Abstract

Multicore processors have paved the way to increase the performance of any application by the virtue of benefits of parallelization. However, exploiting parallelism from a program is not easy, as it requires parallel programming expertise. In addition, manual parallelization is a cumbersome, time consuming and inefficient process. A number of tools proposed in the past ease the effort of parallel programming. This paper presents a classification of such parallelization tools. The classification is based on different eras of tool development, role played by these tools in various parallelization stages, and features provided by parallel program assistance tools. Classification of tools concludes with a discussion on requirements of futuristic parallelization tools. Finally, this paper proposes our on-going work about the development of a parallel program assistance tool called EasyPar, which is a parallel program assistance tool.

References

- official home page - Open MP, Available : www.openmp.org
A Review of Parallelization Tools and Introduction to Easypar

- Vivek Sarkar, Programming challenges for multicore Parallel systems, Presentation for Computer Science Department, Rice University. http://www.rice.edu
- S. Sah and Vinay G. Vaidya, Paradyn: A Dynamic Parallel Programming Tool, ICDCN, 12th International Conference on Distributed Computing and Networking, PhD forum (2010)
- Par4All homepage: URL: http://www.par4all.org
- Andrew R. Bernat, Barton P. Miller, "Incremental Call-Path Profiling," Computer Sciences Department, University of Wisconsin. Madison, WI
- Cathal Boogerd, Leon Moonen, "On the Use of Data Flow Analysis in Static Profiling," Software Evolution Research Lab, Delft University of Technology. The Netherlands


- Ravichandran K. M., Bhaskar P, Annamalai. S. P. and Dr. A. P. Shanthi, Automatic Inter-procesural Parallelism, Department of Computer Science, College of Engineering, Guindy, Anna University


- NirSavit, Dan Touitou, Software Transactional Memory, ACM-PODC, Ottawa Ontario, CA, 1995

- Official Website : &quot;The Multicore Association&quot;, http://www.multicore-association.org/


- Sudhakar Sah, Vinay G. Vaidya, A GPU Based Novel Design of Side Effect Analysis, ICOMECE, Goa, India, (2011) 137-143


- Karl Fuerlinger, Michael Gerndt, Jack Dongarra, On Using Incremental Profiling for the
- U. Ismail, "Incremental call-graph construction for the eclipse IDE", University of Waterloo Technical Report No. CS-2009-07, David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, Canada
- Hannu-MattiJärvinen, MikkoTiusanen, and Antti T. x`Virtanen, Convit, a Tool for Learning Concurrent Programming, Software Systems Institute, Department of Information Science Tampere University of Technology, Finland
- Yu, Y., Rodeheffer, T., and Chen, W., "Racetrack: efficient detection of data race conditions via adaptive tracking", in Proceedings of the twentieth ACM symposium on Operating systems principles, SOSP '05, (New York, NY, USA), pp. 221–234, ACM, 2005
- Prism: an analysis exploration and verification environment for software implementation and optimization on multicore architectures from CriticalBlue. http://www.criticalblue.com
- vfAnalyst: Analyze your sequential C code to create an optimized parallel implementation from VectorFabrics, http://www.vectorfabrics.com/
- Das, D. and Wu, P., "Experiences of using a dependence profiler to assist parallelization for multi-cores," in IPDPS Workshops, pp. 1–8, 2010
- M. Kulkarni, M. Burtscher, R. Inkulu, K. Pingali, and C. Cascaval. "How much parallelism is there in irregular applications?" In PPoPP &apos;09: 2009
A Review of Parallelization Tools and Introduction to EasyPar

- M. W. Hall, B. R. Murphy, S. P. Amarasinghe, S. Liao, and M. S. Lam. "Interprocedural Analysis for Parallelization"; In LCPC’s06, pp 61–80

Index Terms

Computer Science Parallel Processing

Keywords

Interactive Parallelization Parallel Program Assist Automatic Parallelization

Parallel Programming Tools

Multicore