# Energy Consumption Analysis and Verification by Transformation into Horn Clauses and Abstract Interpretation*

M. V. Hermenegildo M. Klemen U. Liqat
IMDEA Software Institute
and
Universidad Politécnica de Madrid (UPM), Departamento de Inteligencia Artificial

P. Lopez Garcia
IMDEA Software Institute
and
Spanish Council for Scientific Research (CSIC)

{manuel.hermenegildo,maximiliano.klemen,umer.liqat,pedro.lopez}@imdea.org

## Extended Abstract

The static estimation of the energy consumed by program executions has applications in program optimization and verification, and is instrumental in energy-aware software development. We describe our approach for estimating such energy consumption statically (i.e., at compile-time, without running the program) in the form of *functions on the input data sizes of procedures* (and possibly other hardware-dependent parameters), and for *using such functions for verifying and finding errors* with respect to a rich class of energy consumption specifications for programs. We also present the implementation of this approach within the CiaoPP system.

Determining program energy consumption requires the analysis of low-level program representations, since sufficiently accurate energy models are only available at the instruction set architecture level or, with some precision loss, at intermediate levels (such as, e.g., LLVM). We have developed an approach to the analysis and verification of energy consumption [14, 8, 10, 7, 6] that is based on a transformation of instruction set architecture- (or LLVM-)level programs into Horn clauses [13, 4]. The Horn clauses encode the semantics of these low-level programs, at different levels of abstraction, as determined by different abstract domains. Such abstract domains approximate properties that are instrumental for energy analysis, such as sized types, determinacy, or non-failure, as well as the energy consumption itself [17]. The latter contrasts with previous approaches to cost analysis that are not based directly on abstract interpretation.

Given such a set of Horn clauses, the objective of our analysis is to compute their abstract minimal model for each abstract domain or combination of domains. For this purpose we use the PLAI analyzer of the CiaoPP system [5]. PLAI computes the abstract minimal model of a set of Horn clauses using a top-down, memo table-based fixpoint algorithm –which can be seen as an extension of a highly optimized SLDT resolution engine with the abstract domains taking the traditional role of constraint domains in

*The research leading to these results has received funding from the European Union 7th Framework Programme under grant agreement no 318337, ENTRA - Whole-Systems Energy Transparency, Spanish MINECO TIN2015-67522-C3-1-R *TRACES* project, and the Madrid M141047003 *N-GREENS* program.

A. Lisitsa, A. Nemytykh, M. Proietti (Eds.):
Verification and Program Transformation (VPT 2017)
EPTCS ??, 2017, pp. 4–6, doi:10.4204/EPTCS.??.2

the logic. This extended abstract tabling algorithm includes optimizations for fixpoint acceleration such as dependency tracking or dynamic strongly connected component detection. It is incremental and, in contrast to bottom-up algorithms (also available in CiaoPP), multivariant (context sensitive). The result of the abstract model computation –the analysis– is the set of memo tables which store all the abstract call-success pairs that occur in the program (and their dependencies). In the case of energy analysis this includes, for each procedure, and for each possible abstract call state and path, a function that returns the corresponding energy consumed by that procedure and class of calls, as a function of input data sizes.

This work builds on our earlier work on cost analysis, initially developed for granularity control during automatic program parallelization [1, 11], and which was capable of inferring both upper- and lower-bound functions [3, 2], for a wide class of *user-definable* resources [16, 15]. This configureability of the system is instrumental in the energy application for representing low-level energy models. As mentioned before, this work was recently extended by us to be fully based on multivariant abstract interpretation using sized shapes (sized types) [17], and to perform static profiling [12].

For verification and (performance) error detection the inferred abstract models of energy consumption are compared to the energy specifications [9, 5]. This can optionally be done during the analysis or after it. In our approach specifications can include both lower and upper bounds on energy usage, and they can express intervals within which energy usage is to be certified to be within such bounds. The bounds of the intervals can be given in general as functions on input data sizes. Our verification system can prove whether such energy usage specifications are met or not. It can also infer the particular conditions under which the specifications hold. To this end, these conditions are also expressed as intervals of functions of input data sizes, such that a given specification can be proved for some intervals but disproved for others. The specifications themselves can also include preconditions expressing intervals for input data sizes.

We also report on a prototype implementation of our approach within the CiaoPP system for the XC language and XS1-L architecture, with an emphasis on embedded programs. We provide experimental results from energy analysis of such programs at the instruction set architecture level and compare them to the actual energy consumption measured on the hardware. We also illustrate with an example how embedded software developers can use this tool for determining values for program parameters that ensure meeting a given energy budget while minimizing losses in quality of service.

## References

[1] S . K. Debray, N.-W. Lin & M. Hermenegildo (1990): *Task Granularity Analysis in Logic Programs*. In: *Proc. 1990 ACM Conf. on Programming Language Design and Implementation(PLDI)*, ACM Press, pp. 174–188, doi:10.1145/93548.93564.

[2] S . K. Debray, P. López-García, M. Hermenegildo & N.-W. Lin (1994): *Estimating the Computational Cost of Logic Programs*. In: *SAS'94*, *LNCS* 864, Springer-Verlag, pp. 255–265, doi:10.1007/3-540-58485-4_45.

[3] S . K. Debray, P. López-García, M. Hermenegildo & N.-W. Lin (1997): *Lower Bound Cost Estimation for Logic Programs*. In: *ILPS'97*, MIT Press, pp. 291–305.

[4] K. S. Henriksen & J. P. Gallagher (2006): *Abstract Interpretation of PIC Programs through Logic Programming*. In: *SCAM'06: Proceedings of the Sixth IEEE International Workshop onSource Code Analysis and Manipulation*, IEEE Computer Society, pp. 184–196, doi:10.1109/SCAM.2006.1.

[5] M. Hermenegildo, G. Puebla, F. Bueno & P. López-García (2005): *Integrated Program Debugging, Verification, and Optimization Using Abstract Interpretation (and The Ciao System Preprocessor)*. *Science of Comp. Progr.* 58(1–2), pp. 115–140, doi:10.1016/j.scico.2005.02.006.

[6] U . Liqat, Z. Banković, P. López-García & M. Hermenegildo (2016): *Inferring Energy Bounds Statically by Evolutionary Analysis of Basic Blocks*. In: *(HIP3ES'16)*, IEEE Computer Society. arXiv:1601.02800.

[7] U . Liqat, K. Georgiou, P. López-García, M. Hermenegildo, J. P. Gallagher & K. Eder (2016): *Inferring Parametric Energy Consumption Functions at Different Software Levels: ISA vs. LLVM IR*. In: *Proc. of FOPARA-2015*, *LNCS* 9964, Springer, pp. 81–100, doi:10.1007/978-3-319-46559-3_5.

[8] U . Liqat, S. Kerrison, A. Serrano, K. Georgiou, P. López-García, N. Grech, M. Hermenegildo & K. Eder (2014): *Energy Consumption Analysis of Programs based on XMOS ISA-level Models*. In: *Proceedings of LOPSTR'13*, *LNCS* 8901, Springer, pp. 72–90, doi:10.1007/978-3-319-14125-1_5.

[9] P. López-García, L. Darmawan, F. Bueno & M. Hermenegildo (2012): *Interval-Based Resource Usage Verification: Formalization and Prototype*. In: *Proc. of FOPARA-2011*, *LNCS* 7177, Springer-Verlag, pp. 54–71, doi:10.1007/978-3-642-32495-6_4.

[10] P. López-García, R. Haemmerlé, M. Klemen, U . Liqat & M. Hermenegildo (2015): *Towards Energy Consumption Verification via Static Analysis*. In: *WS on High Perf. Energy Efficient Embedded Sys. (HIP3ES)*. arXiv:1512.09369.

[11] P. López-García, M. Hermenegildo & S . K.Debray (1996): *A Methodology for Granularity Based Control of Parallelism in Logic Programs*. *J. of Symbolic Computation, Special Issue on Parallel Symbolic Computation* 21(4–6), pp. 715–734, doi:10.1006/jsco.1996.0038.

[12] P. López-García, M. Klemen, U . Liqat & M. Hermenegildo (2016): *A General Framework for Static Profiling of Parametric Resource Usage*. *Theory and Practice of Logic Programming* 16(5–6 (32nd International Conference on Logic Programming)), pp. 849–865, doi:10.1017/S1471068416000442.

[13] M. Méndez-Lojo, J. Navas & M. Hermenegildo (2007): *A Flexible (C)LP-Based Approach to the Analysis of Object-Oriented Programs*. In: *Proc. of LOPSTR-2007*, *LNCS* 4915, Springer-Verlag, pp. 154–168, doi:10.1007/978-3-540-78769-3_11.

[14] J. Navas, M. Méndez-Lojo & M. Hermenegildo (2008): *Safe Upper-bounds Inference of Energy Consumption for Java Bytecode Applications*. In: *The Sixth NASA Langley Formal Methods Workshop (LFM 08)*, NASA, pp. 29–32.

[15] J. Navas, M. Méndez-Lojo & M. Hermenegildo (2009): *User-Definable Resource Usage Bounds Analysis for Java Bytecode*. In: *BYTECODE'09*, *ENTCS* 253, Elsevier, pp. 65–82, doi:10.1016/j.entcs.2009.11.015.

[16] J. Navas, E. Mera, P. López-García & M. Hermenegildo (2007): *User-Definable Resource Bounds Analysis for Logic Programs*. In: *Proc. of ICLP'07*, *LNCS* 4670, Springer, pp. 348–363, doi:10.1007/978-3-540-74610-2_24.

[17] A .Serrano, P. López-García & M. Hermenegildo (2014): *Resource Usage Analysis of Logic Programs via Abstract Interpretation Using Sized Types*. *Theory and Practice of Logic Programming* 14(4–5 (30th International Conference on Logic Programming)), pp. 739–754, doi:10.1017/S147106841400057X.