Report on Second Project Workshop (Year 2).

D8.5

Due date of deliverable: 30th September 2013

Start date of project: October 1st, 2011

Type: Deliverable
WP number: WP8
Task number: T8.3

Responsible institution: –
Editor and editor’s address: John M’Call

Project co-funded by the European Commission within the Seventh Framework Programme

<table>
<thead>
<tr>
<th>Dissemination Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PU</td>
<td>Public</td>
</tr>
<tr>
<td>PP</td>
<td>Restricted to other programme participants (including the Commission Services)</td>
</tr>
<tr>
<td>RE</td>
<td>Restricted to a group specified by the consortium (including the Commission Services)</td>
</tr>
<tr>
<td>CO</td>
<td>Confidential, only for members of the consortium (including the Commission Services)</td>
</tr>
</tbody>
</table>
Executive Summary

This deliverable presents a summary of the second ParaPhrase workshop, which was co-located with HiPEAC 2013 and took place in Berlin (Germany) on Monday 24 January 2013.

The workshop was entitled "HLPGPU: High-level Programming for Heterogeneous and Hierarchical Parallel Systems", and consisted of 6 presentations. Chapter 1 contains a brief description of the workshop and its aims. In Chapter 2, we provide the programme and abstracts of each presentation. A list of publications arising from this work is given in Chapter 3.
Chapter 1

Introduction

As the ParaPhrase project ends its second year, we provide a report on the second ParaPhrase workshop. The annual workshops, co-located with a leading conference is part of our wider programme of dissemination of ParaPhrase results to the wider community of scientists, researchers, students, and the public in general.

Each ParaPhrase Workshop is intended to be a showcase for the research results in the project, as well as an open forum for the discussion of possible avenues for future work. In order to maximise its impact, we intend that that each ParaPhrase Workshop should:

- be multi-disciplinary, featuring a well-balanced and succinct agenda, which displays the project progress in key areas such as parallel patterns, multi-core architectures, applications, and refactoring.
- help to attract interest in the project findings; and,
- allow project members to swiftly publish their results.

Having been conceived to allow the scientific community access to the project results, any effective workshop must include a strong component of diffusion. Such diffusion efforts must be carried out during and after workshop, and can be typically divided into:

1. The workshop event
2. Subsequent publication of results

In Chapter 2 of this report, we present the Workshop Programme, with abstracts of each accepted presentation. Finally in Chapter 3 we list publications by ParaPhrase authors arising from work presented at the workshop.
Chapter 2

Second ParaPhrase Workshop: Programme and Abstracts

As with the first ParaPhrase Workshop (reported in Deliverable D8.3) the second ParaPhrase Workshop was an open event co-located with the HiPEAC conference http://www.hipeac.net/content/hipeac-2013, which took place in Berlin in January 2013. The workshop was entitled "High-Level Programming for Heterogeneous and Hierarchical Parallel Systems" (HLPGPU).

The HiPEAC network is “an FP7 project which gathers more than 1000 researchers in computing systems in Europe. It is the biggest such network in the world, offering training, mobility support, dissemination services, and abundant networking facilities to its members.” Further information is available at www.hipeac.net.

2.1 HLPGPU Programme

The workshop contained six presentations and ran 10:00 - 13:00 on Tuesday 22 January 2013. Around 30 HiPEAC delegates were in attendance. They were as follows:


   **Abstract:** Parallelization of Machine Learning methods is an active research area, fuelled by the need for acceleration of complex computations, and the constant growth of numbers of samples and features in available data sets. Because several Machine Learning methods are general in the sense that they can be reused again and again for new learning tasks, it is common to collect these methods in libraries, e.g. the library mlpp at SCCH. Such libraries are intended to be used by several users on different hardware.
platforms. As a result, it is important that their parallelization does not introduce dependence on a restricted set of deployment environments. The ParaPhrase approach, besides having advantages in the modeling of parallelism and the parallelization process, promises to provide the needed flexibility with respect to supported hardware, by targeting multicore machines, distributed clusters, and hardware accelerators like GPGPUs. The process of parallelizing one method in mlpp, Coordinate Descent, is illustrated in the presentation.

2. Intelligent Static Mapping for Heterogeneous Environments using MCTS Methods, M. Goli, J. McCall

Abstract: Skeletal based programming is a high level parallel programming model for parallel and distributed computing. It abstracts commonly used patterns of parallel computation, communication, and interaction. Moreover, it provides top-down design composition and control inheritance throughout the whole structure. Applying a heterogeneous environment to parallel computing applications based on algorithmic skeletons has long been of interest to the research community. However, utilising the environment and selecting the most appropriate combination of patterns is challenging and not trivial at all. This can be categorised as a form of static mapping problem. MCTS is an approach developed in AI Games to efficiently select moves in large game trees where full evaluation of the tree is computationally intractable. An extensive recent survey of MCTS applications can be found in [1]. MCTS uses Monte Carlo random walks to evaluate states only at the endpoints of each walk. Recently MCTS has been applied to planning and scheduling problems [2], hence the relevance to the static mapping problem. Our aim is to make available, in design time, a means of identifying an optimal static mapping with regards to the throughput by using MCTS approaches to generate an optimum pattern based on the available architecture.


Abstract: In this talk we present a new programming methodology for introducing and tuning parallelism in Erlang programs, using source-level code refactoring from sequential source programs to parallel programs written using our skeleton library, Skel. High-level cost models allow us to predict with reasonable accuracy the parallel performance of the refactored program,
enabling programmers to make informed decisions about which refactorings to apply. Using our approach, we demonstrate easily obtainable, significant and scalable speedups of up to 21 on a 24-core machine over the sequential code.


Abstract: A methodology suitable to split computations from structured data parallel pattern (map and reduce) is shown, suitable to distribute the generated tasks on CPU and GPU cores. The methodology uses profiling and CPU/GPU performance models to devise the percentage of tasks to be directed to CPU and GPU cores such that the overall execution time of the parallel pattern is optimized. Experimental results are shown that demonstrate feasibility and soundness of the approach.

5. Memory affinity for scalable lock-free memory allocator, F. Tordini, M. Aldinucci and M. Torquati

Abstract: Modern computers are built upon multi-core architectures. Achieving peak performance on these architectures is hard and may require a substantial programming effort. The synchronisation of many processes racing to access a common resource (the shared memory) has been a fundamental problem on parallel computing for years, and many solutions have been proposed to address this issue. Non-blocking synchronisation and transactional primitives have been envisioned as a way to reduce memory wall problem. Despite sometimes effective (and exhibiting a great momentum in the research community), they are only one facet of the problem, as their exploitation still requires non-trivial programming skills. With non-blocking philosophy in mind, we propose high-level programming patterns that will relieve the programmer from worrying about low-level details such as synchronisation of racing processes as well as those fine tunings needed to improve the overall performance, like proper (distributed) dynamic memory allocation and effective exploitation of the memory hierarchy.

6. Towards the development of FastFlow on distributed virtual architectures, S. Campa, H. Gonzalez-Velez, A. M. Popescu and M. Torquati

Abstract: In this paper we investigate the deployment of FastFlow applications on multi-core virtual platforms. The overhead introduced by the virtual environment has been measured using a
well-known benchmark both in the sequential and in the FastFlow parallel setting. The overhead introduced for both sequential and parallel execution CPU and memory-intensive applications is in the range of 2-30 %, while execution speedup is almost preserved. Additionally, we have ported the FastFlow benchmark to a cloud-distributed environment in which a task-intensive application has been tested and the performance compared with the corresponding run on a smaller cluster of multi-core machines without virtualisation. From a parallel programming perspective, we have demonstrated how a unique programming framework based on the structured parallel programming paradigm can cope with very different kind of target architectures without any (or minimal) code intervention.

In addition to the HLPGPU workshop, a project poster was exhibited in the main HiPEAC conference area. The poster presented recent ParaPhrase information and was manned at all times by ParaPhrase participants.
Chapter 3

Related Publications

In this section we list publications co-authored by ParaPhrase participants that have flowed from the work presented a HLPGPU 2013.


4. M. Danelutto, Attacking the programming model wall, keynote talk at PDP 2013, Belfast

5. Marco Danelutto, Structured parallel programming in FastFlow, Domain specific languages summer school, 8-20 July 2013, Babeș-Bolyai University, Cluj-Napoca, Romania


7. M Goli, J McCall, C Brown, V Janjic, K Hammond, Mapping parallel programs to heterogeneous CPU/GPU architectures using a Monte Carlo Tree Search, 2013 IEEE Congress on Evolutionary Computation (CEC), 2932-2939