Università
della
Svizzera
italiana

**Facoltà
di
scienze
informatiche**

Edge Computing in the IoT

Fault-tolerant Computer Systems

Alberto Ferrante (alberto.ferrante@usi.ch)

Fault → Error → Failure

- A defect, imperfection or flaw that occurs in hardware or software components may cause an error which, in turn, may lead to a failure

Dhiraj K. Pradhan (Ed.). 1996. Fault-Tolerant Computer System Design. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Fault

- A defect, imperfection or flaw that occurs in hardware or software components
  - **Nature**: type of fault
  - **Duration**: length in time
    - Permanent
    - Transient
    - Intermittent
  - **Extent**: localized to a component or not
  - **Value**:
    - Determinate: status remains unchanged through time
    - Indeterminate: different status at different times

Dhiraj K. Pradhan (Ed.). 1996. Fault-Tolerant Computer System Design. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Error

- The manifestation of a fault
  - Deviation from accuracy or correctness

Dhiraj K. Pradhan (Ed.). 1996. Fault-Tolerant Computer System Design. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Failure

- System performs one of its functions incorrectly
  - Non performance of some actions
  - Performance of actions in subnormal quantity or quality

Dhiraj K. Pradhan (Ed.). 1996. Fault-Tolerant Computer System Design. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Terminology

- *Fault latency*: length in time between the occurrence of a fault and the appearance of an error due to that fault
- *Error latency*: length in time between the occurrence of an error and the appearance of the resulting failure

- ***Fault-Tolerant System****: a system that can continue the correct performances of its tasks in the presence of hardware and/or software failures*

Dhiraj K. Pradhan (Ed.). 1996. Fault-Tolerant Computer System Design. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Example

- Let us consider a system that measures the temperature of a fluid every 10 minutes and uses a resistor to heat the fluid when the temperature goes below 40°C
- Let us suppose that the temperature sensor becomes faulty, being stuck at 42°C
  - The fault is
    - Permanent
    - Localized
    - Determinate
  - The fault becomes an error as soon as the temperature is measured (an it is not 42°C in reality)
    - Fault latency: up to 10 minutes
  - The error causes a failure as soon as the real temperature falls below 40°
    - Error latency depends on the system dynamics

Type of Techniques to Deal With Faults

- Fault avoidance
  - Attempt to prevent faults
- Fault Masking
  - Prevents faults from introducing errors into the information structure
- Fault tolerance
  - Ability of the system to continue performing its tasks after the occurrence of faults

Dhiraj K. Pradhan (Ed.). 1996. Fault-Tolerant Computer System Design. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Reconfiguration

- Fault detection: the process of recognizing that a fault has occurred
- Fault location: the process of locating the fault
- Fault containment: the process of isolating the fault
- Fault recovery: the process of remaining operational/regaining operational status

Dhiraj K. Pradhan (Ed.). 1996. Fault-Tolerant Computer System Design. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Redundancy

- Redundancy is used to provide fault tolerance and fault detection for hardware components
- Three types:
  - Hardware redundancy
  - Information redundancy
  - Time redundancy

Dhiraj K. Pradhan (Ed.). 1996. Fault-Tolerant Computer System Design. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

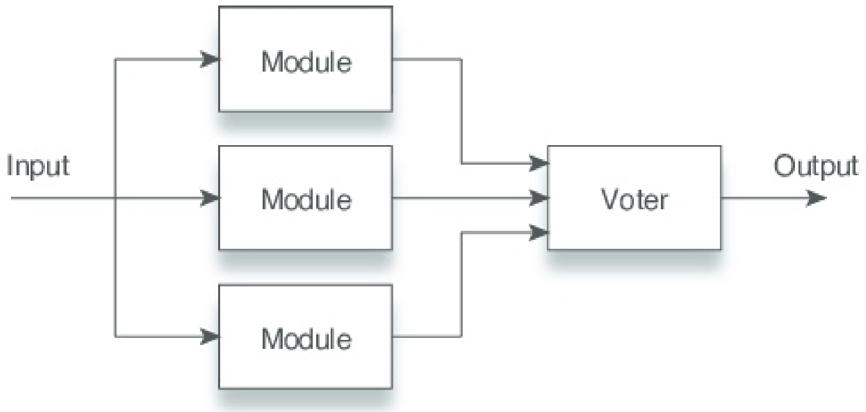
Hardware Redundancy

- Physical replication of components
- Passive redundancy:
  - Achieve fault tolerance without requiring any action from system/operator
  - No detection of faults
  - Fault masking
- Active/Dynamic redundancy
  - Detect faults and perform actions to remove the faulty hardware (i.e., reconfiguration to tolerate faults)
- Hybrid redundancy
  - Combines passive and active approaches

Dhiraj K. Pradhan (Ed.). 1996. Fault-Tolerant Computer System Design. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Passive Hardware Redundancy

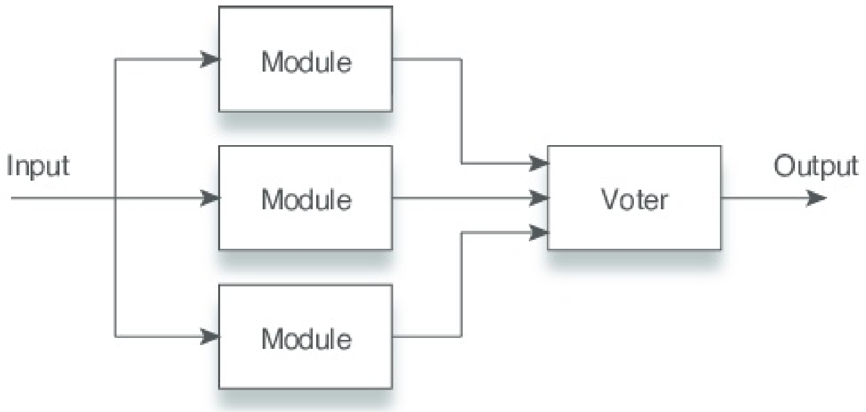
- Multiple copies of a component + voting
- Triple Modular Redundancy (TMR):
  - Triplicate the hardware and perform a majority vote
    - Tolerate failure of one of the three redundant components
    - The voter is a ***single point of failure***



Dhiraj K. Pradhan (Ed.). 1996. Fault-Tolerant Computer System Design. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Passive Hardware Redundancy

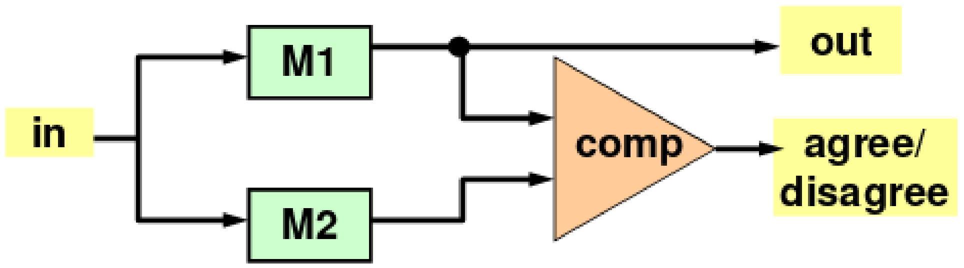
- N-modular Redundancy (NMR):
  - Generalization of TMR
  - N copies of a given module
    - N should be odd
  - Provides the ability to tolerate more faults
    - 5-MR: can tolerate two faulty redundant components



Dhiraj K. Pradhan (Ed.). 1996. Fault-Tolerant Computer System Design. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Active Hardware Redundancy

- Fault detection + fault location + fault recovery
  - In many applications: error detection + error location + error recovery
  - No fault masking: i.e., no attempts to prevent faults from producing errors
    - Common in applications where temporary errors are tolerable



- Duplication with comparison
  - Cannot tolerate faults but only detect them
  - Exact comparison vs. minor differences tolerated, depending on applications

Dhiraj K. Pradhan (Ed.). 1996. Fault-Tolerant Computer System Design. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

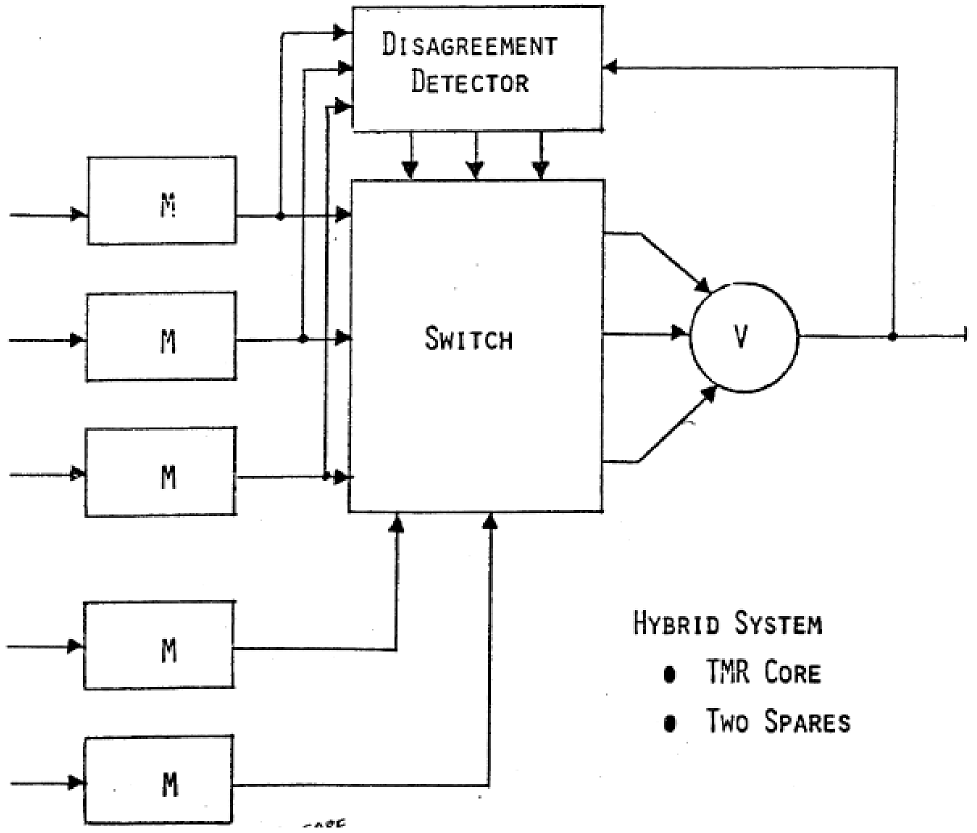
Active Hardware Redundancy

- Standby sparing / standby replacement
  - One module is operational
  - Other modules are used as spares
  - Fault/error detection techniques are used to detect faults
  - Faults are located
  - A spare is used instead of the faulty component

Dhiraj K. Pradhan (Ed.). 1996. Fault-Tolerant Computer System Design. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Hybrid Hardware Redundancy

- N-modular Redundancy with Spares
  - Basic core of N modules arranged in a voting configuration
  - Spares are provided to replace the failing units in the NMR core



Dhiraj K. Pradhan (Ed.). 1996. Fault-Tolerant Computer System Design. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Information Redundancy

- Addition of redundant information to data to allow:
  - Fault detection
  - Fault masking
  - Fault tolerance
- Error *detecting* codes
- Error *correcting* codes

Dhiraj K. Pradhan (Ed.). 1996. Fault-Tolerant Computer System Design. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Information Redundancy – Error Detecting Codes

- Parity codes
  - Add an extra bit to check the parity (number of 1s)
    - Odd parity
    - Even parity
  - Can detect an error in 1 bit of the word
    - If multiple bits are affected, it may not detect the error!
  - No error correction

Data	No. of ones	Even parity	Odd parity
1010001	3	1010001**1**	1010001**0**

Dhiraj K. Pradhan (Ed.). 1996. Fault-Tolerant Computer System Design. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

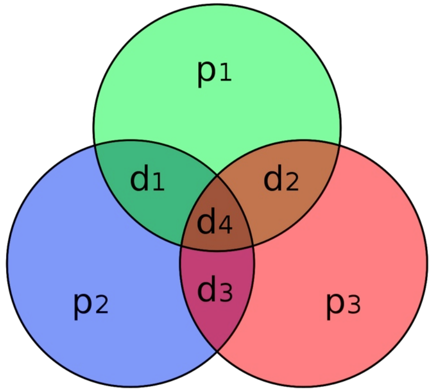
Error Detecting Codes - Checksum

- Checksum
  - A small-sized datum derived from a block of digital data
    - E.g., longitudinal parity check:
      - Breaks the data into "words" with a fixed number n of bits
      - Computes the XOR of all those words
  - When storing/sending a chunk of data, a checksum is created
  - The checksum is calculated again at reception/read and compared with the one computed in the storage/sending phase
  - Depending on the method used to create the checksum, different types and amount of errors can be detected

Dhiraj K. Pradhan (Ed.). 1996. Fault-Tolerant Computer System Design. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Information Redundancy – Error Correcting Codes

- Hamming error-correcting codes
  - Partition the information bits into groups
  - Specify a parity bit for each group
  - Groups are overlapped to be able to correct errors
  - Often used in memories (Error Detection an Correction Unit, EDAC)
  - E.g., Hamming(7,4)
    - Encodes four bits of data into seven bits by adding three parity bits
    - Can correct any single-bit error
    - Can detect all single-bit and two-bit errors



Data	Parity bits (even)	Received data	
1010	101	1110	p1, incorrect; p2, OK; p3, incorrect –> d2 incorrect, can be corrected

Error Detecting Codes - CRC

- Cyclic redundancy check
- Form of checksum
  - A CRC-enabled device calculates a short, fixed-length binary sequence for each block of data to be sent or stored and appends it to the data, forming a codeword.
  - When a codeword is received or read, the device compares its check value with one freshly calculated from the data block
  - If the CRC values do not match, then the block contains a data error.
- Simple to implement in binary hardware
  - Based on modulo-2 divisions

Error Detecting Codes - CRC

- Easy to analyze mathematically
- Particularly good at detecting common errors caused by noise in transmission channels
  - Not suitable for protecting against intentional alteration of data
- Performance
  - Can detect all burst errors that affect an odd number of bits.
  - Can detect all burst errors of length less than or equal to the degree of the polynomials
    - The divisor is represented not as a string of 1s and 0s, but as an algebraic polynomials
  - Can detect with a very high probability burst errors of length greater than the degree of the polynomials

Time Redundancy

- Basic concept: repetition of computations in a way that allows faults to be detected
  - E.g., repeat two or more times and compare results
    - If fault is detected, repeat again an choose the result
- Reduces extra hardware for redundancy at the expenses of time
- If input data has been corrupted by a transient fault, the fault goes undetected
- With small extra hardware can be used to detect permanent faults

Dhiraj K. Pradhan (Ed.). 1996. Fault-Tolerant Computer System Design. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Fault Tolerance in Software

- The ability of computer software to continue its normal operation despite the presence of system or hardware faults
- Software fault tolerance methods are designed to overcome execution errors by modifying variable values to create an acceptable program state

Dhiraj K. Pradhan (Ed.). 1996. Fault-Tolerant Computer System Design. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Software Fault Tolerance

- Recovery blocks
  - The system is broken down into fault recoverable blocks
  - Each block contains at least a primary, secondary, and exceptional case code along with an adjudicator
  - The adjudicator first executes the primary alternate; if it determines that the primary block failed, it tries to roll back the state of the system and tries the secondary alternate.
  - If the adjudicator does not accept the results of any of the alternates, it then invokes the exception handler, which indicates that the software could not perform the requested operation

Dhiraj K. Pradhan (Ed.). 1996. Fault-Tolerant Computer System Design. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Software Fault Tolerance

- N-version software
  - Parallels the traditional hardware fault tolerance concept of N-way redundant hardware
  - Each module is made of up to N different implementations
    - Each variant accomplishes the same task in a different way
  - Each version then submits its answer to a voter which determines the correct answer
- Self-checking software
  - Extra checks, often including checkpointing and rollback recovery methods, added into fault-tolerant or safety critical systems

Dhiraj K. Pradhan (Ed.). 1996. Fault-Tolerant Computer System Design. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Fault-tolerance in Practice

- Provide complete fault tolerance may be expensive
- In many systems, fault detection may be enough
  - Some faults/errors may be difficult to detect and may require specific techniques
  - Some other faults/errors may be detected by means of simple checks
    - E.g., If a speed sensor of a city car detects a speed of 300km/h, something is obviously wrong

Fault-tolerance in Practice

- Once the fault/error is detected, the system may need to go in a **failsafe status**:
  - Stop the system
  - Disable certain functions
  - Notify user/operator
  - E.g., abnormal values are detected on the steering wheel sensor of a car (e.g., angle of 1200° when the wheel maximum rotation is 1000°)
    - The power steering, along with other electronic systems that use the steering wheel angle value, is disabled
    - The driver is notified with a message on the dashboard and an alert sound
    - Restarting the system (switch the engine off and on again) may re-enable the power steering, if the fault was transient