Modular Software Model Checking for Distributed Systems

Abstract—

Distributed systems are complex, being usually composed of several subsystems running in parallel. Concurrent execution and inter-process communication in these systems are prone to errors that are difficult to detect by traditional testing, which does not cover every possible program execution. Unlike testing, model checking can detect such faults in a concurrent system by exploring every possible state of the system. However, most model-checking techniques require that a system be described in a modeling language. Although this simplifies verification, faults may be introduced in the implementation. Recently, some model checkers verify program code at runtime but tend to be limited to stand-alone programs. This article proposes cache-based model checking, which relaxes this limitation to some extent by verifying one process at a time and running other processes in another execution environment. This approach has been implemented as an extension of Java Path Finder, a Java model checker. It is a scalable and promising technique to handle distributed systems. To support a larger class of distributed systems, a checkpointing tool is also integrated into the verification system. Experimental results on various distributed systems show the capability and scalability of cache-based model checking.