Chapter 110
D-Cloud:
Software Testing Environment for Dependable Distributed Systems Using Cloud Computing Technology

Toshihiro Hanawa
University of Tsukuba, Japan

Mitsuhisa Sato
University of Tsukuba, Japan

ABSTRACT

Various information systems are widely used in the information society era, and the demand for highly dependable system is increasing year after year. However, software testing for such a system becomes more difficult due to the enlargement and the complexity of the system. In particular, it is often difficult to test parallel and distributed systems in the real world after deployment, although reliable systems, such as high-availability servers, are parallel and distributed systems. To solve these problems, the authors propose a software testing environment for dependable parallel and distributed system using the cloud computing technology, named D-Cloud. D-Cloud consists of the cloud management software as the role of the resource management, and a lot of virtual machine monitors with fault injection facility in order to simulate hardware faults. In addition, D-Cloud introduces the scenario manager, and it makes a number of different tests perform automatically. Currently, D-Cloud is realized by the use of Eucalyptus as the cloud management software. Furthermore, the authors introduce FaultVM based on QEMU as the virtualization software, and D-Cloud frontend that interprets test scenario, constructs test environment, and dispatches commands. D-Cloud enables automating the system configuration and the test procedure as well as performing a number of test cases simultaneously and emulating hardware faults flexibly. This chapter presents the concept and design of D-Cloud, and describes how to specify the system configuration and the test scenario. Furthermore, the preliminary test example as the software testing using D-Cloud is presented. As the result, the authors show that D-Cloud allows easy setup of the environment, and to test the software testing for the distributed system.

DOI: 10.4018/978-1-4666-6539-2.ch110

INTRODUCTION

According to the formulation of advanced information society, various information systems are widely used. Since such systems are closely related to daily life, they must employ highly dependable facilities to avoid undesirable behavior caused by the underlying bugs and interference from the external environment. In order to certificate the dependability of such systems, these systems should be tested sufficiently. However, as recent information system becomes larger and more complicated, software testing for such a system becomes more difficult. In order to check whether components work correctly or not, tremendous test cases are needed for various input patterns, and environment to execute a great number of tests immediately should be provided.

Although highly dependable systems such as high-availability servers especially likely to form parallel and distributed systems, the testing of large-scale parallel and distributed system is troublesome job in the real world after deployment. When a failure occurs in such complex systems, the reproducibility of the failure in the actual system is so poor that the detection of the defective part has been a serious problem.

On the other hand, a highly dependable system should be equipped with the combination of multiple functions of fault tolerance against hardware faults. Even though testing fault tolerant facilities should be done under hardware fault conditions or anomaly loads, it is too difficult to destroy a specific part of actual hardware or to concentrate an unrealistic overload in a hardware device.

To solve these problems, we proposed a software testing environment for reliable distributed systems using cloud computing technology, named "D-Cloud" (Hanawa, 2010, Banzai, 2010). In this chapter, we present the concept and design D-Cloud, discuss the description of the system configuration and the test scenario, and demonstrate the preliminary test example using D-Cloud.

After that, we explain related works, then we explain the concept of D-Cloud. Next, we describe the design of D-Cloud as a software testing environment, and the test configuration and test scenario is denoted. In addition, preliminary test examples using D-Cloud are demonstrated. Finally, we conclude our study and discuss future works.

RELATED WORK

A number of fault injection techniques in program tests have been proposed. DOCTOR (Han, 1995) is a software fault injector, which supports memory faults, CPU faults, and communication faults. However, software fault injection in the DOCTOR requires modification of the source codes to be tested. Xception (Carreira, 1998) uses the kernel module to inject the fault, and the tester sets the hardware breakpoint to the specified address. When the process reaches the breakpoint, the fault is injected into the register or memory as an incorrect value.

On the other hand, BOND (Baldini, 2000) uses a special agent to intercept the system call of Windows NT 4.0 OS. The agent hooks the system call from the target application and returns an incorrect value to the application. Although this scheme does not require any modification of either the OS or the target application, the fault can only be injected to a location that can be accessed through the software, such as the register and memory. Moreover, this scheme cannot be applied to test the OS.

FAUmachine (Potyra, 2007) performs a software test using virtual machines for the fault injection mechanism. However, since FAUmachine does not provide an automated test environment, the tester must configure the test environment manually.

Meanwhile, large-scale software testing has been studied. GridUnit (Duarte, 2006) executes software tests automatically on the grid referred