threading model remains a very low-level abstration, parallelizing Java code with Ateji PX often amounts to inserting only a couple of parallel bars (" || ") in the source. For the matrix multiplication example, a speedup of 12.5x on a 16-core server was achieved with the single addition of a parallel bar in the source code. Ateji PX performs most of its work at compile-time and incurs practically no overhead at run-time.

Ateji is currently developing implementations of the Ateji PX language for distributed architectures (grids, clusters and cloud) where message passing is integrated at the language level and mapped by the compiler on any chosen infrastructure, such as sockets or MPI. A version dedicated to GPU accelerators is also in preparation. Because the compiler understands parallel composition and message-passing, the same source code can be ported to very different architectures without changes.



Matrix multiplication example with the parallel bars

Ateji PX has been selected for presentation in the Disruptive Technologies exhibit at SC'10 in New Orleans.

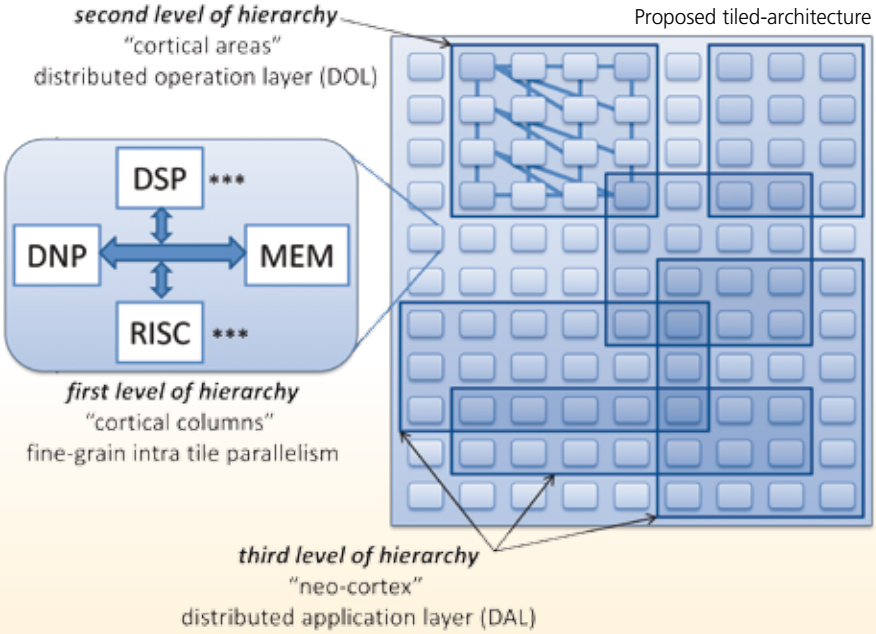
Whitepapers, online demos and evaluation licenses are available at http://www.ateji.com/px
Academic licenses are free.

Patrick Viry
patrick.viry@ateji.com

FP7 EURETILE Project: EUropean REference TILed architecture Experiment

Coordinator:
Pier Stanislao Paolucci
Istituto Nazionale di Fisica Nucleare (INFN)
Website: www.euretile.eu
Partners:
INFN (IT)
ETH Zurich (CH)
RWTH Aachen University (DE)
UJF-TIMA (FR)
TARGET (BE)
Duration: 3 years

EURETILE investigates and implements brain-inspired foundational innovations to the system architecture of massively-parallel tiled computer architectures and the corresponding programming paradigm. The execution target is a fault-tolerant many-tile HW platform, equipped with a many-tile simulator. A set of SW process - HW tile mapping candidates are generated by the holistic SW tool-chain using a combination of analytic and bio-inspired methods. The Hardware-dependent Software is then generated, providing OS services with maximum efficiency/minimal overhead. The many-tile simulator collects profiling data, closing the loop of the SW tool chain. Fine-grain parallelism inside processes is exploited by optimized intra-tile compilation techniques, but the project focus is above the level of the elementary tile. The elementary HW tile is a multi-processor, which includes a fault tolerant Distributed Network Processor (for inter-tile communication), a floating-point numerical engine (for computations) and a RISC processor (for control, user interface and sequential computations). Furthermore, EURETILE investigates and implements the innovations for equipping the elementary HW tile with high-bandwidth, low-latency brain-like inter-tile communication (emulating 3 levels of connection hierarchy, namely neural columns, cortical areas and cortex. EURETILE leverages on the working SW and HW prototypes of the innovative multi-tile HW paradigm and SW tool-chain developed by the FET-ACA SHAPES Integrated Project (2006-2009). This background knowledge includes working tile silicon and board, a multi-tile simulator (running up to eight tiles), and a complete SW tool-chain including a parallel programming and an automatic mapping/optimization environment (Distributed Operation Layer), a specialized OS (DNA-OS automatically generated for both RISC and VLIW floating-point numerical processor) integrated with Linux RT, and an optimizing compiler co-designed with the floating-point engine. INFN (Istituto Nazionale di Fisica Nucleare) is the coordinator of the project. The APE Parallel Computing Lab of INFN Roma is in charge of the EURETILE HW Architecture and Scientific Application Benchmarks. The Computer Engineering and Networks Laboratory (TIK) of ETH Zurich (Swiss Federal Institute of Technology) designs the high-level explicit parallel programming and automatic mapping tool (DOL/DAL). The Software for Systems on Silicon (SSS) group of the ISS institute of RWTH Aachen University, investigates and provides the parallel simulation technology. The TIMA Laboratory of the University Joseph Fourier in Grenoble explores and deploys the HdS (Hardware dependent Software) including the distributed OS architecture. TARGET Compiler Technologies, the Belgian leading provider of retargetable software tools for the design, programming, and verification of application-specific processors (ASIPs), is in charge of the optimizing C compiler for custom components of the EURETILE architecture.



Project Coordinator
Pier Stanislao Paolucci
Istituto Nazionale di Fisica Nucleare, Roma
pier.paolucci@roma1.infn.it

FP7 PEPPHER Project: Performance Portability and Programmability for Heterogeneous Manycore Architectures

Coordinator: Sabri Pllana, University of Vienna (AT)
Website: www.peppher.eu
Partners:
University of Vienna (AT)
Chalmers University (SE)
Codeplay Software Ltd. (UK)
INRIA (FR)
Intel GmbH (DE)
Linköping University (SE)
Movidius Ltd. (IE)
Karlsruhe Institute of Technology (DE)
Duration: 3 years

The pervasiveness of multi-core processors in Europe impacts on the whole spectrum of systems from embedded and general-purpose to high-end computing systems. However, leveraging the inherent performance of such multi-core systems, in particular heterogeneous many-core systems, with a reasonable effort is a challenging task.

Consortium members

The three-year European FP7 project PEPPHER, which started in January 2010, addresses the vital issue of maintaining application performance portability in today's different, highly complex, often heterogeneous parallel many-core architectures. This means that the necessary adaptations of an application to ensure good absolute performance or preserving specified performance requirements when porting from one, possibly heterogeneous many-core architecture, to another are performed automatically by static and dynamic reorganizations of programmer identified components of the code. PEPPHER addresses this challenge by a component annotation and coordination language with which the programmer identifies components of the application, specifies their interaction and performance requirements, possibly by relating to abstract machine models stored in a PEPPHER repository. PEPPHER assumes that these components may already have been parallelized using a conventional language or framework.

A major challenge in the project is to make this parallelization-agnostic approach work with different existing parallelization models and languages. The coordination of components enables auto-tuning at a high-level of abstraction by finding the optimal composition of the identified components. The static optimization of more or less coarse-grained components is complemented by an efficient, heterogeneous run-time scheduling system that is able to select the most promising implementation of a component and schedule this among the immediately available processing elements, based on performance information derived from the component annotations, and/or from the history based performance information. The static and dynamic optimization possibilities exposed through the annotation language that permit transformation and compilation into component variants, are complemented by compilation techniques and efficient adaptive algorithms library support. For special, commonly used, or performance-critical components, a framework for library support of auto-tuned algorithms that can adapt to different types of many-core architectures is likewise being developed. Finally, PEPPHER is investigating hardware support for performance portability and programmability based on a simulator developed in the project.

Coordinated by the University of Vienna, PEPPHER unites forces of the Universities of Chalmers and Linköping, Karlsruhe Institute of Technology, an INRIA research centre, European SMEs Codeplay and Movidius, and one of Intel's European labs.

Project Coordinator
Prof. Sabri Pllana,
University of Vienna, Austria
pllana@par.univie.ac.at

An Update on MILEPOST

The original MILEPOST project, a collaboration supported by HiPEAC finished last year. This is an update on work since then.

The Challenge

Modern compilers offer hundreds of possible optimization passes but not all optimizations work well on all code. Selecting the best optimization setting is a huge challenge to the programmer.

The solution is *iterative compilation*. The user repeatedly compiles their program, trying different sets of optimization options and finally selecting the best set. This can sometimes double performance, but is too laborious for everyday use.

What is MILEPOST?

MILEPOST "learns" the characteristics of programs and associates them with the best set of optimization passes. The infrastructure provides a database to hold information "learned" and uses this to select the optimization passes to be used with any new program.

The database is trained using a representative set of programs, which are compiled many times with different options and their performance measured. Standard machine learning techniques are used to construct a probabilistic model, from which predictions for new (unseen) programs can be made.

The approach is designed to be generic, but has initially been implemented for GCC, with MILEPOST GCC 4.4 released in June 2009. Early results showed that average performance benefits ranged from 11% to 40% for the different processors, with the best programs more than doubling in speed. That is comparable with the results from iterative compilation, but with a compiler that is as easy and quick to use as standard GCC.

Collective Tuning and Recent Work on MILEPOST

Since the completion of the original MILEPOST project, the Collective Tuning website (www.ctuning.org) has acted as the repository for the technology.

MILEPOST was developed as a generic solution, although the first released version was for GCC. For MILEPOST to be widely adopted, we need it to use standard features of compilers such as GCC. This will involve both changes to MILEPOST and changes to standard GCC.

Over the past year, Jörn Rennecke of Embecosm has been working with INRIA to update MILEPOST, supported by HiPEAC. Rather than use GCC plugins, we have adopted the GCC *target hook* infrastructure, which is more suited to MILEPOST. At present adding a new target hook is a very laborious task, so GCC needs to be updated to make target hook implementation simpler.

Jörn Rennecke achieved two goals during the past year:

- *MILEPOST GCC was updated to GCC 4.5.*
- The basic improvements to the target hook infrastructure were accepted into GCC mainline.

The Future: MILEPOST II

The target hook improvements to GCC are only a first step. MILEPOST also needs a robust multi-target infrastructure in GCC. Having achieved these changes to GCC, we must modify MILEPOST to use them.

MILEPOST also needs comprehensive tutorial and porting documentation, so it can easily be installed using standard tools, ported to new architectures and run with private databases.

We have outlined a new three phase project to achieve these goals.

1. Phase One will develop the new infrastructure needed in GCC and modify MILEPOST to use it over six months
2. Phase Two will push those changes through the GCC adoption process, a part-time effort of up to two years.
3. Phase Three will run concurrently with Phases One and Two making MILEPOST easier to use, and demonstrating its portability on partner's architectures.

Embecosm is able to provide the resource to do the work, and will underwrite a proportion of the cost of that resource. However we need a lead technology partner, with an investment of up to €70k over the 2-3 year lifespan of the project and other technology partners with an investment of up to €30k during Phase Three of the project.

We are in discussions with HiPEAC and INRIA over matched funding to reduce the contribution needed from the partners.

A formal announcement about this project will follow shortly. In the meantime, organizations may express their interest by contacting Dr Jeremy Bennett at Embecosm, jeremy.bennett@embecosm.com.

A more detailed version of this report is available on Embecosm's website at www.embecosm.com/articles/ear4/milepost-update.pdf.

Jeremy Bennett

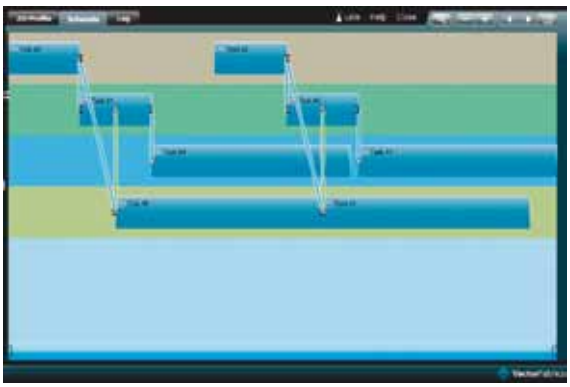
Vector Fabrics

Parallelizing C Code for Embedded Systems

The need for more speed with less power means we are no longer increasing clock frequencies. Instead, we are keeping frequencies constant, but increasing the amount of things that can be done at the same time. It is all about going parallel, especially within embedded systems.

Parallelizing C programs is particularly challenging because pointers are allowed to run rampant, creating data dependencies that cannot be analyzed statically. Whatever C's faults, however, the fact remains that C is extremely prevalent in embedded systems and shows no signs of going away. The problem cries out for tools to help make sure that multi-threaded programs work correctly.

vfAnalyst is the first of a series of tools created by startup Vector Fabrics to address this challenge. This tool, through a combination of static and dynamic analysis, identifies and characterizes the dependencies inherent in a program. This allows two key possibilities: you can use the insights gained to change the program to make it more amenable to parallelization, and you can make decisions as to how best to parallelize the program.



Application scheduling view

code or the algorithms will allow you to massage the program into something more manageable.

Given a program that can be parallelized, there are typically many ways to do that. vfAnalyst works not only by displaying the dependencies, which give visual cues as to where to break the code, but also by showing the cost and benefit of various loop parallelization strategies. The key here is that vfAnalyst is not a magic intelligent tool that can figure out how best to parallelize the program on its own – no tool can do that. Instead, it gathers information that is very hard to obtain by hand so that you can make the decisions.

Because this is a cloud-hosted tool accessed through a Flash browser application, you can view your program in some novel ways. In one view, you see a kind of 3D-looking display of the functions and loops in the program.

Another view shows an ASAP (or eager) schedule for code that has been parallelized. You can explicitly see the data dependencies as they go back and forth between the different threads.

3D View of the program structure (upper part) and data dependences (bottom): red dependencies create a serious challenge for partitioning; green ones can be managed by adding streaming channels after partitioning

In the first case, if a program is poorly designed, it will have numerous intertwining data dependencies, making it difficult to find places where you can pull it apart. Restructuring the

This visibility into your program – abstracting out lots of unnecessary detail and making the code accessible even if you did not write it and do not know it – has obvious utility for developers trying to parallelize existing sequential code and for re-optimizing programs that have already been parallelized.

But it is also useful to architects who are simply playing with high-level functions that have not been written yet. The basic data interactions can be sketched and run, even with dummy algorithms, to determine the most efficient way to assemble a parallel system while minimizing overhead for moving data between threads.

Vector Fabrics will be building on this basic technology as new capabilities are added. The goal is to reduce the time and work it takes to parallelize code and to make the process less error-prone. By focusing developers on what they do best – evaluating options and making decisions – the hope is that engineers get to spend more time doing the interesting part of the work even as they get their projects done in less time and with higher quality.

You can try out vfAnalyst at www.vectorfabrics.com.

Bryon Moyer
bryon@vectorfabrics.com

ACACES 2010 Report

Vivek Sarkar during his lecture at ACACES

This summer I was lucky to spend one week at ACACES, the HiPEAC summer school that was held at the La Mola conference center some 30 km North-West from Barcelona, Spain.

To set the tone for this report, I say that ACACES stands for learning, socialization, and relaxation.

I decided to break the dependency between my flights and the ACACES buses. Also, I wanted to explore beaches in Barcelona. So I stayed a couple of nights in Barcelona before and after the summer school. In other words, ACACES was a very good reason to pay a visit to Barcelona!

The conference centre was located in the middle of nowhere, and it was rather hot outside all the time, so everybody stayed focused on classes and socialization. Conveniently, the conference center was all ours.

I liked the architecture and design of the venue and rooms. They could have easily been the focus of an article in the Wallpaper magazine.

The classes were excellent. You can look up the website of ACACES 2010 for a detailed list and description of them. I personally enjoyed those I attended: Multicore Programming Models and Their Compilation Challenges, Variation-Aware Processor Design, FPGA-Based Reconfigurable Computing, and How to Transform Research Results Into a Business. What made the summer school special was the style of presenting the information. It was easy to comprehend the basics given on the first day and then follow the line throughout the week. Obviously, decent attention was required, just like in any learning process. Occasional questions from the lecturers made it even more interesting and involving. The handouts were handy, the classrooms were nice, the chairs were comfortable. Moreover, shortly after the summer school, all the materials were made available for download.

To maintain the sugar and caffeine levels throughout the day, lavish coffee breakes were provided.
Another thing that I really appreciated was abundance of free still water: there were refrigerators on each floor, always stuffed with bottles.

In addition to the classes, there was a poster session. I did not present a poster, so had the freedom and time to move around and interrogate others. That was extremely interesting and informative.

There were two official evening outings to the city during the week. I did not attend them – prefered the dinners at the summer school. I enjoyed the privacy of those two evenings and exceptionally nice chats with others who stayed in.

Here I have to explicitly say that the meals were awesome. I got all the fruits, meat, wine and cakes I needed to tick the checkbox against "food" on my satisfaction list. As far as I am concerned, the vegetarian menu was good too.

To burn excessive calories, I went to the on-site gym and refreshing outdoor swimming pool.

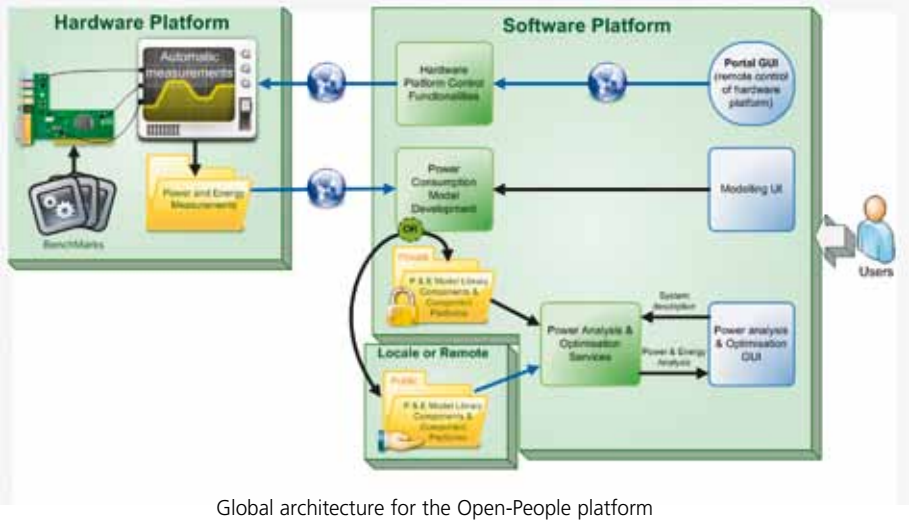
Although swimming was officially not allowed after 8pm, the final evening of the summer school was an exception: a pool party was thrown after the closing barbecue. That was a perfect time for pictures and networking without business cards.

To conclude with, ACACES was a great experience during the great summer of 2010. I learned a lot, met many sharp people, and recharged my solar batteries.
It was my first HiPEAC summer school; the organizers set a high standard. I look forward to coming back next year and am sure that my expectations will be met.

Dmitry Knyaginin

(Open-People) ANR Project
Open Power and Energy Optimization Platform and Estimator

Website: www.open-people.fr
Partners:
Lab-STICC
(University of South Brittany)
Cairn-INRIA Rennes Bretagne Atlantique
(University of Rennes 1, Lannion)
INRIA-Lille
(University of Valenciennes)
INRIA-Nancy, LEAT
(University of Nice)
Thales (Colombes)
InPixal (Rennes)



Global architecture for the Open-People platform

Designing low power complex electronic devices is now a key challenge for corporations in a number of electronic domains. The motivations, which lead designers to consider low power designs, are multiple: increase lifetime, increase use time without recharge of battery, limit battery capacity, limit temperature, etc.
Unfortunately, there are currently neither methodologies nor tools available to obtain an accurate estimation of consumption of a complete system at different levels of description, i.e. at different design steps.

The Open-People project aims at providing a complete platform

- to allow rapid power/energy estimation for complex heterogeneous systems,
- to test different optimizations in order to significantly reduce the power consumption of the system.

The goal of the Open-People platform is to provide an access to hardware execution boards (processor, dsp, fpga, logic analyzers, etc.) and the control of power estimations and optimizations.

Through a secured web portal, the platform provides an access to the power measurements and helps the designer to define models of energy consumption for hardware and software components of a complete system. These models can be included in the component library in order to be used for estimation and optimization design steps. At the end of Open-People project, the platform will propose a set of optimization tools at different levels of description and/or for the different target boards (architectural optimizations, operating system optimizations, etc).

Various research and development works are currently done in the Open-People project. These works include the definition of new methods and tools to model the different components of a heterogeneous system architecture: processors, hardware accelerator, memories, reconfigurable circuits, operating system services, IP blocks, etc. For reconfigurable system, the dynamic reconfiguration paradigm will be modeled to estimate how this feature can be used by operating system to reduce the energy consumption. Furthermore, this project studies how the complete estimation and validation can be ensured for very complex system in a small simulation time.

The software platform can be used in standalone or connected mode, to drive estimation and optimization and is coupled to an automated hardware platform for physical measurements. This hardware platform is also accessible through an Internet portal, allowing to remotely realize the measurements needed to build models for new components. In addition, a library of benchmarks will be proposed, to help building models for new components and architectures.

The project is open to new collaborators, which are interested by using the platform or by contributing with some power consumption component models or by methodologies and tools for estimation and/or optimizations.

The Open-People project is funded by the French ANR framework (2009-2012).

Daniel Chillet

9 Million Euros for Invasive Computing

Coordinator:
Professor Dr.-Ing. Jürgen Teich
University of Erlangen-Nuremberg
Website: www.invasic.de
Partners:
University of Erlangen-Nuremberg
Karlsruhe Institute of Technology
Technische Universität München,
Duration: 4 years
Start: July 1, 2010

In July 2010, the German Research Foundation (DFG) has established the new Transregional Collaborative Research Centre (TCRC) 89, "Invasive Computing", coordinated by the speaker Jürgen Teich, head of the Chair of Hardware/Software Co-Design at the Friedrich-Alexander-University Erlangen-Nuremberg. In collaboration with colleagues from the Karlsruhe Institute of Technology and the Technische Universität München, the researchers are ambitious to find new ways to design and to program novel parallel computing systems. In a first phase, TCRC is initially funded for four years and is supported with around nine million euros in this time.
"This is a fantastic success for all participating scientists", Jürgen Teich commented the decision of the DFG. "The TCRC gives us the opportunity to shape cutting-edge research in the field of new multi-processor technology and programming methods."

At TCRC, we intend to investigate a novel paradigm for designing and programming future parallel computing systems, called "Invasive Computing". Within the next few years, multi-core computers will have hundreds or even thousands of processor cores integrated in a single chip. The main idea and novelty of invasive computing is to introduce resource-aware programming support by giving programs the ability to explore and dynamically spread their computations to neighbour processors in a phase called invasion, then to execute portions of code in parallel based on the available (invasible) region on a given, heterogeneous multi-processor architecture. Afterwards, once the program terminates or if the degree of parallelism should be decreased, the program may enter a "retreat" phase, deallocates resources and continues execution, for example, sequentially on a single processor. In order to support this idea of self-adaptive and resource-aware programming, not only new programming concepts, languages, compilers and operating systems are necessary but also revolutionary architectural changes in the design of MPSoCs (Multi-Processor Systems-on-a-Chip) have to be provided in order to efficiently support invasion, infection and retreat operations involving concepts for dynamic processor, interconnect and memory reconfiguration.
This process should be fully automated and it is our goal to keep it as flexible as possible.
Additionally, it is our policy that software is able to adapt to hardware and vice versa. In this way, the computing systems can be designed to operate much more efficiently, particularly in terms of their energy budget and their computing power.

Project Coordinator
Prof. Jürgen Teich,
University of Erlangen-Nuremberg,
Germany
teich@cs.fau.de

PhD News

Power Modeling of Embedded Systems and Wireless Sensor Networks

By Daniel Schmidt
(schmidt@eit.uni-kl.de)
Advisor: Prof. Dr. Norbert Wehn
TU Kaiserslautern, Germany
July 2010

The technological progress of the past decade has triggered a tremendous growth of the market of mobile embedded systems. This market poses numerous demands, especially small size, very low cost, and long battery life time. Hence, energy efficiency is a key design challenge and design optimization for this goal has to be part of all system design phases. This thesis deals with the creation of accurate power models to be used in early design phases for different kinds of embedded systems. In a purely measurement-based approach we show the need of system simulation and optimization and identify many of the common assumptions of power consumption based on theoretical models as overly simplified and optimistic.

We present XEEMU, a power simulator for the Intel XScale, a typical representative of an embedded systems microprocessor. The power model and behavioral model are based on runtime and power measurements on a test platform that allows dynamic voltage scaling and SDRAM power management. We compare XEEMU to state-of-the-art simulators and assess aggressive SDRAM power management strategies. We show that previously published SDRAM power models are inaccurate

and propose a new model, which is integrated into XEEMU. XEEMU proves to be the most accurate simulator and, to the best of our knowledge, the only system simulator including a precise SDRAM power model.
Wireless sensor networks (WSN) are the enabling technology for e.g. Ambient Intellgence (AmI) systems. While early research always promoted to reduce the transmission power and range of the sensor nodes, we show that the energy consumption can be effectively reduced by the application of forward error correcting (FEC) channel codes. This hypothesis is based on a detailed analysis of the power consumption of typical nodes and application scenarios. We are also the first to give experimental data on the efficiency of FEC codes in WSNs.

Architectural Support For Parallel Computers with Fair Reader/Writer Synchronization

By Enrique Vallejo
(enrique.vallejo@unican.es)
Advisors: Dr. Ramon Beivide and Dr. Fernando Vallejo
University of Cantabria, Spain
June 2010

Technological evolution in microprocessor design has led to parallel systems with multiple execution threads. These systems are more difficult to program and present higher performance overheads than the traditional uniprocessor systems, what may limit their performance and scalability. These overheads are due to the synchronization, coherence, consistency and other mechanisms required to guarantee a correct execution.

Parallel systems require a deeper knowledge of the system from the programmer in order to achieve good performance and scalability. Traditional parallel programming has been based on synchronization primitives such as barriers, critical sections and reader/writer locks, highly prone to programming errors. Transactional Memory (TM) is a relatively recent proposal that seeks to remove the synchronization problems from the programmer. However, many TM systems still rely on reader/writer locks, and would get benefited from an efficient implementation.

This thesis presents new hardware techniques to accelerate the execution of such parallel programs. We propose a Hybrid TM system based on reader/writer locks, which minimizes the software overheads when acceleration hardware is present, but still allows for correct software-only execution. The fairness of the system is studied, and a mechanism to guarantee fairness between hardware and software transactions is provided. We introduce a low-cost distributed mechanism named the Lock Control Unit to handle fine-grain reader-writer locks. Finally, we propose an organization of a parallel architecture based on Kilo-Instruction Processors, which helps to simplify the consistency model while allowing for high performance thanks to the speculative large instruction window.

Reducing Memory Latency by Improving Resource Utilization

By Marius Grannaes
(grannas@idi.ntnu.no)
Advisor: Prof. Lasse Natvig
Norwegian University of Science and Technology, Norway
June 2010

Integrated circuits have been in constant progression since the first prototype in 1958, with the semiconductor industry maintaining a constant rate of miniaturisation of transistors and wires. Up until about the year 2002, processor performance increased by about 55% per year. Since then, limitations on power, ILP and memory latency have slowed the increase in uniprocessor performance to about 20% per year. Although the capacity of DRAM increases by about 40% per year, the latency only decreases by about 6 - 7% per year. This performance gap between the processor and DRAM leads to a problem known as the memory wall.
The main goal of this thesis is to improve system memory latency by leveraging available resources with excess capacity. The interaction between the memory controller and the prefetcher is especially important, because of the complex 3D structure of modern DRAM. Utilizing open pages can increase the performance of the system significantly. Memory controllers can increase bandwidth utilization and reduce latency at the same time by scheduling prefetches such that the number of page hits are maximized. In addition, when memory subsystem resources are shared, one core might interfere with another core's execution by delaying memory requests or displacing useful data in the cache.
Prefetching predicts what data is needed in the future and fetches that data into the cache before it is referenced. This dissertation presents a technique for generating highly accurate prefetches with good timeliness called DCPT. DCPT uses a table indexed by the load's address to store the delta history of individual loads. Delta correlation is then used to predict future misses. DCPT-P extends DCPT by introducing L1 hoisting which moves data from the L2 to the L1 to further increase performance. In addition, DCPT-P leverages partial matching which reduces the spatial resolution of deltas to expose more patterns.

On the Programmability of Heterogeneous Massively-Parallel Computing Systems

By Isaac Gelado (igelado@ac.upc.edu)
Advisors: Prof. Dr. Nacho Navarro and Prof. Dr. Wen-mei W. Hwu
Universitat Politecnica de Catalunya
July 2010

This thesis deals with the programmability problems due to separate physical memories for CPUs and accelerators, which are key to accomplish high performance. These separate memories are presented to application programmers as disjoint virtual address spaces, which harms programmability in two different ways. First, extra code is needed to transfer data between system and accelerator memories. Second, data structures are double-allocated in both memories, which results in using different memory addresses (pointers) in CPU and accelerator code to reference the same data structure.

This thesis proposes a programming model that integrates accelerator and system memories into a single virtual address space, allowing the CPU code to access accelerator memories using regular load/store instructions. Moreover, because a single copy of data structures exists in the application virtual address space, applications can use the same virtual memory address (pointer) to access such data structures.

The Non-Uniform Accelerator Memory Access (NUAMA) architecture is proposed as an efficient hardware implementation of this programming model. This architecture incorporates mechanisms to buffer and coalesce memory requests from the CPU to the accelerator memory to reduce the performance penalty produced by long-latency memory accesses to accelerator memory.

The Asymmetric Distributed Shared Memory Model (ADSM) is also introduced in this thesis, as a run-time system that implements the proposed programming model. In ADSM, the CPU code can access all memory locations in the virtual address space, but accelerator code is restricted to access those memory addresses mapped to the accelerator physical memory. This asymmetry allows all required memory coherence actions to be executed by the CPU, which allows the usage of simple accelerators.

Finally, this thesis introduces the Heterogeneous Parallel Execution (HPE) model, which allows a seamless integration of accelerators in the existent sequential execution model offered by most modern operating systems. The HPE model introduces the execution mode abstraction to define the processor (i.e., CPU or accelerator) where the application code is being executed.

Multithreaded Programming and Execution Models for Reconfigurable Hardware

By Enno Lübbers (enno.luebbers@uni-paderborn.de)
Advisor: Prof. Dr. Marco Platzner
University of Paderborn, Germany
March 2010

Rising logic densities together with the inclusion of dedicated processor cores have increasingly promoted reconfigurable logic devices, such as FPGAs, from traditional application areas such as glue logic, emulation, and prototyping to powerful implementation platforms for complete reconfigurable systems-on-chip.

However, traditional design techniques that view specialized hardware circuits as passive coprocessors are ill-suited for programming the combination of fast CPU cores and fine-grained reconfigurable logic within these reconfigurable computers. In particular, the programming models for software (running on an embedded operating system) and digital hardware (synthesized to an FPGA) lack commonalities which hinders design space exploration and severely impairs the potential for code re-use.

In this thesis, we present a novel programming model and run-time infrastructure for multithreaded programming of reconfigurable logic devices called ReconOS. It is based on existing embedded operating systems and extends the multithreaded programming model – already established and highly successful in the software domain – to reconfigurable hardware. Using threads and common synchronization and communication services as an abstraction layer, ReconOS allows for the creation of portable and flexible multithreaded HW/SW applications for CPU/FPGA systems.

ReconOS uses a pervasive programming paradigm to model the interaction of interdependent executable units (threads), regardless of their execution domain (hardware or software). This common programming model has several benefits:

- First, it enables seamless run-time sharing of computational resources not only for software threads on a microprocessor, but also for hardware threads within the reconfigurable fabric. This is accomplished by utilizing partial reconfiguration together with non-preemptive and cooperative scheduling mechanisms.
- Second, it improves code reusability and portability of individual threads and the execution environment as a whole; our reference implementation is based on modern platform FPGAs and able to target different software host operating systems (e.g., eCos and Linux) and processor architectures (e.g., PowerPC and MicroBlaze).
- Third, it allows the transparent integration of dedicated hardware accelerators and established software frameworks, thereby providing access to a vast library of existing software libraries, network protocol stacks, and legacy code.

ReconOS is open-source and available at www.reconos.de.

Upcoming Events

The 43rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO-43)
4–8 December 2010, Atlanta, Georgia http://www.microarch.org/micro43/

The 6th Int. Conference on High-Performance and Embedded Architectures and Compilers (HiPEAC 2011),
24-26 January 2011, Heraklion, Crete, Greece, http://www.hipeac.net/

The Asia and South Pacific Design Automation Conference 2011 (ASP-DAC 2011)
25-28 January 2011, Yokohama, Japan, http://www.aspdac.com/aspdac2011/

The 38th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages (POPL 2011),
28-28 January 2011, Austin, Texas, USA http://www.cse.psu.edu/popl/11

The 17th IEEE International Symposium on High-Performance Computer Architecture (HPCA-17),
12-16 February 2011, San Antonio, Texas, USA http://hpca17.ac.upc.edu/

The 24th International Conference on Architecture of Computing Systems (ARCS 2011),
22-25 February 2011, Lake Como, Italy, http://conferences.dei.polimi.it/arcs2011/

The Design, Automation, and Test in Europe Conference (DATE 2011),
14-18 March 2011, Grenoble, France, http://www.date-conference.com/

The 16th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2011), 5-11 March 2011, Newport Beach, CA, USA, http://asplos11.cs.ucr.edu/

The 14th IEEE International Symposium on Object/component/service-oriented Real-time distributed computing (ISORC 2011), 28-31 March 2011, Newport Beach, CA, USA, http://dream.eng.uci.edu/isorc2011/

The IEEE/ACM International Symposium on Code Generation Optimization (CGO 2011),
2-6 April 2011, Chamonix, France, http://www.cgo.org/cgo2011/workshops.html

The 5th ACM/IEEE International Symposium on Networks-on-Chip (NOCS 2011),
1-4 May 2011, Pittsburgh, Pennsylvania, USA, http://www.nocsymposium.org

The 32nd ACM SIGPLAN conference on Programming Language Design and Implementation (PLDI 2011),
4-8 June 2011, San Jose, CA, USA, http://pldi11.cs.utah.edu/

The 38th International Symposium on Computer Architecture (ISCA 2011),
4-8 June 2011, San Jose, CA, USA, http://isca2011.umaine.edu/

The 48th Design Automation Conference (DAC2011),
5-10 June 2011, San Diego, CA, http://www2.dac.com/dac+2011.aspx

Contributions

If you are a HiPEAC member and would like to contribute to future HiPEAC newsletters, please contact Rainer Leupers at **leupers@iss.rwth-aachen.de**

HiPEAC Info is a quarterly newsletter published by the HiPEAC Network of Excellence, funded by the 7th European Framework Programme (FP7) under contract no. IST-217068.
Website: http://www.HiPEAC.net
Subscriptions: http://www.HiPEAC.net/newsletter