{tag}

{/tag}
International Journal of Computer Applications

Volume 67 - Number 18

Year of Publication: 2013

Authors:

N. Rajasekaran

K. K. Sureshkumar

N. M. Elango

10.5120/11493-7200

{bibtex}pxc3887200.bib{/bibtex}

Abstract

Randomized testing has been shown to be an effective method for testing software units. However, the thoroughness of randomized unit testing varies widely according to the input which is provided by the user. Such as the relative frequencies with which methods are called to be improved the thoroughness of randomized unit testing. In this paper the system, describes genetic algorithm based parameter finding for randomized unit testing that optimizes test coverage. Here the unit test data will be generated by nighthawk system. The system can be viewed as two levels, lower level and upper level. Randomized unit testing engine is a lower level, which tests a set of methods according to parameter values specified as genes in a chromosome, including parameters that encode a value reuse policy. The upper level is a genetic algorithm (GA) which uses fitness evaluation, selection, mutation and recombination of chromosomes in order to find out good values for the genes. Integrity is evaluated on the basis of test coverage and number of method calls performed. To find good parameters users can use Nighthawk and then perform with randomized unit testing based on those parameters. Many new test cases can quickly generate by randomized testing that achieve high coverage, and can continue to do so for as long as users wish to run it. In this research the test coverage

results of Nighthawk are compared with manual unit testing results [6]. The Nighthawk system produced maximum test coverage results in less timing based on the genetic algorithm comparing with manual unit testing results.

References

- Andrews J. H, S. Haldar, Y. Lei, and C. H. F. Li, "Tool Support for Randomized Unit Testing," Proc. First Int'l Workshop Randomized Testing, pp. 36-45, July 2006.
- Andrews J. H and Y. Zhang, "General Test Result Checking with Log File Analysis," IEEE Trans. Software Eng. , vol. 29, no. 7, pp. 634-648, July 2003.
- Andrews. J, F. Li, and T. Menzies, "Nighthawk: A Two-Level Genetic-Random Unit Test Data Generator," Proc. 22nd IEEE/ACM Int'l Conf. Automated Software Eng. ,
- Andrews. J and T. Menzies, "On the Value of Combining Feature Subset Selection with Genetic Algorithms: Faster Learning of Coverage Models," Proc. Fifth Int'l Conf. Predictor Models in Software Eng. ,
- Anatoly's and R. G. Hamlet, "Automatically Checking an Implementation against its Formal Specification," IEEE Trans. Software Eng. , vol. 26, no. 1, pp. 55-69, Jan. 2000.
- Ball. T, "A Theory of Predicate-Complete Test Coverage and Generation," Proc. Third Int'l Symp. Formal Methods for Components and Objects, pp. 1-22, Nov. 2004.
- Ciupa. I, A. Leitner, M. Oriol, and B. Meyer, "Artoo: Adaptive Random Testing for Object-Oriented Software," Proc. 30th ACM/ IEEE Int'l Conf. Software Eng. , pp. 71-80, May 2008.
- Claessen. K and J. Hughes, "QuickCheck: A Lightweight Tool for Random Testing of Haskell Programs," Proc. Fifth ACM SIGPLAN Int'l Conf. Functional Programming, pp. 268-279, Sept. 2000.
- Clarke L. A, "A System to Generate Test Data and Symbolically Execute Programs," IEEE Trans. Software Eng. , vol. 2, no. 3, pp. 215-222, Sept. 1976.
- Csallner. C and Y. Smaragdakis, "JCrasher: An Automatic Robustness Tester for Java," Software Practice and Experience, vol. 34, no. 11, pp. 1025-1050, 2004.
- Doong. R. K and P. G. Frankl,"The ASTOOT Approach to Testing Object-Oriented Programs," ACM Trans. Software Eng. and Methodology, vol. 3, no. 2, pp. 101-130, Apr. 1994.
- Ernst M. D. , J. Cockrell, W. G. Griswold, and D. Notkin, "Dynamically Discovering Likely Program Invariants to Support Program Evolution," IEEE Trans. Software Eng. , vol. 27, no. 2, pp. 99-123, Feb. 2001.
- Godefroid. P, N. Klarlund, and K. Sen, "DART: Directed Automated Random Testing," Proc. ACM SIGPLAN Conf. Programming Language Design and Implementation, pp. 213-223, June 2005.
- Goldberg. D. E, Genetic Algorithm in Search, Optimization, and Machine Learning. Addison-Wesley, 1989.

- Groce . A, G. J. Holzmann, and R. Joshi, "Randomized Differential Testing as a Prelude to Formal Verification," Proc. 29th Int'l Conf. Software Eng. , pp. 621-631, May 2007.
- Guo. Q, R. M. Hierons, M. Harman, and K. Derderian, "Computing Unique Input/Output Sequences Using Genetic Algorithms," Proc. Third Int'l Workshop Formal Approaches to Testing of Software, pp. 164-177, 2004.
- Gupta. N, A. P. Matcher, and M. L. Soffa, "Automated Test Data Generation Using an Iterative Relaxation Method," Proc. Sixth Int'l Symp. Foundations of Software Eng. , pp. 224-232, Nov. 1998.
- Hamlet. R, "Random Testing," Encyclopedia of Software Eng. , Wiley, pp. 970-978, 1994.
- Holland J. H, Adaptation in Natural and Artificial Systems. University of Michigan Press, 1975.
- King J. C, "Symbolic Execution and Program Testing," Comm. ACM, vol. 19, no. 7, pp. 385-394, 1976. Kira. K and L. Rendell, "A Practical Approach to Feature Selection," Proc. Ninth Int'l Conf. Machine Learning, pp. 249-256, 1992.
- Korel. B, "Automated Software Test Generation," IEEE Trans. Software Eng. , vol. 16, no. 8, pp. 870-879, Aug. 1990.
- Leow W. K, S. C. Khoo, and Y. Sun, "Automated Generation of Test Programs from Closed Specifications of Classes and Test Cases," Proc. 26th Int'l Conf. Software Eng. , pp. 96-105, May 2004.
- Michael C. C, G. McGraw, and M. A. Schatz, "Generating Software Test Data by Evolution," IEEE Trans. Software Eng. , vol. 27, no. 12, pp. 1085-1110, Dec. 2001.
- Miller B. P, L. Fredriksen, and B. So, "An Empirical Study of the Reliability of UNIX Utilities," Comm. ACM, vol. 33, no. 12, pp. 3244, Dec. 1990.
- Owen. D and T. Menzies, "Lurch: A Lightweight Alternative to Model Checking," Proc. 15th Int'l Conf. Software Eng. and Knowledge Eng. , pp. 158-165, July 2003.

Index Terms

Computer Science Software Testing

Keywords

Randomized Unit Testing Feature Subset Selection Nighthawk. Software under Test (SUT)