Verifying Networked Programs Using a Model Checker Extension

Abstract
Model checking finds failures in software by exploring every possible execution schedule. Until recently it has been mainly applied on stand-alone applications. We propose an extension for a Java model checker to support networked programs. It contains a cache module, which captures data streams between a target process and a peer process. Captured data are replayed by the cache module when a duplicate request is sent. This demonstration shows how we found a defect in a WebDAV client with a model checker and our extension.

Background
Software model checking verifies software by exploring every possible schedule whereas software testing only executes the program through one thread schedule for each run. Java Pathfinder (JPF), a model checker for Java, is used as a base model checker for our development. It includes its own Java Virtual Machine, which explores all thread schedules of the program. Although it is designed to verify only a single process at a time, we extend its functionalities via several mechanisms to support multi-process networked applications.

Concept of Cache
Cache can be used as a proxy to the real external process. Net-locache, our JPF extension, makes use of requests and responses in the past and sends already known responses back to the target application instead of dispatching request messages to the peer process. As a result, peer processes do not become aware of the target application being driven by the model checker. If the request is not cached, the JPF-cache will physically send the request to the peer, wait for a response, and remember it.

Architecture
- Implemented as a JPF extension.
- Network-related classes are rewritten as abstract classes.
- Abstract output stream redirects outputs of the SUT to the cache.
- Abstract input stream accepts input from the cache.
- Listener signals the cache on the state transition event.
- Cache saves/restores pointer positions and the number of active connections.

Tool Download
- javapathfinder from http://jpf.sourceforge.net

References