NFTAPE: Networked Fault Tolerance and Performance Evaluator


Center for Reliable and High Performance Computing
University of Illinois at Urbana-Champaign
1308 W. Main St., Urbana, IL 61801
Email: [dstott, ph-jones, m-hamman, kalbar, iyer]@csrc.uiuc.edu

1 What is NFTAPE?
The NFTAPE is a software implemented, highly flexible fault injection environment for conducting automated fault/error injection-based dependability characterization. NFTAPE: (1) enables a user: (i) to specify a fault/error injection plan, (ii) to carry on injection experiments, and (iii) to collect the experimental results for analysis; (2) targets assessment of a broad set of dependability metrics, e.g., availability, reliability, coverage; (3) operates in a distributed environment; (4) can be configured to implement a variety of fault/error injection strategies and thus to serve multiple users and target systems; (5) imposes minimal disturbance of target systems.

A measure of the effectiveness of the NFTAPE environment is the breadth of the fault space (fault location, fault type, and fault trigger) the tool can assess. To attain a high coverage in the sense mentioned above, it is essential to separate the fault injection component, the fault trigger component, and the control mechanisms. With this separation (or modularity), components – Lightweight Fault Injector (LWFI) or Lightweight Trigger (LWT) – can be added with little effort and any combination of trigger and fault injection can be configured.

To support these components and other experimental processes (e.g., target application), NFTAPE provides a common control mechanism and a scripting language to automate executing fault injection campaigns. The NFTAPE API helps developers write new LWFI and LWT components so that these components can be swapped at run-time. Figure 1 illustrates components and a typical setup of NFTAPE-based fault/error injection experiment.

2 NFTAPE Features
Fault injector types: (1) debugger-based (e.g., Solaris, Linux, Lynx) – injection to the target process memory and registers, (2) driver-based (e.g., Linux, Solaris) – injection to memory, registers, OS functions, I/O devices, (3) network injector – injection to network cards/controllers; corrupting messages (e.g., VxWorks), (4) use of performance monitors (in processors) to trigger fault injection.

Fault injection targets: processor registers, memory, network, application, specific OS function.
Fault injection triggers: random (based on time), application supplied breakpoints, externally supplied breakpoints
Fault/error models: single/selected multiple bit flips, transient and permanent faults

3 Example Applications of NFTAPE
Motorola IDEN MicroLite: critical base station controller (call-processing application and database) in digital mobile telephone network.
DHCP (Dynamic Host Configuration Protocol) server – evaluation of application control flow checking.
Software implemented fault tolerance (SIFT) environment on REE testbed – evaluation of recovery coverage and performance overhead of the SIFT environment.
Internet server applications: ftp and ssh (secure shell) - evaluation of error induced security vulnerabilities in ftp and ssh applications.
Voltan and Chameleon ARMors software middlewares - evaluation of fail-silence provided by process duplication (Voltan) versus internal error detection (Chameleon).

REFERENCES

Figure 1: Typical NFTAPE-based Fault Injection Experiment Setup