Using Bayesian Learning to Estimate How Hot an Execution Path is

Karim Ali
{karim}@cs.uwaterloo.ca
David R. Cheriton School of Computer Science
University of Waterloo

Abstract—The abstract goes here.

REFERENCES

[1] C. Boogerd and L. Moonen, "On the Use of Data Flow Analysis in Static Profiling," in *Source Code Analysis and Manipulation, 2008 Eighth IEEE International Working Conference on*. IEEE, 2008, pp. 79–88.

[2] S. Heckman and L. Williams, "A Systematic Literature Review of Actionable Alert Identification Techniques for Automated Static Code Analysis."

[3] R. Buse and W. Weimer, "The road not taken: Estimating path execution frequency statically," in *Proceedings of the 2009 IEEE 31st International Conference on Software Engineering*. IEEE Computer Society, 2009, pp. 144–154.

[4] B. Calder, D. Grunwald, M. Jones, D. Lindsay, J. Martin, M. Mozer, and B. Zorn, "Evidence-based static branch prediction using machine learning," *ACM Transactions on Programming Languages and Systems (TOPLAS)*, vol. 19, no. 1, pp. 188–222, 1997.

[5] T. Baba, T. Masuho, T. Yokota, and K. Ootsu, "Design of a two-level hot path detector for path-based loop optimizations," in *Proceedings of the third conference on IASTED International Conference: Advances in Computer Science and Technology*. ACTA Press, 2007, p. 28.

[6] T. Ball and J. Larus, "Efficient path profiling," in *micro*. Published by the IEEE Computer Society, 1996, p. 46.

[7] E. Duesterwald and V. Bala, "Software profiling for hot path prediction: Less is more," *ACM SIGARCH Computer Architecture News*, vol. 28, no. 5, pp. 202–211, 2000.

[8] M. Merten, A. Trick, C. George, J. Gyllenhaal, and W. Hwu, "A hardware-driven profiling scheme for identifying program hot spots to support runtime optimization," in *Proceedings of the 26th annual international symposium on Computer architecture*. IEEE Computer Society, 1999, p. 147.

[9] G. Ammons and J. Larus, "Improving data-flow analysis with path profiles," in *Proceedings of the ACM SIGPLAN 1998 conference on Programming language design and implementation*. ACM, 1998, pp. 72–84.

[10] S. Graham, P. Kessler, and M. Mckusick, "Gprof: A call graph execution profiler," *ACM Sigplan Notices*, vol. 17, no. 6, pp. 120–126, 1982.

[11] W. Weimer and G. Necula, "Exceptional situations and program reliability," *ACM Transactions on Programming Languages and Systems (TOPLAS)*, vol. 30, no. 2, pp. 1–51, 2008.

[12] J. Larus, "Whole program paths," in *Proceedings of the ACM SIGPLAN 1999 conference on Programming language design and implementation*. ACM, 1999, pp. 259–269.

[13] T. Ball and J. Larus, "Branch prediction for free," *ACM SIGPLAN Notices*, vol. 28, no. 6, pp. 300–313, 1993.

[14] P. Mehra and B. Wah, "Synthetic Workload Generation for Load-Balancing Experiments," *IEEE Parallel & Distributed Technology: Systems & Technology*, vol. 3, no. 3, pp. 4–19, 1995.

[15] A. Nanda and L. Ni, "Benchmark workload generation and performance characterization of multiprocessors," in *Proceedings of the 1992 ACM/IEEE conference on Supercomputing*. IEEE Computer Society Press, 1992, pp. 20–29.

[16] K. Sen, "Concolic testing," in *Proceedings of the twenty-second IEEE/ACM international conference on Automated software engineering*. ACM, 2007, pp. 571–572.

[17] K. Sen and G. Agha, "CUTE and jCUTE: Concolic unit testing and explicit path model-checking tools," in *Computer Aided Verification*. Springer, 2006, pp. 419–423.

[18] S. Goldsmith, A. Aiken, and D. Wilkerson, "Measuring empirical computational complexity," in *Proceedings of the the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering*. ACM, 2007, pp. 395–404.

[19] R. Vallée-Rai, P. Co, E. Gagnon, L. Hendren, P. Lam, and

V. Sundaresan, “Soot-a Java bytecode optimization framework,” in *Proceedings of the 1999 conference of the Centre for Advanced Studies on Collaborative research*. IBM Press, 1999, p. 13.

[20] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten, “The WEKA data mining software: An update,” *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.