HIGH PERFORMANCE AND EMBEDDED ARCHITECTURE AND COMPILATION

Project Acronym: *HiPEAC*
Project full title: High Performance and Embedded Architecture and Compilation
Grant agreement no: *ICT- 217068*

DELIVERABLE 2.1

RESEARCH REPORT I

1. Introduction

The WP2 research programme consists of a number of clusters and task forces that coordinate part of the HiPEAC strategic research agenda.

The proposed clusters are mostly non-overlapping, and where there is overlap, it has been created intentionally to stimulate inter-cluster collaboration. The topics of the clusters are broad enough to integrate the research topics of most members in one or more clusters. The aim is to gradually steer and focus the research of the cluster members on the topics of the HiPEAC strategic research agenda, as detailed in the HiPEAC roadmap.

On top of the cluster structure there are several cross-cutting topics like power, application focus, training, reliability and availability. These cross-cutting topics are dealt with by task forces. These task forces periodically meet, coordinate and organize activities to further develop the crosscutting topics. The task forces are also involved in the development of the roadmap.

Each cluster is coordinated by an academic member institution. The coordinating partner has a (senior) researcher to assist him in coordinating and animating the cluster.

The objectives of the clusters and task forces are:

- **Building a cluster/task force community.** Attract all excellent researchers in Europe and beyond to participate in the cluster research (including industrial researchers with special emphasis on researchers from SMEs and application companies). Every cluster must continuously look for companies willing to actively collaborate in a cluster. We especially welcome SMEs and application companies. Only HiPEAC members are entitled to have their cluster expenses being reimbursed by HiPEAC. Non-HiPEAC members are still welcome to collaborate or to attend the general cluster meetings, but their travel expenses will not be reimbursed.
- **Coordinating joint research and animate cluster/task force activities**. Organize the general cluster meetings, stimulate collaboration between the cluster members, and monitor the progress indicators for the clusters. Set up internal and external communication channels for the clusters.
- **Doing research in the domain of the cluster/task force**. Coordinating and animating a HiPEAC cluster creates a unique opportunity for an ambitious researcher to build a network and to create visibility. The cluster coordinator and the HiPEAC researcher working in the cluster should take a leading role in the domain of the cluster.
- **Managing cluster membership.** HiPEAC membership is proposed by the cluster coordinators based on quality criteria and on actual collaboration in the cluster. In order to keep control over the HiPEAC membership, proposed members must be approved by the HiPEAC steering committee. Yearly, the cluster coordinator will provide the HiPEAC steering committee with a list of active cluster members. Non-active members will be dropped from the cluster membership.
- **Stimulating links with non-European research groups.** The clusters should not work in isolation, but maintain research relationships with non-European centres of excellence in its domain. Especially collaboration with important research centres in the USA (via NSF) is considered important. These members will be called *affiliate members* in the rest of this project.

- **Stimulating publications in top conferences and journals.** An important metric for a cluster is the publication outcome. Clusters represent communities specialized in a particular topic, and the cluster collaboration should lead to co-authored publications in top- conferences and in top-journals.
- **Stimulating serving as referee, PC-member, journal editor**. An important metric for the success of a cluster is the role of cluster members as referees, PC-members or journal editor. The cluster members should stimulate and help each other to fill in these positions.
- **Promoting young talent.** Yearly, tens of PhD-students graduate and start a research career. For these young researchers, being able to rely on a network like HiPEAC can mean a lot. Clusters should promote young talented researchers in the outside world. Positive actions are: recommending them as reviewers, inviting them for program committees or as reviewer for journals, inviting them to co-organize a workshop.
- **Contributing to the HiPEAC roadmap**. One of the tasks for a cluster (including the affiliate members) is to reflect on its own roadmap, and to contribute to the regular updates of the HiPEAC roadmap. Since the union of all clusters is covering the HiPEAC strategic research agenda, the union of the cluster roadmap results in the HiPEAC roadmap.
- **Contributing to the ACACES summer school**. The clusters should yearly propose a few names of top-ranked teachers for the ACACES summer school, and stimulate their members to attend the summer school.
- **Organizing public events**. All clusters should be visible and spread their excellence publicly. Per cluster there should be regular public events. This can be a workshop, a tutorial, a course or any other event that creates visibility for the cluster and for the research activities of the cluster.
- **Reporting on the cluster activities**. The cluster coordinator is responsible for the yearly scientific and administrator reporting on the cluster activities.

The general activities of the clusters are the cluster meetings, held during the HiPEAC Computing Systems Weeks, or in conjunction with another main HiPEAC activity such as the conference. In 2008, 3 cluster meetings were held, in Goteborg, in Barcelona and in Paris.

Between 150 and 200 people attended these meetings, making them very successful events. Clusters have between 36 and 64 members. All clusters have industrial members. The extensive publication output, of which a great part joint papers, is the result of the collaboration within the cluster.

Next to the general cluster meetings, clusters also organize additional activities: bilateral meetings between members, international workshops and seminars, brainstorming on research challenges in the domain of the cluster, etc. The cluster members are stimulated to submit joint project proposals, participate in international program committees, etc. The total cluster budget is distributed over the different clusters, and managed by the cluster coordinator.

Most HiPEAC members attend more than one cluster/task force. The following table shows how many cluster members for a particular cluster are also registered in the other clusters/task forces.

	T2.1 Multi-core architecture	T2.2 Programming models and operating systems	T2.3 Adaptive compilation	T2.4 Interconnects	T2.5 Reconfigurable computing	T2.6 Design methodology and tools	T2.7 Binary translation and virtualization	T2.8 Modelling and Simulation Framework	T2.9 Compilation platform	Task Force on Applications	Task Force on Education and training	Task Force on Low power	Task Force on Reliability and availability
T2.1 Multi-core architecture	63	25	17	9	20	18	15	18	17	7	4	7	11
T2.2 Programming models and operating systems		55	19	2	10	9	14	15	20	15	3	1	5
T2.3 Adaptive compilation			50	0	10	10	21	21	31	5	3	2	4
T2.4 Interconnects				34	6	6	0	4	0	1	2	6	5
T2.5 Reconfigurable computing					40	18	8	11	10	3	1	4	4
T2.6 Design methodology and tools						42	6	21	8	2	1	6	4
T2.7 Binary translation and virtualization							41	13	18	4	2	1	6
T2.8 Modelling and Simulation Framework								58	17	6	3	7	5
T2.9 Compilation platform									46	5	1	2	3
Task Force on Applications										22	2	1	1
Task Force on Education and training											7	1	1
Task Force on Low power												15	5
Task Force on Reliability and availability													26

The cluster research report gives an overview of the cluster achievements in 2008, activities and future plans of the research clusters:

1. Multi-core Architecture (Chalmers, Per Stenström)
2. Programming models and operating systems (BSC, Mateo Valero)
3. Adaptive compilation (Edinburgh, Michael O'Boyle)
4. Interconnects (Forth, Manolis Katevenis)
5. Reconfigurable Computing (TU Delft, Georgi Gaydadjiev)
6. Design methodology and tools (RWTH Aachen, Rainer Leupers)
7. Binary translation and virtualization (UGent, Koen De Bosschere)
8. Simulation platform (INRIA, Olivier Temam)
9. Compilation platform (INRIA, Albert Cohen; IBM, Ayal Zaks)
10. Reliability and availability task force (University of Cyprus, Yanos Sazeides)
11. Applications task force (UPC, Nacho Navarro)
12. Low power task force (University of Patras, Stefanos Kaxiras)
13. Education and training task force (FORTH, Manolis Katevenis)

2. Multi-Core Architecture Cluster

2.1. Description of scientific issues

The shift towards multi-core architectures poses several challenges for computer architects. With each new technology generation, there will be a significant increase of the number of transistors. The question is how computer architects can establish computational structures that can transform the increase in transistors into an equal increase in computational performance efficiency. This challenge must be attacked from several fronts, namely, the basic architecture of each processor (core) to increase single thread performance, via the architecture of the memory system, to a holistic approach to support emerging programming models that aim at reducing the efforts the programmers have to invest in designing software amenable to multi-core architectures. As a consequence, this cluster is engaged in advancing state-of-the-art in the following core areas of (multicore) computer architecture:

- Processor architecture
- Memory hierarchy
- Technology impact on architecture
- Application impact on architecture
- Architecture support for parallel programming models
- Performance/power evaluation

As a preparation for the first cluster meeting that was held in Goteborg on January 30, 2008, an initial poll was done as far as interests in different topics on multicore architecture issues. About 50 responses were received of which about 40 were from academia and about 10 from industry. This poll resulted in the following distribution across different topics:

Processor architecture	28
Memory architecture	26
Application impact	23
Programmability support	19
Perf/power evaluation	18
Technology impact	14

It has become clear that the initial sets of topics are a good starting point for the cluster.

The cluster on multicore architectures has strong links to almost any of the other clusters. Most notably, the cluster has a strong link to the Programming Models and Operating System Cluster because future multicore architectures must provide a productive abstraction to the software and must support emerging programming models. To stimulate such inter-cluster discussions, we have over the year arranged three international workshops that are described below.

There are of course links to other clusters as well. For example, interconnects which play a fundamental role in future multicore architectures is dealt with in a separate cluster. Another architectural approach that is important and that join a significant community is reconfigurable

computing. This topic is also dealt with in another cluster. Compilation issues for multicore architectures and especially as far as simplifying the efforts in parallelization, which is dealt with in the Adaptive Compilation cluster, have also strong links to the multicore architecture cluster. It is a topic for future activities to promote interaction among clusters by thematic activities and will be discussed later.

2.2. Summary of activities

During 2008, the cluster activities can be summarized as follows:

- Compilation of research activities in the cluster
- Formulation of a roadmap
- Stimulation of research interaction
- Stimulation of creating research consortia for upcoming calls

Three cluster meetings have been arranged during the year. Apart from the inaugural meeting in Goteborg on January 27, 2008, which attracted close to a hundred attendees, the second meeting was held in Barcelona during the HiPEAC Computing Systems week. The purpose of that meeting was twofold. A first purpose was to give all members of the cluster a chance to present what they are doing by giving a 5-minute presentation. As many as 18 presentations were given spanning the following topics: multicore architecture, locality management, power/performance issues, application impact on architecture. So indeed, the active presentations matched the set of topics for the multicore architecture cluster quite well. The second purpose of the cluster meeting was to have a roundtable discussion on the roadmap. Based on Dr. Duranton's questionnaire sent out to each cluster leader, a discussion took place that distilled the major challenges facing the multicore architecture area.

A task force to prepare the roadmap for the multicore architecture cluster was put together with the following individuals as members: Andre Seznec (INRIA), Sami Yehia (Thales), Emre Ozer (ARM), Yiannakis Sazeides (Univ. of Cyprus), Stefanos Kaxiras (Univ. of Patras), Alex Ramirez (Barcelona Supercomputing Center/UPC). This group distilled the first set of challenges for the roadmap which are summarized below:

- The Scalability, Power, Reliability, and Verification Challenge
- The Parallel Programming Challenge
- The Memory System Challenge
- The Heterogeneity Challenge
- The Resource Management and Instrumentation Challenge

The third cluster meeting was organized in the HiPEAC Computing Systems Week in Paris on November 27 2008. The main theme of this meeting was to present the new FP7 Call on Computing Systems that had been announced a couple of days before. Some important take-aways from the call are that parallelization and reliability issues are stressed is relevant for many researchers in the cluster. It is however important that a successful consortium addresses **all** aspects from the application down to the architecture which is why successful consortia should be formed by groups from different HiPEAC clusters. It was encouraged that work on forming strong consortia should be initiated immediately.

To foster interaction between the multicore architecture and the programming model communities, three international workshops have been organized by the cluster leaders of the multicore architecture cluster (Per Stenström, Chalmers) and the programming model and operating system cluster (Osman Unsal and Eduard Ayguade of Barcelona Supercomputing Center). The first one, MULTIPROG 2008 took place at the 3rd International Conference on High-Performance and Embedded Architectures and Compilers in Goteborg on January 27, 2008. It attracted close to a hundred attendees and featured a keynote by Jesus Labarta as well as a set of technical sessions spanning topics across both clusters. A second international workshop – the Barcelona Multicore Workshop (BMW 08) – was arranged in Barcelona during the HiPEAC Computing Systems week with more than a hundred attendees.

This workshop featured inspiring invited talks by international leaders in the community including leaders of the newly established centres at Stanford, Berkeley, and Illinois. Also, a panel with the theme "50 Billion Transistor Chips: What should the Hardware provide?" moderated by Yale Patt and "What Terrifies You More: Multicore Hardware or Software?" By Daniel Reed were arranged. Finally, based on the success of the first MULTIPROG workshop, a second edition will take place at the 4th HiPEAC Conference in Paphos, Cyprus on January 25. Apart from technical sessions spanning architecture as well as programming model issues, it also features an invited talk on the FET SARC project as well as a panel on multicore programming issues moderated by Per Stenström, Chalmers.

This cluster has over the year been led by Per Stenström Chalmers. He is conducting research on memory system and programmability issues for multicore computers in the framework of two European projects (the FET SARC IP and the FP7 VELOX project). Chalmers finalized the recruitment of a new faculty member who will help Per to animate and work on cluster activities in 2009. The new faculty member started to work on November 1.

2.3. Inter-cluster collaborations

The VELOX project aims at implementing transactional memory all the way from the programming model to the architecture level. It is a EU FP7 project involving partners from the Multicore Architecture Cluster (Per Stenstrom, Chalmers) and the Programming Model and Operating System Cluster (Osman Unsal, Eduard Auyguade, and Mateo Valero, BSC)

In the context of the EU FET SARC project there is collaboration between the following clusters: Multicore Architecture (Stenstrom, Chalmers), Programming Models (Martorell, UPC/BSC), Compilation (O'Boyle, Edinburgh), Interconnect (Katevenis, FORTH), and Reconfigurable Computing (Gaydadyiev, TUD)

2.4. Future plans

The activities over the first year have successfully joined a community around the central topics. However, future activities must address several important actions:

- It must foster more collaboration inside the cluster
- It must create tangible links with other clusters

To this end, we will organize the cluster meetings differently. We will arrange several parallel thematic roundtable discussions in smaller groups. It is simply not feasible to have discussions during sessions with close to a hundred attendees. In a similar way, cross-cluster discussions can be promoted by arranging round-table discussions around central themes for two clusters. For

example, a discussion on architectural support for programming model X could involve groups of researchers from three clusters (multicore architecture, adaptive compilation, and programming models and operating systems).

2.5. Progress Indicators

- # active members: 64
- # companies involved: 8
- # publication output: 68
- # joint papers: 16 (3 industry)

3. Programming Models and Operating Systems Cluster

3.1. Description of scientific issues

Despite all the experience on parallel programming gathered in the last decades and the experience in implementing runtime systems to support parallel execution, the productive expression and efficient exploitation of parallelism for current multicore architectures are not obvious tasks.

Many-/multi-core architectures currently include tens of cores and will include hundreds of them in the near future, with memory organizations that will probably escape from the easy-to-program shared-memory address space. In addition, resource specialization is considered as the path to built power efficient architectures, leading to heterogeneous architectures. New programming paradigms and OS/hardware support to enable effective programming of these architectures with 100+ cores, in terms of scalability and portability, are necessary.

In the cluster the research has been organized along the following 5 directions:

1. Novel programming models (or evolutions of the current ones) for future homogeneous/heterogeneous many-/multi-core architectures. Expressiveness and productivity are equally important. Accelerators (SIMD units, GPUs, FPGAs,...) have been considered.
2. Compiler and runtime support to parallel programming models, hiding as much as possible the particularities of the target architecture to the programmer.
3. New architectural features to support the programming model and its runtime implementation: thread creation, task off-loading, transactional memory.
4. OS support for architectures including heterogeneous cores (CPU, SIMD, GPU), reconfigurable cores (FPGA),... that guarantees fast task creation and synchronization in the many-/multi-core environment, provides efficient task scheduling and allocation mechanism, thread management for power, temperature, and reliability, provides QoS in real time systems,...
5. Benchmarks and methodologies to be used as a basis for the evaluation of ideas proposed in our community.

3.2. Summary of activities

The activities carried out in the cluster are classified in three groups:

1. Joint research activities originated from the interaction within the cluster. The most significant ones are:
   (a) Exploration and evaluation of currently available programming models for Cell, proposing extensions that increase their efficiency, portability and programming productivity. The current set of applications written and evaluated with such programming model are matrix multiply, HPCC Stream and RandomAccess, PBPI, DiskIO, and Fixedgrid. An exhaustive analysis of some bio-informatic applications under the Cell BE architecture has been done. Groups involved: FORTH, BSC-UPC.
   (b) Implementation of taking and extensions to OpenMP 3.0 towards making easy the use of accelerators in heterogeneous environments (multicore, Cell and GPUs). New pragmas to express target device accelerators, data movement, program and data

transformations as well as alternative function implementations have been explored. Groups involved: BSC-UPC, U. Castellon.

(c) Explore and propose novel paradigms based on the exploitation of parallelism at runtime, supported with program annotations. Implementation of GPUSs, a version of StarSs targeting GPU-based systems, e.g. StarSs (BSC-UPC), TFlux (U. Cyprus), FLAME (U. Castellon).

(d) Programming models oriented towards the acceleration of applications on reconfigurable hardware and automatic hardware/software partitioning of high-performance computing applications. In addition, the support provided by custom instruction processors and operating systems has been investigated. Groups involved: U. Delft and BSC-UPC.

(e) Promotion of common research activities around transactional memory, including language extensions, libraries and hardware support. The application of transactional memory to novel domains in which the combination of TM and dataflow execution models is feasible. Groups involved: U. Manchester, U. Siena, BSC-CNS. In the context of Haskell, there has been collaboration with Microsoft Research Cambridge U.K.

(f) Promotion of common tools to support our research activities. For example, BSC-UPC's Mercurium compiler, Nanos and StarSs runtimes and TaskSim simulator.

(g) The group at KTH started a research group around primarily multicore software development issues together with researchers at Swedish Institute of Computer Science and Uppsala University. The group started the Swedish Multicore Initiative which it coordinates together with a number of other research sites in Sweden.

2. Organization of workshops on programming models and runtime support for multi-cores systems:
   - MULTIPROG-2008 in Goteborg (January 2008, in conjunction with HiPEAC-2008)
   - BMW'08 in Barcelona (June 2008)
   - MULTIPROG-2009 in Paphos (January 2009, in conjunction with HiPEAC-2009)

   In addition to these major dissemination events, three cluster meetings have taken place. The format for these meetings has been quite similar, with the purpose of detecting common topics of research. The person hired by the cluster has been involved in the organization of the three major events as well as the cluster meetings.
3. Personal research activities of the person hired by the cluster. Roberto Gioiosa focused on:

   (a) Scheduling algorithms for periodic soft-real time systems in the presence of SMT processors. Adapting existing scheduling algorithms like EDF so that they explicitly manage SMT hardware resources.

   (b) Scheduling issues for massive multithreading processors like the UltraSparc Niagara T2. Determining the overhead of the operating system and the hypervisor in massive multithreading systems.

3.3. Summary of inter-cluster collaborations

The cluster has done activities in collaboration with two other clusters and one taskforce, as follows:

1. Multi-core Architecture cluster:

- We have jointly organized two workshops (MULTIPROG-2008 and MULTIPROG-2009) in conjunction with the HiPEAC conference and the Barcelona Multicore Workshop (BMW-2008).
- Collaboration between the groups at University of Manchester (Mikel Lujan) and BSC (Osman Unsal) about hardware support for transactional memory and efficient implementation of transactional memory libraries in support for different kind of application domains. Common development of benchmarks to support the evaluation of the research proposals and implementations.
- Collaborations inside FP6 SARC project. In SARC, the collaboration is in the definition of a clear interface between the architecture and the programming model, proposing extensions to the OpenMP programming model to express the offloading of tasks to accelerators in the SARC architecture, including transparent management of local memories. A prototype implementation has been done and its evaluation with some NAS benchmarks.
- Collaborations inside FP7 Velox project. In Velox, the collaboration between Chalmers (Per Stenström) and BSC (Osman Unsal) is on efficient implementation of hardware transactional memory and hardware-supported software transactional memory the hardware.

2. Reconfigurable Computing cluster: Collaboration between the groups at the University of Delft (Georgi Gaydadjiev) and BSC (Xavier Martorell/Daniel Jimenez) about productive programming of high-performance architectures including FPGA accelerators. As a result of the collaboration extensions to the OpenMP 3.0 programming model have been proposed that express the offloading of tasks to FPGA. A prototype version of the runtime system has been implemented, which includes transparent data movement between host and accelerator and data packing and unpacking. In addition, the runtime implements a bit stream cache and a hybrid execution mode to hide the high configuration overheads of the FPGA device.
3. Taskforce on Applications: Collaboration with the taskforce leaders in order to define the requirements that the cluster needs to support the evaluation of the research ideas originated from collaborations and contribute to the HiPEAC benchmark suite.

3.4. Future plans

The planned activities in this cluster seek to create a powerful European research community for programming models and runtime environment for many-/multi-core systems.

To achieve the cluster plans to:

1. Propose novel programming models (or evolutions of the current ones) for future homogeneous/heterogeneous many-/multi-core architectures. Expressiveness and productivity are equally important. Accelerators (SIMD units, GPUs, FPGAs,...) should be considered. In this direction, the following collaborations will continue or will be started:
   a) Exploration and evaluation of currently available programming models for Cell-like architectures, proposing extensions that increase their efficiency, portability and programming productivity. FORTH (Dimitris Nikolopoulos) and BSC (Xavier Martorell).
   b) Explore and propose novel paradigms based on the exploitation of parallelism at runtime, supported with program annotations. Evaluation of these proposals is necessary, proposing a common set of programming productivity metrics and benchmarks to use. We plan to work on StarSs (BSC-UPC), TFlux (U. Cyprus) and FLAME (U. Castellon).

c) Implementation of GPUSs, a version of StarSs targeting GPU-based systems. During this next year the collaboration will focus on the source-to-source compiler and extensive evaluation not just using linear algebra library kernels. U. of Castellon (Enrique Quintana) and BSC (Rosa M. Badia).
d) The implementation of the runtime for the OpenMP 3.0 extensions for FPGA-based accelerators will continue, porting and evaluating more applications as well as extending the implementation to support multiple-FPGA devices working in conjunction with the multicore hosts. The collaboration is with the U. of Delft (Georgi Gaydadjiev) and BSC (Xavier Martorell/Daniel Jimenez) groups.

2. Explore and propose new architectural features to support the programming model and its runtime implementation (e.g. thread creation and task offloading, transactional memory). In particular:
   a) The collaboration between FORTH (Dimitris Nikolopoulos) and BSC (Xavier Martorell/Alex Ramirez) will continue, exploring possible hardware support for resolution of dependences between tasks and managing task offloading. In the same direction, collaboration with U. Cyprus (Pedro Trancoso) is planned.
   b) The collaboration between the U. of Manchester (Mikel Lujan) and BSC (Osman Unsal) in runtime-adaptable software/hardware transactional memory systems will start. This also includes the development of realistic transactional memory applications that a non-expert programmer would likely develop, which is the expected use case for the future.
   c) The collaboration between U. of Siena (Roberto Giorgi) and BSC (Adrian Cristal) will continue, investigating novel programming paradigms for transactional memory systems, expanding the range of applicability of transactional memory to other application domains and parallel structures.
3. Evaluate and propose OS support for architectures including heterogeneous cores that guarantees fast task creation and synchronization in the many-/multi-core environment, provides efficient task scheduling and allocation mechanism, thread management for power, temperature, and reliability, provides QoS in real time systems,... This collaboration will involve partners in the current FP7 Merasa project.
4. Continue the contribution to the Taskforce on Applications with new benchmarks and methodologies that could serve as a basis for the evaluation of ideas proposed in our community.
5. Form interdisciplinary (intra- and inter-cluster) research groups that prepare join project proposals on programmability and parallelism of homogeneous or heterogeneous multi-core and/or reconfigurable architectures, with a holistic approach that addresses issues related to the underlying hardware, the operating system and the system software. A number of such projects have been submitted to the FP7 Call 4 (April 2009).

3.5. Progress Indicators

- # active members: 56
- # companies involved: 7
- # publication output: 52
- # joint papers: 15

4. Adaptive Compilation Cluster

This section describes the activities of the adaptive compilation cluster during 2008. It starts by describing the issues facing the community, and then reports on the projects funded by the cluster this year. The activities undertaken by the cluster are described in section 3, and are followed by the future plans for the cluster in the next year.

4.1. Description of scientific issues

One of the key challenges of future multi-core architectures is programmability and scalability. As compilers link the code written by programmers to the underlying parallel hardware, this cluster has a pivotal role to play. It focuses on versatility by adapting the user code to the ever changing underlying hardware. It pursues a system wide perspective on adapting programs and, more generally, workloads, to both short-term architecture variation, such as cache misses, and longer-term changes, such as the increasing number of processors available. It includes the following initial research topics:

- Software adapts to hardware
    - Iterative compilation: auto-tuning/parallelising applications to new parallel hardware
    - Machine learning based compilation: using machine learning to learn new optimisation strategies for multi-cores
    - Dynamic compilation and virtual machines: combining ahead of time optimisation plus JIT compilation for Java and .NET-like applications on multi-cores
- Hardware adapts to software
    - Compiler/runtime fusion: Both make resource and scheduling decision. Combining allows static knowledge about future behaviour and previous dynamic behaviour to be exploited.
    - Resource-aware compilation. Use knowledge of the program to configure the processor to maximise ED^2. Specialise the ISA to give ASIP solutions.
    - Reconfiguration: dynamically change the processor structure based on either off-line analysis or dynamic knowledge.

4.2. Supported Projects

The first call for funding from this cluster was announced in February 2008, immediately following the HiPEAC2 Kick-off meetings in Goteborg. Since then funds have been committed to support six research collaborations within the adaptive compilation cluster. Each of these is described briefly in the following sections.

Context-Aware Optimisation and Runtime Adaptation of Sequential Libraries for Multi-core Systems

As multi-core systems have become mainstream, system and software developers face a challenge of efficiently parallelising existing sequential programs. Many programs use specialised libraries such as ATLAS, FFTW, SPIRAL, the Intel Math Kernel Library and the AMD Core Math Library, which are carefully optimised to achieve high performance on a target system. Furthermore, these libraries often include multiple versions of key functions, each specialised for a different set of

input data. The exact version to be executed can be chosen at runtime depending on some features of the input data (typically its size). These optimisation and adaption techniques, however, largely ignore possible system and program behaviour at invocation time, which is particularly important on multi-core systems where multiple threads compete for shared resources.

This research aims to evaluate iterative compilation of sequential libraries with multiple datasets under realistic system conditions. Based on this evaluation and previous research considering program multi-versioning and run-time adaptation, it aims to develop new techniques for runtime predictive adaptation, which will allow the selection of appropriate library versions depending on the 'context': i.e., system and program behaviour, as well as dynamic features of the input.

People Involved

Grigori Fursin (INRIA), Anton Lokmotov (Imperial College), Paul Kelly (Imperial College)

Using Adaptive Compilation to Produce High Performance Sparse Computations

Matrix operations appear frequently in many areas of science and engineering. Often, the solution to real life problems requires efficient computations on sparse matrices. The performance of a code which operates on sparse matrices depends on many parameters, such as the density and sparsity pattern of the input matrix, the data structures used, or the processor's cache sizes.

This research aims to explore the use of adaptive compilation to produce high performance implementations of codes working on sparse matrices. Using an iterative compilation approach, it will investigate high performance implementations of operations dealing with sparse matrices. In addition, it will explore the potential for multi-versioning: selecting which version of the operation to use at runtime depending on the features of the input.

People Involved

Georgios Goumas (Athens), José Herrero (UPC)

Split Compilation and Code Specialisation

Split compilation is a promising research direction that attempts to partition code optimisation into runtime-independent and runtime-dependent parts. As much optimisation as possible is performed in the runtime-independent phase, leaving only the bare minimum to do in the runtime-dependent stage when the execution environment is known.

Orthogonally, code specialisation takes advantage of runtime information to improve program performance by specialising parts of the application specifically for the underlying target. However, since this is performed during execution, the optimiser must decide when the extra work is worth the effort (i.e., the overheads in compilation time will be outweighed by the benefits from optimised code). By statically pre-computing the predicates that will impact the performance and embedding them in the byte code as annotations, this research plans to simplify the decision process of the just-in-time compiler. This enables the optimiser to make faster and better informed decisions about when to perform specialisation.

People Involved

Erven Rohou (INRIA), Stefano Crespi (Milano)

Split Vectorisation Using Gcc and Mono

The current tool chain for static languages suffers from a number of major deficiencies, stemming from a lack of information at pre-processing time. One of these deficiencies is a lack of (detailed) information about the target platform, and its specific SIMD capabilities. In addition to this, small changes in the SIMD architecture may require major rewriting of parts of the compiler's assembly code or the full compiler tool chain.

As explained, split compilation attempts leverage machine-independent optimisations wherever possible, deferring any machine-dependent work until more is known about the actual target. This research will investigate the use of split compilation for SIMD architectures to enable a platform-independent SIMD compiling environment.

People Involved

Albert Cohen (INRIA), Erven Rohou (INRIA), Ayal Zaks (IBM), Sami Yehia (Thales)

Value-Based Optimisation

In the past, many researchers have discovered that the majority of data passed around a processor belongs to a set of values which is both small and predictable. Using this knowledge, value-based optimisations can be developed which take advantage of the redundancy available when recurring values of data are seen. This work will harness the compiler to exploit both hardware and software techniques of frequent value reuse optimisation to improve the performance and power consumption of general purpose and embedded applications. The research will investigate the trade-offs between software and hardware techniques for frequent value-based optimisation, considering frequent values and sparsity patterns in sparse matrix vector multiplication codes.

People Involved

Andy Nisbet (Manchester Metropolitan), Yiannakis Sazeides (Cyprus), David Moloney (Movidia)

Performance Counter-Based Power and Temperature Prediction

This project builds on performance counter-based power and temperature prediction work that is being performed in Cornell. The work consists of a validated the power prediction model for an AMD four-core Phenom and an Intel Core 2 Quad Q6600. The temperature model is being actively developed.

This research has two directions. Firstly, the use of continuous optimisation systems developed in Siena will be investigated to see if they can benefit from the information gathered from the performance counters. Secondly, similar performance counters will be added to an ARM simulator to see if the approach works equally well for embedded systems.

People Involved

Sally McKee (Chalmers), Sandro Bartolini (Siena), Timothy Jones (Edinburgh)

4.3. Summary of activities

During 2008 the adaptive compilation cluster has held three meetings. The kick-off meeting, in Goteborg in January, was a great success with enthusiastic support from all aspects of the compilation community. The meeting was devoted to presentations from cluster members to allow the community to better understand the current research being performed in this cluster's area. A

discussion was also held on the future directions of the cluster and the challenges facing the community in the next 10 years.

The second cluster meeting was held in Barcelona in June. David Bernstein from IBM was invited to make a keynote speech, giving an industrial perspective on the key challenges for compiler research and the areas that IBM is interested in. Talks were also given by seven cluster members to advertise their research and identify potential collaborations for the coming year.

Finally, in Paris in November the last cluster meeting of the year was held. Again, four presentations were given by cluster members, and then we split the meeting into three groups, each focussing on a different challenge facing the community. The aim of this was to foster collaborations between the cluster members to tackle the challenges ahead. This exercise successfully produced several collaborations and requests for cluster funding to support joint projects in the coming year.

Meeting Structure

Over the year this cluster has experimented with several formats to the cluster meetings. It has been found that splitting the participants into three subgroups has created a more dynamic environment, enabling better, more focussed discussions between group members. This has received positive support from cluster members and this format will be developed and improved in the coming year.

4.4. Future Plans

In the coming year, work will continue on the research topics detailed in section 3.2. In particular split compilation, where optimisations partitioned into runtime-independent and runtime-dependent parts, will be a focus of a number of cluster members, in collaboration with the Compilation Platform cluster. This will consider code specialisation in conjunction to split compilation, as well as split vectorisation for SIMD architectures.

4.5. Progress Indicators

- # active members: 51
- # companies involved: 7
- # publication output: 44
- # joint papers: 6 (2 industry)

5. Interconnects

5.1. Description of scientific issues

Interconnection networks play a critical role in almost every aspect of modern computer systems. They are key components of high-speed network routers, large-scale parallel computers, and storage systems, while recently they are becoming a critical part of general-purpose chip multiprocessors and of large systems-on-chip.

Hardware complexity is growing at a fast pace every year both at the chip and at the system level. Practically, designers must integrate an enormous amount of components implementing – in many cases – heterogeneous functions, satisfying at the same time strict performance-energy constraints. In this rapidly changing environment, interconnects research will become more prevalent. Efficient methods are required to allow easy system plug-and-play connectivity along with new power/performance efficient communication mechanisms either for on-chip or system wide networks. Still, the room for innovation and improvement is large and new ideas are needed to solve some critical problems that arise from various application domains and new technology-imposed constraints.

The members of the Interconnects cluster are highly active in solving some of the major research problems that arise in the field. Based on the issues discussed periodically during the Interconnects cluster meetings and the recent publication record of the cluster members, the three most challenging subjects (categories) identified by the cluster's research community are the following:

- Networks-on-chips.
- Inter-processor communication architectures and scalable network interfaces.
- Automated design tools and methodologies

5.2. Description of investigated approaches

Networks-on-chip is a new open research field that has attracted most of the members' interest. Technological advances allow the integration of more functionality on the same chip, which in turn makes the on-chip communication infrastructure a critical parameter of overall system performance. The members of the Interconnects cluster community have presented promising solutions concerning several aspects of the on-chip networks. New switch architectures have been proposed that reduce message delivery latency and improve system throughput. In parallel, new topologies and routing mechanisms have been presented that improve system performance whilst offering viable implementations. Additionally, several design alternatives have been presented that target other critical design constraints such as interconnect power efficiency, scalability and fault tolerance.

Inter-processor communication architectures clearly affect the scalability of future many core systems. Interconnect cluster members are actively involved in the design of new tightly coupled network interfaces integrated with the cache controller and the definition of an efficient event-response mechanism that will serve at the same time cache-coherence protocols, remote-DMA transfers, and synchronization primitives. Preliminary results appear promising and future enhancements are expected.

Increasing design productivity and improving system quality clearly depends on the available design tools that will automate the design process and let the designer focus on critical decision making. Interconnects cluster members have presented significant results in the area of application-specific interconnection networks where design parameters, especially routing algorithms and topologies, are calibrated for the specific design environment, thus reducing drastically hardware complexity. Also, automated approaches and tools have emerged for application-specific task mapping that efficiently balance the workload to different parts of the network.

5.3. Summary of activities

The promising research results of the Interconnects cluster members come either through the research efforts of individual groups or through joint research activities that began and continue with the support of the HIPEAC network. Interconnects cluster meetings are the starting point of many of these collaborations. The open environment and the stimulating discussions have clearly helped the members identify common research interests with other colleagues and build a common understanding of their future research goals. Briefly, the major currently active collaborations between cluster members that emerged via HIPEAC are the following: UPV (Spain) - Univ. Ferrara (Italy); FORTH (Greece) - Barcelona Supercomputing Center (Spain) - TU DELFT (Netherlands); TU DELFT (Netherlands) - Univ. Ferrara (Italy); UPV (Spain) - Simula Labs (Norway); FORTH (Greece) - Univ. Cantabria (Spain); Univ. Catania (Italy) - Jönköping University (Sweden); FORTH (Greece) - Technion (Israel).

Also, several of these collaborations have taken the form of student visits to other HIPEAC institutes for a short period. The majority of such visits aimed at combining the expertise of the two collaborating research groups in order to improve the validity and the quality of the research results that in many cases were published in widely appreciated journals and conferences. A characteristic example of such combinations is students with a solid architectural background visiting institutes such as Univ. of Ferrara and FORTH in order to perform actual chip designs and discovering the issues involved in building real hardware. Pablo Abad of University of Cantabria visited FORTH, while Samuel Rodrigo and Francisco Gilabert Villamon from UPV and Daniele Ludovici from TU Delft visited Univ. Ferrara.

The 3 cluster meetings organized in the first year of HIPEAC2 were based on short members' presentations of their recent work followed by questions and comments by the audience. In this way, the majority of the members were able to show their work to the cluster and get in many situations valuable feedback. In order to increase the visibility of those research results, all presentations, were made available to the subscribers of the Interconnects mailing list, which is a broader group of researchers and students not necessarily HIPEAC members, via the HIPEAC website http://www.hipeac.net/Interconnects. The contacts with the members, the organization of the meetings, their administration, and the filing of the material were handled by the cluster researcher; the co-ordinator chaired the cluster meetings.

The meetings took place in Goteborg Jan. 2008 and Barcelona June 2008, both funded by HIPEAC1, as well as in Paris Nov. 2008. The organization of the upcoming meeting in Cyprus Jan. 2009 is pending. The talks given during the meetings are the following:

Goteborg meeting

- "Performance of NOC topologies under synchronization intensive traffic patterns", Martino Ruggiero - University of Bologna (Italy) and Davide Bertozzi University of Ferrara (Italy)

- "Congestion management", Pedro Garcia University Castilla-La-Mancha (Spain)
- "Technology constraints and router micro-architecture", Giorgos Dimitrakopoulos, FORTH (Greece)
- "Topological mapping for NOCs", Juan Manuel Orduna Huertas, UPV (Spain)
- "The new face of on-chip interconnect", Chrysostomos Nicopoulos, EPFL (Switzerland)
- "On-Chip networking", Isaac Keslassy, Technion (Israel),

Barcelona meeting

- "Rotary Router: a Suitable router for CMP systems", Valentin Puente, Universidad de Cantabria, Spain
- "Secure Memory Accesses on Network on-Chip", Leandro Fiorin ALaRI-USI and Gianluca Palermo Politecnico di Milano
- "Network on Chip topologies for 64-tiles general purpose MPSoC platforms: Latest advances in the cooperation between University of Ferrara and Universidad Politecnica de Valencia", Davide Bertozzi, University of Ferrara
- "Thales Research and technology: Embedded System Lab contribution", Sami Yehia, THALES Research and Technology
- "Three Dimensional Networks on Chip", Igor Loi University of Bologna
- "Power efficient NOCs and Interconnects Roadmap", Giorgos Dimitrakopoulos, ICS - FORTH

Paris meeting

- "A silicon-aware design platform for nano-scale networks-on-chip", Davide Bertozzi, University of Ferrara
- "2D Crossbar VLSI Layouts", Giorgos Passas, FORTH
- "Arbiter Design Revisited", Giorgos Dimitrakopoulos, FORTH

Another meeting point of the cluster members is the workshop on Interconnection Network Architectures: On-Chip, Multi-Chip organized by the cluster members in parallel to the HIPEAC conference. The attendance and the presentations so far have been of high quality.

The cluster researcher also dedicated a considerable part of his time to stimulating an open discussion trying to identify future research problems in the interconnects research area. This activity was performed either during the cluster meetings or through e-mails where the majority of the members eagerly responded sharing new ideas. This repeated procedure helped significantly in building a common consensus among the cluster members, which fed the Interconnects section of the HIPEAC Roadmap preparation. The cluster researcher (hired at FORTH) played a leading role in gathering the opinions and preparing several drafts of the cluster's contribution to the roadmap.

5.4. Future plans

In the future, the major goal of the HIPEAC Interconnects cluster is to continue being the reference point for all European researchers in the area of interconnection networks. To achieve this goal we plan to enhance our cluster meetings in order to further stimulate research collaborations and attract more junior researchers. Additionally, we will continue stimulating long-term visits, collaborative research, and steering and influencing the research directions of members. More specifically:

- Besides academic members, we will also seek to attract industrial companies that work on interconnection networks. For example, we hope to soon have the HyperTransport Consortium (http://www.hypertransport.org) participation in the cluster activities.
- Especially for the student members of the cluster, our goal is to organize meetings with specific subjects such technology limitations of networks-on-chip so as to give them the opportunity to enhance their knowledge and maybe present also their work.
- Finally, the Interconnection cluster is collaborating closely with others HiPEAC clusters (especially, Multi-core Architecture, Programming models, and Reconfigurable Computing). Collaboration topics include:
    - NoC topologies and their performance as these integrate with multicore architectures
    - fault tolerant and irregular routing in NoC's as these relate to future unreliable transistors and heterogeneous multi-cores
    - network interface architecture as it relates to parallel programming model
    - network prototyping using FPGA's, reconfigurable networks

5.5. Progress Indicators

- # active members: 36
- # companies involved: 3
- # publication output: 52
- # joint papers: 22 (3 industry)

6. Cluster: Reconfigurable Computing

6.1. Description of scientific issues

Reconfigurable Computing (RC) is an emerging paradigm which combines the flexibility of software with the high-performance of hardware by enabling a massive number of programmable computational nodes to execute in parallel. In effect, the power/performance ratio for reconfigurable devices is considerably better than ordinary general-purpose processors. This is because reconfigurable devices such as FPGAs support customization of the datapath and control-flow of hardware in order to adapt to the specific needs of the application. Thereby enabling faster execution times than it is currently possible with general-purposed processors, while at the same time being more power efficient. For these reasons, reconfigurable hardware is becoming popular in embedded systems and application-specific design, while it can also provide an attractive solution for high performance computing. However, in order to deploy this technology at a larger scale, there is a need to address several specific challenges.

The first year of the project we have identified the challenging issues in the field of reconfigurable computing. In order to identify the Reconfigurable Computing (RC) challenges and put them in the right context, we first described the abstraction layers and properties of Reconfigurable Computing. More precisely, we consider the following abstraction layers:

- Applications;
- Tools, Methods and Runtime Support;
- Systems architecture, micro-architecture, and implementation;
- Realization/Technology.

We denote as system architecture, micro-architecture, and implementation the elements of a system built (partially or entirely) in reconfigurable hardware rather than the reconfigurable device itself. Furthermore, we denote as Realization/Technology, the technology of a reconfigurable platform, built in silicon or any other future fabrication technology. A Reconfigurable System has or is desired to have the following properties:

- Low overhead Reconfiguration;
- Power;
- High Reliability;
- Adequate Performance.

LayersProperties	**Reconfiguration**	**Power**	**Reliability**	**Performance**
Applications	Adaptive, Parallel	Embedded	Safety critical	HPC (industrial and scientific)
Tools, Methods and Runtime Support	technology mapping, Low Overhead Runtime Support	Extension of run-time systems, Profiling/ Modelling	Runtime support for detection, self-testing, self-repairing, graceful	Hiding RC overheads/ penalties, efficient HW/SW partitioning

			degradation etc.	
System Architecture, micro-architecture, and implementation	adaptive architectures, ISA extensions, Memory architectures	customized data path and control-flow	isolation, correction, redundancy, etc.	customized data path and control-flow
Realization/ Technology	Novel reconfiguration mechanisms	Power efficient reconfigurable platforms	Novel error prone components	Coarser-grain, emerging technologies: Nanowires, 3D

Table 5.1 depicts the above abstraction layers and properties of RC. It further provides more precise information regarding each RC property at every abstraction layer. The main RC challenges that we have identified span across the above RC abstraction layers and properties and are the following:

1. **Application domain extensions**;
2. **Improved run-time reconfiguration**;
3. **Power-efficient computation**;
4. **Reliability**;
5. **Tools, Methods and Runtime Support**;
6. **Targeted Systems (Embedded/High Performance-Multicores)**;
7. **RC Specific Memory architectures**;
8. **Realization/Technology**.

Challenges 1-5 and 8 constitute a row (abstraction layer) or a column (RC desired property) on the table. Challenge 7 is an important issue in the architecture of any reconfigurable system. Finally, Challenge 6, although it overlaps with most of the other challenges, it is meant for identifying the different requirements/characteristics/issues of the embedded and high performance systems that exploit reconfigurable hardware. Several challenges, such as Challenge 6, overlap with each other. In general, two challenges overlap when they cross each other on Table 5.1. In essence, challenges that are RC properties overlap with RC abstraction layers.

The above challenges in Reconfigurable Computing are discussed in detail in the first draft of the RC chapter of HiPEAC2 Roadmap.

6.2. Description of investigated approaches

Besides identifying challenging future issues in RC, in the first year of the project, the cluster members have continued their research on Reconfigurable Computing, several findings of which were presented in our cluster meetings.

Some of the investigated approaches covered by within our cluster are the following:

- Compilation for Reconfigurable Architectures;
- Applications for Reconfigurable Computing;
- Dynamic adaptation and runtime reconfiguration;
- Reconfigurable MPSoC, Multi/Many-cores, Parallel Processing;

- Reconfigurable embedded systems and tools for various applications (e.g., signal/image processing);
- Operating systems for Reconfigurable Computing;
- Evolvable Components;
- Custom Computing;
- Power efficiency;

The above works were presented and discussed during our cluster meetings:

- *"From Software Programming Languages to Reconfigurable Computing Architectures"* (presented by Joao M.P. Cardoso, IST/INESC-ID, Goteborg, January 2008)
- *"HPC applications for RC"* (presented by Miquel Pericas, UPC/BSC Goteborg, January 2008)
- *"Dynamic Adaptation"* (presented by Wolfgang Karl, Karlsruhe Uni, Goteborg, January 2008)
- *"Reconfigurable MPSoC Architectures for Automotive Applications Driver Assistance"* (presented by S. NIAR, Goteborg, January 2008)
- *"On run-time reconfiguration research @ UGhent* (presented by Dirk Stroobandt, U Ghent, Barcelona June 2008)
- *"Architectured-FPGA and RC activities at TRT"* (presented by Sami Yehia and Philippe Bonnot, Thales, Barcelona June 2008)
- *"Reconfigurable and Custom Computing at University of Paderborn",* (presented by Christian Plessl, U. Paderborn, Barcelona June 2008)
- "The TFlux Parallel Processing System Accelerator" (presented by Pedro Trancoso, U. Cyprus, Cyprus, Paris November 2008)
- *"Power-Efficient Design of Tightly Coupled Parallel Processors with Dynamic Reconfiguration Capabilities"* (presented by Frank Hannig, U. of Erlangen-Nuremberg, Paris November 2008)
- "*Can reconfigurable be integrated in General Purpose Computers?"* (presented by Georgi Gaydadjiev TU Delft, Paris November 2008)

6.3. Summary of activities

The following activities took place in the RC cluster. All of these were performed or initiated by the cluster coordinators Georgi Gaydadjiev and Ioannis Sourdis.

Organizing a cluster board We organized the cluster creating a cluster board which consists of 5 senior cluster members. The cluster board among other activities performs the following: suggest new members, decides about the cluster funds distribution and the annual evaluation of existing members, and suggests RC research directions. The board consists of the following members: *Prof.dr.-Ing Juergen Becker, U. Karlsruhe, Prof. Georgi Gaydadjiev, TUDelft (cluster coordinator), Prof. Wayne Luk, Imperial, Prof. Nacho Navarro, UPC, Prof. Donatella Sciuto, PoliMi.*

Roadmap chapter in RC The first version of the RC chapter in the roadmap is written. The actual writing and the integration of the members' suggestions was performed by the coordinators.

Special Interest Groups It was decided to form several groups of special interest. These groups have special research interests and/or assignments that are important for our cluster. Some of them are: (i) *Emerging domains for RC: Identify new applications for RC*; (ii) *Related technology developments*; (iii) *Runtime reconfiguration*.

Organizing Workshops Ioannis Sourdis and other cluster members organized the HiPEAC workshops on Reconfigurable Computing of 2008 and 2009, held together with the HiPEAC conference. The workshop was very successful having over 20 submissions each year although the proceedings were informal. The submissions were authored by researchers from all over the world (many of which were outside of HiPEAC).

cluster meetings The cluster meetings were successful, and gave the chance to our members to present their work on Reconfigurable Computing, discuss challenging issues and future directions for the scientific community.

cluster wikipage A wikipage for the cluster was created by Ioannis Sourdis `http://ce.et.tudelft.nl/HiPEACRC/wiki/`. The wikipage provides a forum for discussion among the cluster members. It also provides useful information regarding the expertise of each research group involved in the cluster, and hence promotes collaborations between the cluster members. It provides the means to exchange and share ideas and knowhow, and to collaborate in between the cluster meetings when the members do not have the chance to physically meet.

Evaluation of cluster members The coordinators and the cluster board evaluate the members and advises HiPEAC2 steering committee regarding the activities of the cluster members. The coordinators collect information (mostly through the RC wiki page) regarding the above activities.

6.4. Future plans

As part of our future we plan the following:

- Promote collaborations and writing proposals in the field of Reconfigurable Computing for the next FP7 call;
- Improve and enhance collaborations with industrial partners. Find new industrial partners interested in Reconfigurable Computing;
- Improve and Finalize the RC chapter for the HiPEAC2 Roadmap;
- Have formal proceedings in the next HiPEAC workshop on Reconfigurable Computing 2010;
- Continue and work out in more detail our past activities;

6.5. Progress Indicators

- # active members: 41
- # companies involved: 3
- # publication output: 112
- # joint papers: 26 (5 industry)

7. Design Methodology and Tools

7.1. Description of scientific issues

The domain of design methodology and tools is presently driven by the trend towards multiprocessor system-on-chip (MPSoC) architectures. From an application point of view, various domains of embedded systems are of highest interest, such as wireless communications, multimedia, and automotive. Corresponding MPSoC architectures have to meet very high efficiency goals (measured in MIPS/Watt or Joule/bit), which results in a preference for programmable, yet application-specific and heterogeneous MPSoCs. In particular in the embedded domains, there is currently a shift towards Electronic System Level (ESL) design methodologies, where high-level building blocks (processors, buses, memories etc.) form the basic components. It is expected that ESL design will provide the required next productivity boost beyond RTL design, which is key for managing the complexity of today´s and future digital chip designs in advanced CMOS technologies (45nm, 32nm, and beyond).

Generally, ESL design involves various major tasks, such as architecture exploration and optimization at system and block level, software optimization, HW/SW partitioning, synthesis, and validation. More specifically, during the cluster kickoff meeting (Goteborg, Jan 2008), the cluster members agreed on the following areas of common interest:

- Embedded system applications: e.g. wireless, multimedia, biomedical
- Embedded processor design: ASIP design, design space exploration, configurable and reconfigurable processors, SW tools generation, retargetable compilation, processor/compiler co-design, loop parallelization, template based ASIP design, instruction set extensions, ultra-low power ASIPs, open source processor cores
- MPSoC modelling and verification: modelling languages, SystemC based modelling, verification and design, high-speed instruction set simulation, virtual platforms, path from model to implementation, NoC simulation
- MPSoC programming: application-to-architecture mapping, RTOS, task graph scheduling, sequential-to-parallel code generation, benchmarking
- MPSoC HW/SW architectures: SW performance estimation, HW/SW integration, tightly coupled processor architectures, memory hierarchy, HW/SW interface synthesis, fault tolerance, loop transformations

7.2. Description of investigated approaches

The cluster members are pursuing a multitude of ESL approaches along the lines sketched above, depending on their particular interests and current local projects environments. Due to the limited space, we just provide a representative snapshot below, summarizing recent member presentations given at the last cluster meeting (Paris, Nov 2008):

- Prof. Juergen Teich, Erlangen University: Proposes a new paradigm for parallel programming: "Invasive Programming". In this new paradigm algorithms replicate themselves and claim resources from the platform at run time. Prof. Teich's claim is that with this paradigm one achieves: scalability, robustness and reliability. This approach has been illustrated at the loop-level on a weakly programmable array. They are working on implementing invasive programming at the thread and at the job level.

- Veerle Desmet, U. Ghent: Motivated by the non-continuity of improvements along the years of different algorithms, Ghent developed a web-based application that helps researchers comparing their results against those of other people in a fair way. The application is called ArchExplorer and provides a repository for people to upload their code. Under the hood, ArchExplorer performs a fully automatic architecture exploration based on genetic algorithms optimizing performance. In order for a researcher to use ArchExplorer, SW and HW interfaces need to be compatible to those provided by the framework.
- Thuyen Le, Infineon: Infineon is already using MPSoCs in their products. They are expectantly looking at the MPSoC developments and the usability that future chips will provide. He stressed the difference with the server computing community. They have several cores, but of different nature and with several HW accelerators. He showed a sample platform based on ARM11. They make use of virtual platforms. They use a real-time kernel for the protocol stacks and a normal OS for multimedia-like applications. He discussed the matter of instruction reuse in protocol stacks and argued that bigger caches are not improving performance as expected.
- Martino Ruggiero, U. Bologna: Presented a "Resource Optimizer" which is part of the “CellFlow”? The application is represented as a directed acyclic graph that communicates through FIFO communication channels. The optimizer allocates and schedules the task graph on to a platform by using the principle of Problem Decomposition were a part of the problem is solved via ILP (allocation) and another part via Constrained Programming (scheduling). He compared results of performance with 2 and 3 phases of decomposition.
- Jeronimo Castrillon, RWTH Aachen University: presented two research projects running at the SSS institute. The first one, called MAPS (MPSoC Application Programming Studio), address the issue of programming an MPSoC taking into account:
    - Multiple applications with different representations
    - Semi-automatic program partitioning
    - Temporal and spatial allocation
    - Code generation

  He presented a concrete proof of concept developed together with HiPEAC associate member Prof. Isshiki from Tokyo Institute of Technology. The second one, OSIP (OS Instruction Set Processor), discussed an approach of offloading OS services (scheduling and mapping) onto a specialized core (ASIP).
- Francesco Regazzoni, ALaRI: presented a flow for designing ASIPs, i.e. determining potential Instruction Set Extensions, targeting security instead of performance. In this flow, the main components were missing: metric, tools at spice level, CMOS with protected logic. In the frame of a HiPEAC collaboration grant with the EPFL, they implemented these components and exercised the complete flow.

7.3. Summary of activities

Major activities of the cluster include the following:

- Cluster meetings: Three regular cluster meetings were held in 2008 (Goteborg, Barcelona, and Paris), which were well attended. During the meetings, organizational issues and joint research directions have been discussed. Several new members have been accepted, among them associate members from USA and Japan. Thanks to a relatively high number of company members, there is a good degree of academia/industry interaction.

- Web site: A cluster web site (at www.hipeac.net) has been set up. It contains e.g. member research profiles, supporting material, as well as cluster meeting minutes.
- HiPEAC workshops and schools: Cluster members attended major HiPEAC events such as the ACACES summer school and various industrial workshops.
- Bilateral research meetings: Numerous bilateral meetings took place in 2008 in order to discuss particular research issues. For instance, Aachen had private meetings with groups from TU Delft, Tokyo Institute of Technology, CoWare, TU Braunschweig, Thales, TU Kaiserslautern, NXP, INRIA etc.
- MPSoC Forum: The cluster provided sponsorship for the MPSoC Forum (Aachen, June 2008). The MPSoC Forum (www.mpsoc-forum.org) is the premier international event in application specific MPSoC architectures and design tools.
- RAPIDO workshop: Together with people from the HiPEAC simulation cluster, a new workshop on simulation and architecture exploration has been organized, scheduled to take place at the HiPEAC conference in Jan 2009.
- ECSI workshops: Aachen's research results on Virtual Platform technology, developed jointly with CoWare, have been successfully demonstrated at various ECSI workshops, e.g. at the Design Automation Conference (DAC) in June 2008.
- Infineon MPSoC workshop: German HiPEAC partners (Aachen, Erlangen, Augsburg) have organized a joint workshop with Infineon Technologies (Nov 2008) in Munich. Focus has been on advanced MPSoC design technologies and how these could help industry in their forthcoming design challenges.
- Industry internships: Students from Aachen have performed 6 months industry internships at ACE and Infineon.

Most of these activities have been coordinated by Aachen's HiPEAC researcher M.Sc. Jeronimo Castrillon. He has also been involved in the production of the quarterly HiPEAC newsletter. Due to the high workload, for 2009 it is envisioned to include another researcher from Aachen (M.Sc. Anastasia Stulova) in the cluster activities and organizational issues.

7.4. Inter-cluster collaborations

The simulation and EDA clusters have been collaborating on simulation technology. As a first step, an effort was launched, and is still on-going, to identify the different typical simulation usage scenarios spanning the different domains of HiPEAC (embedded, general-purpose and high-performance), with the prospect of better focusing the simulation efforts. The two clusters also aim to join their simulation efforts in order to avoid redundancy and to learn from the different domains involved, as it turns out the embedded systems domain and the general-purpose domain have come up with different solutions to the main simulation issues (especially simulation speed). A workshop on that topic (RAPIDO) has been organized by Smail Niar and the two cluster leaders, at the HiPEAC conference in Cyprus. Recently, several members of the EDA and simulation clusters have submitted a joint proposal for addressing the simulation speed issues of multi-cores. Finally, the two cluster coordinators are contemplating editing a book together on simulation.

7.5. Future plans

After completing the essential cluster formation in 2008, it is expected that more and more bilateral or multilateral co operations will be established. For instance, various cluster members are planning to submit joint FP7 project proposals within different consortium configurations.

At the same time, the cluster will remain open for further memberships, encouraging especially more company members, in order to facilitate technology transfer. We also expect that the ESL related co operations stimulated by HiPEAC will be consolidated, in the sense that a lower number of key topics will be pursued.

An example is the domain of MPSoC programming. Aachen will organize a special session on this topic at the DATE 2009 conference, where HiPEAC members and external speakers will discuss the state-of-the-art. After more and more SW prototypes are becoming available and mature, it is anticipated that real tools integration can take place. In the domain of MPSoC programming, there is also cooperation with a corresponding cluster in the ARTIST-DESIGN NoE, which will be intensified in the future.

7.6. Progress Indicators

- # active members: 43
- # companies involved: 6
- # publication output: 38
- # joint papers: 8 (2 industry)

8. Binary Translation & Virtualization Cluster

The economic lifetime of software is many times longer than the economic lifetime of hardware. The real value (and also the complexity) of an information processing system lies in the specialised software, not in the generic hardware that is replaced at regular times anyhow. Therefore, companies want to protect their investments in software.

This evolution leads to the hardware/software paradox. In the past, hardware was considered to be complex and inflexible whereas software was simple and flexible. In spite of hardware's exponential performance evolution as illustrated by Moore's law, the software stack has in time overtaken hardware in terms of complexity and associated inflexibility. Therefore, rather than looking for software to run on their hardware platform, people are looking for hardware platforms to run their software on. Since platforms are evolving fast (and are expected to evolve even faster in the coming years with the advent of many-cores, accelerators and reconfigurable hardware), old platforms will become obsolete, requiring extra investments to port hard-to-change software to new hardware platforms.

The big challenge is therefore to let the hardware evolve at its current rapid pace while keeping the software maintainable, stable and efficient.

One family of techniques that can help with solving these challenges is binary translation and virtualization (BTV). These techniques can assist with automatically transforming software stacks, ranging from dynamic optimization and resource management to abstracting away the entire underlying hardware architecture or software layers. They can also help with providing greater insight into the software stacks. Furthermore, future co-designed hardware/software BTV platforms can offer additional advantages over the current approach of making the BTV layer, the other software layers and the programmer as oblivious to each other's existence as possible.

8.1. Description of scientific issues

Hardware

The cluster identified the following main hardware challenges:

Hardware support for BTV Today, hardware support for BTV often means that a few extras have been bolted on an existing platform to address some of the worst performance problems. This means that the BTV layer often needs to work around the hardware in order to offer reasonable performance and reliability.

Dealing with hardware variability Hardware variability comes in many forms: heterogeneous multi-cores, heterogeneous clouds, reconfigurable cores and process variability. In all of these cases, a programmer ideally should be able to write software in a processor-independent manner. This will ensure that the software can run on all available target processors, that partially defective parts can be used, and potentially that a running program can migrate between different cores.

The cluster is currently looking into the following approaches:

Hardware/software co-design A co-designed hardware/software platform, built from the ground up for running a virtualized environment, offers many exciting prospects both in terms of

power and performance. The cluster is investigating the design of such a platform, based on a VLIW architecture and dynamic compilation techniques.

Adaptability to hardware variability BTV can be used to adapt code and data layout per execution or even during a single execution, enabling workarounds for stuck bits and similar issues in ranch predictors, cache lines, memory, … Furthermore, dynamic (re)compilation can generate alternate code to avoid using non-functional CPU instructions and registers. The cluster is investigating the feasibility of various approaches and the kinds of faults that can be dealt with using these techniques.

Software

The cluster identified the following main software challenges:

Meta-information about client programs Many transformations and opportunities are only available when the appropriate information about the program is available. This information can sometimes be automatically reverse-engineered from the binary level, but such techniques are seldom 100% reliable and complete.

Interacting with the hosts Until now very little work has gone into specifically designing applications and programming models for running inside BTV or system virtualization environments. BTV is almost invariably an independent add-on technique used for solving various unforeseen problems. A generic and extensible interface supported by the various BTV systems, to enable application design that takes into account the presence of and functionality offered by underlying BTV systems.

The cluster is currently looking into the following approaches:

A generic intermediate format and annotations A generic, standardised intermediate format to express programs would go a long way towards improving the analysability and portability of computer programs. Such a format would probably incorporate annotations to express extra semantic information about the program. The BTV cluster is investigating the necessary properties of such a format and the usability/extensibility of existing formats together with the Compilation Platform cluster.

Vertical instrumentation Vertical instrumentation means that the information gathered at different layers of the execution stack is coupled together to provide a holistic view of the system. As part of the previous work, we are interested in defining the necessary hooks to enable efficient communication and cooperation between the various execution layers, as well as amongst different programs running concurrently.

Requirements challenges

The cluster identified the following main software challenges:

Understanding system behaviour Modelling and understanding the behaviour of a program is hard. Modelling the behaviour of an entire BTV stack with different client programs running on top of them is even harder. Nevertheless, insight into the low level behaviour of such stacks, predicting performance consequences of software and hardware decisions and determining bottlenecks (preferably ahead of run time) is of paramount importance to tune and improve such environments.

Certification and validation The complexity of BTV systems means that the validation required before they can be used in mission-critical situations is extremely hard. Nevertheless, such

validation and subsequent certification is very important given the increasing ubiquity of BTV.

The cluster is currently looking into the following approaches:

Performance modelling One killer application for a BTV environment is without doubt the consolidation of server farms. However, to ensure an optimal consolidation, it is desirable to take performance requirements of both virtualized systems and applications into account. For this purpose, cluster members are working on performance models that deal with the multitude of layers in the virtualized execution stack, including the host OS, VM, guest OS, etc.

8.2. Activities summary

Cluster meetings The cluster has met three times in 2008, exchanging ideas, giving short presentations and preparing its contribution to the HiPEAC2 Roadmap. These meetings were prepared by the cluster researcher, who also gave several presentations on his work since the previous meeting.

The presentations held during the meetings were the following:

Goteborg meeting

- IBM Haifa interest items – Bilha Mendelson (IBM Haifa)
- Diota and Diablo, Partners in Rewriting – Jonas Maebe (Ghent University)
- The GACOP Research Group – Jos M. Garcia (University of Murcia)
- Binary Translation and Virtualisation Cluster – Koen De Bosschere (Ghent University)
- VM Research at Ghent University – Lieven Eeckhout (Ghent University)
- Virtualization: from Mobiles to Super Computers – Marco Cornero (ST Microelectronics)
- Overview of the Edinburgh High Speed Simulator – Nigel Topham (University of Edinburgh)
- A Performance Study of KVM – Shlomo Weiss (Tel Aviv University)
- AbsInt Products and Research Interests – Daniel Kastner (AbsInt)
- Technology-aware Virtualisation – Bjorn De Sutter (Ghent University)

Barcelona meeting

- Diablo: from one ISA to another – Jonas Maebe (Ghent University)
- Cache Content Duplication – Marios Kleanthous (University of Cyprus)
- ANCORA – Pedro Diniz (INESC-ID)

Paris meeting

- Report on Lyons ICT 2008 – Koen De Bosschere (Ghent University)
- Link time code compaction for the ARM architecture – Jonas Maebe (Ghent University)
- Report on the Dagstuhl Seminar on DBT – Koen De Bosschere (Ghent University)
- Presentation of the BTV Roadmap – Koen De Bosschere (Ghent University)

- Incorporating higher-level abstractions in intermediate byte code formats – Bjorn De Sutter (Ghent University)

Dagstuhl Seminar The cluster co-organised a Dagstuhl Seminar on "Emerging Uses and Paradigms for Dynamic Binary Translation". The cluster researcher cooperated in the preparation and reporting of this seminar.

Tools overview The cluster researcher made a web page on the cluster website providing information about and links to BTV-related software available from the cluster members.

Roadmap The cluster discussed and prepared two drafts of its contribution to the HiPEAC2 Roadmap. These drafts were largely prepared by the cluster researcher.

ARM internship The cluster collaborated with the SARC project on a research internship by the cluster researcher at ARM. This work, on the topic of static binary translation, was presented at the last cluster meeting of 2008.

Diablo The cluster researcher has further improved, maintained and supported Diablo, a static binary rewriting framework. This framework was/is used by the University of Edinburgh, the University of Siena and UPC.

Summer school teacher The cluster invited Leendert van Doorn (AMD) to give course on virtualization at the ACACES 2008 Summer school.

Cross-ISA optimization Ghent University and UPC cooperated in the investigation of cross-ISA optimizations at link time for the Cell BE.

8.3. Inter-cluster collaborations

- Compiling source code into an intermediate format and translating it into native code at run time using a Just-in-Time compiler embedded in a process virtual machine is commonplace nowadays (e.g., Java Virtual Machines). Together with the compilation cluster, we are looking into extending this approach to virtual machines for entire system stacks, by designing or extending an intermediate format to include high level meta-information about
  - program features that are not easily expressible in a generic way using traditional byte code formats (such as loop nest information)
  - ways to interact with the with the various programs and layers in the system, to either monitor or influence their behaviour
- Together with the adaptive compilation cluster, we are investigating the use of binary rewriting to reduce energy usage by instruction caches. We use profiling information to group hot code into regions, and then disable cache tag checking for these regions. As a result, as long as the execution remains in such an area, instruction cache accesses use much less energy.

8.4. Future plans

- Maintaining and supporting the Diablo binary rewriting framework.

  The Diablo rewriting framework is widely used all around the world as an aid while researching program diversification, program obfuscation, code compaction and energy consumption reduction. We will keep assisting people in using it, updating it for new operating system and library revisions, and maintaining it as both an educational and research tool.
- Researching challenges and opportunities related to hardware variability.

As mentioned in the first section, we will investigate how BTV can contribute to dealing with the increasing hardware variability, and how it can enable more easily changing the underlying hardware platforms. Full system virtualization and dynamic code generation and optimization are core components of this research, and we plan on designing such a prototype system and implementing a proof-of-concept.

- Researching the incorporation of non-determinism into performance models for virtualized systems.

  Performance models of virtualized systems are very hard to device, because non-determinism has even more effects in this case than in other systems. For example, which methods are optimized in a Java Virtual Machine can vary significantly from one execution to the next, depending on the used input set. Similarly, the input can also significantly influence the memory access pattern of certain programs, thereby placing a different kind of stress on page table handling in system virtual machines. We will investigate such influences and attempt to model them.

- Cooperation with the Compilation Platform cluster on a generic and common intermediate binary format.

  The acceptance in the market and the energy-efficient use of new kinds of heterogeneous multi-core systems requires that pre-existing software can take proper advantage of such systems. For this purpose, we need a very flexible intermediate format that encodes a lot of semantic information, thereby enabling lower layers to transform and adapt the software to the available hardware resources. Together with the Compilation Platform cluster, we will investigate the requirements of such an intermediate format and either adapt an existing format, or design a new one, that addresses these challenges.

- Support for a BTV course at ACACES 2009.

8.5. Progress Indicators

- # active members: 42
- # companies involved: 10
- # publication output: 7
- # joint papers: 4 (1 industry)

9. Cluster on Simulation and Modelling Platform

9.1. Description of scientific issues

The goal of this cluster is to bring together the community around a common approach for modelling architectures. The motivation for this cluster is twofold: (1) modelling methodologies are facing significant challenges, especially due to the complexity of architectures and the emergence of multi-cores, (2) modelling/simulation tools are largely fragmented within the academic and industrial communities, making it exceedingly difficult to share, reuse or compare research or product achievements. Within this cluster, we will put an emphasis on simulation, like in HiPEAC1, but we also want to investigate analytical/statistical modelling, and FPGA prototyping, as two major alternative approaches.

Within HiPEAC1, we had kick-started this effort, then exclusively focused on simulation, which led to the development of the UNISIM simulation framework. Within HiPEAC2, our first goal is less to promote a single all-encompassing environment than to define methods for the different simulator efforts and tools to interoperate. The second goal of HiPEAC2 is to address a set of urgent research challenges. Some of the key challenges are:

- Simulation speed. It is the #1 challenge as architectures evolve to multi-cores with tens of cores. Numerous solutions are emerging, such as transaction-level modelling, sampling, parallel simulators, native execution, statistical simulation, FPGA prototyping, but they should be rapidly investigated.
- Technology. Numerous technology issues are now creeping up into system design and can no longer be ignored: besides power and area cost issues, they include wire delays, clock domains, temperature,...
- Application complexity. With the advent of smartphones used for 3D games and GPS navigation, embedded applications are getting almost as complex as desktop applications. Moreover, these smartphones are now running Linux or MacOS, so that full-system simulation is a must have, especially with multi-core architectures where the scheduler is playing an increasingly important role.
- Heterogeneous architectures. Embedded architectures and even the new general-purpose architectures are heterogeneous architectures. Modelling/Simulating such architectures raises special challenges. These challenges will be considered, especially through a tight cooperation with the Design cluster.

9.2. Summary of activities and main investigated approaches

During the first year, the cluster has focused on two issues: simulation/modelling speed and interoperability.

For *simulation speed*, there is no dominant approach and it is still very unclear which approach will ultimately prevail. So we focused on identifying all the potential approaches existing within the HiPEAC community. As a result, we had multiple presentations on the topic at the 1st and 3rd meetings. These presentations span most known approaches on the topic. The different approaches and talks are listed below.

- Fast functional and timing simulation, Nigel Topham, Edinburgh University, UK

- Hybrid simulation (toggling between native execution and timing simulation), Stefan Kraemer, Aachen University, Germany
- Sampling for multi-cores/MPSoC, Smail Niar, INRIA Lille, France
- Statistical simulation, benchmark generation, analytical modelling and workload characterization, Lieven Eeckhout, Ghent University, Belgium
- Fast analysis of program behaviour on architectures, Erik Hagersten, Uppsala University/ACUMEM, Sweden
- Transaction-Level Modelling, Daniel Gracia-Perez, CEA, France
- Sampling for full-system simulation, Ayose Falcon, HP Labs Barcelona, Spain
- Hybrid FPGA/software simulation, and statistical sampling, Babak Falsafi, EPFL, Switzerland

For *interoperability*, the discussions have started at the 2nd meeting and are still under way. The general idea is to promote a *bottom-up* approach where, instead of imposing a single platform, we progressively create APIs allowing the different existing tools to connect together. For instance, if an existing full-system simulation supports a given power modelling API, any API-compliant power modelling tool can be plugged into that simulation. We believe this approach creates a win-win situation. First, no one has to drop his/her effort invested in developing or mastering the utilization of an existing platform or tool, everyone can keep their own tools. Second, anyone who has developed a simulator or tool can potentially have many complementary tools at his/her disposal. For instance, no single group can afford to develop a full-system simulator, a power model and sampling techniques to speed it up. However, using the APIs approach, the full-system simulator developer can leverage existing and complementary power models and simulation speed techniques. Third and conversely, the audience of any given simulator tool/platform becomes much larger because other researchers do not have to drop their own tools/simulators in order to leverage the works of others, they can combine it with their own work. In the next few years, we will evaluate and possibly implement this synergetic approach at building a common set of tools.

There were also several discussions and presentations revolving around *full-system simulation* which is one of the key aforementioned issues:

- Full-system multi-core simulation, Babak Falsafi, EPFL, Switzerland
- A full-system simulator based on the Cell processor, Alex Ramirez, UPC/BSC, Spain (no slide, presentation of SARCSim)
- Virtual platforms, Daniel Gracia-Perez, CEA, France

Defining needs. At the 2nd meeting, a lively discussion erupted on the necessity to clearly define needs before selecting any speed or interoperability approach. While HiPEAC1 was very much focused on micro-architectures, HiPEAC2 spans more deeply into the MPSoC domains with partners like Aachen University, and in data centres with partners like HP Labs Barcelona. As a result, there are multiple different use cases for simulation tools, each with different requirements. Consequently, it was decided to first identify and list these different use cases, and to progressively draw conclusions from these use cases on which approaches we should preferentially push within HiPEAC2.

We identified three applications scenarios so far, which are summarized below.

Use Case 1: Simulation for the Datacenter (Paolo Faraboschi, HP Labs, Barcelona, Spain) The current trends towards “cloud computing” call for a new approach to model the computing

systems in a consolidated datacenter. The datacenter itself becomes the computer, that is, the "object" to optimize. The way in which datacenters are measured and optimized go beyond the traditional power and energy dimension, and need to take into account SLAs (Service Level Agreements with the customers) and TCO (Total Cost of Ownership). In first approximation, SLAs are related to performance and "ilities" and TCO is related to cost and energy.

The workloads of interest in this domain are a compound of consolidated standard applications (e.g., SAP, Oracle, BEA), together with emerging applications covering the so-called "web 2.0" space (e.g., server-side email, searching and indexing, distributed map-reduce jobs, content aggregation pipelines, rich media transformation and distribution).

What do we want to achieve? Model the entire computing system to approximate and predict its behaviour. This is to generate understanding or validate intuition on new architectural or system ideas.

Why do we need simulation? Because some components may not exist, or it may not be feasible to add instrumentation to a real system

Who are the target users? System architects (for component evaluations, what- if scenarios), pre-sale engineers (for right-sizing bids), IT administrators (for capacity planning).

How? Cycle-accurate simulation is not the way to model new solutions in this new world. What's needed is a combination of multiple simulation levels including

- Fast emulation-based full-system execution-driven simulation of parts of the datacenter. This would model an isolated cluster of a few hundred nodes with however cores/node the CPU technology offers (4-8 multicore today, 64 manycore tomorrow). This simulation platform can be used to extract detailed statistics about the behaviour of certain component combination for a given workload.
- Model-based, trace driven simulator for larger (possibly datacenter- level) systems. This does not execute the workloads in a traditional sense, but works off high level customer traces collected on a live system, together with component characterization (possibly extracted by the cluster simulator) to estimate the different target metrics under different scenarios.

The full-system simulator characteristics could be summarized as follows

- Speed of hundred of simulated MIPS, i.e. a 10x-20x slowdown vs. native execution. Support for parallel/distributed execution at the scale-out (cluster) level and if possible at the scale-up (multiprocessor) level
- Accuracy: absolute error 10%-20% vs. a real machine, relative error 2%-5% vs. other simulated measurements
- Target ISA: industry standard general purpose CPUs: x86 (32b/64b) is by far the most important, followed by Power, Itanium, Sparc (much less important)
- Abstraction level: full-system modelling, capability to run unmodified commercial OS and application stack, including enough device details (disks, memory, NICs, etc.) to be compatible with standard OS drivers.
- Validation: functional level must be fully validated to run unmodified apps, metric-level (performance, energy) must be validated vs. real hardware on few standard benchmark and a few machine configurations

- Applications: standard software stack, unmodified binaries, standard reference inputs, typical benchmarks running for a few minutes of real time (corresponding to a few hours of simulation time)

Use case 2: MPSoC HW Platform Design: Early Performance Estimation (Rainer Leupers, Aachen University, Germany) The design methodology for application specific Multiprocessor Systems-on-Chip (MPSoC) platforms involves an early architecture exploration phase. During this phase, design options for implementing new platforms from scratch, or deriving enhanced variants of legacy platforms to support added functionalities, are evaluated w.r.t. performance. Key parameters are e.g. number and type of processing elements (PEs), spatial and temporal task-to-PE mapping, memory hierarchy, as well as topology and estimated bandwidth for the communication architecture. Such explorations typically take place by means of a Virtual Prototype (VP) of the platform, i.e. a high-level pure SW model of the SoC. The components of the VP may be modelled at various abstraction levels, ranging from purely statistical traffic generators for new, yet to be mapped, functionalities down to accurately timed components in case of predefined legacy IP blocks. Using the VP as an executable system model, the platform designer is enabled to rapidly explore various points of interest in the MPSoC design space in order to meet the system performance requirements in a cost and energy efficient way.

What do we want to achieve? Model the entire MPSoC at various (even mixed) abstraction levels to come up with an executable HW/SW platform specification. Sufficiently fast for rapid system-level exploration, yet accurate enough to gain confidence on the performance impact of major architectural and mapping decisions. Being able to accommodate legacy architectures, software, and IP blocks.

Why do we need simulation? Because purely analytical models do not deliver sufficient accuracy or may impose severe practical problems (e.g. task-to-PE mapping, model reuse and refinement).

Who are the target users? Primary: MPSoC platform architects. Secondary: algorithm and SW designers How?

- Use of modelling and interface standards such as SystemC and TLM 2.0 in order to enable fast IP integration
- Use of abstract (e.g. traffic generator) or concrete (e.g. instruction-accurate ISS) application task and PE models
- Use of various communication abstraction levels (packet-level TLM down to timed concrete NoC model)
- Orthogonal modelling of computation and communication
- Abstract modelling of key OS services
- Being able to connect to existing "downstream" MPSoC design flows via refinement procedure

The MPSoC VP characteristics could be summarized as follows

- Speed of hundred of simulated MIPS, close to (possibly even faster than) real time.
- Accuracy: no promise on absolute accuracy, but being able to predict trends correctly ("platform variant A performs better than variant B", "task mapping C is faster than D" etc.)
- Target ISA: heterogeneous MPSoC, including off-the-shelf embedded processors (RISC. DSP) and custom HW accelerators (ASIPs, ASICs, eFPGA), bus and NoC models

- Abstraction level: statistical traffic generators down to instruction-accurate simulators, loosely timed down to approximately timed communication architecture models
- Validation: executing applications (natively or target-compiled), utilizing host I/O facilities (LCD, USB etc.) connected to the VP
- Applications: wireless communication terminals, consumer electronics, automotive

Use case 3: Micro-Architecture Exploration (Olivier Temam, INRIA, France) Context and trends. The design of a high-performance general-purpose processor starts with a micro-architecture exploration phase. The goal is to max one or several performance objectives, within the constraints imposed by technology. Performance is evaluated over a set of benchmarks, serving as "representative applications" for a typical general-purpose workload. The micro-architecture exploration often occurs for a known ISA, though ISA extensions can be part of the process. Up to now, the evaluation of the various performance tradeoffs heavily rely on detailed, typically cycle-level, simulations.

The micro-architecture design domain is experiencing several shifts that are particularly challenging for evaluation methodology. A first trend is the evolution towards multi/many-cores (simulation speed, role of the O/S, interplays with the programming model,...). A second trend is technology-related issues creeping up in the design process (power, temperature, process variability, clock domains,...). Overall, there is increased complexity, which makes it difficult to anticipate the impact of a local micro-architecture modification.

What do we want to achieve? Compare micro-architecture design choices. From coarse-grain overall structure to fine-grain parameterization decisions. Accurately reflect the relative merit of two design options. Full runs are not necessary if partial runs can provide reliable performance trends.

Why do we need simulation? In fact, for high-level decisions, it is possible to start with analytical modelling. Simulation needed to explore design options, to anticipate interplays between all architecture parts, to confirm high-level decisions.

How? Cycle-level simulation remains a major approach, but alternatives or enhancements (badly) needed due to speed issues, operating systems effects, technology impact and complexity. No obvious solution emerges as of now.

- Speed is responsible for the greatest fragmentation of evaluation methods; many but also very distinct options (FPGA prototyping, parallel simulators, sampling and statistical simulation, abstract simulation, analytical and machine-learning models,...).
- Full-System simulators more widespread, but full-system simulation and fast simulation together are not necessarily trivial; full-system simulators = significant development investment.
- Factoring technology impact is still in infancy. Reasonable power models for memory; no good/generic model for logic. Some emerging models for process variability.
- Complexity calls for design automation and sharing simulator components/tools as researchers often specialize (NoC, cache coherence, temperature, etc).

The micro-architecture simulator characteristics could be summarized as follows:

- Speed: one benchmark should be evaluated in a few minutes to a few hours. Page 2

- Accuracy: mostly relative accuracy (between two design options) needed here; of the order of 1
- Target ISA: the most popular general-purpose and embedded ISAs (x86, PPC, ARM,...).
- Abstraction level: depends on goal; cycle-level for one or a few cores; TLM probably OK for many-cores; similarly, user-level still OK for one core, full-system required for multi/many cores.
- Applications: the future "typical" workloads; multiple data sets badly needed for all feedback-based studies.

RAPIDO Workshop

In order to achieve further cross-fertilization on the key topic of simulation speed, the HiPEAC cluster on Simulation and the cluster on Design have together supported the RAPIDO workshop on the topic. This workshop was launched at the initiative of Smail Niar, University of Valenciennes/INRIA, France, and the first edition will take place in the form of a 1-day workshop at the HiPEAC 2009 conference in Cyprus. The workshop will include 6 invited presentations and 9 selected presentations.

9.3. Future plans

The future plans have been largely mentioned before and briefly summarized here:

- Finalize the use case discussion and draw conclusions for the cluster orientations.
- Attempt to implement, possibly through a number of experiments, the interoperability approach.
- From an organizational standpoint, now privilege a few long presentations of tools and simulators, allowing participants to understand how they could connect their own tools and how APIs should be designed.
- Further confront the different simulation speed approaches.

9.4. Progress Indicators

- # active members: 59
- # companies involved: 9
- # publication output: 26
- # joint papers: 1 (1 industry)

10. Cluster on Compilation Platform

Coordinator: Albert Cohen, INRIA Saclay and Ayal Zaks, IBM Israel.

GCC has been selected as the default compiler platform, with the option to use additional software such as compiler plug-ins or completely separate compilation tools for just-in-time or dynamic compilation.

HiPEAC members have been conducting research and contributing to GCC in the past 4 years, but we expect the impact of the network to increase in the context of HiPEAC2, using GCC as a powerful dissemination vehicle of HiPEAC compiler research.

10.1. Collaborative Research Activities

- Practical iterative and machine-learning compilation in a production compiler. Contacts: Grigori Fursin, INRIA Saclay; Bilha Mendelson, IBM Haifa.

  This task builds on the Interactive Compilation Interface designed in the context of HiPEAC1, looking for more extensive instrumentation of the compilation flow.
- Extend the SSA-based internal representation to support non-scalar data-flow. Contact: Albert Cohen, INRIA Saclay; Skevos Evripidou, University of Cyprus. Proposals: data-flow streams, polyhedral model.

  This task follows the ongoing work on polyhedral compilation in the official graphite branch of GCC, in collaboration with SuSE (Germany, Czech Republic) and AMD Austin.
- Compiler support for Transactional Memory (TM); support optimistic concurrency for parallel programming in GCC. Contacts: Osman Unsal, UPC/BSC; Albert Cohen, INRIA Saclay; Wolfgang Karl, T. U. Karlsruhe; Avi Mendelson, Intel Haifa.

  The work of Martin Schindewolf has led to the opening of an official transactional memory branch of GCC, maintained by Richard Henderson from Red Hat.
- Instruction set customization and online configuration of the machine description. Contacts: Phil Barnard, ARC; Mike O'Boyle and Björn Franke, U. Edinburgh.
- Support whole-program and link-time optimization on a high-level representation. Contacts: Ayal Zaks, IBM Haifa; Paul Kelly, Imperial College.
- Design and implement a portable performance model for the multi-core era. Contacts: Basilio Fraguela, U. A Coruña; Mike O'Boyle, U. Edinburgh.
- Meta-programming support for active libraries (a.k.a. library generators). Contact: Paul Kelly, Imperial College.

Static-dynamic split compilation

GCC is not (currently at least) a Just-In-Time (JIT) compiler, and it only provides incomplete support for managed languages. A strong motivation for JIT-compilation in HiPEAC is the convergence of processor virtualization and performance portability objectives, in the context of heterogeneous multicores and hardware accelerators. There is a clear need for delayed code generation and optimization, taking into account dynamic information (including system load parameters, library calling context, data sets), and the exact target architecture instance. The compilation platform cluster is currently working on the selection of a platform to work in this area,

aiming for maximal synergy with the GCC platform; this is a joint activity with the Virtualization and Adaptive Compilation clusters. Initial participants: Ghent U., Thales, INRIA and IBM Haifa.

Transactional memory in GCC

Past efforts on transactional memory led Martin Schindewolf to start a PhD thesis at T. U. Karlsruhe, in collaboration with INRIA Saclay, U. of Bologna and RedHat (Richard Henderson, one of the most active GCC developers). We expect this will lead to joint work with the Adaptive Compilation and Programming Model clusters on automatic parallelization and data-flow parallel programming. A proposal was submitted to the fourth call of FP7 (FET proactive "teradevice computing").

10.2. Platform Mutualisation and Transfer Actions

- GCC as a common research platform: animation and training. Contact: Albert Cohen, INRIA Saclay.
    - Upcoming SSA seminar, April 2009, co-organized by Fabrice Rastello.
    - GROW workshop (with HiPEAC'09): Paul Kelly, Imperial College; Sebastian Pop, AMD Austin.
    - Planned training week (technical and hands-on tutorial) in spring 2009.
- Build an API to plug research tools into GCC. Contact: Grigori Fursin, INRIA Saclay.
- GIMPLE type system: defer lowering of abstract data structures. Contact: Paul Kelly, Imperial College.
- Single-source, multi-ISA compilation. Contact: Albert Cohen, Xavier Martorell, Ayal Zaks, Bilha Mendelson.
- Improve GCC baseline performance: priorities in alias analysis, competitive back-end (VLIW-friendly). Contact: Harm Munk, NXP Eindhoven.
- Copyright and GPL issues. Contact: Harm Munk, NXP Eindhoven.
- Interactions with the greater GCC community. GCC Summit. Research, development, coordination and training in the US, Canada, Russia, India. Contact: Ayal Zaks, IBM Haifa.
- Contributions to the HiPEAC2 roadmap. Cluster delegate: Albert Cohen, INRIA Saclay.

10.3. Summary of Activities

1st meeting, Goteborg, January 31 2008

- Invitation to Dr. Ayal Zaks from IBM Research Haifa to jointly coordinate the cluster activities.
- Setting up the cluster's initial research topics, mutualisation and transfer activities; identification of the key contact persons for each topic and activity.
- Opening of the engineering position to launch a "transactional memory for GCC" initiative, in support for more productive parallel programming and compilation research in this architecture- and runtime-system-dominated field. Martin Schindewolf from T. U. Karlsruhe was appointed to this position from April 1st to December 31st (funded by HiPEAC1).
- Re-assessment of the interaction principles within the GCC community, and the legal issues. Of course, all developments within the platform should be free software. Contributions to GCC may only make it to the official development branches if the copyright-holding organization signs a transfer agreement with the FSF.

2nd meeting, Barcelona, June 2 2008

- Jan Hubicka from SuSE (Prague) presented "Link time compilation and interprocedural optimization in GCC".
- Martin Schindewolf from T. U. Karlsruhe presented "Transactional Memory Support in the GCC", co-authored with Albert Cohen and Wolfgang Karl.
- Antoniu Pop from École des Mines presented "Improving GCC Infrastructure for Streamization", co-authored with Sebastian Pop, Harsha Jagasia, Jan Sjödin and Paul H. J. Kelly.

3rd meeting, Paris, November 28 2008

- Presentations by cluster members: Grigori Fursin from INRIA Saclay; Arutyun Avetisian from ISP Russian Academy of Sciences.
- Status report of the transactional enhancements to GCC.
- Interaction with Richard Henderson (Red Hat) who opened a new transactional-memory branch based on our design.
- Focus on the next year(s) for the cluster engineer. Proposal: CLI portability layer based on GCC4NET.
- Looking for a new cluster engineer (starting January 1st).
- Support and participation to the 20 years of SSA seminar, with a presentation of the seminar proposal by Fabrice Rastello from INRIA Rhône-Alpes.
- Coordination of GCC-based developments in the upcoming FP7 call 4 for Embedded and Computing Systems.
- Opportunity for a multi-day GCC training session sponsored by the cluster in 2009.

4th meeting, Paphos, January 29 2008

A large part of this upcoming meeting will be dedicated to the collaboration with the virtualization cluster, see "Future Work" below.

GROW workshop

The workshop on GCC for Research Opportunities (GROW) was organized on Jan. 25, 2009, in Paphos, Cyprus, together with the HiPEAC'09 conference. It was chaired by Paul H. Kelly from Imperial College and Sebastian Pop from AMD. It follows the GREPS workshop organized in September 2007 together with the PACT'07 conference, in Brasov, Romania. The workshop was coordinated with the Adaptive Compilation cluster which jointly organised the third SMART workshop.

A second GROW workshop is planned for the upcoming HiPEAC'10 conference.

Organization

Martin Schindewolf was the platform engineer from April 1st 2008 to December 31st 2008. There has not been a postdoc/engineer for the cluster since January 1st.

Funds have been allocated about equally to support travels cluster meetings and to support point-to-point collaborations. We expect more collaborations to be funded in the context of the cluster in 2009, as well as direct support for cluster-sponsored events.

The cluster had 4 meetings. The following list recalls the main events and milestones achieved during those meetings.

- Goteborg, January 31, 2008. Initial list of topics, identification of the main contact persons for each topic, commitment to support the transactional memory initiative.
- Barcelona, June 2, 2008. Two Invited speakers: Ian Hubicka from SuSE/Prague on recent developments on interprocedural and link-time optimization in GCC, and Antoniu Pop from École des Mines on automatic exploitation of stream parallelism in GCC.
- Paris, November 28, 2008. Presentations by Arutyun Avetisian from ISP, Russian Academy of Sciences, Fabrice Rastello from ENS Lyon and INRIA, and Grigori Fursin from INRIA Saclay. Adoption of HiPEAC's transactional memory developments by RedHat and opening of a new branch of GCC.
- Paphos, January 29, 2009. Launch of the inter-cluster static-dynamic compilation initiative with the virtualization cluster.

10.4. Inter-cluster collaborations

- HiPEAC PhD internship on GCC research at IBM Haifa.
- U. of Cyprus Ph.D. student visit to INRIA.
- Multiple exchanges with Imperial College (Anton Lokhmotov, Grigori Fursin, Albert Cohen, Mohammed Fellahi).
- Joint work between T. U. Karlsruhe cluster engineer and U. Bologna PhD student.
- Support U. of Cyprus and T. U. Delft student to cluster meetings.

10.5. Future Plans

Follow-up on current tasks and activities

The main collaborative and mutualisation tasks will continue.

In particular, the HiPEAC-sponsored effort on transactional memory will be extended as an independently funded PhD thesis at T. U. Karlsruhe (Prof. Wolfgang Karl and Martin Schindewolf), in collaboration with INRIA Saclay, University of Bologna (Prof. Luca Benini and Andrea Marongiu) and Red Hat (Richard Henderson, now in charge of the merge into the GCC mainline).

A complementary effort is also under way to leverage the graphite branch of GCC (polyhedral compilation) for automatic parallelization. The cluster provides an ideal context to bring together expert in program transformations, coarse-grain parallelization and run-time systems. This will include both data-level parallelism and task-level (pipeline) parallelism, through a collaboration with IBM Research (Ayal Zaks, Razya Ladelsky), Ecole des Mines (Prof. François Irigoin, Antoniu Pop), and INRIA Saclay.

Performance portability across heterogeneous multiprocessors

Moore's law drives the design many-core processors with very diverse micro architectures. To make efficient use of such processors, a parallel program often needs to be recompiled to suit the synchronization grain and memory constraints of the target. In addition, Amdahl's law drives the design of heterogeneous multiprocessors with hardware accelerators dedicated to certain kinds of computations (e.g., vector/SIMD computing). Again, recompilation is required to make use of such accelerators. Independently, system-level virtualization has emerged as a pragmatic solution to

portability, security and resource-efficiency issues. It can transparently migrate applications and running processes across different ISAs, but lacks the recompilation capabilities to address the performance issues. Conversely, system-level virtualization brings much of the required technology to enable dynamic recompilation for performance portability. In collaboration with the Virtualization Cluster, we wish to investigate technical solutions, conduct advanced research and mutualise platform development to enable this convergence of system-level virtualization and dynamic compilation. This approach also involves adaptive compilation across the byte code intermediate language. We call it *split compilation*, as it splits optimization algorithms into an offline and an online stage, attempting to leverage machine-independent optimization as much as possible, deferring the machine-dependent part for when we know (more about) the actual target. See "Split Compilation: an Application to Just-in-Time Vectorization", P. Lesnicki et al., GREPS PACT Workshop 2007; and "Liquid SIMD: Abstracting SIMD Hardware Using Lightweight Dynamic Mapping", HPCA 2007. Split compilation led to collaboration with Thales, IBM Research Haifa and INRIA supported by the Adaptive Compilation cluster.

In the short term, the compilation cluster looks forward to complement GCC with a software platform for dynamic and just-in-time compilation. Currently, CLI (ECMA standard) is considered as the starting point for a portable byte code language. But many options exist for dynamic compilation of CLI, including LLVM, Mono, STMicroelectronics' PVM, Polytechnico di Milano's ILDJIT, and GCC itself. We are also looking for a platform engineer or postdoc for duration of 12 to 24 months to conduct part of the research in development, and help animate this task.

Contact: Jonas Maebe and Bjorn De Sutter, Ghent University; Albert Cohen and Erven Rohou, INRIA; Ayal Zaks, IBM Research Haifa; Sami Yehia, Thales R&T.

Postdoc activities

We plan to hire a postdoc by the end of May 2009. The postdoc will focus on the JIT-compilation initiative of the cluster.

June-July 2009: the cluster postdoc will first work on a detailed survey of JIT-compilation and language-virtualization platforms. Such a survey is under way in collaboration with the virtualization and adaptive compilation clusters, but a deeper examination of the compilation-side of the available infrastructures is necessary. A common platform will be selected at the end of July, aiming for maximal interoperability with the existing GCC platform.

August-January 2009: the cluster postdoc will support the platform efforts on *split compilation*, an emerging research in between static, adaptive and dynamic compilation. Currently, this will directly involve 12 people from INRIA, U. Ghent, U. Edinburgh, IBM, Thales and STMicroelectronics. We hope this work will result in paper submissions to the best conferences this fall.

10.6. Progress Indicators

- # active members: 47
- # companies involved: 7
- # publication output: 38
- # joint papers: 12 (2 industry)

11. Reliability and availability task force

11.1. Description of scientific issues

Technological developments are confronting us with the challenge of building computer systems made of unreliable parts. In tomorrow's computing world failures will not be exceptional due to many and frequent causes such as: soft-errors, process variation, wear-out, hardware and software bugs, incomplete specifications, impossibility to simulate all use cases etc. An additional challenge stems from the wide diversity of computing systems, high-end supercomputers to low end embedded devices, each having different reliability and availability (R&A) requirements. The worst case forecast is that R&A constraints may obviate the benefits of technology scaling and slow down economic growth in several ICT driven markets. Consequently, future computing systems need to be dependable. More specifically, need to operate correctly and satisfactory in the presence of faults, capable of on-line detect, repair and recovery from faults, have small down time, and do so at high dependability/euro ratio. The purpose of this task force is to promote awareness and research activity within HiPEAC related to R&A. The instruments we are using to achieve our goal are:

- Task-force meetings, usually to be held in conjunction with other HiPEAC activities (cluster meetings or conference).
- Task-Force Web page.
- Task-Force Mailing List.
- Contributing to the HiPEAC roadmap.
- Management.
- Organize courses, workshops, tutorials related to R&A within HiPEAC such as summer school, conference and cluster meetings as well at international forums.

11.2. Summary of activities

Meetings

The task force had three meetings in Barcelona, June 3 2008, in Paris, December 5 2008, and in Paphos, January 29, 2009. Each meeting was attended by around 30 researchers both form industry and academia. All meetings include a short introduction about reliability and availability and an overview of the role/goals of the task force. Other issues in the agenda of the various meetings are as follows:

1st meeting in Barcelona:

1) an open discussion about important issues relevant to R&A, and

2) short presentations by M. Vinov IBM, E. Ozer ARM, A. Billas Forth, B. De Sutter Ghent, Osman Unsal BSC, Veerle Desmet Ghent, D. Bertozzi Ferrera.

One of the key conclusions of the meeting is that R&A issues are quite diverse and cover the whole spectrum of any computing system, including programming, compiler, operating system, architecture, hardware and circuits.

2nd meeting in Paris:

1) a report about a reliability related workshop at Micro 2008 about Process Variation.

2) short presentations by B. Falsafi EPFL, A. Zaks IBM, M. Michael UCY.

During the meeting it was suggested to arrange a TF meeting mainly devoted on input/presentations from industrial partners.

3rd meeting in Paphos was mainly devoted to input from industrial partners that included presentations by D. Pandini ST, S. Mamagkakis IMEC and B. Mendelson IBM.

Other Output

Some other results that are direct outcome of the task force activity are:

- HiPEAC members organized the Workshop on Dependable Architectures in conjunction with Micro-41 in Lake Como, Italy, Nov. 2008. www.cs.ucy.ac.cy/carch/wda08
- During HiPEAC conference 2009 they were two R&A activities taking place for the first time: the 1st Workshop on Design for Reliability (DFR) www.eng.ucy.ac.cy/theocharides/dfr09 and a tutorial by Intel experts Joel Emer and Shubu Mukherjee on reliability.
- Contributed to the preparation of the TF reports for HiPEAC deliverables.
- The web page of the task-force as well as a mailing-list is up.
- The task-force has presently 27 registered members.
- More details about the meetings and the presentations can be found at the task-force's web http://www.hipeac.net/TF_reliability

11.3. Future Plans

As part of the next meeting we will have an open discussion to assess the output of the TF and discuss other means to achieve the TF objectives. One instrument we will propose is to encourage members to highlight using the TF mailing list interesting developments/news/research results related to R&A.

In the next meeting we will also have a brief report on SELSE5, an annual IEEE reliability related workshop held in US.

There is intention by HiPEAC members to organize the Workshop on Dependable Architectures (WDA) in conjunction with Micro-42 in December 2009 and the Workshop on Design for Reliability (DFR) with HiPEAC 2010. We will encourage HiPEAC members and in particular members of the TF on R&A to contribute to both events.

11.4. Progress Indicators

- # active members: 28
- # companies involved: 4
- # publication output: 14
- # joint papers: 5

12. Applications task force

12.1. General Objectives

HIPEAC has a unique strength in bringing together research on architectures, programming models, compilers and tools. In order to use this strength wisely, we want to go against the tide and connect software research to architectural research using applications and algorithms as drivers. We will promote some – carefully selected, realistic and representative of user domains – applications to be open to the HiPEAC community, the EU funded projects and also to the public domain.

Our goal is to increase source code sharing and collaboration within the HiPEAC community, in order to provide a common set of applications to evaluate new research ideas. A concrete effort that all HiPEAC clusters will benefit from is to define a benchmark suite composed by a set of well characterized applications and input data sets.

Due to the HiPEAC organization in clusters, one of the most crosscutting topics is the need for representative applications on top of which we can show our research progress in a way that can be compared to actual technologies, reproduced in multiple sites, and discussed among academia and industry. We would like to agree on an applications-driven research agenda.

To promote a real impact on all clusters, we look forward to have at least one active contact person at each of the nine HiPEAC clusters. Our Task Force meets periodically, coordinates and organizes activities to further develop activities on promotion and dissemination of high quality benchmarks for new computer architectures.

During the time span of the Task Force, we would like to have enough momentum to be able to cover (and have enough member's teams to focus on) scientific parallel applications, industrial high-performance applications, as well as applications for embedded systems, for low power systems, for real time, etc . We know that good application classification is crucial to our success, for there is a workload convergence in benchmark candidates, meaning that they share a common set of mathematical models and techniques, and further down, similar numerical algorithms and primitives.

Being a Network of Excellence, we want to nurture the EU HPC ecosystem and especially companies that are active in the area. We are aware that new technologies enable new applications, and that we have to look at new user requirements. We want to give a first class status to new metrics like: scalable performance, programmability, compile/optimization time, power efficiency, etc. But we also want to address what really matters to users, like response time. That's why we will be involved in the development of the HiPEAC Roadmap. In fact, it will allow us to double check which areas are considered important by the Roadmap and the Task Force.

As an example, we consider that benchmarking is needed to evaluate new research solutions on heterogeneous architectures. There is a debate if new programming models are needed, how compilers have to deal with multiple ISA applications, and how loaders and runtime/OS have to manage the heterogeneity of these new systems. Benchmark suites for such heterogeneous architectures (like the Cell BE or the GPU platforms) are missing and we don't know any other body that is trying to build them.

We all have experienced that porting applications to heterogeneous CMP is a time-consuming and error-prone task. To optimize them and get expected speedups has only been done by expert

architects and programmers for very few applications. Those codes are rarely shared. We would like to change this behaviour, and encourage people to devote time to clean up their codes, make them available to the Task Force first, which will act as a discussion forum on the appropriateness of the code to become a benchmark, to be ported to other platforms, to validate realistic input data sets, and to agree on a set of experiments that can be reproduced and compared. Then, after a voting process, we will promote some of the most representative benchmarks to be open to the HiPEAC community, the EU funded projects and also to the public domain. This final selection can be guided by a Task Force Board, composed by the representatives of the nine HiPEAC clusters, the more active members in the Task Force, the industry representatives, and the Task Force Chair.

12.2. Current status

The task force already has a couple of applications that are being explored, mainly NAMD tool for computing the molecular dynamics of bio-molecules, and WRF, extensively used for detailed weather and weather-related simulations (such as pollution patterns).

Chosen applications support different programming models - both shared-memory and distributed-memory ones. This in turn enables a methodological comparative study, characterizing the applications at all levels: from an underlying system characterization based on CPU and memory behaviour, through examining the effectiveness of a programming model to tackle an algorithmic problem, and up to the software representation of the algorithmic and mathematical formulation of the problem at hand.

We are currently setting up the TF's website at http://hpc.ac.upc.edu/tfa/index.php/Main_Page

12.3. Original work plan

During the project duration, we plan to achieve:

- Identify the set of Application Domains that can benefit from this effort and decide which ones we can deal with. Look at realistic applications from financial, science, medical, educational, consumer, etc environments.
- Identify Programming Models and Computer Architectures that are relevant to this study in our community
- Select the platforms we are going to test taking into account the features that are significant: multi-core processors, heterogeneity, computation scalability, memory hierarchy, application domain accelerators, communication and synchronization mechanisms, network-on-chip, power constraints, etc
- Define and establish a consensus in our field, the adequate metrics to evaluate new platforms that have made obsolete the speedup only previous measurements.
- Create groups in charge of encouraging the port of applications to different programming models and to different platforms
- Promote a repository of applications, where HiPEAC members contribute, download and use project management and bug tracking tools
- Start the discussions at the Task Force Board to select applications that fulfil some quality criteria and are representative of an area of knowledge
- Promote the set of selected applications to become benchmarks suites supported by HiPEAC in order to use them in our collaborative research and publications.
- Be proud of and disseminate the “EU HiPEAC Benchmark Suite for ..." (programming model, platform, domain…)

- Participate at the HiPEAC Roadmap meetings and contribute to the writing.

Task Force meetings and related activities

- Regular conference call meetings of the Task Technical Applications Board
- Task Force on Applications meetings at the HiPEAC Computing Systems Weeks and HiPEAC Conference.
- Hands on Tutorial for PhD. Students and feedback session from benchmark users at the HiPEAC Summer School.
- Organize tutorials and promote participation at the International Workshop on OpenMP (IWOMP), the Programmability Issues for Multi-Core Computers (MULTIPROG) Workshop, and other similar conferences and events that our task force members organize.

Task Force Repository and Forum

- Increase source code sharing and collaboration within HiPEAC community
- Provide a common set of applications to evaluate new research ideas
- Define a benchmark suite composed by a set of characterized applications and input data sets
- Provide a portable/extensible framework to set up experiments and retrieve experimental results
- Establish a discussion forum to get opinions about the status of the applications, the requirements to run on a specific platform, the applicability to a domain, the accuracy of the performance results, etc
- Setup the voting and promotion mechanisms to select the applications that will be part of the Benchmark Suites.

Task Force Web presence and mailing lists

The web page of the task-force as well as a mailing-list is up. The task-force has presently more than twenty registered members.

12.4. Summary of Activities

First meeting at the HIPEAC Computing Systems Week, Barcelona, Spain, June 2008

At the Barcelona Clusters meeting, we had our first presentation of the Task Force goals in front of public. More than 50 people attended. Then, in the afternoon, we had a session on performance evaluation, where seven presentations (from BSC, UPC, IBM, Thales, FORTH) on ongoing work and tools showed the interest to collaborate.

At this meeting, Thales joined the Task Force. Their interests are the following: “The project you are describing is quite appealing for us, and I like its application centric approach: providing an adequate and sufficiently large application base for heterogeneous platforms and considering future application challenges and their different constraints (real time dynamicity, power,...) can greatly enhance the research in this area. I would also add to what you propose a characterization of the different computation patterns in these real life applications. This is much in the line of the dwarfs proposed by Berkeley but in the context of more complex applications (I am not sure doing the research based on isolated dwarfs or kernel is adequate for the heterogeneous multicore platforms we are thinking of). I also agree that properly describing the applications and finding a way to representing it independently of the parallel programming model used (unlike PARSEC benchmarks) would be of great help to the community.”

Meeting at the ACACES Summer School, L'Aquila, Italy, July 2008

Among the participants at the ACACES Summer School, we had bilateral contacts (UPC, BSC, Intel, IBM, FORTH, NTNU and Thales) and organized a formal meeting to continue to define the goals of the Task Force. Some of the participants agreed on preparing a proposal for a STREP at next call of the EU Framework Program 7.

Our partner from NTNU, Norway, has introduced Numerical Rocks to participate in the Task Force: "We met at ACACES 2007 and in Barcelona this June … we discussed the difficulty of getting companies to contribute with applications to the application task force. Now, my friends at Numerical Rocks … will attempt to contribute with a parallel HPC application from the oil industry. The application will be adapted so that it can become a part of the HIPEAC benchmark. The underlying algorithm will be a 3D Random Walk simulation"

2nd Workshop on Programmability Issues for Multi-Core Computers (MULTIPROG-2009)

The organization of this workshop was collocated with the HiPEAC 2009 Conference, Paphos, CYPRUS, January 25-28, 2009

Preliminary status on the Task Force on Applications repository

We currently have a first preliminary SVN repository with close to 20 applications for the Cell BE architecture. We have developed a framework to run them on the IBM blades as well as the Sony PS3. They have been contributed by HiPEAC partners that were using them to run their research: BSC, UPC, FORTH, IBM, and UEDIN. That means we have already started to get inputs (code) and comments from the Programming Models and Runtime Cluster, from the Interconnects Cluster, the Multi-core Architecture Cluster, and the Adaptive compilation Cluster.

12.5. Planned activities

Following an intense porting effort, we expect to begin the system characterization phase in the coming weeks, working on two inherently different architectures: the SGI ccNUMA machine with 128 processors, and the MareNostrum supercomputer supporting both shared-memory and distributed-memory applications with its ~2000 4-way nodes.

12.6. Progress Indicators

- # active members: 23
- # companies involved: 2
- # publication output: 46
- # joint papers: 21

13. Low power task force

13.1. Description of scientific issues

The Task Force on Low Power is concerned with supporting research on Low Power orthogonally to the many other research areas within HiPEAC. In this respect, the TF is concerned with the required infrastructure and methodologies for enabling such research. In particular, central to the TF are: modelling, simulation, and measurement of power consumption in computer systems. In addition, ground braking low-power techniques are also of special interest to the TF for quick dissemination among members and application of such techniques in a wide range of contexts.

13.2. Description of investigated approaches

Towards these goals the TF thought its meetings is developing a strategy to create (by combining the work of its members) a common simulation/modelling/measurement infrastructure, freely available in HiPEAC.

The website of the TF will play a central role in the dissemination and promotion of this infrastructure and we are in the process of building it for future use.

13.3. Summary of activities

The main vehicle for the TF work are its meetings. The task force already held three major meetings:

Kickoff meeting: Computer System Week, Barcelona Monday, June 2 2008. Presentation of the aims of the task force. Presentation of work on low power by several participants.

2nd meeting: Computer System Week, Paris. Nov 28 2008. Discussion on specific targets of the task force. Rainer Leupers (RWTH Aachen) presentation on low power directions. Pierre Michaud (INRIA) presentation of the implications and projections of technology scaling on low/power and temperature.

3rd meeting: HiPEAC Conference, Jan 29 2009. Presentation of work by Yanos Sazeides (University of Cyprus) and presentation of methodologies for measuring power consumption using performance counters (in multicores) by Sally McKee (Chalmers). Discussion on research interests of the participans for possible collaborations.

13.4. Future plans

There is significant on-going work on modelling, simulation, and measurement of power consumption by different HiPEAC members. In the immediate future, TF work will integrate, to the extent possible, such approaches for the benefit of the HiPEAC community.

In terms of organization the TF will hold its regular meetings coinciding with the major HiPEAC events (Computing System Weeks, Conference, Summer School). In addition the TF will promote a Workshop on low power targeted to the interests of the HiPEAC community.

13.5. Progress Indicators

- # active members: 16
- # companies involved: 1

- # publication output: 11
- # joint papers: 3

14. Education and training

Education and training is the preparation of the future – of those who will carry on the work of the current generation; hence, it is at least as important, if not more important, than current work. In addition, graduate education is not only the preparation of the future researchers, but also one of the main constituents that support current research. The concern of this Task Force is University Education, undergraduate and graduate, and the training of the senior students or of the young researchers, in all the sub-fields of Computer Science and Engineering that HiPEAC covers.

14.1. Description of issues

Objectives of this task force: promote communication and hence sharing of experiences, recommendations, teaching methods, material, and tools among the HiPEAC members and more generally the community at large, on all aspects of education and training. Out of this communication, additional actions may hopefully arise in the future, outside the scope of the HiPEAC project itself but complementary to its goals, such as training networks or joint graduate programs.

Method of Work: the members of the Education and Training TF meet at least biannually (during cluster meetings: 3 June 2008 in Barcelona, 27 Nov. 2008 at Thales; at the conference: 29 Jan. 2009 in Paphos, Cyprus), they communicate through the TF Mailing List, they plan and organize activities, and they construct the TF Web Page which is intended to become something like a repository, providing to the community at large pointers to and information about: teaching material, tools, methods, experiences, recommendations, announcements, etc. Below we list the year-1 activities, and then the future plans that resulted from of our discussions during meetings.

14.2. Summary of activities

besides our three meetings above, and besides initial set-up of the mailing list and web site, we have:

- **Instructors List:** We have collected a substantial number and will collect more names and addresses of instructors in Europe who teach "HiPEAC" courses, so as to be able to send information to them; it is expected that this instructor set will be much wider that the HiPEAC members, since there are many instructors who are not themselves actively doing research in HiPEAC topics.
- **Marie-Curie Initial Training Network (ITN) Proposal** several of the HiPEAC partners submitted (2 Sep. 2008) the ``Training through Research in Advanced Computing Systems" (TRACeS) proposal to the ``People" Programme, for training in the HiPEAC topics (unfortunately, it was not approved, but we do hope for the future).
- **Marie-Curie ITN Projects:** three HiPEAC partners, FORTH, BSC, INRIA and two HiPEAC member organizations Technion, EPFL participate in the ``SCALing by means of Ubiquitous Storage" (SCALUS) ITN and in the ``Theoretical Foundations of Transactional Memory" (TransForm) ITN (the latter one coordinated by a HiPEAC partner); both ITN's are starting in 2009.
- **Erasmus Mundus Proposal:** a HiPEAC partner and a HiPEAC member, KTH, BSC together with other ``Cluster" Consortium Universities, have been preparing an Erasmus Mundus II proposal in Parallel and Distributed Systems.

14.3. Future plans (both in terms of science and organization)

Besides a continuation of the above activities, also consider the following ideas:

- **Course(s) on How to Teach** e.g. next to the HiPEAC Conference, during the Workshops day, for half or full day, 2 to 3 hours per topic. Topics may include e.g.: bottom-up versus top-down teaching of digital design, organization, architecture; parallel programming; computer architecture; virtualization; graphics processing units.
- **Competitions on How to Explain a topic:** during the above Workshops/Tutorial days, or during Cluster Meetings, organize competitions on how to best explain, in 5 or 10 minutes, a difficult topic, like e.g. out-of-order processor architecture, superscalar, virtualization.

14.4. Progress Indicators

- # active members: 8
- # companies involved: 2
- # publication output: 2
- # joint papers: -