Volume 4, Issue 4, April 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Fault Injection and Its Techniques

Ruchi Gaba*
PG Scholar
KITM, Kurukshetra, India

Er. Shobhit Gupta
Assistant Professor,
Department of CSE & IT, KITM Kurukshetra, India

Abstract— A fault is a defect in a program, usually difficult to pinpoint .A faults may occur at single points or distributed points. In software testing, fault injection is a technique of introducing faults into the code for improving the coverage and usually used with stress testing for robustness of the developed software. When the fault-tolerance mechanisms detect an error, they may initiate several actions to handle the faults and contain its errors. The research objective is to survey fault injection techniques that are used to validate the dependability of a system by analysing the behaviour of the devices when a fault occurs. This paper presents an introduction to fault injection and techniques.

Keywords — Fault, Fault tolerance, Fault injection, Dependability, System reliability, System robustness.

I. INTRODUCTION

In software testing, **fault injection** is a technique by introducing faults into the code for improving the coverage and usually used with stress testing for robustness of the developed software. Fault injection is done by introducing faults to test code paths in particular error handling code paths that might otherwise rarely be followed. A system may not always perform the function it is intended for. The causes and consequences of deviations from the expected function of a system are called the factors to dependability:

- Fault is a cause of an error. It may be defined as imperfection or flaw, physical defect that occurs within some hardware or software component.
- Error is a deviation from correctness and accuracy and is the manifestation of fault
- Failure is deviation of the component or system from its expected delivery, service or result that is due or expected

When a fault causes an incorrect change in a machine stage, an error occurs. The process of introducing faults in a system in order to assess its behaviour and to measure the efficiency (coverage, latency, etc.) of fault tolerance mechanisms is referred to as fault injection.

Fault injection is the process of deliberately inserting an error into an application under test and then running the application to determine whether the application deals with the error properly. Fault injection covers a variety of techniques for introducing faults in systems to measure their response to those faults. Fault injection is best for measuring the fault tolerance or robustness of a system. In particular, it can be used in both electronic hardware systems and software systems to measure the fault tolerance of the system.

Hardware or Physical faults that arise during system operations are best classified by their duration:

- ***Permanent faults:*** Caused by irreversible component damage, recovery can only be accomplished by replacing or repairing the damaged component or subsystem.
- ***Transient faults:*** Triggered by environmental conditions such as power-line fluctuation, electromagnetic interference, or radiation.
- ***Intermittent faults:*** Caused by unstable hardware or varying hardware states. They can be repaired by replacement or redesign.

During the process of software development, faults can be created in every step such as requirement definition, requirement specifications, design, and implementation. **Software faults** can be cataloged as:

- ***Function faults:*** Incorrect / missing implementation that requires a design to be corrected or changed.
- ***Algorithm faults:*** Incorrect /missing implementation that can be fixed without requiring change in design.
- ***Timing/Serialization faults:*** Missing / incorrect serialization of shared resources.
- ***Checking fault:*** Missing/ incorrect validation of data, incorrect loop, and incorrect conditional statement.
- ***Assignment fault:*** Values are incorrectly assigned or not assigned.

II. FAULT INJECTION ENVIRONMENT

A fault injection environment consists of the following components:

- ***Fault injector:*** it is used to inject faults into the target system so it executes commands from the workload generator.
- ***Fault library:*** It stores different types of fault, locations of fault, fault times, and appropriate hardware semantics or software structure.

- ***Workload generator:*** It provides the workload for the targeted system as input
- ***Workload library:*** In this sample workloads are stored for the target system
- ***Controller:*** Controls the experiment.
- ***Monitor:*** It keeps the tracks of execution of commands and initiates data collection whenever necessary.
- ***Data collector:*** It performs the online collection of data.
- ***Data analyzer:*** Performs data processing and analysis.

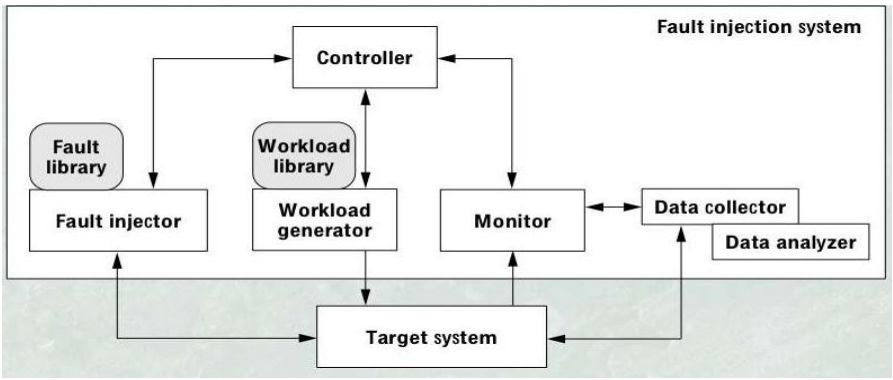

Fig.1 Basic components of a fault injection environment [1].

III. FAULT INJECTION TECHNIQUES

Fault injection reduces the impact of anomalies by introducing artificial anomalies to see the tolerance of the targeted system. The fault injection techniques have been recognized for a long time as necessary to validate the dependability of a system by analysing the behaviour of the devices when a fault occurs. Several efforts have been made to develop techniques for injecting faults into a system prototype or model. Fault injection tests fault detection ,fault isolation, and reconfiguration and recovery capabilities.

Hardware-based fault injection: It is accomplished at physical level. It uses additional hardware to introduce fault into the target system. Depending on the faults and their locations, hardware fault injection methods is of two types:

- Hardware fault injection *with contact (pin level injection):* The fault injector has direct contact with the target system, producing current or voltage changes externally in the targeted chip.
- Hardware fault injection *without contact*: faults is injected by creating heavy ion radiation.

Benefits:

- Using hardware fault injection technique can access locations which are hard to be accessed by other means. For example, the Heavy-ion radiation method can inject fault into VLSI circuits.
- It works well for the system which needs high resolution of time for hardware monitoring and triggering.
- To evaluate the experiments by injection into actual hardware is many cases the only practical way to estimate coverage and latency accurately.
- Faults injects by this technique have low perturbation.
- This technique is best suitable for the low-level fault models.
- No modification of the target system is required to inject faults.
- Experiments done by this technique are fast.
- Experiments can be run in near real-time, allowing for the possibility of running a large number of fault injection experiments.
- Using the same software that will run in the field, fault injection experiments are performed
- To inject fault, no development of model or validation required.
- It can model the permanent fault at pin-level.

Drawbacks:

- It can introduce high risk of damage for the injected system.
- Some hardware fault injection methods, such a state mutation, require stopping and restarting the processor to inject a fault, it is not always effective for measuring latencies in the physical systems.
- Low portability and observability.
- Limited set of injection points and injectable faults.
- The setup time for each experiment might, in fact, offset the time gained by the ability to perform the experiments in near real time.
- Requires special-purpose hardware in order to perform the fault injection experiments. This hardware is used to inject faults into the processor by applying the rail voltages to the Input/Output (I/O) pins of the processor.
- Limited observability and controllability. At best, it is able to corrupt the I/O pins of the processor and the internal processor registers.

Software-based fault injection or software implemented fault injection: Software Fault Injection consists in the deliberate introduction of software faults into a system, in order to assess the impact of hidden faults in the software, and in order to improve fault tolerance algorithms and mechanisms.The objective of this technique consists of reproducing at software level the errors that would have been produced upon occurring faults in the hardware. Software fault injection method can be categorize on the basis of when the faults are injected:

- *Compile-Time Injections* - It is a fault injection technique where source code is modified to inject simulated faults into a system.
- *Run-Time Injections* - It makes use of software trigger to inject a fault into a software system during run time. The Trigger can be time based triggers and interrupt based trigger.

In order to perform fault injection in software three aspects need to be defined:

i. **What to inject** ,this type of aspect is related to kind of errors to be injected by replacing the correct values with the incorrect one
ii. **When to inject**, this type of aspect is related to the instant in which an error is injected can be time or event dependent.
iii. **Where to inject,** this type of aspect is related to the location of errors to be injected.

Benefits:

- This fault injection technique can be targeted to applications and operating systems which are not done by the hardware fault injection.
- This technique can run real time scenarios.
- This technique does not require any special-purpose hardware for fault injection;
- Low complexity, low development and low implementation cost.
- No model for development or validation required.
- It can be expanded for new classes of faults.

Drawbacks:

- This technique is not able to insert faults into location that are not accessible to software
- To inject fault modification of the source code is required to support the fault injection.
- Limited observability and controllability.
- To model permanent fault is impossible.

Table1. Comparison of fault injection techniques [2]

	Hardware		**Software**	
	With contact	**Without contact**	**Compile time**	**Runtime**
Cost	High	High	Low	Low
Perturbation	None	None	Low	High
Risk of Change	High	Low	None	None
Monitoring Time Resolution	High	High	High	Low
Accessibility of fault injection points	Chip pin	Chip Internal	Register Memory Software	Register Memory I/O Controller/Port
Controllability	High	Low	High	High
Trigger	Yes	No	Yes	Yes
Repeatability	High	Low	High	High

IV. FAULT INJECTION TOOLS

Hardware Fault Injection Tools:

RIFLE

This tool is developed at University of Coimbra, Portugal .It is a pin-level fault injection system for dependability validation. This system can be adapted to a wide range of target systems and the faults are mainly injected in the processor pins.

FOCUS

It is developed at University of Illinois at Urbana-Champaign [9] .A design automation environment used for analyzing a microprocessor- based jet-engine controller FOCUS uses a hierarchical simulation environment based on SPLICE for tracing the impact of transient faults.

MARS

Maintainable Real-time System tool developed at Technical University of Vienna Austria. Its system is a time-triggered, fault-tolerant, distributed system. It consists of several computer nodes communicating by means of a synchronous time division multiple access strategy.

Software Fault Injection Tools:

Software tolerance/reliability is a major standard for judging safety critical software. Having knowledge of when and where a fault might occur is essential to figuring out the severity and frequency of a fault. Fault injection provides the extra layer of testing coverage need to uphold the standard of software reliability. The tools required for software fault injection must be able to parse a program automatically and insert faults. Some major tools are discussed below.

FERRARI

Fault and Error Automatic Real-Time Injection tool developed at University of Texas at Austin.. It is a run-time fault injection tool that's why it uses time-triggers and program counters. This software inject memory, bus and CPU faults The faults injected can be transient or permanent faults. Ferrari can emulate hardware faults.

DOCTOR

Integrated Software Fault Injection Environment emulates the occurrence of faults in distributed applications. It can inject memory faults, CPU faults and network communication faults. It can use three different triggers: time-out triggers, traps and code insertion.It also collects reliability information.

XCEPTION

It requires no modification of source code or insertion of traps. It uses the features of debugging on modern processors to trigger fault injection. The fault scenarios are reproducible. Xception uses a fault mask while injecting a fault into a target system.

FTAPE

Fault Tolerance and Performance Evaluator tool developed at the University of Illinois. This tool is used to measure the fault tolerance of the systems. It combines the system wide fault injection with the controllable workload.

V. **RELATED STUDY**

Haissam Ziade, RaficAyoubi, and Raoul Velazco[1] Fault tolerant circuits are currently required in several major application sectors. Fault injection provides techniques for assessing the dependability of a system under test. Fault injection includes inserting faults into a system and monitoring the system to determine its behavior in response to a fault. There are several fault injection techniques have been proposed and practically experimented. Mei-ChenHsueh, TimothyK.Tsai and Ravishankar K.Iyer[2] describe Fault injection is important to evaluating the dependability of computer systems under test. Engineers use fault injection to test fault-tolerant systems or components.

Roberto Natella, Joao A. Duraes, S. Madeira[3]The injection of software faults in software components to assess the impact of these faults on other components or on the system as a whole. It allows the evaluation of fault tolerance. In this paper author presents an experimental study to evaluate the representativeness of faults injected by a state-of-the-art approach (G-SWFIT).

Andréas Johansson[4]It is the sad truth that no matter how good a programmer all software systems will contain faults and be exposed to faults from the environment! An important aspect of any software component is how resilient it is to faults. Using software implemented fault injection (SWIFI) for software component evaluation seems very appealing. There are however many questions that needs to be resolved before any real use of such a technique can be gained. Questions like what type of faults that arise in real systems and how to emulate them in software. Is the system suitable for testing with SWIFI? What effectdoes the tool have on the experiment and effect does the workload have?

Fanping Zeng Juan Li Ling Li Xufa Wang,[5] Fault injection devotes an efficient way for verifying fault tolerance of computer and detecting the vulnerability of software system. The author presents a Xen-based fault injection technique to test vulnerability of software to build an efficient and general-purpose software test model. The model injects faults into interactive layer between software applications and their environments. This technique has two contributions: First, to detect the vulnerability of software according to model which requires less number of fault test cases. Second, it enhances the flexibility and the robustness of the fault injection tools

JereyM. Voas & Anup K. Ghosh[6] defines information survivability which means the ability of an information system to operate continuous in the presence of faults, irregular system behavior, or destructive attacks. Software testing ensures that software behaves correctly in an intractable problem for any non-trivial piece of software. In this paper, the authors present off-nominal" testing techniques, which are not bother with the correctness of the software, but with the behavior of the software in the face of irregular events and destructive attack.

Stefan Winter, Michael Tretter, Benjamin Sattler, Neeraj Suri,[7],Software fault injection is a technique to evaluate the robustness of software systems. A large number of software fault injection frameworks exist, all are based on a single-fault assumption. As software systems containing more than a single fault often are the norm than an exception and current safety standards require the consideration of "multi-point faults". To address the issue of simultaneous SWIFI (simFI), authors analyze how independent faults can manifest in a generic software composition model and extend an existing SWIFI tool to support some characteristic simultaneous fault types

VI. CONCLUSION

Fault injection is a testing technique used for the evaluation of design metrics such as reliability, safety and fault coverage of the target system. Fault injection injects fault into the system and monitor the system to determine its behaviour in response to the fault. Fault injection techniques fall into two categories: Hardware based Fault injection, Software–based fault injection. In table 1, summarize the advantage, disadvantage and tools used for these techniques.

Table1. Summary of main advantages disadvantages and tool

Techniques	Advantages	Disadvantages	Tools
Hardware-Based	• Can access locations which are not accessed by other means. • Permanent faults are model at the pin level. • Experiments are performed fast. • Non intrusive. • Suited for the low level fault model.	• Risk of damage for the injected system is high. • Set of injected faults are limited and limited injection points. • To perform fault injection special hardware is required.	RIFLE, FOCUS,MARS
Software-Based	• No need of any special hardware • Operating system and applications are target • Experiments can run real time scenarios. • Low development and low implementation cost. • Add new classes of faults later.	• Modification of the source code is essential for the fault injection. • Permanent faults are most difficult to model. • faults are not injected in location in accessible to software	FERRARI,FTAPE,DOCTOR,XCEPTION

REFERENCES

[1] Haissam Ziade 1, Rafic Ayoubi2, and Raoul Velazco 3 1 Faculty of Engineering I, Lebanese University, Lebanon2 Faculty of Engineering, University of Balamand, Lebanon3 IMAG Institute, TIMA Laboratory, France"A Survey on Fault Injection Techniques"2004.

[2] Mei-ChenHsueh,TimothyK. Tsai,andRavishankar K. IyerUniversity of Illinois atUrbanaChampaign"Fault Injection Techniques and Tools"1997.

[3] Roberto Natella, Member, IEEE, DomenicoCotroneo, Member, IEEE,Joao A. Duraes, and Henrique S. Madeira, Member, IEEE"On Fault Representativeness of Software Fault Injection"2013.

[4] Andréas Johansson Department of Computer Engineering Chalmers University of Technology Gothenburg, Sweden "Software Implemented Fault Injection Used for Software Evaluation"2002.

[5] FanpingZeng Juan Li Ling Li Xufa Wang Department of Computer Science University of Science and Technology of China, "Fault Injection Technology for Software Vulnerability Testing Based on Xen"2009.

[6] Jerey M. Voas & Anup K. GhoshReliable Software Technologies" Software Fault Injection for Survivability"2000

[7] Stefan Winter, Michael Tretter, Benjamin Sattler, Neeraj Suri DEEDS Group, Dept. of CS, TU Darmstadt, Germany "simFI: From Single to Simultaneous Software Fault Injections".IEEE 2013.

[8] Jean Arlat, Member, IEEE, Yves Crouzet, Johan Karlsson, Member, IEEE,PeterFolkesson, Member, IEEE, Emmerich Fuchs, Member, IEEE Computer Society, andGu¨ nther H. Leber, Member, IEEE"Comparison of Physical and Software-Implemented Fault Injection Techniques"2003.

[9] Robert Slater Carnegie Mellon University 18-849b Dependable Embedded Systems"Fault Injection" Spring 1998.

