

The development cycle of large software is necessarily prone to introducing software errors that are hard to find and fix. Automated regular testing (regression testing) is a popular method used to reduce the cost of finding and fixing functionality errors, but it neglects software performance. The thesis focuses on performance errors, enabling automated detection of performance changes during software development (regression benchmarking). The key investigated problem is non-determinism in computer systems, which causes performance fluctuations. The problem is addressed by a novel benchmarking methodology based on statistical methods. The methodology is evaluated on a large open-source project Mono, detecting daily performance changes since August 2004, and on open-source CORBA implementations omniORB and TAO. The benchmark automation is a complex task in itself. As suggested by experience with compilation of weather forecast model Arpege/Aladin and implementation of component model SOFA, large systems place distinguishing demands on tasks such as automated compilation or execution. Complemented by experience from Mono benchmarking, the thesis proposes an architecture of a generic environment for automated regression benchmarking. The environment is being implemented by master students under supervision of the author of the thesis.