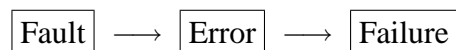


2 Dependability

2.1 Terminology

Systems. A specification describes the ideal behavior of a *system* at its *interfaces*. A system should follow its specification as closely as possible, even in faults occur, for being called *dependable*.

The following terms are causally related:



Fault: (hypothetic) cause of an error

Error: internal system state that does not correspond to specification, not visible at interfaces.

Failure: Deviation of system from specification at interfaces.

Recursive!

Example 1. Fan in power supply congested by dust, fan is slowed down, airflow massively reduced; power supply overheats, a part burns out and supply stops working.

Fan system: dust = fault, slowdown = error, reduced airflow = failure.

Power supply system: fan failure = fault, overheating = error, deliver failure.

Example 2. RAID storage system (bits with ECC, disks with RAID mirroring).

2.2 Attributes

- availability — readiness for correct service
- reliability — continuity of correct service
- safety — absence of catastrophic failures
- confidentiality — no unauthorized disclosure
- integrity — no improper states or state changes

2.3 Techniques

Prevention. formal design, quality control, access control.

Tolerance.

- error & fault detection
 - failure detectors (ping)
 - intrusion detection sensors & systems
- recovery
 - isolation
 - rollback, reboot
 - compensation, fail-over
 - ACID, database transactions
 - masking, redundancy (voting, ECC, RAID; N-version programming; redundant hardware)
- removal
 - formal verification of implementation w.r.t. specification
 - validation of specification w.r.t. real environment
 - fault injection and testing
- forecasting
 - evaluation & prediction
 - fault trees, attack trees

Commercial fault-tolerant systems employ a combination of the above techniques, but focus on sophisticated designs for fault-tolerance through hardware redundancy, combined with isolation and recovery methods for software [BS04]. Examples include the HP NonStop Architecture (formerly built by Tandem Corp.) [BBV⁺05] and IBM's zSeries mainframe servers and z/OS operating system (successors of S/390 servers and OS/390) [Hof97].

2.4 Measures

Assumptions are that all rates are constant, independent of time and that failures occur independently of each other.

Define the *failure rate* to be λ ; then the mean time to failure, $MTTF = 1/\lambda$. Similarly, define the *repair rate* to be μ ; then the mean time to repair, $MTTR = 1/\mu$.

Now we have

$$Availability = \frac{MTTF}{MTTF + MTTR}.$$

With a high MTTF, availability can be further increased by reducing the MTTR — this is the credo of the Berkeley/Stanford Recovery-Oriented Computing (ROC) Project, see <http://roc.cs.berkeley.edu/>.

For a system S_{ys} representing a combination of components C_1, \dots, C_n (without redundancy), we have

$$\lambda_{S_{ys}} = \sum_i \lambda_{C_i}.$$

Hence,

$$MTTF_{S_{ys}} = \frac{1}{\sum_i \frac{1}{\lambda_{MTTF_i}}}.$$

For a system S_{ys} in which components C_1, \dots, C_n are redundant, we have

$$\lambda_{S_{ys}} = \prod_i \lambda_{C_i}.$$

Problems of MTTF & MTTR. There are 8760h per year. Does a system with MTTF of 500'000h on average run for 57 years without failures, even though its manufacturer specifies a system lifetime of 5 years? No. MTTF and MTTR are only statistical measures that are relevant in a large population of samples, i.e., if you have 100 systems expect a faulty one every 0.57 years. Note the fundamental assumption that failures are independent.

References

- [ALR00] A. Avizienis, J.-C. Laprie, and B. Randell, *Fundamental concepts of dependability*, UCLA CSD Report no. 010028, LAAS Report no. 01-145, Newcastle University Report no. CS-TR-739, 2000.
- [BBV⁺05] D. Bernick, B. Bruckert, P. D. Vigna, D. Garcia, R. Jardine, J. Klecka, and J. Smullen, *NonStop[®] Advanced Architecture*, Proc. International Conference on Dependable Systems and Networks (DSN-2005), June 2005, To appear.
- [BS04] W. Bartlett and L. Spainhower, *Commercial fault tolerance: A tale of two systems*, IEEE Transactions on Dependable and Secure Computing **1** (2004), no. 1, 87–96.
- [Gra85] J. Gray, *Why do computers stop and what can be done about it?*, Tech. Report 85.7, Tandem Corp., June 1985, Available from <http://research.microsoft.com/~gray/>.
- [Hof97] G. F. Hoffnagle, *Preface to the special issue on S/390 Parallel Sysplex Cluster*, IBM Systems Journal **36** (1997), no. 2, 170–371, On-line at <http://www.research.ibm.com/journal/sj36-2.html>.
- [HP02] J. L. Hennessy and D. A. Patterson, *Computer architecture: A quantitative approach*, 3rd ed., Morgan Kaufmann, 2002.
- [Pat02] D. A. Patterson, *An introduction to dependability*, ;login: — The Magazine of the USENIX Association **27** (2002), no. 4, 61–65.