

Conclusion to the Special Issue on Parallel SAT Solving

Youssef Hamadi

youssefh@microsoft.com

Microsoft Research

7 J J Thomson Avenue, CB3 0FB Cambridge

United Kingdom

Recent years have shown a major architectural shift in computer hardware. The traditional efficiency gains upcoming from the relentless raise of chips frequencies has been stopped by a thermal wall and performance improvements have to be found elsewhere. The new direction is to add more computing units (cores) to a chip in order to raise its computational power. The products resulting from this multicore strategy are now on every desktop and yet the horizon is wide open since the number of cores is expected to grow exponentially. This technological shift represents an important challenge for many computer sciences fields whose best algorithms have to be rethought for multicore-based parallelization. The goal of this special issue was to officially acknowledge this evolution, and to present recent advances in the parallel processing of SAT problems. Three works were finally selected.

- *PaMiraXT: Parallel SAT Solving with Threads and Message Passing*, by Tobias Schubert, Matthew Lewis, and Bernd Becker extends MiraXT to clusters of workstations.
- *Incorporating Clause Learning in Grid-based Randomized SAT Solving*, by Antti E. J. Hyvärinen, Tommi Juntilla, and Ilkka Niemelä presents an algorithmic framework for learning-enhanced randomized SAT solving in Grid environments.
- *ManySAT: a Parallel SAT Solver*, by Youssef Hamadi, Said Jabbour, and Lakhdar Sais thoroughly describes the winner of the 2008 parallel SAT-Race.

These articles demonstrate the recent progresses achieved in the parallel resolution of the SAT problem. Parallel SAT solvers performance is now significant, and thanks to the generalization of multicore platforms, their integration into existing applications can be done transparently, and without any adaptation. This special issue described three different approaches, two of them are mixing distributed and shared memory architectures, one of them is focusing on shared memory systems. The algorithms presented here were thoroughly described and evaluated. I hope that this level of presentation collected into a single special issue, will help clarify and demystify them, and allow the spread and development of new generations of parallel SAT solvers.

I would like to thank all the anonymous reviewers for their work, the authors for their contribution, and Hans van Maaren for giving me this opportunity of publishing this special issue.