DSM

Datenerstellung zunehmend einfacher / schneller
⇒ größere Datenmengen
Durch Experimente, Observation, Messung sehr wertvoll, einmalig
80% unstructured 20% rows & columns
Information Lifecycle for business data: created → processed → delivered → warranty void → disposed
Scientific " : Datenakquisition → Analyse → Lehre → Idee
↑ Projekt / Messung ←
Datenbedarf in der Forschung steigend: LHC → 50PB in 4 Jahren
Challenges: Managing Storage Growth
Proper Forecasting
Managing Costs
Backup administration
Managing Complexity
Was man dafür tun kann:
Tiered Storage Build-out
Consolidation
Technology refresh
Backup Redesign
Virtualization adoption
Improving Performance
Archiving
Key Requirements for Data Centers
Availability
Security
Data Integrity
Performance
Scalability
Capacity
} + Manageability

Increase storage Efficiency:
Tiered storage: SSDs for caching, critical apps don't migrate critical data to cheap storage (and slow)
consolidate Storage: reduce management complexity
Use virtualization

Data Protection: automatic migration w/ retention policy
migrate crucial data to new storage devices
data deduplication
multisite mirroring
automated failover

Service Management: improve visibility to improve control & forecasting

Data access speed per GB drops

Tape < HDD < SSD

HDDs write magnetic information perpendicular
⇒ higher information density compared to horizontal magnetic field orientation, but slower
⇒ huge, slow disks

Trend: more workload on more reliable SATA disks
SONAS w/ 7200+ disks, Latency distribution
Caching algorithms: LRU < Wise order ~~RAM~~ writes (wow, [illegible]
NVRAM holds write-queue
destage order depends on physical position of block to write

Auto-Tiering: automatically migrate hot data:
SATA < FC < SSD (Like suburb-commuter problem)

CPU <-> Cache <-> RAM | <-> (NVRAM GAP) <-> Flash <-> HDD <-> tap[e]

Volatile | Networked | offlin[e]

Medias: magnet[ic] low c[ost] : slow
optical write [once] (limit[ed])

STORAGE ACCESS & ARCHITECTURES

APPLICATION

File/record Layer — Data Base / FS — Virtual

Block layer — Block aggregation — Host {Network} Device

Storage Device — Physical

Evolution over Time: DAS → JBOD → RAID → SAN/NAS → IP SAN
(+LAN / FC SAN) (Multi-proto[col])

Apps run on host, has CPU, storage, NIC, HBA

User) Host ⊃ Apps ⊃ OS ⊃ FS ⊃ Volume Manager ⊃ Device Driver ⊃ HBA
UI.

LBA: Logical Block Addressing }
File-Level-Access (path & name) } Types of data addressing

LVM: creates logical view on physical disks
supports: physical volumes & volume groups & logical mapping

One or more physical disk may form a volume group
managed as single entity, can be added and removed on demand

Logical Volumes > # physical devices => partitioning
" " < # " " => concatenation

Host has device drivers, to recognize and access device
hardware dependent & OS specific
" has FS. FS = hierarchical structure of files

Physical Components for interconnection between host & device:
Bus + HBA + Cable
~~Protocols used:~~

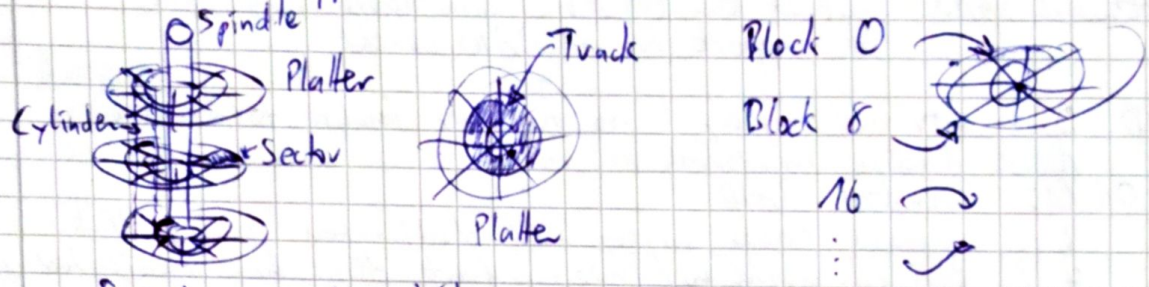
Tightly Connected: @ mainboard, no cable
Directly ": @ case, cable
Network ": @ world, cable, wireless

Medias:

magnetic tape
low cost for long-term data storage
" " per GB
slow until section of desired data is reached, fast after

optical disks
write once read many (WORM)
Limited capacity & speed

Disk Drive
good read/write random access, ideal for performance intensive online application



Speed: cache hit ~ 1 ms
" miss: avg_seek_time + Rotational_Latency + transfer_time
=> head movement + disk movement + data size

$$IOps = \frac{1000}{\text{seek time} + \text{rotation} + \text{transfer time}}$$

Little's Law: $N = a \cdot R$

$N \hat{=}$ ~~X~~ requests
$a \hat{=}$ arrival ~~time~~ rate
$R \hat{=}$ avg-response time

Utilization Law: $U = a \cdot R_s$
$U \hat{=}$ Utilization
$R_s \hat{=}$ service time

FC 15k RPM ~ 160 IOps
FC 10k RPM ~ 120 IOps } Random IO
SATA 72k RPM ~ 75 IOps

Command Queuing: re-ordering IO commands to reduce seek time. Fetch blocks that are ~~needed~~ requested, but would be passed by in order to maintain queue ordering

Short stroking: don't use the innermost Tracks.
+ Data can be accessed faster on the outer ones, with less head movement.
- less capacity used
Can be ideal for small critical data

Workload characteristics:
Transaction processing: small, but lots of IOps (73GB FC 15k RPM)
Throughput processing: few, but large Data requested (SATA)

performance characteristics: avg response time, read cache hit ratio

Flash Disks: high IOps, energy efficient, small, expensive, no moving parts

RAID

to overcome single disk limitations
MTBF: 7000 disks ~ 750 h till 1 failuve
RAID increases capacity / availability / performance

Host → RAID Controller → Disk, Disk, Disk (physical array) — logical arrays

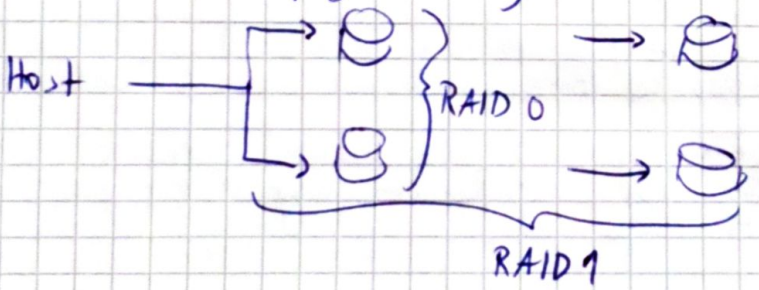
Hardware RAID: special hw controller w/ special software
Software RAID: part of OS, CPU has workload
No support for all RAID Levels

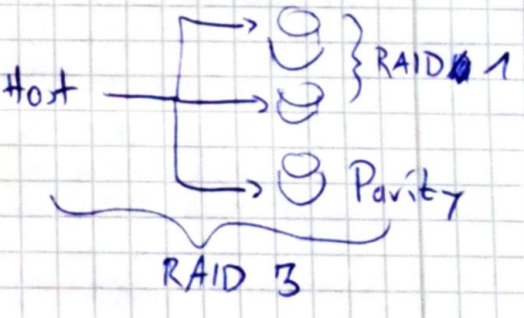
RAID 0: striped array: no further fault tolerance
1: 1 disk mirroring
01/10: nested RAID
3: parallel access array w/ dedicated parity disk
4: striped array w/ independent disks and dedicated PD
5: " " " " " " distributed parity
6: " " " " " " dual " "

Chunk size for example 256kB
Stride size: Capacity of a full stripe, e.g. 2 MiB for a 8+P RAID5
Spare drive: standby drive to jump in in case of drive failure
Rebuild time: duration for spare drive to hold migrated data

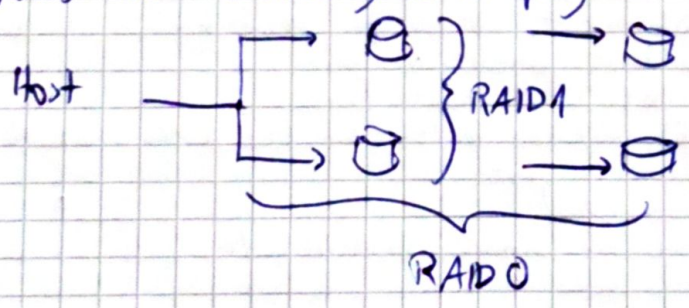
Strip: part of stripe on a disk. Stripes go on over multiple disks

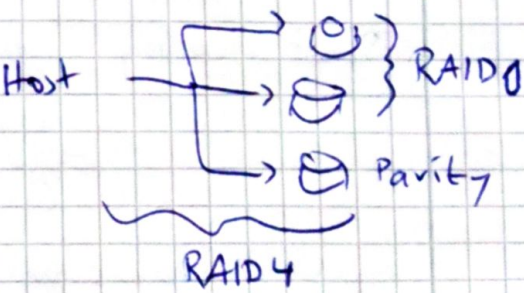
RAID 0+1 Striping & Mirroring

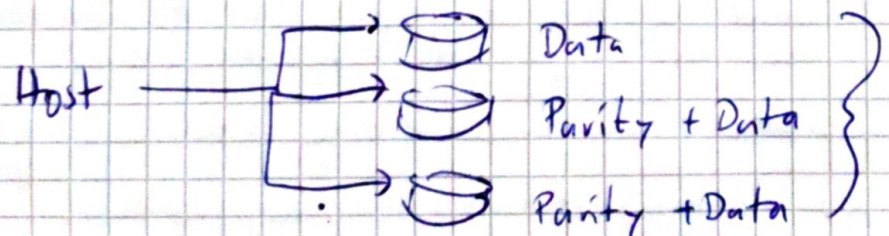




RAID 10 Mirroring & Striping



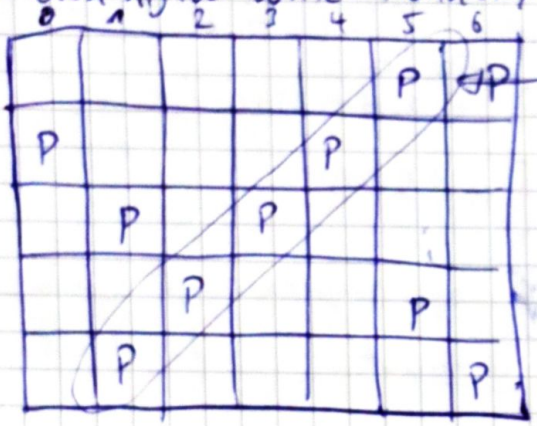




RAID 5 Write Penalty
Write-Update: 2 × Read + 2 × Write
Gegenmaßnahmen:
Fast write-Cache, backend writing asynchronously
Hardwar-Parity-calculation offload
Fullstripe writes
Read old data + old Parity on write
Write new data and parity afterwards
Writes can't have Cache hits!

⇒ Prefer RAID 10 over RAID5, when having lots of writes
Both have good read Performance

RAID 6: Double Parity RAID
survives 2 erasures
has even higher write-Penalty = 6 ops / write, 75% of space usable



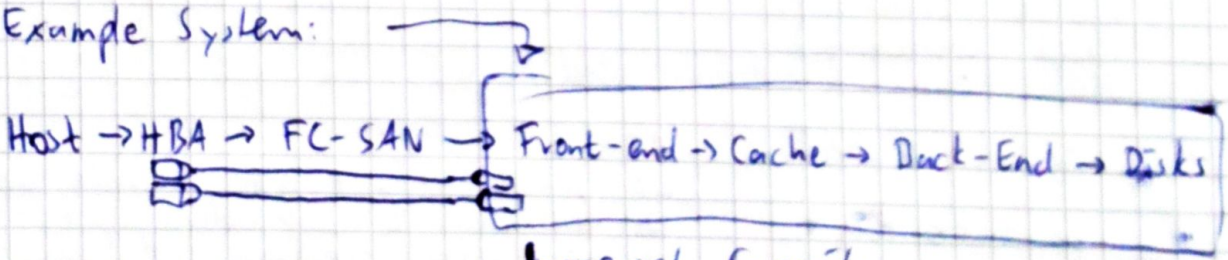
Parity holding the stripe

Other Parities hold crossed information parity

Always can rebuild 100% information until more than 2 drives fail.

Intelligent Systems needed to achieve optimized workload & performance:
1. Workload isolation
use dedicated hardware
2. Workload resource sharing
then distribute the workload among subsets of the dedicated hardware
3. Workload spreading
use all available HW!

Example System:



Increased Capacity
Improved Performance
Easier access & management
Improved availability & protection
Enhanced business continuity support
Improved security & access control

Write-Back: Cache wartet auf günstigen Zeitpunkt, enthaltene Daten auf die Platten zu schreiben. Fällt der Strom vorher aus, sind die Daten trotz gesendeter Bestätigung an den Host, verloren.
Wrik-Through: Quasi ohne Cache-Nutzung. Vermeintlich langsamer. Datenverlust bei Stromausfall nicht üblich.

Write caching
for intelligent destage order
destage data that is likely to be not re-written soon (LWR)
destage data in order that causes least head movement
overall: minimise cost of each destage
What about data loss?
Cache mirroring
dump data to a dedicated set of vault drives

What the host sees:
RAID set & LUNS
LUN masking can be used as a basic access control mechanism
to limit LUN access for individual hosts

High end storage systems use active-active-Link
to handle high amount of IOps, enable redundancy
Midrange storage systems use active-passive-Link
access to LUNs only via the active one
passive path used only in case of active Link failure

STORAGE CONNECTIVITY & NETWORKING

DAS
Simple & easy to deploy, reliable, low complexity & wiring
horrible scaling
distancing limitited, sharing difficult
no load balancing possible

SCSI
Target Devices have SCSI Server to interact with the host (initiator Device)
sends commands for data transfer & management functions
every port has an identifier, 64bit, unique in SCSI Domain
Address = Bus : Target ID : LUN
More virtualization over SAN, RAID & LUNs possible
Commands can be
Non-Data
Write (data-out)
Read (data-in)
Bidirectional (seldomly used).

Why serial (SAS, S-ATA)?

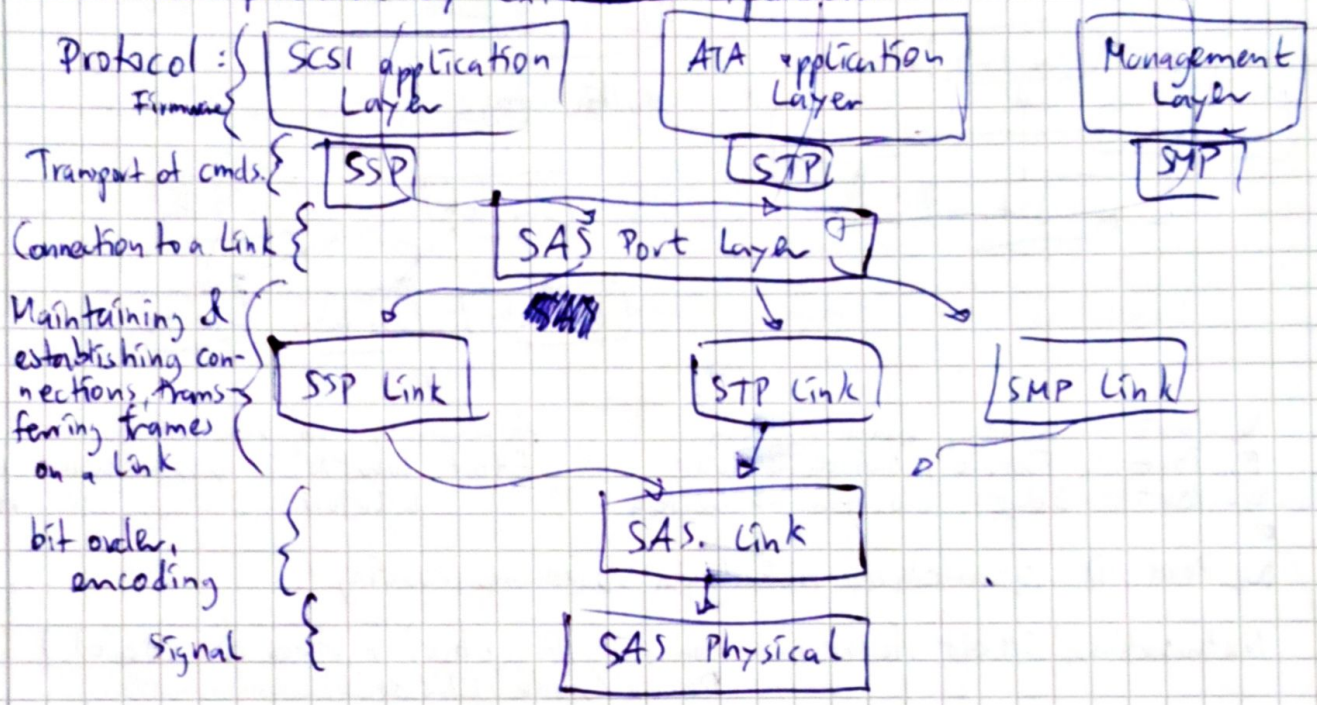
point-to-point → rebuild doesn't affect whole RAID
→ dedicated, scalable throughput
→ no routing overhead
→ no collisions
thin cables → improved airflow
transparent for existing Structure → ease of transition
higher throughput → better performance

New features:
hot-plug-technology
port multiplier (max 1:15)
port selector

SAS: Serial attached SCSI

Designed as a device, not a network interface
also 4-wire, drop in, 8b10b, point-to-point
dual-port support, 16k devices per domain
allows SATA devices in SAS domain

Port multiplication by ~~extenders~~ expanders



SAN (Storage Area Network)

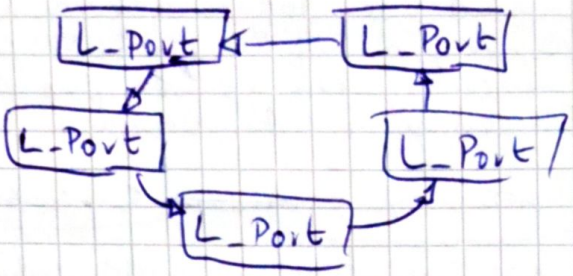
Dedicated high speed network of servers & shared storage devices
Block-level data access
Resource consolidation, centralited storage & management
Secure Access
High Availability
Scalable up to 15m devices

Components
Any-to-any connectivity between servers ~~via~~
Access to net via HBA
Fibre-Channel switches, Director (Router)
FC optimited for storage networks (SAN)
Block access with low protocol overhead

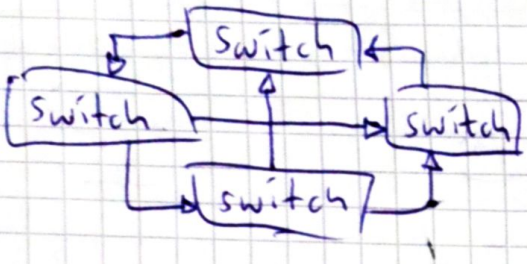
FC-4	Channels / Networks	Mapping to higher level Protoco
FC-3	Common Services	seldomly used, Multicast
FC-2	Framing Protocol / Flow Control	Credit Mechanism
FC-1	Encode / Decode	8b/10b
FC-0	Physical	

P2P

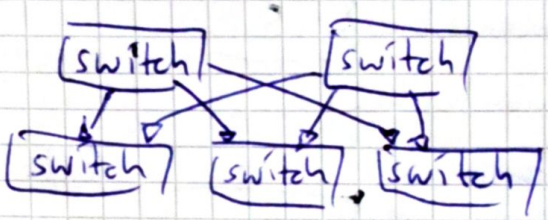
Arbitrated Loop (AL)



Mesh



Multi-stage



N_Port : node, end system
F_Port : Fabric, switch → node
E_Port : Edge, switch → switch

Addressing via MAC
64bit WWPN : world-wide port name
WWNN : " " node name

N_Port ID granted on net-login, used for routing

Networking 24bit Address: Domain ID (8bit) + Area ID (8bit) + Device/node ID (8bit)

ISL (Inter Switch Link)
for transfer host-to-storage & management traffic between switches
also one of the scaling mechanisms in SAN connectivity

FC Service Classes
Class 2 : connectionless P2P ACKs
Class 3 : " no P2P ACKs
Class 1 : reservation of dedicated path
Class 4 : " of shared path
Class 6 : multicast

FC Flow Control (FCFC)
Buffer credits, ~ 1 credit for each 2km @ 1Gb/s
P2P-Link level flow control
credit is reachieved on R_RDY receiving

FC Security
Switches support zoning
zoning describes, which F_Port is reachable from which F_Port
Authentification & access control @ end system
hard zoned ⇒ Packet drop
SANs are mostly closed and therefore more secure than IP networks

IP SAN (iSAN)

over LAN / ethernet

+ no additional network technology required
+ cheap access to 10 Gb/s
+ single integrated network
- security
- performance
- block-level data transfer ??

FCIP

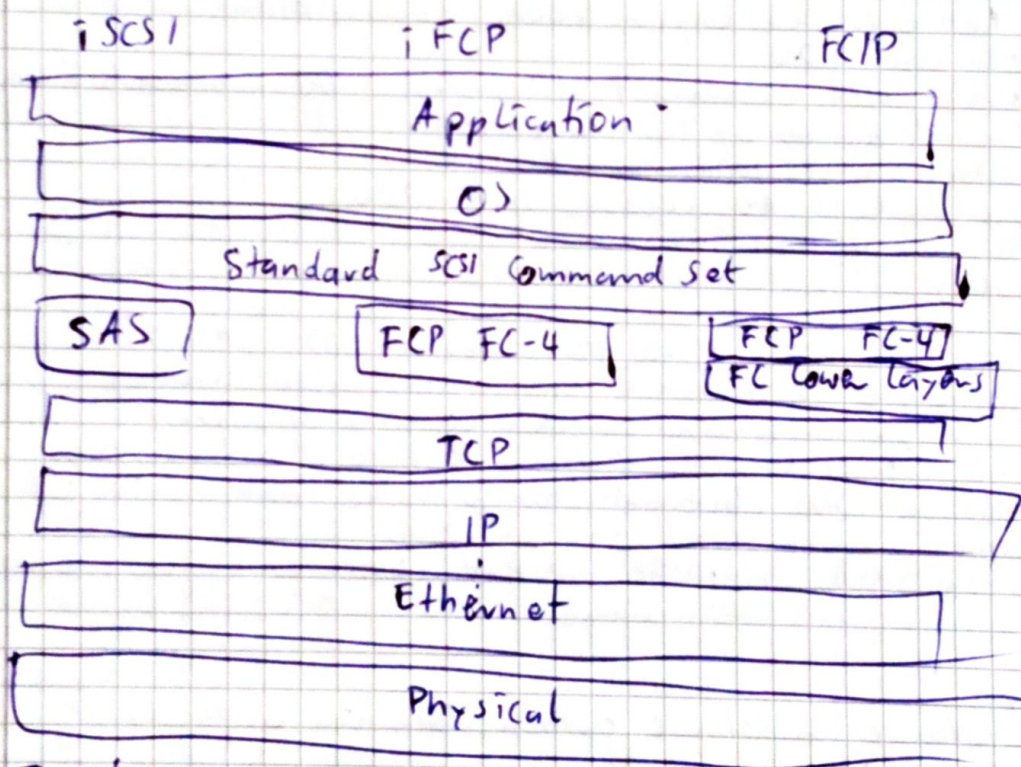
Tunneling FC through IP networks
Transparent for both networks
Used for connecting geographically distributed SANs

iFCP

IP Infrastructure used for routing & switching FC frames

iSCSI

connecting iSCSI devices over TCP/IP networks
Transfer SCSI commands via TCP/IP
IP infrastructure used for routing / switching SCSI commands



Transfer of SCSI Commands via TCP/IP network

Use case: Data Server w/o DAS. Connecting to SAN via iSCSI. Request sent to data server by KIT campus client. Student in WLAN

Implementing FC fabric via TCP/IP switched network
FC4 is mapped to TCP connection for transport
Dynamically creates IP tunnels for FC frames

Interconnecting islands of FC SANs everything transparent for every SAN

PDU : Protocol data Unit

iSCSI session: one or more TCP connections
Deliver SCSI commands in order
Recover from lost connections
Login phase begins each ~~session~~ connection

iSCSI Error Handling
Level 0: on error, bring down TCP connection + restart it
Let the SCSI Layer retry transmission
Level 1: discard PDU
iSCSI will retransmit discarded ~~PDUs~~ commands & data
Level 2: caused by loss of TCP connection
connection reestablishment
use level 1 to recover lost PDUs

Target Discovery
SCSI: polling the bus, give found devices ID (0-15)
iSCSI: 1. configure IP Address & port in initiator (static)
2. send command „send target" to given IP address
3. SLP (Service Location Protocol) to find IP address, then use 1.
4. DHCP or iSNS (Internet Storage Name Service) to get info
Common name schemes: iqn, eui
iSCSI qualified extended unique identifier name

iSCSI Security
CHAP (Challenge handshake authentication protocol)
SRP (Secure remote password)
Kerberos
SPKM (simple public key mechanism)
IPsec can be applied on top of that

TOE / TCP/IP - offload Engines
TCP/IP Protocolstack not efficient for SAN purposes
Protocol processing take time (overhead) checksum, interrupts, memcpy
1 Gb/s requires TOE
HBA implements TCP/IP protocols
ASICs used, sometimes dedicated GPP

FCoE
native map of FC to ethernet
preserving FC protocols and their latency, security, traffic management
preserving investment in FC tools, hardware & employees
FCoE connects widespread & cheap Ethernet w/ SAN-dominant FC

CEE (Converged enhanced ethernet)
(Data-Center ethernet)
(Data-Center bridging)
enables all network data to traverse:
FC, HPC, IP ...
settles on 10 Gb/s physical
lossless Ethernet enables I/O consolidation at the server by transporting storage and network traffic over CEE using a single Converged Network Adapter (CNA)
Ethernet can't support FC, because
a lossless
low latency and
lowly congested medium is required
developed and pushed by the CEE Group

CEE wraps FCP in FCoE before being transported

Ethernet : 48 bit unique, burnt-in MAC address
FC : 24 bit ID, unique in domain, variable
FCoE : Ethernet MAC, so called "session MAC"

CNA as HBA consolidates Ethernet & FC ports
Today: still parallel use of Ethernet & FC
more cables, HBAs, config overhead ⇒ cost!
more points of failure, but maybe more redundancy
slower server provisioning
seperate management realms

FCoE Frame:

12 B	MAC Addresses + 4 B	(normal Ethernet Header)
16 B	FCoE Header	(control information, version, ...)
24 B	FC Header	
up to 2112 B	Payload of FC Frame	
4 B	CRC	
4 B	EOF + Padding	
+ 4 B	FCS	
2180 B max		

Why lossless?
FC manages credit system on link level ⇒ no congestion, when FCoE contains FC Frame
Overall achieved by Layer 2 PAUSE Function

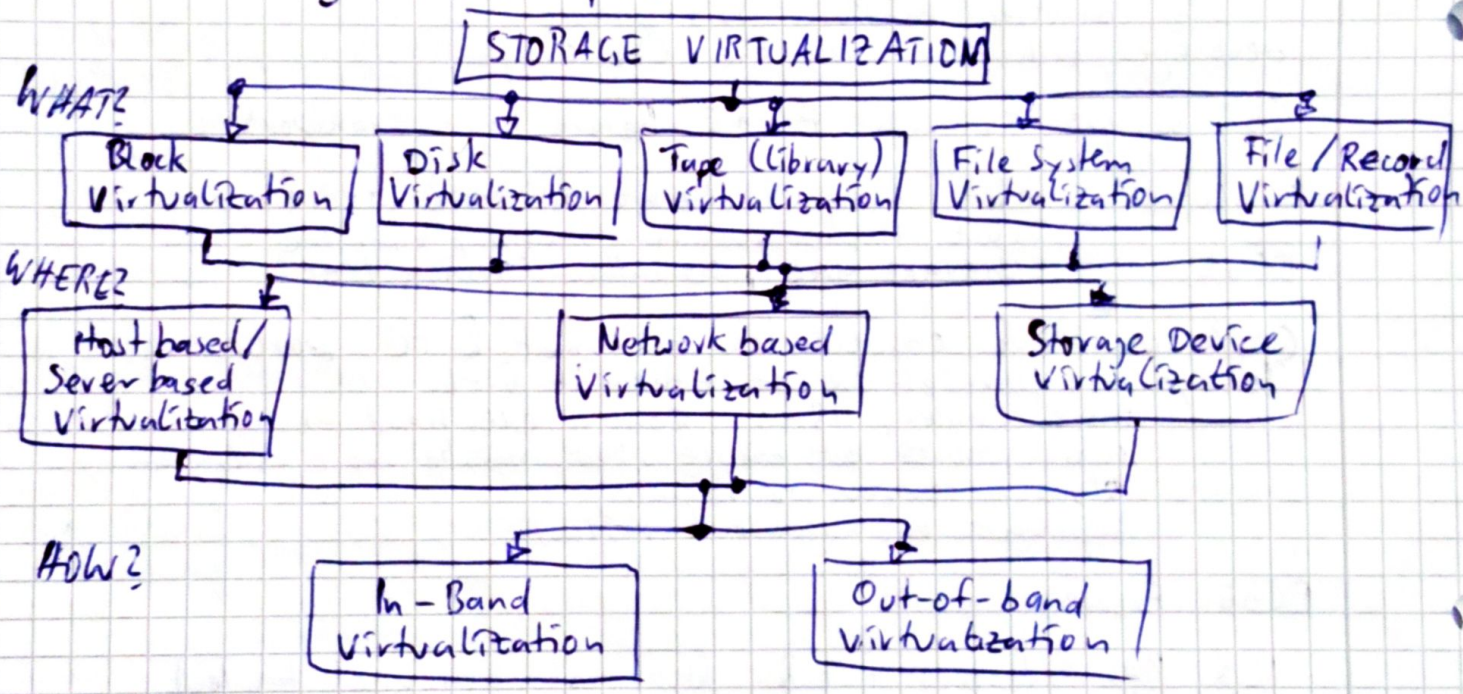
VIRTUALIZATION

Technique of abstracting physical resources to a logical view
increases utilization and capability of IT Resource
simplifies resource management by pooling & sharing resources
⇒improves performance

virtual memory, networks, servers, storage, ...

Why storage virtualization?

Consolidation: easier to manage one "big" instead of thousand "small" drives
Migration: w/o or less downtime
Building: remove or add disks a online
Flexible
Availability
Disaster tolerance
Performance
Minimizing unused disk space



DISK VIRTUALIZATION

C-H-S ↦ LBA (Done by ATA, S-ATA, SCSI, SAS)

BLOCK VIRTUALIZATION

Physical disks
fixed size
bounded performance
do break

Virtualization →

As big / small as user need
as fast as we need
can be (!) reliable
can grow, shrink, morph

Can be done by the OS, host: LVM host block aggregation
in the SN, network: Aggregation appliance

Network aggregation appliance

In-band: aggregation device is the data path
transparent to the host
Switch based
Server based (appliance)
- FC_N Port, F_Port or E_Port functionality
  optimized for little performance impact
- FC N_Port functionality
  Implement storage services flexibly and inexpensively

Out-of-band:
external metadata server stores aggregation map
tells the host where to find block, not-transparent
aggregation path is not the data path

FILE SYSTEM VIRTUALIZATION

Host Mounts FS via network. Linux: NFS, Windows: CIFS

Desktop PC mount local FSs. Access via SATA, SCSI, SAS, ATA, FCS
or via SAN: FC & SCSI
or via iSCSI: IP & SCSI
or a NAS FS: CIFS, NFS, HTTP, FTP

Protocols & Layers

LVM
Manages mapping of logical volumes to physical ones
Concatenation / mirroring / striping of these
Resizing w/o need to reboot
sometimes w/o remounting
usually no file-locking needs to be done by higher levels
VTOC Problem: every OS has its own ~~fun~~ format
most LVMs have own volume header and table definitions
control endianess

Block-virtualization in the storage device
Device-block aggregation
Flexible configuration of RAID sets
" mapping of LUNs to RAID sets
Transparent for hosts
Hot spots
Support for various drive types

Block virtualization in the network device
in band: smart switch
in band appliance can take care of data mirroring, if multiple storage devices are given. active/passive or active/active links possible.
Cluster file systems needed to avoid data integrity issues

All 3 virtualization approaches can be applied at the same time
connection via WAN, for example
Replication can be done by the in-band switch — no host load
or by the RAID system — host platform & network hardware independent, no host load

Performance improvement in SAN
achieved by striping data across independent drives and caching in the SAN appliance

Implementing Storage virtualization

Start w/ hosts w/ DAS
Add SAN appliance (w/ RAID)
Add Metadata Server (out-of-band)
Connect everything w/ FC
Add mirror devices
Move contents of DAS into SAN mirrored drives
Add redundancy to FC network
Consolidate storage
Apply load balancing

File / Record Virtualization
Presents one or more underlying objects as a single composite object, which can be files or directories
Can provide HSM-like (Hierarchical Storage Management) properties in a file system
Presents an integrated file interface - File data and metadata are managed separately in the storage system

File System Virtualization
Aggregates multiple FS into one large virtual FS
User access data through the virtual FS
⇒ underlying FS are transparent
⇒ enables different protocols to access existing FSs

NAS
Client & Server use File Protocols (NFS, CIFS, HDFS, FTP...) to access NAS Appliance / NAS Device

special storage device, IP based, dedicated, high performance file sharing and storage device
using special OS to serve Windows & Linux
optimized OS for I/O

FSs mounted over NAS are virtual FSs w/ File I/O
Clients: NFS Client / CIFS Client
communicating w/ NAS Server (storage device)
Servers: no printer drives and other "standard server apps"
optimized for file I/O, lots of disks attached
NAS ≠ File Servers
why? ⇒ improved efficiency, flexibility, availability, security
central storage eases management
native clustering

NAS System may be attached to LAN

Bringing FC SAN into LAN world by using NAS Gateway
combine their advantages
NAS flexibility
SAN scalability
increase the reach of FC SAN
FC SAN accessible via IP

Address SAN via IP!

NAS Client can mount virtual FS (file-level) via NAS Gateway ← gets block-level I/O from FC SAN attached

Scale-Out NAS

Pool multiple nodes to a single NAS Device
Create a single FS run on all nodes in the cluster
clients connected to any node can access entire FS
FS can grow as drives / nodes are added
striping data across nodes w/ mirroring & parity
use infiniband for interconnection of nodes
apply external switch for access

Client connected to one node sends file:
this node then stripes the file after retrieval

Client requests file from connected node
node reassembles file and sends it out to the client

FILE SYSTEMS

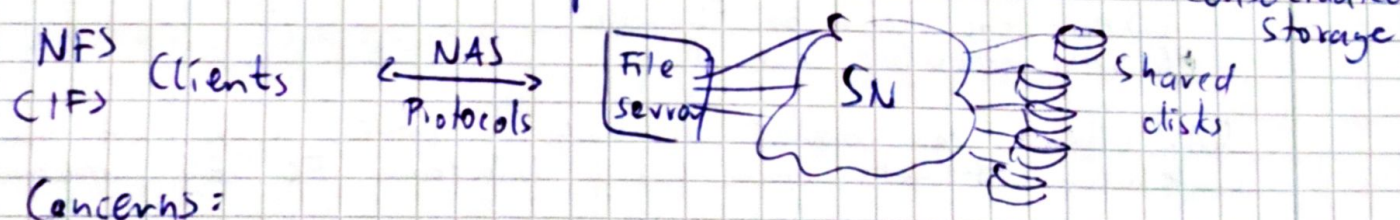
Physical / virtual
distributed FS create logical FSs, transparent for user
file structure (hierarchical) OS & FS dependent
physical system that assembles disk blocks to files
controls access
build a tree of directories & files

Local System
can be shared by different OS (help of LVM)
simple & fast data migration
but different OS can discard metadata because of VTOC problem
awareful conversions needed

Can be hosted by server via storage network
no data replication
cost effective
effective use of servers
no excessive copying

Distributed file Systems
Transparent via network
user mobility enabled
fault tolerance on other side possible
scalable
uses file aggregation / access
replication may be integrated
using uniform protocols

de facto standard: NFS / CIFS / SMB
features like RDMA possible



Concerns:

Locking ⇒ Coherency, read and write is not atomic
can collide on block level
When do changes propagate?
explicit flushing, strong coherency
weak coherency: read after write may return stale data

write ⇒ crash? what now?

stable data: write completion ensures persistance

unstable data: revert to old data

	NFS	CIFS
Locking	Apps, that use API	mandatory
Cache Coherency	weak	Strong
Impact of Coherency	App restart	Usually transparent (re-open of copy of file)
Data Stability	Client: write-back Server: write-thrugh	client: write-back server: write-back permitted, may not be stable

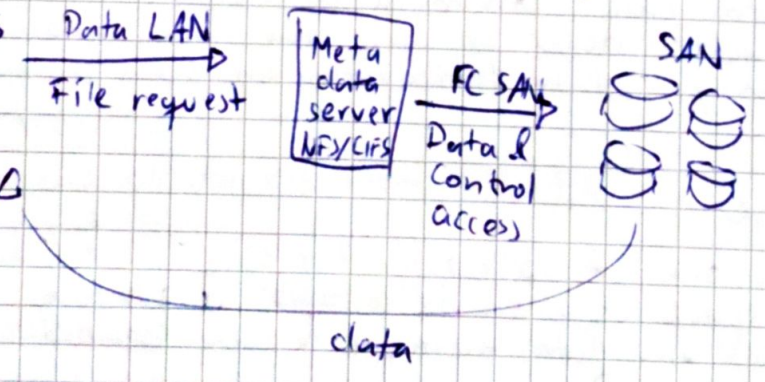
Steps to host a file system:

create volume
assign this " to NAS device
create FS on volume
mount the FS
Access FS using NFS / CIFS

Wide Area File Service
using WAN & NAS extenders Slave / master principle
NAS protocols via ~~A~~ WAN
Goal:
provide single view of the NAS FS and deliver good performance & stability
simplify management

SAN FS

Master / client principle
Master understand and manage metadata
block addresses are distributed to nodes on request
Slaves (clients)
are connected to SAN
no overhead for metadata management
access directly using received block address
designed to support 1000+ nodes in parallel



1 big NFS / CIFS Server w/ FC SAN access

Increased throughput by consolidating storage & FC
more effective use of resources

Asymetric client/server model
Server resolves client requests and possible conflicts
handles FC SAN access

1. Client sends (read/write) request to server
2. Server returns allowed access and SAN device addresses
3. Client accesses SAN Device using received address
4. Server may revoke access of the client, if another client requests the access

Lock mechanism
provided by the server, central point
granularity varies: block, file, byte
might use SMB/NFS semantics
SPOF, Server needs to be protected
either has no cache, or offer validate/invalidate mechanism

Client requests file metadata from server, then accesses SAN device directly.
File request → Server → block access in SAN

Cluster File Systems

Allows files to be shared
All nodes may mount the FS @ same time
they need to understand FS
all have the same view on data
used in Web services: multiple servers mount the same FS

Master mounts FS, could be any node
Remaining nodes act as client ⇒ asymmetric model
they ask for login & locks on files
All have direct read/write access
automatic failover: Master fails: restore locks, client node becomes master
client fails: his Locks are freed
Other approach: symmetric-peer-to-peer implementation
Lock is symmetric across domain
Lock mechanism done by distributed Lock Management (DLM/GLM)
in asymmetric case: controlled by master @ file level
in symmetric case: controlled by all
ship locks across nodes or have one lock repository
Cache Coherency: all modifications are seen by all nodes

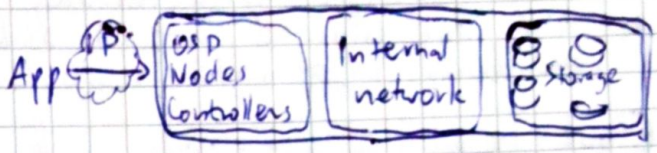
Object Storage

Store data in form of objects
flat address space based on its contents & attributes (not on path & name)
object contains user data, metadata & user-defined attributes
stored using object ID
Object Storage Device (OSD) provide API to interact w/ software-defined data center (& cloud)

Large amount of objects can be stored in flat address space

OSD usually contains 3 components:

- OSD nodes (Controllers)
- Internal network
- Storage



Features:

Scale-out-architecture	Linear scalability, nodes are added independently
Multi-Tennancy	multiple clients / apps can be served from the same infrastructure
Metadata-driven policy	Intelligently drive data placement, protection & services upon service requirement
Global namespace	provide common view, independent of location ⇒ seemingless scaling
Flexible data access method	via SOAP / REST API for mobile access and NFS / CIFS from computer
Automated system management	auto-configuration / auto-healing
Data protection & Geo distribution	use replication or erasure coding technique & distribute copies over different locations

Storing objects

1. App $\xrightarrow{\text{data}}$ OSD
2. OSD splits payload & metadata
3. OSD generates OID from user data
4. OSD stores object metadata & OID using metadata service
5. OSD stores user data using storage service
6. OSD $\xrightarrow{\text{ACK}}$ App

Retrieving objects

1. App $\xrightarrow{\text{request}}$ OSD
2. OSD Locates OID for requested object using metadata service
3. OSD $\xrightarrow{\text{OID}}$ App
4. App $\xrightarrow{\text{OID}}$ OSD storage service
5. OSD retrieves object from storage
6. OSD $\xrightarrow{\text{data}}$ App

Erasure Coding

space-optimal redundancy to protect data against loss, even facing multiple drive failures
n disks are divided into m disks w/ payload, k disk holding redundancy information, calculated from data

Cloud-based object storage Gateway

translate from standard interfaces (iSCSI, FC, NFS, CIFS, ...) to the cloud provider's REST API
presents file & block access to outer world
performs control mechanisms, data conversion
encryption
deduplication
compression
caching to reduce latency

REST (Representational State Transfer)

offers HTTP client-server interaction
allows to combine web resources into applications
HTTP GET, PUT, POST, DELETE
Resource locating using URI
Data & functionality are accessed by URI
Global addressing space, also serving for resource discovery
HTTP standard methods
Self-descriptive resource representation
representation reflects resource state if requested
requester then can act upon the representation
Stateless design
requester includes all data needed by a provider to generate response
provider doesn't need to store state ⇒ performance boost

SOAP (Simple Object Access Protocol)

Messaging protocol for interchanging data in a decentralized and distributed environment
executing remote procedure calls
can work together w/ HTTP, SMTP, TCP, UDP, ...
XML exclusively used for data representation
bound to HTTP, all requests are sent via POST requests

Unified Storage

A single integrated storage serving block (iSCSI, FCoE, FC), file (CIFS, NFS) and object (SOAP, REST) storage
Benefits: reduced expenses (through hardware savings)
central management
increased storage utilization
Integration of software defined storage for mobile, cloud, big data and social applications (desired)

Software Defined Storage (SDS)

Storage management automated by software
pools heterogeneous resources while serving dynamic storage allocation
abstracts from all physical details, storage is software
supports all access protocols and granularities
delivers one big unified view on the storage infrastructure
central management point delivering dynamic, policy- or user driven storage
open, flexible and extensible