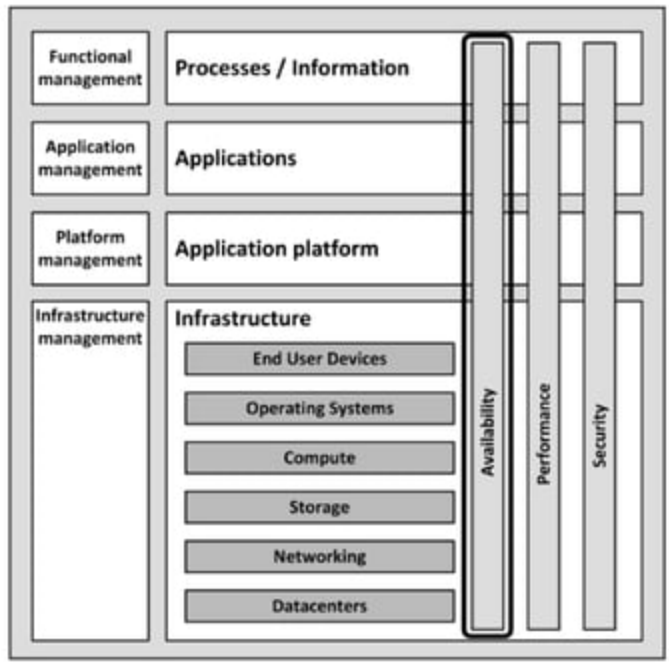
IT Infrastructure Architecture

Infrastructure Building Blocks
and Concepts

Availability Concepts
(chapter 4)

Introduction

- Everyone expects their infrastructure to be available all the time
- A 100% guaranteed availability of an infrastructure is impossible



Calculating availability

- Availability can neither be calculated, nor guaranteed upfront
  - It can only be reported on afterwards, when a system has run for some years
- Over the years, much knowledge and experience is gained on how to design high available systems
  - Failover
  - Redundancy
  - Structured programming
  - Avoiding Single Points of Failures (SPOFs)
  - Implementing systems management

Calculating availability

- The availability of a system is usually expressed as a percentage of uptime in a given time period
  - Usually one year or one month

- Example for downtime expressed as a percentage per year:

Availability %	Downtime per year	Downtime per month	Downtime per week
99.8%	17.5 hours	86.2 minutes	20.2 minutes
99.9% ("three nines")	8.8 hours	43.2 minutes	10.1 minutes
99.99% ("four nines")	52.6 minutes	4.3 minutes	1.0 minutes
99.999% ("five nines")	5.3 minutes	25.9 seconds	6.1 seconds

Calculating availability

- Typical requirements used in service level agreements today are 99.8% or 99.9% availability per month for a full IT system
- The availability of the infrastructure must be much higher
  - Typically in the range of 99.99% or higher
- 99.999% uptime is also known as carrier grade availability
  - For one component
  - Higher availability levels for a complete system are very uncommon, as they are almost impossible to reach

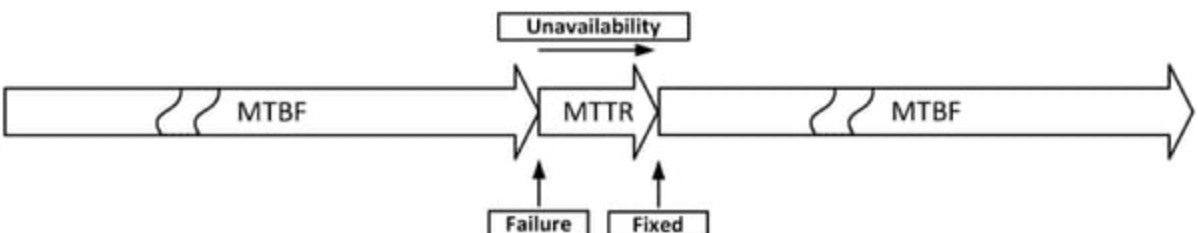
Calculating availability

- It is good practice to agree on the maximum frequency of unavailability

Unavailability (minutes)	**Number of events (per year)**
0 – 5	<= 35
5 – 10	<= 10
10 – 20	<= 5
20 – 30	<=2
> 30	<= 1

MTBF and MTTR

- Mean Time Between Failures (MTBF)
  - The average time that passes between failures
- Mean Time To Repair (MTTR)
  - The time it takes to recover from a failure



MTBF and MTTR

- Some components have higher MTBF than others

- Some typical MTB's:

Component	MTBF (hours)
Hard disk	750,000
Power supply	100,000
Fan	100,000
Ethernet Network Switch	350,000
RAM	1,000,000

MTTR

- MTTR can be kept low by:
  - Having a service contract with the supplier
  - Having spare parts on-site
  - Automated redundancy and failover

MTTR

- Steps to complete repairs:
  - Notification of the fault (time before seeing an alarm message)
  - Processing the alarm
  - Finding the root cause of the error
  - Looking up repair information
  - Getting spare components from storage
  - Having technician come to the datacenter with the spare component
  - Physically repairing the fault
  - Restarting and testing the component

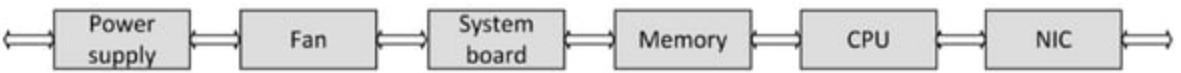
Calculation examples

$$\text{Availability} = \frac{\text{MTBF}}{(\text{MTBF} + \text{MTTR})} \times 100\%$$

Component	MTBF (h)	MTTR (h)	Availability	in %
Power supply	100,000	8	0.9999200	99.99200
Fan	100,000	8	0.9999200	99.99200
System board	300,000	8	0.9999733	99.99733
Memory	1,000,000	8	0,9999920	99.99920
CPU	500,000	8	0.9999840	99.99840
Network Interface Controller (NIC)	250,000	8	0.9999680	99.99680

Calculation examples

- Serial components: One defect leads to downtime



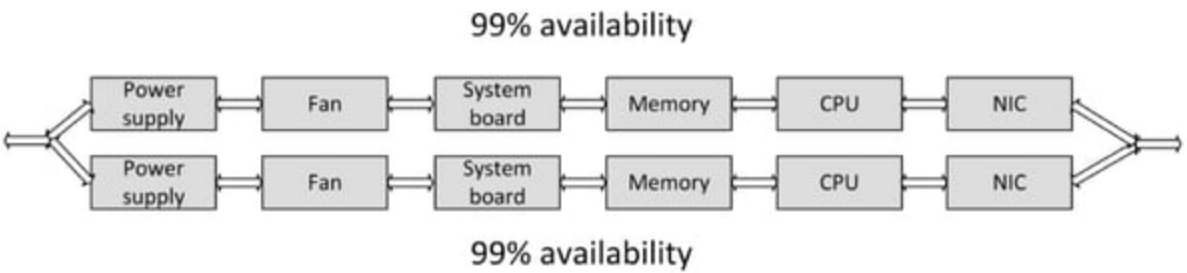
- Example: the above system's availability is:

$$0.9999200 \times 0.9999200 \times 0.9999733 \times 0.9999920 \times 0.9999840 \times 0.9999680 = 0.99977 = \mathbf{99.977}\%$$

(each components' availability is at least 99.99%)

Calculation examples

- Parallel components: One defect: no downtime!
- But beware of SPOFs!



- Calculate availability:

$$A = 1 - (1 - A_1)^n$$

- Total availability $= 1 - (1 - 0.99)^2$ = 99.99%

Sources of unavailability - human errors

- 80% of outages impacting mission-critical services is caused by people and process issues
- Examples:
  - Performing a test in the production environment
  - Switching off the wrong component for repair
  - Swapping a good working disk in a RAID set instead of the defective one
  - Restoring the wrong backup tape to production
  - Accidentally removing files
    - Mail folders, configuration files
  - Accidentally removing database entries
    - Drop table x instead of drop table y

Sources of unavailability - software bugs

- Because of the complexity of most software it is nearly impossible (and very costly) to create bug-free software
- Application software bugs can stop an entire system
- Operating systems are software too
  - Operating systems containing bugs can lead to corrupted file systems, network failures, or other sources of unavailability

Sources of unavailability - planned maintenance

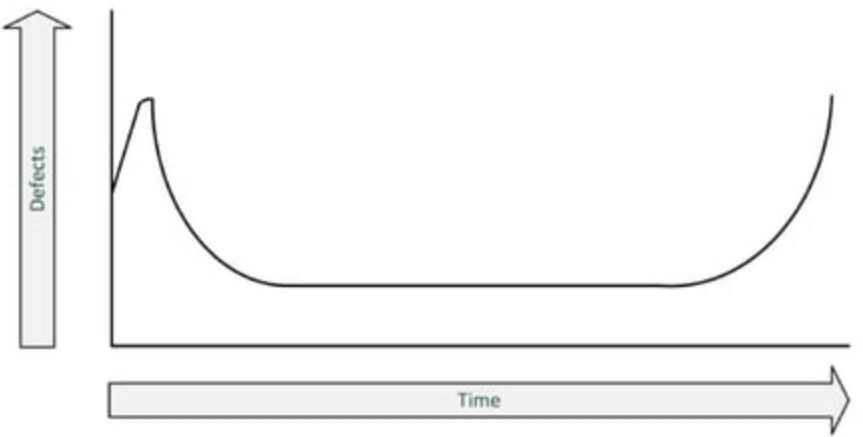
- Sometimes needed to perform systems management tasks:
  - Upgrading hardware or software
  - Implementing software changes
  - Migrating data
  - Creation of backups
- Should only be performed on parts of the infrastructure where other parts keep serving clients
- During planned maintenance the system is more vulnerable to downtime than under normal circumstances
  - A temporary SPOF could be introduced
  - Systems managers could make mistakes

Sources of unavailability - physical defects

- Everything breaks down eventually
- Mechanical parts are most likely to break first
- Examples:
  - **Fans for cooling equipment** usually break because of dust in the bearings
  - **Disk drives** contain moving parts
  - **Tapes** are very vulnerable to defects as the tape is spun on and off the reels all the time
  - **Tape drives** contain very sensitive pieces of mechanics that can break easily

Sources of unavailability - bathtub curve

- A component failure is most likely when the component is new
- When a component still works after the first month, it is likely that it will continue working without failure until the end of its life



Sources of unavailability - environmental issues

- Environmental issues can cause downtime:
  - Failing facilities
    - Power
    - Cooling
  - Disasters
    - Fire
    - Earthquakes
    - Flooding

Sources of unavailability - complexity of the infrastructure

- Adding more components to an overall system design can undermine high availability
  - Even if the extra components are implemented to achieve high availability
- Complex systems
  - Have more potential points of failure
  - Are more difficult to implement correctly
  - Are harder to manage
- Sometimes it is better to just have an extra spare system in the closet than to use complex redundant systems

Redundancy

- Redundancy is the duplication of critical components in a single system, to avoid a single point of failure (SPOF)
- Examples:
  - A single component having two power supplies; if one fails, the other takes over
  - Dual networking interfaces
  - Redundant cabling

Failover

- Failover is the (semi)automatic switch-over to a standby system or component
- Examples:
  - Windows Server failover clustering
  - VMware High Availability
  - Oracle Real Application Cluster (RAC) database

Fallback

- Fallback is the manual switchover to an identical standby computer system in a different location
- Typically used for disaster recovery
- Three basic forms of fallback solutions:
  - Hot site
  - Cold site
  - Warm site

Fallback – hot site

- A hot site is
  - A fully configured fallback datacentre
  - Fully equipped with power and cooling
  - Applications are installed on the servers
  - Data is kept up-to-date to fully mirror the production system
- Requires constant maintenance of the hardware, software, data, and applications to be sure the site accurately mirrors the state of the production site at all times

Fallback - cold site

- Is ready for equipment to be brought in during an emergency, but no computer hardware is available at the site
- Applications will need to be installed and current data fully restored from backups
- If an organization has very little budget for a fallback site, a cold site may be better than nothing

Fallback - warm site

- A computer facility readily available with power, cooling, and computers, but the applications may not be installed or configured
- A mix between a hot site and cold site
- Applications and data must be restored from backup media and tested
  - This typically takes a day

Business Continuity

- An IT disaster is defined as an irreparable problem in a datacenter, making the datacenter unusable
- Natural disasters:
  - Floods
  - Hurricanes
  - Tornadoes
  - Earthquakes
- Manmade disasters:
  - Hazardous material spills
  - Infrastructure failure
  - Bio-terrorism

Business Continuity

- In case of a disaster, the infrastructure could become unavailable, in some cases for a longer period of time
- Business Continuity Management includes:
  - IT
  - Managing business processes
  - Availability of people and work places in disaster situations
- Disaster recovery planning (DRP) contains a set of measures to take in case of a disaster, when (parts of) the IT infrastructure must be accommodated in an alternative location

RTO and RPO

- RTO and RPO are objectives in case of a disaster
- Recovery Time Objective (RTO)
  - The maximum duration of time within which a business process must be restored after a disaster, in order to avoid unacceptable consequences (like bankruptcy)

RTO and RPO

- Recovery Point Objective (RPO)
  - The point in time to which data must be recovered considering some "acceptable loss" in a disaster situation
- RTO and RPO are individual objectives
  - They are not related