OBDD Heuristics Online

Evaluation of OBDD-Heuristics via the Internet

Christoph Meinel, Arno Wagner
Institut für Telematik
Bahnhofstr. 30-32
54292 Trier, Germany
Email: {meinel,wagner}@ti.fhg.de
Keywords: Desicion Diagrams, Internet

1 Introduction

Today the Internet offers the highly interesting possibility of standardized and global communication without a need for special hardware or expensive infrastructure. As the Internet will most likely grow further and become increasingly powerful, it prompts the development of new types of applications. While there is a lot of progress in fields like database systems or electronic commerce, the Electronic Design Automation (EDA) community is only beginning to use these possibilities.

One area that can benefits is the area of data structures and algorithms for EDA. There have been a number of important discoveries in this area recently. Noteworthy are Ordered Binary Decision Diagrams (OBDDs [Bry86], for an overview see [MT98]) and on a higher abstraction level methods such as Symbolic Model Checking [McM93]. These advancements and an ever growing demand for powerful EDA tools have caused many researchers and developers to create tools for verification, simulation, synthesis and similar uses. But one problem remains: Whenever a new algorithm is presented to the public, its potential users only get information on the performance of the method on some limited set of benchmarks. While this is sufficient to demonstrate that a new algorithm is interesting, a general evaluation is not possible in this way. Integration of, e.g., a new heuristic into some existing software package is a lengthy process and success for a specific purpose usually cannot be predicted from limited published performance data. This problem is an important obstacle for the transfer of research results into practical use.

In this paper we present an Internet based solution, that focuses on the area of OBDD heuristics. Section 2 gives a discussion of the need for Web interfaces and explains the specifics of the OBDD setting. Section 3 develops requirements an Internet based approach should satisfy. In section 4 we present our system. Section 5 gives an overview of related work and concludes the paper.

2 Problems of Traditional Approaches

By traditional approach we mean the distribution of software in different forms. While giving away software might be an adequate solution in a commercial setting, our setting is the presentation and demonstration of research results in the area of algorithms for EDA.

General Setting

When new algorithms are developed, they are usually added to some existing software package. After optimization, when (hopefully) a sufficient level of performance is reached, the algorithm is presented to the community. At this stage there often arises the whish to allow others to use the algorithm. But here the problem manifests itself: Several points make giving away the software difficult. Let us consider giving away the source code:

- The code is usually usable only within the software package for it was written for.
- The code has only research level quality and may still lack documentation or even worse contain errors.
- Due to some ideas not yet published, one may not want to give away all the details of the implementation.

So this is clearly not an appealing option. But giving away executables is no improvement either:

- The software would have to be ported to different platforms.
- There would be need for support for every of these platforms.

- There would be need to develop the software to a higher level of stability.
- Software that presents a certain value would have to be given away.

An other option is re-implementation of the algorithm by interested parties. But this requires a larger effort and special experience. Almost nobody is willing to undertake such an effort for an uncertain outcome.

The OBDD Case

Our research group is involved in the development of heuristics for various tasks that arise in the use of OBDDs, e.g. reordering to reduce size or combination of OBDDs with different orderings. OBDDs are by now an essential data-structure in a lot of tasks that arise in connection with the design of boolean circuits, like formal verification or simulation. Whenever a new heuristic has been developed, it is evaluated using benchmarks from a standard benchmark set first, for example the ISCAS 89 ([BBK89]) set of circuits. Due to the nature of the problem, performance both in speed and memory consumption can vary extremely between different circuits of the benchmark. Predictions about the behaviour of a heuristic on other circuits are usually extremely difficult or impossible.

A consequence is that, whenever a new heuristic is presented to the public, researchers and developers will need to evaluate the heuristic on their own benchmarks. Further complications arise because the published results are difficult to reproduce as fine-tuning is often done and usually critical for good results. Specific details of the computer system used for the computations can have a large impact as well. This makes comparisons outside the published results extremely difficult. We think most of these issues can be addressed in a satisfying way by a possibility to access the heuristics using the Internet. In the next section we will develop requirements an Internet based solution should satisfy.

3 Requirements for an Internet based Solution

With the rise of the Internet from an academic curiosity to an increasingly usable tool, a new quality becomes possible: Easy and cheap use of non-local resources. In our case the remote usage of tools containing OBDD heuristics. We belive that some requirements need to be addressed to make such a system truly usable. We will now discuss the main needs and possible solutions for different areas of concern.

Flexibility and Speed

To achieve an appropriate overall speed it is necessary to distribute the actual computations to several computers. The number of computers should be easily adjustable, and it should even be possible to include computers that are remote and only reachable by an Internet connection. To maintain comparability of the results it is necessary that the pool of computers can be divided into groups of machines with equal power.

Reliability

As computations can take in the order of hours and as there might be a number of pending computations, there should be mechanisms that guarantee persistence of submitted requests. This limits the loss in case of software or hardware malfunction and gives the possibility of automated recovery.

Security

Security here means the protection of intellectual property. Certainly nobody would be willing to give away carefully crafted circuits to evaluate some OBDD heuristic on them. A simple and secure solution would be to send OBDDs instead of circuits. As OBDDs are a canonical representation of boolean function, no detail would be given away. BDDs can be transferred encoded as circuits themselves. An other possibility is to do some transformations on the circuit that still allow the desired computations but obscure the actual details of the implementation, see e.g. [HK98]. Of course, if the circuits in question are not critical, circuits can be transferred directly.

Ease of Use

The system should have an interface that is simple and easy to use. A possible solution here is to use the World Wide Web. Any standard browser with file upload capability can be used.

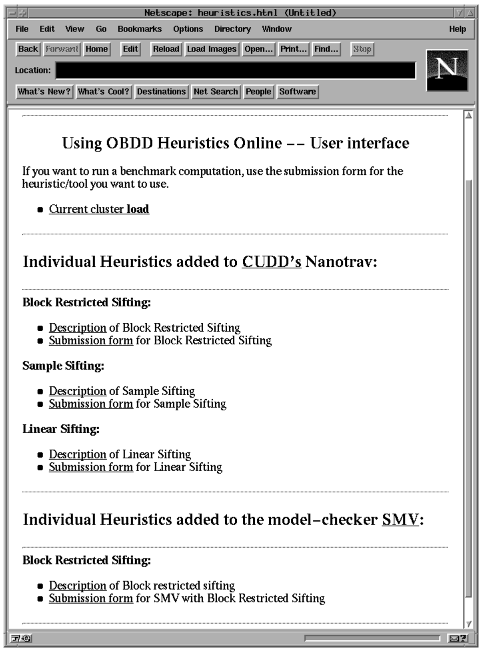

Figure 1: Central Selection Menu

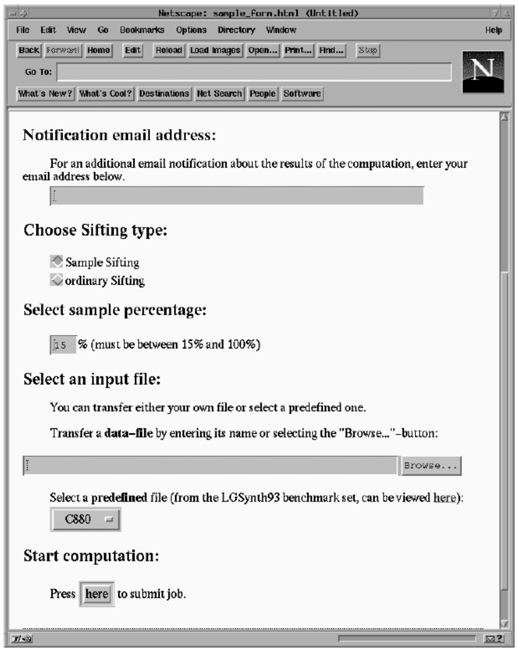

Figure 2: Data entry form

4 OHO – OBDD Heuristics Online

We have implemented a system, called OHO (for OBDD Heuristics Online), that meets the requirements mentioned. We will now describe OHO in more detail.

The WWW Interface

The system is accessed by an easy to use web interface. After an introductory page containing general information, the user is presented with a menu that allows to select submission forms for individual tool and heuristic types, as well as browsing the specific documentation. A part of this central access page is depicted in Figure 1. After having selected one of the possible types of computation, the user is presented with a data entry form (Figure 2), where all the relevant information can be entered. This includes an (optional) e-mail address for notification about results, tool options and selection of input data. Options include the choice between the featured heuristic for reordering or a reference heuristic (usually ordinary sifting). Input data can consist of a predefined circuit or an user transferred file. File upload is done with the browser, a feature found in most modern Web browsers. When the submit button is pressed, the request is entered into a queue of waiting jobs and the computation begins as soon as the needed resources become available. Meanwhile the user is taken to a page that lists the current status of the job, the position of the job in the waiting queue and (after completion) the results. If a notification email address has been given, the results will be e-mailed immediately after completion. The Web interface can be accessed via the following URL: `http://www.informatik.uni-trier.de/TI/OHO.html`

Available OBDD Tools

The tool-functionality that we offer consists of a couple of new and innovative BDD heuristics. At the moment these are *Block Restricted Sifting* ([MS97]) integrated both in the traversal tool Nanotrav (part of the CUDD system, [Som96]) and the well known model checker SMV ([oCS]). Furthermore, *Sample Sifting* ([MS98]) and *Linear Sifting* ([MT96]) integrated into Nanotrav are available and Sample Sifting integrated into SMV will be added shortly. The set of available tools will constantly grow, as new research results will lead to new and interesting heuristics.

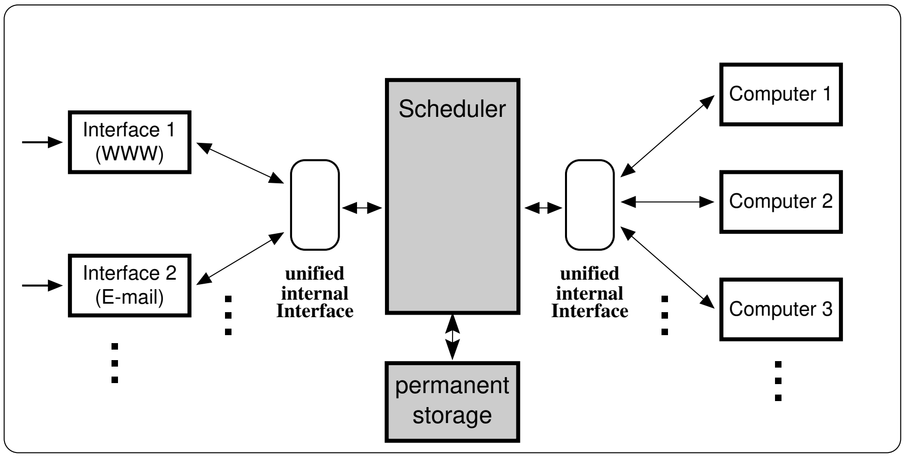

Figure 3: Structure of the system

Technical Details

The basic structure of OHO can be found in Figure 4. Individual tools have individual interfaces, depicted on the left side of the figure. A central scheduler, manages every submitted request during its whole life-cycle. Here, persistent storage of the data, parameters, status and, if available, results is performed. Crash recovery both from software crashes and hardware malfunctions is one of the responsibilities of the scheduler as well.

A number of computers (to the right of the scheduler) perform the actual computations necessary to complete a request. On each of this computers runs a small demon that is given one task at a time by the scheduler, then starts the tool that will perform the computation. When a request is completed (with or without error) a message is sent back to the scheduler. Due to the possibility of rescheduling computations when resources become unavailable, there is no need to use dedicated computers.

The computer cluster used for the computations currently consists of four PCs with Intel Pentium Pro 200 CPUs running Linux. A memory limit of 80 MB is imposed on benchmark computations. There is also a CPU time limit of 2 hours and an upload file size limit of 1 MB. These limits are somewhat arbitrary and might change.

Due to the robustness of the system it is possible to use 'research quality' implementations of the BDD-heuristics. Implementations with this grade of quality are usually available from the necessary experimentation done as part of the actual research.

5 Related Work

There have been efforts to use the WWW as an unified interface to a heterogenous set of EDA tools. In [BBMR96] such an application has been described. The main focus there is the integration of different tools on different platforms with different data formats into a seemingly homogenous environment. The framework is intended to do actual work rather than to allow the comparison of the power of different approaches to a specific problem (like OBDD heuristics) as in our case.

In [Ku97] the vision of integrating a great number of EDA CAD services running in different places into one "global" EDA CAD system using the Internet has been described. The main techniques here are the use of proxies to abstract the actual tool being used.

In [SK95] an other approach to the integration of EDA tools is described. This approach is cantered mostly on the notion of "active messages".

6 Conclusion

We have explained why there is a genuine need for Web interfaces in EDA algorithm research. Especially the performance evaluation and comparison of new methods is significantly facilitated by Web interfaces. In our opinion other approaches are often unsuitable, because they require a large effort either by the developer of some algorithm or by the party interested in the algorithm. This seems to hinder the transfer of mesarch results into actual use.

We presented our system, that specifically allows the comparison and online evaluation of OBDD heuristics and currently features a number of OBDD heuristics developed by our OBDD group. These heuristics are contained in a tool for OBDD synthesis as well as in the well known model checker SMV. Our system can compute more than just very small toy examples, as it uses a pool of computers to do the actual computations. Although it is not intended for computations using a very large amount of memory, we belive that the chosen limits on memory and time are large enough for the successful completion of benchmark computations with have a meaningful size.

Aspects of security where briefly discussed as well. We belive our approach to be an important step to use the power of the Internet in EDA algorithm research. So far as we know no other effort has been made to allow online evaluation and comparison of new OBDD heuristics.

References

[BBK89] F. Brglez, D. Bryan, and K. Kozminski. Combinational profiles of sequential benchmark circuits. In *Proc. IEEE Int. Symposium on Circuits and Systems*, pages 1929–1934, May 1989.

[BBMR96] A. Bogliolo, L. Benini, G. De Micheli, and B. Ricc. Ppp: A gate-level power simulator - a world wide web application. Technical Report Stanford Technical Report No. CSL-TR-96-691, Stanford University, 1996.

[Bry86] R.E. Bryant. Graph Based Algorithms for Boolean Function Manipulation. *IEEE Transaction on Computers*, C-35:677–691, 1986.

[HK98] S. Hauck and S. Knoll. Data security for web-based cad. In *Procceedings of the 35th Design Automation Conference*, 1998.

[Ku97] D. Ku. Embedded tutorial: Eda and the network. In *Procceedings of ICCAD 97*, 1997.

[McM93] K. L. McMillan. *Symbolic Model Checking*. Kluwer Academic Publishers, 1993.

[MS97] C. Meinel and A. Slobodová. Speeding up variable ordering of obdds. In *Proc. of IEEE ICCD'97, Austin(Texas,USA)*, 1997.

[MS98] C. Meinel and A. Slobodová. Sample method for minimization of obdds. IWLS'98, Granlibakken Resort, Lake Tahoe (Kalifornien, USA), 1998.

[MT96] C. Meinel and T. Theobald. Local encoding transformations for optimizing obdd-representations of finite state machines. In *Proc. FMCAD'96, LNCS 1166*, pages 404–418. Springer-Verlag, 1996.

[MT98] C. Meinel and T. Theobald. *Algorithms and Data Structures in VLSI Design*. Springer, 1998.

[oCS] CMU School of Computer Science. Formal methods - model checking. http://www.cs.cmu.edu/˜ modelcheck/.

[SK95] M. J. Silva and R. H. Katz. The case for design using the world wide web. In *Procceedings of the 32th Design Automation Conference*, 1995.

[Som96] F. Somenzi. Colorado university decision diagram package. ftp://vlsi.colorado.edu/pub/, 1996.