Fault-Tolerant Computing

Basic Concepts and Tools
About This Presentation

This presentation has been prepared for the graduate course ECE 257A (Fault-Tolerant Computing) by Behrooz Parhami, Professor of Electrical and Computer Engineering at University of California, Santa Barbara. The material contained herein can be used freely in classroom teaching or any other educational setting. Unauthorized uses are prohibited. © Behrooz Parhami

<table>
<thead>
<tr>
<th>Edition</th>
<th>Released</th>
<th>Revised</th>
<th>Revised</th>
</tr>
</thead>
</table>
Terminology, Models, and Measures for Dependability
The problem is your modem can’t interface with your ISP because your RJ11 cable needs upgrading.

Will it cost much?

That depends on whether you know I just said “you need a longer phone cord.”

“Every time we successfully recover from a technical problem, the computer likes a high five.”

“That foul smell is coming from your computer. You’ve got some old data in there that’s gone bad.”
Impairments to Dependability

- Flaw
- Fault
- Hazard
- Bug
- ERROR
- Degradation
- Defect
- Failure
- Intrusion
- Malfunction
- Crash
The Fault-Error-Failure Cycle

Schematic diagram of the Newcastle hierarchical model and the impairments within one level.

Includes both components and design

Fault

Replaced with NAND?
The Four-Universe Model

Universe  Impairment
Physical  Failure
         ↓
Logical  Fault
        ↓
Informational  Error
         ↓
External  Crash

Cause-effect diagram for Avižienis’ four-universe model of impairments to dependability.
Unrolling the Fault-Error-Failure Cycle

Cause-effect diagram for an extended six-level view of impairments to dependability.
Multilevel Model

Legend:
- Ideal
- Defective
- Faulty
- Erroneous
- Malfunctioning
- Degraded
- Failed
- Low-Level Impaired
- Mid-Level Impaired
- High-Level Impaired

Component

Logic

Information

System

Service

Result

Legend:
- Entry
- Deviation
- Remedy
- Tolerance
An analogy for our multi-level model of dependable computing. Defects, faults, errors, malfunctions, degradations, and failures are represented by pouring water from above. Valves represent avoidance and tolerance techniques. The goal is to avoid overflow.
Why Our Concern with Dependability?

Reliability of $n$-transistor system, each having failure rate $\lambda$

$$R(t) = e^{-n\lambda t}$$

There are only 3 ways of making systems more reliable

- **Reduce** $\lambda$
- **Reduce** $n$
- **Reduce** $t$

**Alternative:**
Change the reliability formula by introducing redundancy in system
Highly Dependable Computer Systems

Long-life systems: Fail-slow, Rugged, High-reliability
Spacecraft with multiyear missions, systems in inaccessible locations
Methods: Replication (spares), error coding, monitoring, shielding

Safety-critical systems: Fail-safe, Sound, High-integrity
Flight control computers, nuclear-plant shutdown, medical monitoring
Methods: Replication with voting, time redundancy, design diversity

Non-stop systems: Fail-soft, Robust, High-availability
Telephone switching centers, transaction processing, e-commerce
Methods: HW/info redundancy, backup schemes, hot-swap, recovery

Just as performance enhancement techniques gradually migrate from supercomputers to desktops, so too dependability enhancement methods find their way from exotic systems into personal computers.
Aspects of Dependability

- Reliability
  - Reliability, $MTTF = MTFF$
- Security
- Safety
  - Risk, consequence
- Resilience
- Performability
  - Performability, $MCBF$
- Availability
  - Pointwise av., Interval av., $MTBF, MTTR$
- Maintainability
- Integrity
- Robustness
- Testability
  - Controllability, observability
- Serviceability
Concepts from Probability Theory

**Probability density function:** pdf  
\[ f(t) = \text{prob}[t \leq x \leq t + dt] / dt = dF(t) / dt \]

**Cumulative distribution function:** CDF  
\[ F(t) = \text{prob}[x \leq t] = \int_0^t f(x) \, dx \]

**Expected value of** \( x \)  
\[ E_x = \int_{-\infty}^{+\infty} x f(x) \, dx = \sum_k x_k f(x_k) \]

**Variance of** \( x \)  
\[ \sigma_x^2 = \int_{-\infty}^{+\infty} (x - E_x)^2 f(x) \, dx = \sum_k (x_k - E_x)^2 f(x_k) \]

**Covariance of** \( x \) **and** \( y \)  
\[ \psi_{x,y} = E [(x - E_x)(y - E_y)] = E [xy] - E_x E_y \]

Lifetimes of 20 identical systems
Some Simple Probability Distributions

- Uniform
- Exponential
- Normal
- Binomial
Reliability and MTTF

**Reliability:** $R(t)$
Probability that system remains in the “Good” state through the interval [0, $t$]

$$R(t + dt) = R(t) [1 - z(t) dt]$$

Hazard function

$$R(t) = 1 - F(t)$$
CDF of the system lifetime, or its unreliability

Constant hazard function $z(t) = \lambda \Rightarrow R(t) = e^{-\lambda t}$
(system failure rate is independent of its age)

**Mean time to failure:** MTTF
MTTF = $\int_0^{\infty} t f(t) dt = \int_0^{\infty} R(t) dt$
Expected value of lifetime

Area under the reliability curve (easily provable)

Two-state nonrepairable system

Start state Up

Failure

Down

Start state

Up

Failure

Down
Failure Distributions of Interest

**Exponential:** \( z(t) = \lambda \)
\[ R(t) = e^{-\lambda t} \]
\[ MTTF = \frac{1}{\lambda} \]

**Rayleigh:** \( z(t) = 2\lambda(\lambda t) \)
\[ R(t) = e^{(-\lambda t)^2} \]
\[ MTTF = (\frac{1}{\lambda}) \frac{\sqrt{\pi}}{2} \]

**Weibull:** \( z(t) = \alpha \lambda (\lambda t)^{\alpha-1} \)
\[ R(t) = e^{(-\lambda t)^\alpha} \]
\[ MTTF = (\frac{1}{\lambda}) \Gamma(1 + 1/\alpha) \]

**Erlang:**
\[ MTTF = k/\lambda \]

**Gamma:**
Erlang and exponential are special cases

**Normal:**
Reliability and MTTF formulas are complicated

**Discrete versions**

**Geometric**
\[ R(k) = q^k \]

**Discrete Weibull**

**Binomial**

Oct. 2007  Terminology, Models, and Measures  Slide 17
Comparing Reliabilities

**Reliability difference:** $R_2 - R_1$

**Reliability gain:** $R_2 / R_1$

**Reliability improvement factor**
$\text{RIF}_{2/1} = [1 - R_1(t_M)] / [1 - R_2(t_M)]$

Example:
$[1 - 0.9] / [1 - 0.99] = 10$

**Reliability improv. index**
$\text{RII} = \log R_1(t_M) / \log R_2(t_M)$

**Mission time extension**
$\text{MTE}_{2/1}(r_G) = T_2(r_G) - T_1(r_G)$

**Mission time improv. factor:**
$\text{MTIF}_{2/1}(r_G) = T_2(r_G) / T_1(r_G)$
Availability, MTTR, and MTBF

**Interval Availability:** $A(t)$
Fraction of time that system is in the “Up” state during the interval $[0, t]$

**Steady-state availability:** $A = \lim_{t \to \infty} A(t)$

**Pointwise availability:** $a(t)$
Probability that system available at time $t$

$A(t) = (1/t) \int_0^t a(x) \, dx$

Availability = Reliability, when there is no repair

Availability is a function not only of how rarely a system fails (reliability) but also of how quickly it can be repaired (time to repair)

$A = \frac{MTTF}{MTTF + MTTR} = \frac{MTTF}{MTBF} = \frac{\mu}{\lambda + \mu}$

In general, $\mu >> \lambda$, leading to $A \approx 1$

Two-state repairable system

MTTF $\mu$
MTTR $\lambda$
MTBF $1/\mu = MTTR$

Repair rate

(Will justify this equation later)
System Up and Down Times

Short repair time implies good **maintainability** (serviceability)

Time to first failure  Time between failures  Repair time

Up  Down

Start state  Repair  Down

Failure

Time

Oct. 2007  Terminology, Models, and Measures  Slide 20
Performability and MCBF

**Performability: $P$**
Composite measure, incorporating both performance and reliability

**Simple example**
Worth of “Up2” twice that of “Up1”
$p_{Upi}$ = probability system is in state $Upi$

$P = 2p_{Up2} + p_{Up1}$

$p_{Up2} = 0.92, p_{Up1} = 0.06, p_{Down} = 0.02, P = 1.90$
(system performance equiv. To that of 1.9 processors on average)

Performability improvement factor of this system (akin to RIF) relative to a fail-hard system that goes down when either processor fails:
$PIF = (2 – 2 \times 0.92) / (2 – 1.90) = 1.6$

**Question:**
What is system availability here?
System Up, Partially Up, and Down Times

Important to prevent direct transitions to the “Down” state (coverage)

Time

- Up 2
- Partial failure
- Repair
- Partial repair

- Up 1
- Failure

- Down

Start state

Partially Up

- MCBF

Total Failure

- Partial Repair

Up

Time

0 t_1 t_2 t'_2 t'_1 t_3 t'_3 t
Integrity and Safety

Risk: Prob. of being in “Unsafe Failed” state
There may be multiple unsafe states, each with a different consequence (cost)

Simple analysis
Lump “Safe Failed” state with “Good” state; proceed as in reliability analysis

More detailed analysis
Even though “Safe Failed” state is more desirable than “Unsafe Failed”, it is still not as desirable as the “Good” state; so keeping it separate makes sense

For example, if a repair transition is introduced between “Safe Failed” and “Good” states, we can tackle questions such as the expected outage of the system in safe mode, and thus its availability