Project no. 248828

ADVANCE

Strategic Research Partnership (STREP)
ASYNCHRONOUS AND **D**YNAMIC **V**IRTUALISATION THROUGH PERFORMANCE
ANALYSIS TO SUPPORT **C**ONCURRENCY **E**NGINEERING

1^{st} User Community Workshop

D13

Due date of deliverable: July 31^{st}, 2011
Actual submission date: July 31^{st}, 2011

Start date of project: February 1^{st}, 2010

Type: Deliverable
WP number: WP8

Responsible institution: HERTS
Editor & and editor's address: Raimund Kirner
University of Hertfordshire, College Lane
Hatfield, AL10 9AB, United Kingdom

Version 1.0 / Last edited by Raimund Kirner / July 30^st, 2011

Project co-funded by the European Commission within the Seventh Framework Programme		
Dissemination Level		
PU	Public	$\surd$
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Revision history:

Version	Date	Authors	Institution	Section affected, comments
0.1	08/06/2011	Raimund Kirner	HERTS	Initial version
1.0	30/07/2011	Raimund Kirner	HERTS	Feedback included

Reviewers:
Frank Penczek, Sven-Bodo Scholz, Alex Shafarenko, Clemens Grelck

Tasks related to this deliverable:

Task No.	Task description	Partners involved°
WP8	Community Building	HERTS*, UvA

°This task list may not be equivalent to the list of partners contributing as authors to the deliverable
*Task leader

Executive Summary

Within the **Advance** project we extent key technologies for the development and execution of concurrent programs to deploy the optimisation potential of runtime optimisations based on statistical program information.

In this document we describe the first user community building activity for the key technology used within the **Advance** project, which was held on 29.-31. March 2010 at the University of Hertfordshire. Attendees of the workshop learned the fundamental concepts of programming parallel applications with S-Net and/or SAC. The workshop was open to the public, but at the same time all **Advance** project partners have been invited in order to learn the technology relevant for carrying out the project.

Contents

Chapter 1

Introduction

From the programmer's perspective, the two main technologies used in the **Advance** project are S-Net and SAC. S-Net [4, 5, 1] is a coordination language that is targeted for course-grain parallelism. S-Net can be also used to reuse legacy code for parallel execution with some code modifications in order to split a program up into different components, called boxes in S-Net terminology. SAC [3, 2] on the other side is a functional programming language that kind of mimics the syntax of ISO C, to give programmers a feeling of familiarity on their first contact. The main source of parallelism in SAC programs is provided by array operations.

Within the **Advance** project the goal is to extend the S-Net and SAC technologies with statistical feedback in order to trigger performance optimisations. The approach is meant to be validated by industrial case studies provided by the industrial partners. Other academic partners will adapt their tools or frameworks to be used within the project.

To get an efficient start on that activity, we decided to organise a user community workshop that introduces participants with the basics of parallel programming with S-Net and SAC. Especially all program partners have been invited to participate. The workshop was organised in a BYOM style (Bring Your Own Mug), which means that participants bring with them their own laptop for programming.

In this document we give a brief summary of the first user community workshop, organised from 29.-31. of March 2010 at the University of Hertfordshire.

Chapter 2

Overview of the Workshop

The first user community workshop was held on three days, starting with 29.03.2010 and ending on 31.03.2010. The first user community workshop was hosted in combination with the first project workshop of the **Advance** project at the University of Hertfordshire. The workshop program is shown in Figure 2.1.

The workshop effectively lasts two full working days. The first day started with an introduction to SAC. The rest of the day was spent on programming some simple SAC programs.

On the second day of the workshop first another SAC programming session has been done in order to finish the examples. The afternoon of the second day was dedicated to the S-Net technology, starting with an introduction to S-Net, followed by the first S-Net programming session.

The third day of the workshop started with an advanced lecture on the design of S-Net. Finally, another S-Net programming session followed that day.

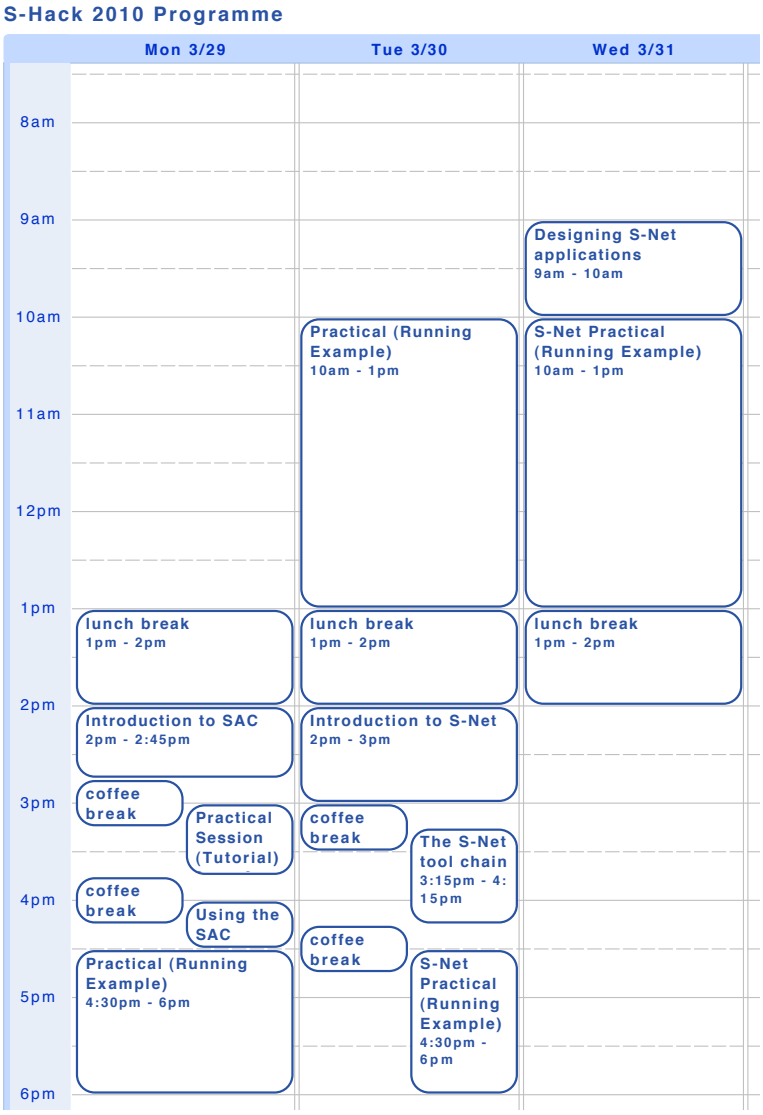

Figure 2.1: Program of the 1^{st} User Community Workshop

Chapter 3

Participants

The user community workshop was organised in accordance to the project work plan and attended by representatives of the project partners according to the table below. One external participant (Prokesch from TU Wien) was present, who was in fact an external project student involved in the ADVANCE work programme.

Name	**Organisation**
Heinz Hertlein	BioID
Daniel Scheibli	SAP
Bernd Scheuermann	SAP
Robert de Groote	UTwente
Mark Westmijze	UTwente/Philips
Bernhard Moser	SCCH
Volkmar Wieser	SCCH
Steffen Jost	St. Andrews
Daniel Prokesch	TU Wien
Clemens Grelck	UvA
Sven-Bodo Scholz	UH
Alex Shafarenko	UH
Frank Penczek	UH
Stephan Herhut	UH
Carl Joslin	UH
Daniel Rolls	UH
Raimund Kirner	UH
Artjom Sinkarov	UH
Jing Guo	UH

Table 3.1: List of Workshop Partcipants

Chapter 4

Summary of the Programming Examples

4.1 Programming in SAC

The SAC programming sessions have been centred around several subtasks of a Mandelbrot programming example. The display part of the Mandelbrot example has been provided, so the participants could focus on the algorithmic part of the Mandelbrot algorithm, expressed in the SAC language.

4.2 Programming in S-Net

The S-Net programming examples then took over the SAC solution to be converted into an S-Net-based program by decomposing the program into several boxes, with the individual boxes still implemented in SAC. This task of decomposition is actually a good introduction for the programming work of the industry partners within the **Advance** project, where legacy applications are also transformed into S-Net-based programs, or the algorithms are re-implemented in SAC.

An overview of the programming tasks for the S-Net part of the workshop is given in Figure 4.1.

Task 1
As a warm-up exercise, take a look at `mandelbrot_simple.snet`. This tiny program uses the mandelbrot implemetation of the SaC exercises as one monolithic box. Open `boxes/mandelbrot.sac` and scroll to the bottom to get an idea of how to "S-Netify" existing source code. Take a look at the provided Makefile to find out how the S-Net application is built.

Task 2
Develop an S-Net that models the data-flow of your SaC mandelbrot implementation. Convert your SaC functions into a set of boxes for use with your developed S-Net.

Task 3
Try to exploit concurrency as much possible by identifying independent tasks of your program and arranging these tasks in parallel.

Task 4
Exploit even more concurrency by decomposing the plane into multiple parts and apply the mandelbrot algorithms to each of the parts independently. After processing, merge the results back into one big plane.

Task 5
Explore the MPI capabilities of S-Net. Make use of the placement combinator and try your program of Task 4 on our cluster.

Task 6
Can you come up with a dynamic scheduling of processing tasks? Investigate what impact different scheduling approaches have on the runtime of your program.

Figure 4.1: Summary of the S-Net Programming Exercises

Chapter 5

Conclusion

In this document we provided a summary of the first user community workshop within the **Advance** project, held on 29.-31. March 2010 at the University of Hertfordshire, the coordinating institution of the project.

The purpose of the workshop was to broaden the user community for S-Net and SAC the two key technologies for parallel programming used in the **Advance** project. The workshop was characterised by a broad attendance of members of institutions of the project consortium.

The programming examples were specifically prepared to allow the participants a fast success experience in programming with S-Net and SAC. The participants gained a first impression of the merit of programming with S-Net and/or SAC.

Bibliography

[1] C. Grelck and A. Shafarenko. Report on S-Net: A Typed Stream Processing Language, Part I: Foundations, Record Types and Networks. Technical report, University of Hertfordshire, Department of Computer Science, Compiler Technology and Computer Architecture Group, Hatfield, England, United Kingdom, 2006.

[2] Clemens Grelck. Shared memory multiprocessor support for functional array processing in SAC. *Journal of Functional Programming*, 15(3):353–401, 2005.

[3] Clemens Grelck and Sven-Bodo Scholz. SAC: A functional array language for efficient multithreaded execution. *International Journal of Parallel Programming*, 34(4):383–427, 2006.

[4] Clemens Grelck, Sven-Bodo Scholz, and Alex Shafarenko. A Gentle Introduction to S-Net: Typed Stream Processing and Declarative Coordination of Asynchronous Components. *Parallel Processing Letters*, 18(2):221–237, 2008.

[5] A. Shafarenko, S.B. Scholz, and C. Grelck. Streaming networks for coordinating data-parallel programs. In I. Virbitskaite and A. Voronkov, editors, *Perspectives of System Informatics, 6th International Andrei Ershov Memorial Conference (PSI'06), Novosibirsk, Russia*, volume 4378 of *Lecture Notes in Computer Science*, pages 441–445. Springer-Verlag, Berlin, Heidelberg, New York, 2007.