Compiler-Assisted Test Acceleration on GPUs for Embedded Software

Title	Compiler-Assisted Test Acceleration on GPUs for Embedded Software
Publication Type	Conference Paper
Year of Publication	2017
Authors	Yaneva, Vanya, Rajan, Ajitha, Dubach, Christophe
Conference Name	Proceedings of the 26th ACM SIGSOFT International Symposium on Software Testing and Analysis
Publisher	ACM
Conference Location	New York, NY, USA
ISBN Number	978-1-4503-5076-1
Keywords	Automated Testing, Collaboration, compilers, Embedded Software, Functional testing, GPUs, human factors, Metrics, policy-based governance, Policy-Governed Secure Collaboration, pubcrawl, resilience, Resiliency, Safe Coding

Abstract

Embedded software is found everywhere from our highly visible mobile devices to the confines of our car in the form of smart sensors. Embedded software companies are under huge pressure to produce safe applications that limit risks, and testing is absolutely critical to alleviate concerns regarding safety and user privacy. This requires using large test suites throughout the development process, increasing time-to-market and ultimately hindering competitivity. Speeding up test execution is, therefore, of paramount importance for embedded software developers. This is traditionally achieved by running, in parallel, multiple tests on large-scale clusters of computers. However, this approach is costly in terms of infrastructure maintenance and energy consumed, and is at times inconvenient as developers have to wait for their tests to be scheduled on a shared resource. We propose to look at exploiting GPUs (Graphics Processing Units) for running embedded software testing. GPUs are readily available in most computers and offer tremendous amounts of parallelism, making them an ideal target for embedded software testing. In this paper, we demonstrate, for the first time, how test executions of embedded C programs can be automatically performed on a GPU, without involving the end user. We take a compiler-assisted approach which automatically compiles the C program into GPU kernels for parallel execution of the input tests. Using this technique, we achieve an average speedup of 16A when compared to CPU execution of input tests across nine programs from an industry standard embedded benchmark suite.

URL https://dl.acm.org/citation.cfm?doid=3092703.3092720
DOI 10.1145/3092703.3092720
Citation Key yaneva_compiler-assisted_2017

Automated Testing collaboration compilers embedded software Functional testing GPUs Human Factors Metrics policy-based governance Policy-Governed Secure Collaboration pubcrawl resilience Resiliency Safe Coding