# Exploiting parallel skeletons in an all-purpose parallel programming system

Javier Fresno
Dept. Informática
University of Valladolid
jfresno@infor.uva.es

Arturo Gonzalez-Escribano
Dept. Informática
University of Valladolid
arturo@infor.uva.es

Diego R. Llanos
Dept. Informática
University of Valladolid
diego@infor.uva.es

## Abstract

Parallel skeletons are a high-level parallel programming model that specify the overall structure of common computation patterns, hiding the complex details of parallel applications.

In this work we introduce an extension to the parallel source-to-source compiler system Trasgo. It introduces a new abstraction level that bridges between synchronization and communication structures detected at the application level, and the low level primitives for communication.

## 1 Introduction

Development of parallel software is a quite complicated task. The programmer has to deal with several non-trivial issues such as problem decomposition, data distribution, local computation, data exchanges, load balancing, or synchronization.

Parallel Skeletons [3] are a high-level parallel programming model that abstract commonly-used patterns of parallel computation, communication, and interaction. Programmers do not have to write the code to perform the communications or synchronizations between processors. They only have to provide the specific code to solve the problem, using the skeleton like a template.

Trasgo [4] is a source-to-source compiler system that translates simple high-level specifications of parallel algorithms to lower-level native programs, with data partition and communication details generated automatically. Hitmap is the run-time library used by the back-ends of Trasgo, for hierarchical tiling and mapping of arrays or sparse data structures.

## 2 Skeleton support

After studying the state of the art of parallel skeletons and analyzing several parallel frameworks, we have integrated the skeleton approach with the data partition, mapping, and scheduling plug-in system of the Hitmap library.

Our solution consists in a skeleton interface using Hitmap functionalities. It adds a new abstraction layer for skeletons and uses a generic work pool to manage tasks. This interface will be used by our Trasgo system to generate communications for dynamic data dependences and structures which appear in many applications.

The new extension of the Hitmap library takes concepts from state of the art skeleton implementations such as The Edinburgh Skeleton Library (eSkel) [1] or the The Munster Skeleton Library Muesli [2]. Different task scheduling and distribution schemes have been implemented.

## 3 Experimental results

Experimental work has been conducted to validate our solution. We have implemented a new benchmark using the skeleton support. The benchmark solve a

Mandelbrot set fractal [5]. It is a problem easily paralelizable but introduces a load-balancing problem. Figure 1 shows execution times obtained in a homogeneous cluster at the EPCC (Hector). The results show that the new extension offers a good scalability and efficiency.

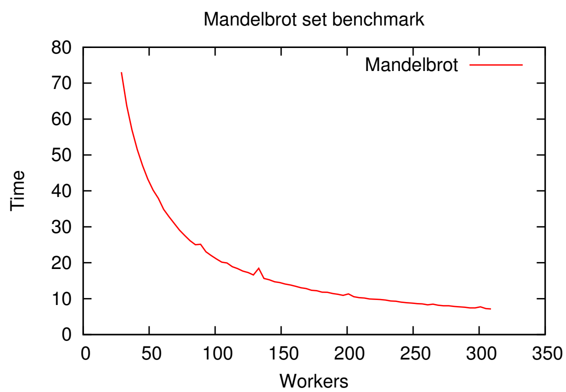


Figure 1: Execution time for the Mandelbrot set program benchmark in Hitmap.

# 4 Conclusions

Parallel skeletons present an interesting approach for parallel programming. In this work, we have added an extension to Hitmap, a back-end library to the parallel source-to-source compiler system Trasgo. This extension adds a new abstraction layer to the system that will help to produce efficient implementations for codes of higher abstraction level.

# Acknowledgements


The work was carried out under the HPC-EUROPA2 project (project number: 228398) with the support of the European Commission - Capacities Area - Research Infrastructures.

This work made use of the facilities of HECToR, the UK's national high-performance computing service, which is provided by UoE HPCx Ltd at the University of Edinburgh, Cray Inc and NAG Ltd, and funded by the Office of Science and Technology through EPSRC's High End Computing Programme.


# References


[1] Anne Benoit, Murray Cole, Stephen Gilmore, and Jane Hillston. Flexible skeletal programming with eSkel. In *Euro-Par'05 Proceedings of the 11th international Euro-Par conference on Parallel Processing*, pages 761–770, Lisbon, Portugal, 2005.

[2] Philipp Ciechanowicz, Michael Poldner, and Herbert Kuchen. The Münster Skeleton Library Muesli – A Comprehensive Overview. Technical Report 7, Westfälsche Wilhelms-Universität Münster (WWU) - European Research Center for Information Systems (ERCIS), 2009.

[3] Murray Cole. *Algorithmic Skeletons: Structured Management of Parallel Computation*. MIT Press, 1989.

[4] Arturo Gonzalez-Escribano and Diego R. Llanos. Trasgo: a nested-parallel programming system. *The Journal of Supercomputing*, 2009.

[5] Benoit B Mandelbrot. Fractal aspects of the iteration of $z \mapsto \lambda z(1-z)$. *Annals of the New York Academy of Sciences*, 357:249–259, 1980.