Abstract

With the introduction of GPGPUs, parallel programming has become simple and affordable. APIs such as NVIDIA’s CUDA have attracted many programmers to port their applications to GPGPUs. But writing CUDA codes still remains a challenging task. Moreover, the vast repositories of legacy serial C codes, which are still in wide use in the industry, are unable to take any advantage of this extra computing power available. Lot of attempts have thus been made at developing auto-parallelization techniques to convert a serial C code to a corresponding parallel CUDA code. Some parallelizes, allow programmers to add "hints" to their serial programs, while another approach has been to build an interactive system between programmers and parallelizing tools/compilers. But none of these are really automatic techniques, since the programmer is fully involved in the process. In this paper, we present an automatic parallelization tool that completely relieves the programmer of
any involvement in the parallelization process. Preliminary results with a basic set of usual C codes show that the tool is able to provide a significant speedup of ~10 times.

References

- VectorFabrics. vfAnalyst: Analyze your sequential C code to create an optimized parallel implementation. http://www.vectorfabrics.com/
- M. Hall, J. Anderson, S. Amarasinghe, B. Murphy, S. -W. Liao, E. Bugnion, and M.


Index Terms

Computer Science
Information Technology

Keywords

Auto parallelization parallelization C CUDA hiCUDA GPU