Fault Tolerant Software Development

Presented by:

Hussam Bakdash

Main Topics

- Definition of Fault Tolerance
- Importance of Fault Tolerance
- Source of fault
- Propagation of Fault
- Type of fault
- Application of Fault Tolerance
- Fault Tolerance Technics
- Example of Fault Tolerant Systems

Definition of Fault Tolerance

Fault tolerance is the ability of the system (Automation process, Machine, Computer, Network, cloud, Cluster, …etc.) to continue its operation without interruption in case of the failure of one or more than one of its components.

Objective:

To prevent the disruption arising from a single point of failure.

Importance of Fault Tolerance

Prevent Failure of the system

High Fidelity

High Performance where the system continue its operation in case of component failure.

Maintain the quality of a computer system that gracefully handles the failure of component hardware or software

Continue to operate satisfactorily in the presence of one or more system failure conditions

Example: Ariane 5 Rocket, Bug in the computer program due to 16 bits signed integer variable

Source of Fault

Physical Faults: These kinds of faults are caused by physical problem such as aging or defect in components.

Internal Faults: These kinds of faults are caused by defect in the system itself such as connection cut.

External Faults: These kinds of faults are caused by the environment that affect into the system such as power down.

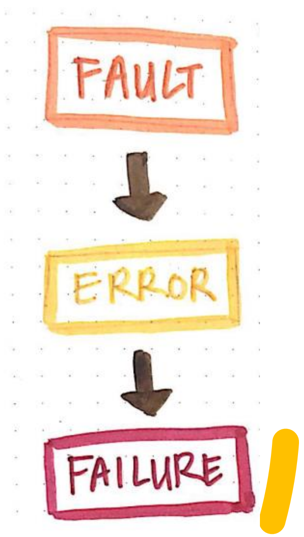
Design Faults: These kinds of faults are caused by to bad or not well studied design.

Human Faults: Human can make mistakes which can take the system into failure or damage (Developer or User).

Propagation of Fault

Fault as a physical defect, imperfection, or flaw that occurs in some hardware or software component

- ***Fault:*** The system continues to work with unwanted or bad performance
- ***Error:*** The system continues to work with incorrect results
- ***Failure:*** The system stops and fails to continue its working



Types of Fault

- ***Permanent Faults:*** once they occur, they do not disappear. They remain active until a corrective action is taken.
- ***Transient Faults:*** appear and remain active for a short period of time and then disappear.
- ***Intermittent Faults:*** they appear, disappear and then reappear. They are transient fault that become active periodically or randomly.

Application of Fault Tolerance

- Transportation (Airplanes, UAVs, Ships, Trains, Cars)
- Factories and Automation Process
- Home Appliance
- Networking and Banking Systems
- E-Government, E-commerce, E-Business
- Computer Systems
- Information Technologies
- Mobile Applications

Fault Tolerance Technics

Redundancy:

- Static: Two or more modules running together at all time with a voting system
- Dynamic: Two or more modules, one module is running and the other module are ready to operate in case of its failure
- Standby: Two or more modules are running together where one of them is active and the other is ready to operate in case of failure
- Duplication: Two or more modules are running together and the result is the average of them. In case of failure of one of them, it will be excluded.
- Memory Architecture: System to organize the memory configuration such as RAID.

Fault Tolerance Technics

Fault Detection

- State estimation: Reference module is used to check the module performance
- Fault estimation: Reference modules with possible fault cases are used to check the module performance.
- Thresholds: Checking the range of output
- Comparison: Way to determine the failed module when use redundancy
- Acceptance Test: Check the validity of the output or system result
- Validation Test: Check the user inputs
- Reversal Check: Backword calculation to check the failure of the system.

Fault Tolerance Technics

Fault Masking

It is the way of insuring that the correct values are passed in case of the presence of a fault.

- Error Correction: Prevent the system to be affected by fault
- Triple Modular Redundancy: Three modules running together with a voting system used to rectify the error in one of them.
- N-Modular Redundancy: more than Three modules running together with a voting system
- Voting System: Smart module which manipulates the results of redundant module and select or calculate the final output
- Neural Network: Commonly used as one of voting system

Fault Tolerance Technics

Fault Recovery

- Exception Handling: Process the case of failure
- Check point and restart: Check the system performance in the critical points and take an action depending on the failure case
- Process pair: Realizing of two hardware in different way
- Data Diversity: Technic used to determine that the input is incorrect depending on the result or the output of the system.

Fault Tolerance Technics

Software Consideration

- Built in self test: It runs automatically before each operation to check the functionality of the system.
- Watchdog timer: Used to monitor the performance and the lock or stuck of the program.
- Exception handling: Commonly used to process abnormal case that may cause during normal operation
- Cyclic Redundancy Check (CRC): Commonly used in communication and transferring data
- Process Pair: Two versions of programs are running as redundancy
- N-Version Programming: More than two version of program are running as redundancy
- Software Diversity: Different versions of program developed by different teams and running on different hardware.

Fault Tolerance Technics

- Fault Containment: process of isolating a fault and preventing the propagation of its effects throughout the system.
- Fault Tree Analysis: method to list the faults that can happened, determine their effects, and the events to reconstruct the system.

Example of Fault Tolerant Systems

- Boeing 777 has three programs developed by three teams built in three processors (AMD29050, Motorola 68040, and Intel 80486) running at same time where the results or actions are compared and gotten using voting program.
- Transactional log files that protect the Microsoft Windows registry and allow recovery of hives
- Uninterruptible power supply (UPS) to protect the system against primary power failure
- RAID (1, 2, 3, 4, 5, 6, 10,53) are good examples of fault tolerance for Data is distributed.
- Load Balancing is a good example of fault tolerant servers routing

THANKS FOR YOUR ATTENDANCE

QUESTIONS ARE WELCOME