Abstract

A large-scale distributed system needs to be tested thoroughly before production deployment. It is crucial to inject several kinds of faults to assess application behaviour, fault tolerance and scalability. This is often done with the help of various network simulators like NS3, Omnet++, Mininet and Opnet. Generally this involves a considerable amount of coding within the network simulator code in order to facilitate the desired test scenarios, involving its own learning curve that requires substantial time and effort. This in turn affects and prolongs the overall life-cycle of the software development and deployment. It is this process that we seek to simplify with this project. In this dissertation, we present a proof of concept in which a developer/tester can easily test the behaviour any large-scale application, independent of language and architecture with the help of a graphical user interface.

Apart from the network simulator, a major component of the approach is to use an application containerization framework like Linux containers (LXC) or Docker for a dynamic and configurable deployment of any application in the test environment. One of the most important uses of containerization frameworks to to maintain a clear separation between the network simulator and the test application code bases. This involves facilitating containers with work in conjunction with the network simulator.

For the proof of concept, our algorithm demonstrates a few specific functionalities that we think are critical in the process of testing any networked application namely: topology generation, link up/down, adding/removing network nodes during application run-time. We illustrate the effectiveness of the system by testing how dynamically we can create nodes, run applications on them, how easily we can inject faults and test the behaviour of the application being simulated.