School of Computing Science & Engineering

Course Code : CSCV4021

Course Name: Service Oriented Architecture

Software Fault Tolerance

Name of the Faculty: Mr. Kalidass S

Programme: B.Tech(Spl)

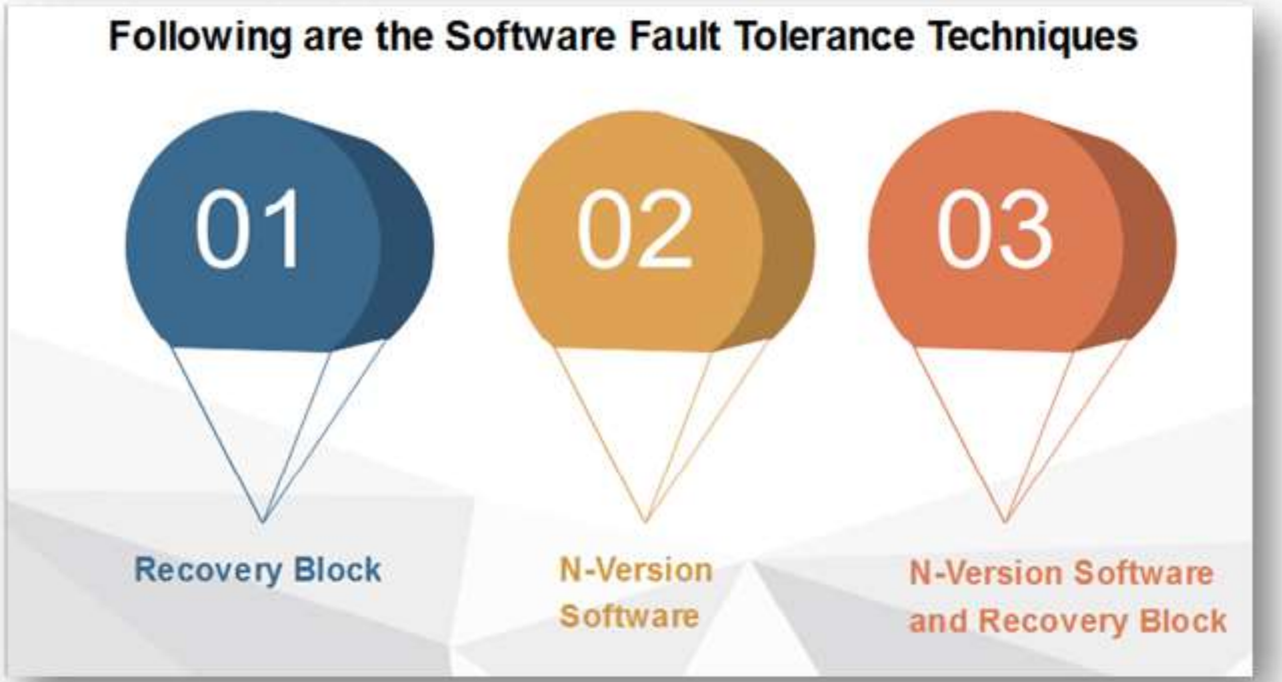
Software Fault Tolerance:

Software fault tolerance is the ability for software to detect and recover from a fault that is happening or has already happened in either the software or hardware in the system in which the software is running to provide service by the specification.

Software fault tolerance is a necessary component to construct the next generation of highly available and reliable computing systems from embedded systems to data warehouse systems.

To adequately understand software fault tolerance it is important to understand the nature of the problem that software fault tolerance is supposed to solve.

Software faults are all design faults. Software manufacturing, the reproduction of software, is considered to be perfect. The source of the problem being solely designed faults is very different than almost any other system in which fault tolerance is the desired property.



Following are the Software Fault Tolerance Techniques

Recovery Block

The **recovery block method** is a simple technique developed by Randel. The recovery block operates with an adjudicator, which confirms the results of various implementations of the same algorithm. In a system with recovery blocks, the system view is broken down into fault recoverable blocks.

The entire system is constructed of these fault-tolerant blocks. Each block contains at least a primary, secondary, and exceptional case code along with an adjudicator. The adjudicator is the component, which determines the correctness of the various blocks to try.

If the adjudicator does not accept the results of any of the alternates, it then invokes the exception handler, which then indicates the fact that the software could not perform the requested operation.

The **recovery block technique** increases the pressure on the specification to be specific enough to create various multiple alternatives that are functionally the same. This problem is further discussed in the context of the **N-version software method**.

N-Version Software:

The **N-version software** methods attempt to parallel the traditional hardware fault tolerance concept of N-way redundant hardware. In an **N-version software system**, every module is done with up to N different methods. Each variant accomplishes the same function, but hopefully in a various way. Each version then submits its answer to voter or decider, which decides the correct answer, and returns that as the result of the module.

This system can hopefully overcome the design faults present in most software by relying upon the design diversity concept. An essential distinction in **N-version software** is the fact that the system could include multiple types of hardware using numerous versions of the software.

N-version software can only be successful and successfully tolerate faults if the required design diversity is met. The dependence on appropriate specifications in N-version software, (and recovery blocks,) cannot be stressed enough.

N-Version Software and Recovery Blocks:

The differences between the **recovery block technique** and the **N-version technique** are not too numerous, but they are essential. In traditional recovery blocks, each alternative would be executed serially until an acceptable solution is found as determined by the adjudicator. The recovery block method has been extended to contain concurrent execution of the various alternatives.

The **N-version techniques** have always been designed to be implemented using N-way hardware concurrently. In a serial retry system, the cost in time of trying multiple methods may be too expensive, especially for a real-time system. Conversely, concurrent systems need the expense of N-way hardware and a communications network to connect them.

The recovery block technique requires that each module build a specific adjudicator; in the N-version method, a single decider may be used. The recovery block technique, assuming that the programmer can create a sufficiently simple adjudicator, will create a system, which is challenging to enter into an incorrect state.