AFRL-AFOSR-UK-TR-2011-0048

Programming Models for Heterogeneous Multicore Systems

Jesus J. Labarta

Barcelona Supercomputing Center (BSC)
Computer Science Department
Jordi Girona 29
Barcelona, Spain 08034

EOARD GRANT 09-3075

August 2012

Final Report for 02 May 2009 to 02 May 2011

Distribution Statement A: Approved for public release distribution is unlimited.

Air Force Research Laboratory
Air Force Office of Scientific Research
European Office of Aerospace Research and Development
Unit 4515 Box 14, APO AE 09421

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188

1. REPORT DATE ***(DD-MM-YYYY)***	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
08-08-2011	Final Report	2 May 2009 – 02 May 2011

4. TITLE AND SUBTITLE **Programming Models for Heterogeneous Multicore Systems**	**5a. CONTRACT NUMBER** FA8655-09-1-3075
	5b. GRANT NUMBER Grant 09-3075
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Professor Jesus J. Labarta	**5d. PROJECT NUMBER**
	5d. TASK NUMBER
	5e. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Barcelona Supercomputing Center (BSC) Computer Science Department Jordi Girona 29 Barcelona, Spain 08034	**8. PERFORMING ORGANIZATION REPORT NUMBER** N/A
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) EOARD Unit 4515 BOX 14 APO AE 09421	**10. SPONSOR/MONITOR'S ACRONYM(S)** AFRL/AFOSR/RSW (EOARD)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S) **AFRL-AFOSR-UK-TR-2011-0048**

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This report results from a contract tasking BSC as followsThe objective of this grant was to study a programming model and performance analysis tools to productively exploit the parallelism in multicore systems. The work is based on the StarSs programming model proposal by BSC and the performance analysis tools also developed at BSC, The focus was on three areas: 1) Intelligent analysis techniques in performance data analysis, 2) Locality aware scheduling, and 3) Applications. The StarSs programming models is a general node level parallel programming model based on pragmas annotating otherwise standard C programs. The annotations encapsulate certain computations as tasks and specify the directionality of their arguments (input/output/inout) in such a way that dependences between different tasks can be computed at run time and the algorithm executed in a dataflow manner. Different runtime components have been developed implementing such functionalities on different platforms. CellSs is the runtime implementation of StarSs for the Cell/B.E. processor. It was the first one available and has been followed by SMPSs for general purpose homogeneous multicores and SMPs and GPUSs for NVIDIA GPUs. During the span of this project, significant progress has taken place in the development and use of the StarSs programming model and BSC performance tools. We are involved in an important effort to integrate into the OmpSs implementation of all the features that have been identified as relevant, but the evidences seem to be that the StarSs model supports an appropriate programming methodology for the heterogeneous multicore systems to come. The improvements taking place in the performance tools area, by using more intelligent data processing techniques show very promising results in terms of delivering real insight on the application behaviour and will help focus the optimization/parallelization efforts in the most productive direction. We consider that continued development in the two areas will soon result in huge improvements in the productivity of programmers as well as in the efficiency we will be able to achieve form the myriad of heterogeneous target architectures that we are starting to see.

15. SUBJECT TERMS

EOARD, Modeling & Simulation, High Performance Computing, Parallel Computing

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18, NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			JAMES LAWTON Ph. D.
UNCLAS	UNCLAS	UNCLAS	SAR	9	**19b. TELEPHONE NUMBER** *(Include area code)* +44 (0)1895 616187

Programming Models for Heterogeneous Mulitcore Systems

Grant FA8655-09-1-3075

Final Report

1 Summary

The document summarizes the technical work developed during grant number FA8655-09-1-3075 from May 2009 to June 2011. The aim of the grant was to study programming model extensions to exploit the parallelism in multicore nodes.

The work performed focused on three major lines: further development of the BSC performance tools environment; further development of the StarSs programming model and runtime; and port of some applications to StarSs.

Papers with the most relevant results of the project are attached to this report.

2 Introduction

The objective of grant FA8655-09-1-3075 was to study the programming models to exploit the parallelism in multicore nodes. The work extends the StarSs programming model proposal by BSC and evaluates its appropriateness to address the following points:

- Handling of dependences
- Heterogeneity
- Memory association.
- Hybrid use of StarSs within MPI

The work also addresses the analysis of some applications suggested by AFRL to understand their performance and propose ways of parallelization.

The StarSs programming models is a general node level parallel programming model based on pragmas annotating otherwise standard C programs. The annotations encapsulate certain computations as tasks and specify the directionality of their arguments (input/output/inout) in such a way that dependences between different tasks can be computed at run time and the algorithm executed in a dataflow manner.

Different run times were available at the start of the project, supporting StarSs on different platforms and different functionality level. CellSs is the runtime implementation of StarSs for the Cell/B.E. processor. It was the first one available and has been followed by SMPSs for general purpose homogeneous multicores and SMPs and GPUSs for NVIDIA GPUs.

3 Methods, assumptions and procedures

The project carried out several concurrent activities in the three major areas that are described in the following subsections. We describe the different activities and methods in this section.

3.1 Evolution of the StarSs infrastructure

During the project we have proposed several new functionalities in the StarSs model. Each of the proposals was implemented and tested on a different infrastructure (compiler and/or runtime) targeting a specific platform. The objective was to perform rapid prototyping of the ideas in order to explore their potential.

CellSs targeted the Cell processor and has been used for some applications in that platform. The CellSs version was also used as starting porting for a first support of GPUs.

The SMPSs version has been used as starting point for locality aware scheduling optimizations, the introduction of a new clause for reduction support, the support of strided and partially aliased regions as arguments and the hybrid integration of MPI/StarSs

Towards the second part of the project the decision was made that those features identified as useful will be integrated in the OmpSs version that integrates OpenMP and the StarSs concepts in a single infrastructure. This implementation will be the only one maintained in the long term. It allows the same OmpSs program to run on an SMP, a node with GPUs or a cluster of nodes each of them possibly with several GPUs.

Our runtime developments target existing machines and we do not require any specific hardware. We do require CUDA on GPU based platforms and do not yet support OpenCL based accelerator systems.

3.2 Performance tools

Our performance tools development has been based on the original infrastructure consisting of an instrumentation package (renamed to Extrae during the project lifetime), Paraver (an extremely flexible trace browser) and Dimemas (a simulator to replay the behaviour of parallel program under new architectural characteristics).

The use of traces and Paraver let us dig into the fine grain details of program behaviour and by gathering the experience in analysing many codes with it we have been able to develop techniques to automatically squeeze the information from the raw trace data. These techniques that have been embedded in external utilities that interoperate with the rest of the environment, but could also be integrated into other tools. Our visualization environment has also been very useful to assess the quality of the results of the automatic analysis.

We have focused on the use of clustering techniques to identify regions of similar behaviour and to obtain the complete and precise set of hardware counts for each such region with just one run of the program. This has been useful to derive CPI stack models that give deeper insight on the code performance than the hardware counts themselves.

All this analysis relies on the target machine having access to hardware counters through eh PAPI interface.

Another technique that has been developed consists in capturing both instrumented and sampled data and by correlating their timestamps being able to obtain extremely precise information on the instantaneous evolution of all metrics.

The use of the different techniques in conjunction with each other does result in extremely powerful analysis under development.

3.3 Applications

Different applications already available to BSC have been used to demonstrate the different improvements in the model. Some of them are linear algebra kernels.

During the last year of the project a close cooperation with Prof Palaniappany from U. of Missouri has taken place. P. Bellens from BSC visited did a stay of 1 month (November 2010) at U. of Missouri. Kernels in the area of image processing and tracking have been ported to StarSs as part of this cooperation and evaluated on both Cell and SMP based machines.

The applications that were evaluated in this collaboration include:

- Two implementations of the Flux Tensor in StarSs. The first two steps of this algorithm use a convolution operation and a temporal derivative. These linear operators can be interchanged, resulting in two different versions with different characteristics.
- A morphology kernel for CellSs, containing one opening and one closing operation.
- Two implementations of the Integral Histogram for StarSs, using a wavefront scan and a cross-weave scan.
- An implementation of the Integral Histogram specialized for the Cell/B.E. for performance comparisons with the versions for StarSs.
- A kernel that implements Otsu-Thresholding (using the Integral Histogram) for StarSs.

A tutorial lecture on StarSs was given in August 2010 at the Griffis Institute. Rome. NY.

4 Results and discussion

In this section we summarize the major results of the project

4.1 StarSs model and runtime

Different proposals for extensions of the StarSs model as compared to the one at the start of the project were done. We proposed the reduction clause as a mechanism to allow the scheduler execute concurrently sequences of commutative operations as long as the atomicity is maintained at application level. We also extended the SMPSs model to support strided and partially aliased arguments [9]. The hierarchical integration between SMPSs and GPU support was studied in [10]. This feature matches the hierarchical support in OpenMP and will this be naturally supported by the OmpSs implementation. We also proposed ways to integrate OpenMP and StarSs [6]**Error! Reference source not found.** and to handle GPU based systems [1]. The potential of leveraging OpenCL kernels in StarSs was investigated in [12].

Improvements in the CellSs runtime reported in [7] showed the potential of renaming and write modes. Other lazy renaming mechanism and locality aware scheduling for SMPSs was investigated in [11]. The write-back mode is used now in the GPU implementation of OmpSs. Renaming does have a huge potential to improve performance although we still consider that it is still necessary to research more on conditions where to restrict it.

The interaction of MPI and SMPSs in a hybrid programming model was published in [8] reporting the good characteristics of the approach not only to deliver high efficiency but also tolerance to low interconnect bandwidth and to operating system noise.

4.2 Performance tools

The description of how to extrapolate hardware counters for individual regions of a parallel program with high precision was published in [16]. The combined usage of instrumentation and sampling first appeared in [17].

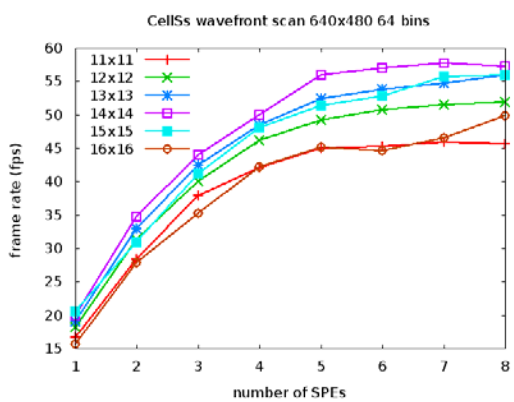
4.3 Applications

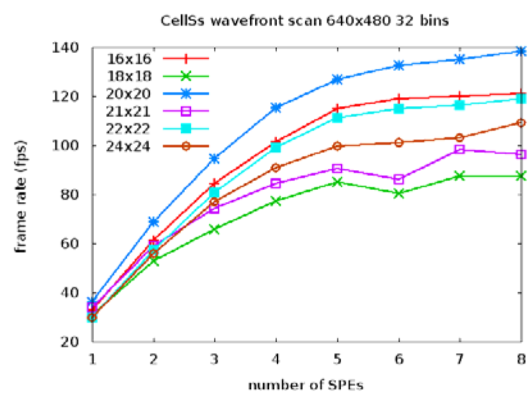
Examples of applications developing applications in StarSs were published in[4][5][6]. Four kernels are described in our submission to the HPC Challenge at the Supercomputing conference is described in [15].

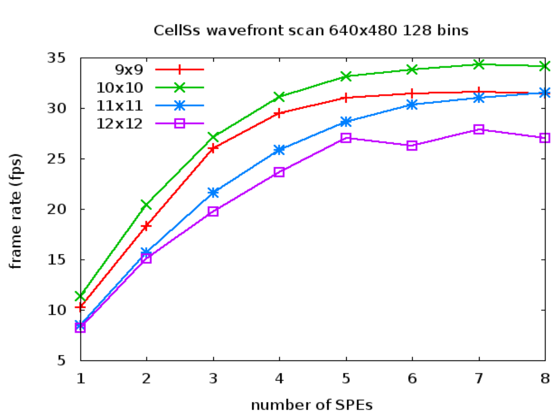
The first analysis of applications being developed by other AFRL collaborators was done in [2]. Further cooperation with U. of Missouri has resulted in the following paper [18] accepted and two papers in preparation [19][20]. The following table summarizes the different ports performed and some of the obtained results.

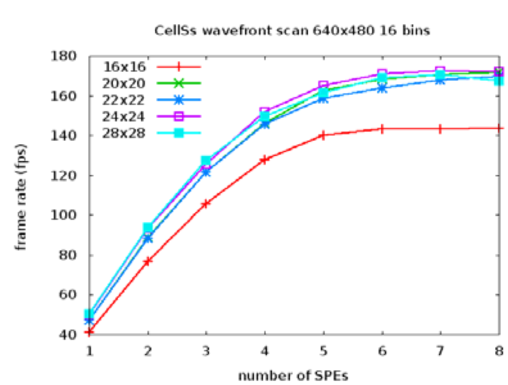
	CellSs	SMPSs	GPUSs	specialized
Flux Tensor (two versions)	Yes	Yes	No	
Morphology Kernel	Yes	Yes	No	
Integral Histogram (cross-weave)	Yes	Yes	Yes	Specialized for the Cell/B.E.
Integral Histogram (wavefront)	Yes	Yes	Yes	Specialized for the Cell/B.E.
Otsu-Thresholding	Yes	Yes	Yes	Specialized for the Cell/B.E.

The Integral Histogram on CellSs sustains a better than real-time performance of 220 frames per second. Performance evaluation for the other implementations of StarSs, as well as the comparison with the optimized version for the Cell/B.E. is ongoing at the time of writing.









5 *Conclusion*

During the span of the project, a significant progress has taken place in the development and use of the StarSs programming model and BSC performance tools. We are involved in an important effort to integrate into the OmpSs implementation of all the features that have been identified as relevant, but the evidences seem to be that the StarSs model supports an appropriate programming methodology for the heterogeneous multicore systems to come.

The improvements taking place in the performance tools area, by using more intelligent data processing techniques show very promising results in terms of delivering real insight on the application behaviour and will help focus the optimization/parallelization efforts in the most productive direction.

We consider that continued development in the two areas will soon result in huge improvements in the productivity of programmers as well as in the efficiency we will be able to achieve form the myriad of heterogeneous target architectures that we are starting to see.

6 *References*

[1] E. Ayguade, R.M. Badia, F.D. Igual, J. Labarta, R. Mayo and E.S. Quintana-Orti. “An extension of the StarSs Programming Model for Platforms with Multiple GPUs”. Procs. of the 15th international Euro-Par Conference (Euro-Par 2009). 851—862.

[2] J. Gimenez, J. Labarta. Report on the flux analysis. Submitted to Dr. Seetharaman and prof. Palaniappany

[3] E. Ayguade, R.M. Badia, P. Bellens, D. Cabrera, A. Duran, R.Ferrer, M. Gonzalez, J. Labarta, X. Martorell, R. Mayo, J.M. Perez, J. Planas, E.S. Quintana-Orti. “A Proposal to Extend the OpenMP Tasking Model for Heterogeneous Architectures ” LNCS Vol. 5568. 2009. pp. 154-167

[4] H. Ltaief, J. Kurzak, J. Dongarra, R.M. Badia. “Scheduling Two-sided Transformations using Tile Algorithms on Multicore Architectures”. Scientific Programming, vol 18(1), pages 35-50, January 2010.

[5] J. Kurzak, H. Ltaief, J. Dongarra, R.M. Badia. “Scheduling dense linear algebra operations on multicore processors”. Concurrency and Computation: Practice and Experience, volume 22, issue 1, January 2010.

[6] R. M. Badia, J. R. Herrero, J. Labarta, J. M. Pérez, E. S. Quintana-Ortí, G. Quintana-Ortí. “Parallelizing Dense and Banded Linear Algebra Libraries Using SMPSs”. Concurrency and Computation: Practice and Experience, volume 21, issue 18, December 2009.

[7] Bellens, P.; Perez, J.M.; Badia, R.M.; Labarta, J. “Just-in-Time Renaming and Lazy Write-Back on the Cell/B.E.” Parallel Processing Workshops. ICPPW’09. Viena, Sept. 2009.

[8] Marjanovic V.; Labarta J.; Ayguade E.; Valero M. “Overlapping Communication and Computation by Using a Hybrid MPI/SMPSs Approach” 24th International Conference on Supercomputing. Epochal Tsukuba, Tsukuba, Japan. June 2010

[9] J. M Perez, R. M. Badia and J. Labarta. “Handling task dependencies under strided and aliased references”. Proceedings of the 24th ACM International Conference on Supercomputing, June 2010.

[10] J. Planas, R. M. Badia, E. Ayguadé and J. Labarta, "Hierarchical task based programming with StarSs", in International Journal of High Performance Computing Applications, vol. 23, no. 3, 284 - 299, 2009.

[11] L. Martinell. “Memory usage improvements for the SMPSs runtime”. Master Thesis UPC. Jan 2010

[12] R. Ferrer, J. Planas, P. Bellens, A. Duran, M. Gonzalez, X. Martorell, R. Badia. E. Ayguade, J. Labarta. “Optimizing the Exploitation of Multicore Processors and GPUs with OpenMP and OpenCL”. The 23rd International Workshop on Languages and Compilers for Parallel Computing (LCPC2010). Rice Univ. Houston. October 2010

[13] R. Ferrer, P. Bellens, J.-S. Yeom, S. Schneider, K. Koukos, M. Alvanos, V. Beltran, M. González, X. Martorell, R. M. Badia, D. S. Nikolopoulos, A. Bilas, E. Ayguadé. “Parallel Programming Models for Heterogeneous Multicore Architectures”. IEEE Micro, vol. 30, no. 5, September 2010.

[14] E. Ayguade, R. M. Badia, P. Bellens, D. Cabrera, A. Duran, M. Gonzalez, F. Igual, D. Jimenez-Gonzalez, J. Labarta, L. Martinell, X. Martorell, R. Mayo, J. M. Perez, J. Planas and E.S. Quintana-Ortí. “Extending OpenMP to Survive the Heterogeneous Multi-core Era”. International Journal of Parallel Programming, vol. 38, n 5-6, pages: 440-459, June 2010.

[15] J. M Perez, R. M. Badia, J. Labarta and E. Ayguade. “SMPSs Submission to the HPC Challenge Class 2 Competition 2010”

[16] J. Gonzalez, J. Gimenez, and J. Labarta. “Performance Data Extrapolation in Parallel Codes”.. Proceedings of the 16th International Conference on Parallel and Distributed Systems (ICPADS '10), Shanghai, China December 2010.

[17] H. Servat, G. Llort, J. Giménez and J. Labarta. “Detailed Performance Analysis Using Coarse Grain Sampling”. PROPER 2009 workshop @ Europar 2009

[18] Pieter Bellens, Kannappan Palaniappany, Rosa M. Badia, Guna Seetharaman, Jesus Labarta. “Parallel Implementation of the Integral Histogram” in ACIVS 2011, Ghent, Belgium

[19] Pieter Bellens, Kannappan Palaniappany, Rosa M. Badia, Guna Seetharaman, Jesus Labarta. “A Portable Implementation of the Integral Histogram in StarSs” under revision for HIPC 2011, Bangalore, India

[20] Pieter Bellens, Kannappan Palaniappany, Rosa M. Badia, Guna Seetharaman, Jesus Labarta. “Comparison between an Optimized Implementation of the Integral Histogram on the Cell/B.E. And its Implementation on the Cell/B.E.”, being written