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
- Peter Kacsuk, Gabor Dozsa, Tibor Fadgyas, Robert Lovas, &apos;The GRED Graphical Editor for the GRADE Parallel Program Development Environment&apos;, Budapest, Hungary : MTA-SZTAKI Computer and Automation Research Institute, Hungarian Academy of Science.
- Andrew R. Bernat, Barton P. Miller, &apos;Incremental Call-Path Profiling&apos;, Computer Sciences Department, University of Wisconsin. Madison, WI
- Cathal Boogerd, Leon Moonen, &apos;On the Use of Data Flow Analysis in Static Profiling&apos;, Software Evolution Research Lab, Delft University of Technology. The Netherlands
- &apos;On Using Incremental Profiling for the Performance Analysis of Shared Memory Parallel Applications&apos;, Innovative Computing Laboratory&apos;, Department of Computer Science, University of Tennessee. Technical Report
- Barbara G. Ryder, A Position Paper on Compiler Time Program Analysis, Computer Science Dept. , Rutgers University (1997)
A Review of Parallelization Tools and Introduction to Easypar

- 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 Toultou, Software Transactional Memory, ACM-PODC, Ottawa Ontario, CA, 1995
- Sudhakar Sah, Vinay G. Vaidya, A GPU Based Novel Design of Side Effect Analysis, ICOMEc, 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-Matti Järvinen, Mikko Tiusanen, and Antti T. x Virtanen, Convit, a Tool for Learning Concurrent Programming, Software Systems Institute, Department of Information Science Tampere University of Technology, Finland
A Review of Parallelization Tools and Introduction to Easypar

- 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., &quot;Experiences of using a dependence profiler to assist parallelization for multi-cores,&quot; in IPDPS Workshops, pp. 1–8, 2010.


- Das, D. and Wu, P., &quot;Experiences of using a dependence profiler to assist parallelization for multi-cores,&quot; in IPDPS Workshops, pp. 1–8, 2010


- Y. He, C. Leiserson, and W. Leiserson. &quot;The Cilkview Scalability Analyzer. &quot; &quot; In SPAA \textcopyright 10, Proceedings of the Symposium on Parallelism in Algorithms and Architectures, 2010

- M. Kulkarni, M. Burtscher, R. Inkulu, K. Pingali, and C. Cascaval. &quot;How much parallelism is there in irregular applications?&quot; &quot; In PPoPP \textcopyright 09: 2009

- T. Harris and K. Fraser,&quot;Language Support for Lightweight Transactions&quot;, In OOPSLA\textcopyright 03, pages 388–402, 2003

- William F. Appelbe, John T. Stasko, Eileen Kraemer, &quot;Applying Program Visualization Techniques to Aid Parallel and Distributed Program Development. &quot; &quot; Dept. of Computer Science, Georgia Institute of Technology. Atlanta, GA, Technical Report. GIT-CC 91/31 (1993)


org
- M. W. Hall, B. R. Murphy, S. P. Amarasinghe, S. Liao, and M. S. Lam. “Interprocedural Analysis for Parallelization”, In LCPC'06, pp 61–80
- Gabriele Jost, Haoqiang Jin, Jesus Labarta, and Judit Gimenez, “Interfacing Computer Aided Parallelization and Performance Analysis”, ICCS'03 International conference on Computational science, 2003 pp 181-190

Index Terms

Computer Science
Parallel Processing

Keywords
Interactive Parallelization
Parallel Program Assist
Automatic Parallelization
Parallel Programming Tools

Multicore