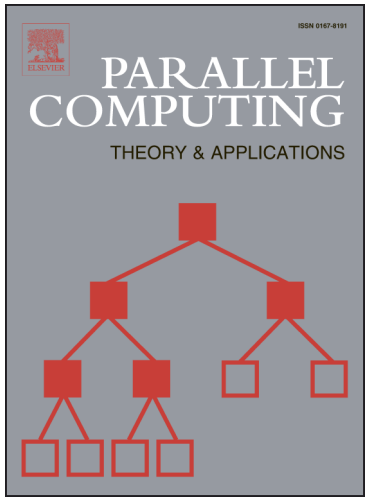

A Case Study of Different Task Implementations for Multioutput Stages in non-trivial Parallel Pipeline Applications

Angeles Navarro, Rafael Asenjo, Francisco Corbera, Antonio J. Dios, Emilio L. Zapata

PII: S0167-8191(14)00063-5
DOI: http://dx.doi.org/10.1016/j.parco.2014.05.003
Reference: PARCO 2188

To appear in: *Parallel Computing*

Please cite this article as: A. Navarro, R. Asenjo, F. Corbera, A.J. Dios, E.L. Zapata, A Case Study of Different Task Implementations for Multioutput Stages in non-trivial Parallel Pipeline Applications, *Parallel Computing* (2014), doi: http://dx.doi.org/10.1016/j.parco.2014.05.003

A Case Study of Different Task Implementations for Multioutput Stages in non-trivial Parallel Pipeline Applications

Angeles Navarro, Rafael Asenjo*, Francisco Corbera, Antonio J. Dios and Emilio L. Zapata*

Department of Computer Architecture, University of Málaga, Spain
{*angeles,asenjo,corbera,ajgarcia,ezapata*}*@ac.uma.es*

Abstract

Task-based libraries, such as Intel's Threading Building Blocks (TBB), are promising tools that help programmers to develop parallel code in a productive way, thanks to high-level constructors which simplify the chore of efficiently exploiting system resources. In this paper we focus on one type of task parallelism, pipeline parallelism, which is becoming an increasingly popular parallel programming pattern for streaming applications in the domain of digital signal processing, graphics, compression and encryption. Specifically, TBB provides a high-level template to express pipeline parallelism, but it is limited to representing simple pipeline structures. We address the issue of non-trivial parallel pipeline structures in which one or more stages in the pipeline have more items leaving than arriving, a problem for which the current TBB pipeline template does not provide support. In this work, we describe a new Multioutput filter that we have incorporated into the TBB pipeline framework to deal with these multioutput stages. Using real world streaming applications from different computational domains (`dedup` and `scenerecog`), we also compare the performance of our implementation using the Multioutput filter in the TBB pipeline template to other more complex TBB task-based implementations that only use the standard filters. We also develop new analytical models for each implementation to better understand the resources utilization in each case. Performance evaluation and analysis shows that the implementation based on the Multioutput filter outperforms the other solutions because: it promotes finer task parallelism, which is more suited to the TBB task-stealing mechanism in order to better exploit the resources; and it also reduces the overheads related to memory and task management.

Keywords: Tasks Library, Parallel Pipeline, Multioutput, Analytical Model

*Corresponding author's contact information. Phone: +34 952 132791, Fax: +34 952 132790, email: asenjo@ac.uma.es