IAF0530 (MSc)
IAF9530 (PhD)

Süsteemide usaldusväärsus ja veakindlus
Dependability and fault tolerance

Lecture 2

Gert Jervan
Department of Computer Engineering (ATI)
Tallinn University of Technology (TTÜ)

Case Studies

- Topic categories:
  - Accident analysis
  - **System safety analysis**
  - Literature survey
  - **Something else (implementation, tool study, etc.)**
  - Requires prior ack.

Literature and sample (!) topics on the webpage

www.pld.ttu.ee/IAF0530

© Gert Jervan
2

Case Studies

- Topic selection:
  - March1 (via e-mail)
- Draft of the report (incl. introductory presentation of the topic):
  - April 4
- Presentations: starting from May 2 (preliminary)
- If in doubt – ASK!!

© Gert Jervan
3

Faults, Errors & Failures

- Fault: a defect within the system or a situation that can lead to the failure
- Error: manifestation of the fault – an unexpected behavior
- Failure: system not performing its intended function

Fault → Error → Failure

© Gert Jervan
4

Measuring

- Failures are measured in FITs
  - 1 FIT (failures in time), is the number of failures in 1 billion device-operation hours. A measurement of 1000 FITs corresponds to a MTTF (mean time to failure) of approximately 114 years.
- Example: Bit flips in hardware due to cosmic radiation
  - A person on an airplane over the Atlantic at 35,000 ft working on a laptop with 256 Mbytes (2 Gbits) of memory. At this altitude, the soft error rate (SER) of 600 FITs per megabit becomes 100,000 FITs per megabit, resulting in a potential error every five hours.

© Gert Jervan
5

Fault Examples

- Year 2000 bug
- Loose wire
- Aircraft retracting its landing gear while on ground

- Effects in time:
  - Permanent
  - Transient
  - Intermittent

6

Permanent

- A permanent fault or failure is one which is stable and continuous.
- Permanent hardware failures require some component to be replaced or repaired.
- An example of a permanent fault would be a VLSI chip with a manufacturing defect, causing one input pin to be stuck high (stuck-at-1).

© Gert Jervan 7

Transient

- A transient fault is one which results from a temporary environmental condition.
- For example, a voltage spike might cause a sensor to report an incorrect value for a few milliseconds before reporting correctly.

© Gert Jervan 8

Transient faults

- Happen for a short time
- **Corruptions of data, miscalculation in logic**
- Do not cause a permanent damage of circuits
- Causes are outside system boundaries

Electromagnetic interference (EMI)

Radiation

Lightning storms

9

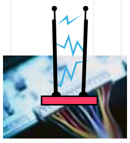
Intermittent

- An intermittent fault is one which only manifests occasionally, due to unstable hardware or certain system states.
- A loose contact on a connector will often cause an intermittent fault.
- Intermittent electrical faults, as a rule, are notoriously difficult to detect. Typically, whenever the fault doctor shows up, the system works fine.

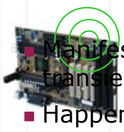
© Gert Jervan 10

Intermittent faults

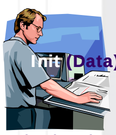
- Manifest similar as transient faults
- Happen repeatedly
- Causes are inside system boundaries



Crosstalk



Internal EMI

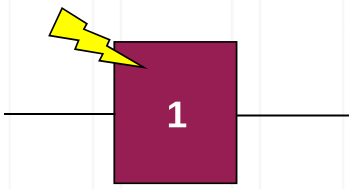


Software errors (Heisenbugs)

Power supply fluctuations

11

Soft Errors



- Transient bit-flip (soft memory error)
  - Random event
  - Corrupts the value but not the cell
  - Can be corrected (in contrast to hard errors caused by faults in the hardware itself)
  - Happen continuously during system lifetime (i.e., can not be screened by burn-in tests)

© Gert Jervan 12

Sources

- First traced to alpha particlce emissions from chip packaging materials
  - Most sources removed (pure materials, different designs, shielding)
- Today's main problem: cosmic radiation
  - Cosmic particles from deep space (actually 5th- or 6th-hand collision particles)
    - At ground level ca 95% neutrons, 5% protons
  - Radioactive material in manufacturing process

© Gert Jervan

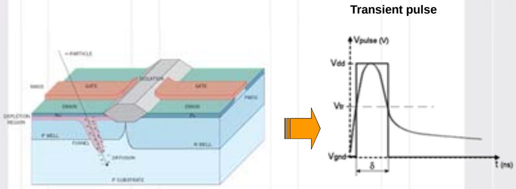
Sources (cont.)

- Four main sources:
  - Low-energy alpha particles
  - High-energy cosmic particles
  - Thermal neutrons
  - Poor system design

SER type	Source	Mechanism	Trend
Alpha	Thorium and uranium contamination in-mold compound, silicon, or lead bumps	2- to 9-MeV alpha particle creating electron-hole funnel traveling 25 microns in silicon	Exponential increase with scaling
Cosmic	Intergalactic sources modulated by solar flares	High-energy neutrons/protons (10 MeV to 1 GeV) colliding with silicon nuclei	Decrease in failures in time per megabit
Thermal neutron	Boron present in BPSG25-meV neutrons	Collision with B10 in BPSG	Highest, always dominates if present

© Gert Jervan

Soft Errors



The electric field in the depletion region directly generates electron-hole pairs in its wake, causing the charges to drift so that the transistor sees a current disturbance

© Gert Jervan

Evidence of Cosmic Ray Strikes

- Documented strikes in large servers found in error logs
  - Normand, "Single Event Upset at Ground Level," IEEE Transactions on Nuclear Science, Vol. 43, No. 6, December 1996.
- Sun Microsystems, 2000 (R. Baumann, Workshop talk)
  - Cosmic ray strikes on L2 cache with defective error protection
    - caused Sun's flagship servers to suddenly and mysteriously crash!
  - Companies affected
    - Baby Bell (Atlanta), America Online, Ebay, & dozens of other corporations
    - Verisign moved to IBM Unix servers (for the most part)
- 2005 – Los Alamos 2048-CPU HP server system crashed frequently due to defective cache
- 2010 Toyota brake problem (still not clear)

© Gert Jervan

Current Situation

- Soft errors induced the highest failure rate of all other reliability mechanisms combined

Rober Baumann, TI

© Gert Jervan

Measuring

- The rate at which SEUs (single-event-upsets) occure is given as SER, measured in FITs (failures in time)
- 1 FIT = 1 failure in 1 billion device-operation hours
- 1000 FIT ≈ MTTF 114 years

© Gert Jervan

Failure Classification

- Domain/Nature
  - Value failure
  - Timing failure
- Perception
  - Consistent failure
  - Inconsistent failure
- Effect
  - Benign failure
  - Malign/catastrophic failure
- Frequency
  - Single failure
  - Repeated failure

19

Failures

- **Crash** Failure: After an error has been detected, the component stops silently.
- **Omission** Failure: Sometimes a result is missing; when result is available, it is correct.
- **Consistent** Failure: If there are multiple receivers, all see the same erroneous result.
- **Byzantine** (Malicious, Asymmetric) Failure: Different receivers see differing results.

© Gert Jervan

20

Failures (cont.)

- **Timing** Failure: A server's response lies outside the specified time interval.
- **Response** Failure: The server's response is incorrect (value of the response is wrong, server deviates from the correct flow of control).
- **Arbitrary** Failure: A server may produce arbitrary responses at arbitrary times.

© Gert Jervan

21

Fault Handling

- Fault avoidance: eliminate problem sources
  - Remove defects: Testing and debugging
  - Robust design: reduce probability of defects
  - Minimize environmental stress: Radiation shielding etc

Impossible to avoid faults completely

- Fault tolerance: add redundancy to mask effect
  - Additional resources needed (more later)
  - Examples:
    - Error correction coding, voting and masking, checksums, ...
    - Backup storage, replication, ...
    - Spare tire, etc

© Gert Jervan

22

Fault Tolerance

- **Fault detection** is the process of recognizing that a fault has occurred. Fault detection is often required before any recovery procedure can be initiated. The techniques include error detection codes, self-checking/failsafe logic, watchdog timers, and others.
- **Fault location** is the process of determining where a fault has occurred so that an appropriate recovery can be initiated.

© Gert Jervan

23

Fault Tolerance (cont.)

- **Fault containment** is the process of isolating a fault and preventing the effects of that fault from propagating throughout the system.
- **Fault recovery** is the process of remaining operational or regaining operational status via reconfiguration even in the presence of faults. A few basic approaches are fault masking, retry, and rollback.

© Gert Jervan

24

Definitions

- Failure rate (λ):
  - Average frequency with which something fails.

$$\frac{6\ failures}{7502\ hrs} = 0.0007998\ failures/hr = 799.8 \times 10^{-6}\ failures/hr$$

- Mean time to failure (MTTF):
  - Average time between failures

$$MTTF = \frac{1}{\lambda}$$

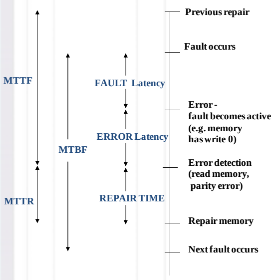
© Gert Jervan 25

Dependability

- Property of a computing system which allows reliance to be justifiably placed on the service it delivers
- Dependability = reliability + availability + safety + security + ...
- Reliability → continuity of correct service
- Availability → readiness of usage
- Safety → no catastrophic consequences
- Security → prevention of unauthorized access

© Gert Jervan 26

Dependability Concepts



Reliability:
a measure of the continuous delivery of service;
R(t) is the probability that the system survives (does not fail) throughout [0, t];
expected value: *MTTF(Mean Time To Failure)*

Maintainability:
a measure of the service interruption
M(t) is the probability that the system will be repaired within a time less than t;
expected value: *MTTR (Mean Time To Repair)*

Availability:
a measure of the service delivery with respect to the alternation of the delivery and interruptions
A(t) is the probability that the system delivers a proper (conforming to specification)service at a given time t.
expected value: *EA = MTTF / (MTTF + MTTR)*

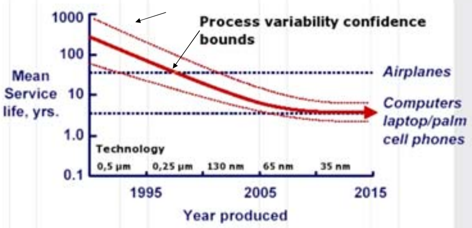
Safety:
a measure of the time to catastrophic failure
S(t) is the probability that no catastrophic failures occur during [0, t];
expected value:
MTTCF(Mean Time To Catastrophic Failure)

Reliability

- A measure of an it performing its intended function satisfactorily for a prescribed time and under given environment conditions.
- Probability that system will survive to time t
  - In aerospace industry the requirement is that failure probability is 10-9 (one failure over 109 hours (114 000 years) of operation)
- Time To Failure (TTF)
- Mean Time To Failure (MTTF)

© Gert Jervan 28

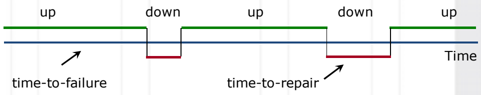
Commercial Chip Reliability Estimation



*Extrapolated from ITRS roadmap, MRQW-2002, Bernstein

29

Availability



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

- Availability:
  - Probability that system is operational at time *t*
- High availability:
  - MTTF → infinity (high reliability)
  - MTTR → zero (fast recovery)

© Gert Jervan 30

Maintainability

- *M(t)* is the probability that a failed system will be restored within a specified period of time *t*.
- Restoration process:
  - locating problem, e.g. via diagnostics
  - physically repairing system
  - bringing system back to its operational condition

© Gert Jervan 31

Graceful Degradation

- The ability of system to automatically decrease its level of performance to compensate for hardware failure and software errors.

© Gert Jervan 32

The Myth of the Nines

Nines	Availability	Downtime per year	Downtime per week	Example
2 nines	99%	3.65 days	1.7 hours	General web site
3 nines	99.9%	8.75 hours	10.1 min	E-commerce site
4 nines	99.99%	52.5 min	1.0 min	Enterprise mail server
5 nines	99.999%	5.25 min	6.0 s	Telephone system
6 nines	99.9999%	31.5 s	0.6 s	Carrier-grade network switch

© Gert Jervan 33

Historical Evaluation

- Mean Time Between Failures:

$$MTBF = MTTR + MTTF$$

  - ENIAC. MTBF: 7 minutes (18000 vacum tubes)
    - ENIAC → TX-2 interactive computer (MIT) → web
  - F-8 Crusader – first fly-by-wire, 375 hours → 750 hours (IBM AP-101)
    - MD-11
    - A320 family
  - Patriot missile defence system
    - 1/3 sec in 100 hours, targeting error: 600 m
    - Needed reboot after 8 hours, was learned in hard way...

© Gert Jervan 34

Ultra-Reliable Systems

- Airbus A320 family fly-by-wire system (1993):
  - computer controls all actuators
  - no control rods, cables in the middle
  - 7 central flight control computers
    - 3 Motorola 68000
    - 2 Intel 80C86
    - 2 Intel 80C286
  - software for hardware written by different software houses (C, ASM, dedicated one, specifically developed)
  - all error checking & debugging performed separately
  - computer allows pilot to fly craft up to certain limits (flight envelope)
    - beyond: computer takes over

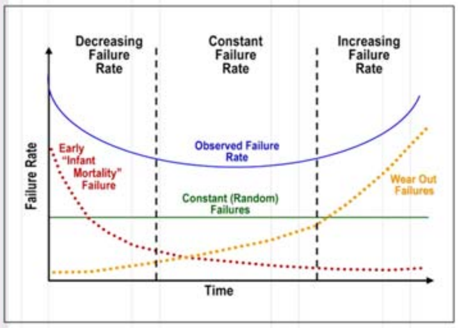
© Gert Jervan 35

Hardware and Environment Failures

- Moving parts, high speed, low tolerance, high complexity: disks, tape drives/libraries
- Lowest MTBF found in fans and power supplies
- Often fans fail gradually → subtle, sporadic failures in CPU, memory, backplane
- Environment: power, cooling, dehumidifying, cables, fire, collapsing racks, ventilation, earthquakes, ...

© Gert Jervan 36

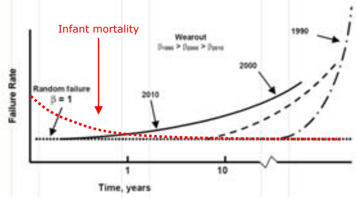
Bathtub Curve



37

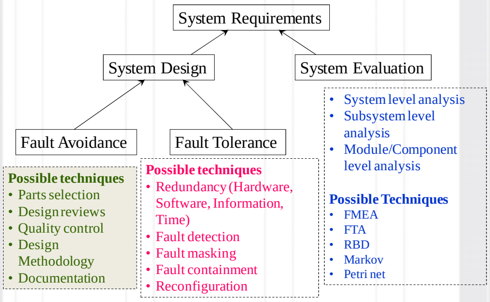
Device Reliability Trends

- As technology progresses, wearout failures become statistically indistinguishable from infant mortality failures with the same wearout drivers.



© Gert Jervan

System Design & Evaluation Top-Level View



39

Safety

- Attribute of a system which either operates correctly or fails in a safe manner
- Freedom from expose to danger, or exemption from hurt, injury or loss.
- "Fail-safe": traffic lights start to blink yellow
- Degrees of safety
- Closely related to risk

© Gert Jervan

40

Risk

- A combination of the likelihood af an accident and the severity of the potential consequences
- The harm that can result if a threat is actualised

- Acceptable/tolerable risk: The Ford Pinto case (1968)

  BENEFITS
  Savings: 180 burn deaths, 180 serious burn injuries, 2,100 burned vehicles.
  Unit Cost: $200,000 per death, $67,000 per injury, $700 per vehicle.
  Total Benefit: 180 X ($200,000) + 180 X ($67,000) + $2,100 X ($700) = $49.5 million.

  COSTS
  Sales: 11 million cars, 1.5 million light trucks.
  Unit Cost: $11 per car, $11 per truck.
  Total Cost: 11,000,000 X ($11) + 1,500,000 X ($11) = $137 million.

© Gert Jervan

41

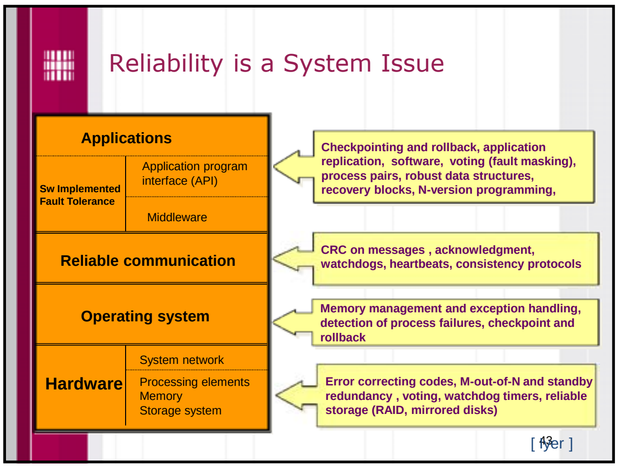
System Safety & Hazards

- Safety:
  - achieved by anticipating accidents and eliminating their causes

- Hazards are potential causes of accidents
  - Conditions in a system which together with other factors in the environment inevitably cause accidents

© Gert Jervan

42


Reliability is a System Issue
Applications
Sw Implemented Fault Tolerance
Application program interface (API)
Middleware
Reliable communication
Operating system
Hardware
System network
Processing elements
Memory
Storage system
Checkpointing and rollback, application replication, software, voting (fault masking), process pairs, robust data structures, recovery blocks, N-version programming,
CRC on messages , acknowledgment, watchdogs, heartbeats, consistency protocols
Memory management and exception handling, detection of process failures, checkpoint and rollback
Error correcting codes, M-out-of-N and standby redundancy , voting, watchdog timers, reliable storage (RAID, mirrored disks)
[fyer]
43