

# "Smart" Manual Software Parallelisation Assistant

Aleksandr Maramzin<sup>\*1</sup>, Björn Franke<sup>\*2</sup>,  
Murray Cole<sup>\*2</sup>

*\* Institute For Computing Systems Architecture, Informatics Forum, The University of Edinburgh, 10 Crichton Street, Edinburgh, UK*

---

## ABSTRACT

Since automatically parallelizing compilers have failed to deliver significant performance improvements, programmers are still forced to parallelize legacy software manually for all but some niche domains. Rather than hoping for an elegant silver bullet, we acknowledge the role of a human expert in the parallelization process and develop a *smart* parallelization assistant.

In its essence our assistant is yet another application of machine learning techniques to the field of optimizing compilers, which tries to predict the parallelisability property of program loops. We use Seoul National University version of NAS Parallel Benchmarks (NPB) [Div12], [Uni12] hand-annotated with OpenMP parallelisation pragmas to train our ML model. We show that the loop parallelisability classification problem can be successfully tackled with machine learning techniques (using only static code features) achieving accuracy of around 90% and outperforming all available baseline random predictors working at an accuracy ranging between 40% and 70%.

To get a real practical application of our techniques, we integrate our trained ML model into an assistant scheme, designed to mitigate the effects of ineradicable statistical errors and make them less critical. Taking application profile our assistant directs a programmer's efforts by pointing the loops, which are highly likely to be parallelisable and profitable as well. Thus, decreasing the efforts and time it takes to parallelize a program manually. As a side effect our assistant extends the capabilities of Intel C/C++ compiler in the task of parallelism discovery by increasing the amount of parallelism found in SNU NPB benchmarks from 81% to 96%.

KEYWORDS: ACACES; poster session; software engineering; parallel programming; compilers; static program dependence analysis; loop iterator recognition; machine learning; programmer feedback;

## 1 Introduction

This document (`guide.pdf`) describes `acaces.cls`, the L<sup>A</sup>T<sub>E</sub>X [Lam86] style file to be used for the abstract of your ACACES poster. The document you are reading right now was also produced with that style file. You are strongly advised to use L<sup>A</sup>T<sub>E</sub>X for preparing your camera-ready paper. If this is impossible, please try to mimic the layout of this document as closely as possible. Word users can use the `ACACEStemplate.doc` template.

The remainder of this document assumes you use L<sup>A</sup>T<sub>E</sub>X with the `acaces.cls` style file.

---

<sup>1</sup>E-mail: `s1736883@sms.ed.ac.uk`

<sup>2</sup>E-mail: `{bfranke,mic}@inf.ed.ac.uk`



Figure 1: An example of a Figure: the HiPEAC logo.

Using the style file is pretty straightforward, just have a look to the source of this file (`guide.tex`).

## 2 Learning Loop Parallelisability Property

## 3 Typesetter

## 4 Assistant Scheme Evaluation

## 5 Typesetter

Use `pdflatex guide.tex` to generate a PDF file, don't use `latex` to produce a dvi file and then `dvipdf` or `dvips` followed by `ps2pdf` or `pdftopdf` as this results in ugly looking PDF.

## 6 Figures

Your figures should also be made in PDF. Unfortunately, not a lot of drawing applications allow you to write a PDF file. You can however create EPS files and transform the resulting files to PDF using `epstopdf figure.eps`. Figure 1 shows an example of a figure. Please bear in mind that your paper will be reduced to 70% of its original size.

## 7 Fonts

Please make sure that your PDF file only contains Type1 fonts. This should be no problem if you use `pdflatex` and the `acaces.cls` style file. If you use another typesetting system, please check if the resulting PDF file only uses Type1 fonts. You can check this using Acrobat Reader: open your file and check all fonts using File→Document Properties→Fonts...

## 8 Final submission

Make sure that your PDF file does not contain page numbers, is up to 4 pages long and is formatted for an A4 page. Upload the final version before June 8 on the website.

## 9 About this style

You are encouraged to send bug reports, remarks, ... about this style to [ronsse@elis.UGent.be](mailto:ronsse@elis.UGent.be).

## 10 End

This is the end!

## References

- [Div12] NASA Advanced Supercomputing (NAS) Division. Nas parallel benchmarks, August 2012.
- [Lam86] Leslie Lamport. *LaTeX: User's guide & reference manual*. Addison-Wesley, 1986.
- [Uni12] Seoul National University. Snu nas parallel benchmarks, August 2012.