Data: stream of bits, bytes to be stored;

Metadata: data about the location of other data;

Bandwidth: the amount of data that can be transferred in a given time [B/s];

Latency: speed of accessing the device [ms];

Energy consumption: amount of energy needed to operate [W];

IO: input, output operations, reading and writing the device;

IO performance: an important quality measure [IO/s, IOPS];

IO model: host accesses data storage;

Structure: the way we understand/interpret data, i.e. meaning of data;

Random access, sequential access: the way of reaching the data in a data set;

- Magnetic recording:
- Magnetic cover on the platters;
- Direction changes in magnetization store bits;
- Disk controller is used to control the rotation as well as the head positioning;
- Data is stored in blocks from 512 Byte (classic) to 4096 Byte (modern);
- Architecture: media - buffer - interface;
- Longitudinal and vertical recording.

Characterized by

- capacity (Bytes, 1-10 Tbytes)
- OS reports less for many reasons;
- error correction: 10% of space;
- redundancy + FS structures;
- decimal or binary prefix;
- speed (rpm, 4200-15000 rpm)
- latency (sec, 2-6 ms);
- throughput (bit/s, 1-3 Gbit/s);
- form factor (2.5" vs 3.5");
- energy consumption (W);

- Desktop: 4-12 Tbytes, 3.5", 0.5 Gbit/s, 5400-10000 rpm;

- Laptop: 1-4 Tbytes, 2.5", 0.5 Gbit/s, 4200-7200 rpm;
- Enterprise: 1-12 Tbytes, any, 1.6 Gbit/s, 10000-15000 rpm;

Interfaces:

the physical + logical link

- IDE, EIDE;
- ATA, PATA, SATA;
- SCSI;
- SAS;
- FC.

- Has a long history, since 1951, IBM + DEC were pioneers.
- Allows sequential data storage.
- Operations: record, play, fast forward, rewind.
- BOT ... EOT.
- Media write: linear vs. helical tracks.
- Exposed tapes ... cartridges (with flash memory).
- Access time: 3 magnitude longer than those of HDD's.
- Compression: redundant data compressed in a limited window buffer (2:1 typical).
- Encryption: for protecting data. Key management!

Linear Tape Open (LTO):

LTO1, 100 GB, 200 GB, 20 MB/s;

LTO2, 200 GB, 400 GB, 40 MB/s;

...

LTO7, 6 TB, 15 TB, 750 MB/s;

LTO8, 12 TB, 30 TB, 900 MB/s;

....

One step backward compatibility.

Tape drives + tape libraries:

- Take tapes in parallel.
- Collect a huge set of cartridges.
- Offer standard interfaces to hosts.

Error correction: CRC + repeated read.

Data blocks: Mbytes of block size.

Designed for data archiving, i.e. 15-30 years.

Data structures: data + metadata.

.tar format.

Labels, BAR-codes.

Cleaning tapes.

- A sort of flash memory (NAND), meant to replace (suplement) HDDs. Same size, similar form factors (exception M.2), same interfaces:
- SAS, 12 Gbit/s;
- PCI-e 3.0, 31 Gbit/s;
- Non-volatile, stores data when losing power.
- Stores data as electrical charges.

Advantages:

- fast random read;
- no moving parts;
- no spin-up time;
- less energy consuming;
- silent.

Disadvantages:

- high cost of capacity;
- limited amount of writes;
- faster, no moving parts, but, have endurance time: 1-2 years (except Intel's 3D Point);
- expensive;
- sophisticated controller.

Controller:

- R/W cache + battery;
- bad block mapping;
- wear leveling;

- Information is stored in floating gate transistors, i.e. read and written, holds charge for a long period.
- Levels of data storage:
- single level cell: 1 bit/cell, fast, endurable (50000-100000 erases), small;
- 2-level cell: 2 bits/cell, slower, less endurable (1000-10000 erases), medium;
- 4-level cell: 4 bits/cell, slower, less endurable, large.
- Capacities: 32 Gbyte - 4 Tbyte.
- Combined HDD + SSD devices.

- Redundant Array of Inexpensive/Independent Disks.
- What was expensive? Mainframe vs. PC worlds.
- Spread data across disks. Single logical disk.
- Reengineered structure: Striped data storage, parity calculation.
- Designed for:
- speedup devices;
- size increase;
- error recovery.
- Redundancy: useful vs. useless.
- RAID levels...
- Use the corresponding RAID to the purpose!
- SCSI RAIDs do better. Why?
- multiple devices on bus;
- device subdivision, device hierarchy.

- Hardware vs. software based;
- Rebuild process and rebuild time;
- Limitations:
- correlated device failures;
- unrecoverable read errors;
- failure during rebuild;
- lack of atomicity of writes;
- write cache reliability.
- TAPE RAID: mirror and concatenate.

RAID0:

- Concatenation and striping disks;
- Benefit: speedup.
- Drawback: failure of any device causes the whole structure to fail;

RAID1:

- Mirroring, no striping;
- Identically written disks;
- Benefits: security, data written 2 places;
- Drawback: performance, slower R/W;

RAID2:

- It has historical relevance;
- Bit-level striping and parity;
- Parity is written on dedicated drive;

RAID3:

- Has no practical relevance.
- Byte-level striping and parity;
- Dedicated parity disk;

RAID4:

- Has no practical relevance.
- Block-level striping and parity;
- Dedicated parity disk;
- Benefits: fast IO;
- Drawback: extra parity disk;

RAID5:

- Most widely used;
- Block-level striping;
- Distributed parity;
- Benefits: fast IO, secure for 1 device failure;
- Drawback: slow rebuild time;

RAID6:

- Widely used;
- Block-level striping;
- Distributed parities (2);
- Benefits: fast IO, secure for 2 devices failure;
- Drawbacks: even slower rebuild time, 2 extra devices;

Special RAID levels:

- RAID 1+0: mirrors first, then concatenate;

- RAID 0+1: concatenates first, then mirrors;

Hot spare!

≠

RAID

Where configuration is stored?

- configuration files (/etc/mdadm.conf);
- superblocks
- 256 Byte;
- stores metadata (RAID level, e.g.)

- Basic purpose: connects servers to disk and tape devices;
- OS senses devices as if they were directly attached;
- Offers only block level operations, no higher structures;
- From 2000s, developed from the mainframe world where multiple servers were connected with storage devices, eliminating Single Point of Faliure;
- Separate network from LAN: own topology and devices;
- Hosts do not own resources! Conflicts.
- As shared device QoS is needed:
- bandwidth;
- latency.

Host Bus Adapter (HBA):

- initiator;
- interface to servers;
- run firmware or software to make use of storage devices;

Fabric:

- active networking layer, such as switches, routers, gateway devices and cables;
- switch:
- gives dedicated port-to-port connection among the devices;
- non-blocking;
- copper vs.

optical cables;

- Chassis;
- Controller(s);

- Cache: RAM + NV;
- Enclosures;
- Power supply;

- Chassis;
- Backplane;
- Connectors;
- Power supply;

NO SPOF!

+

hotswap

Storage:

- target;
- disks, tapes;
- each device has a unique identifier: WWN;
- Logical Unit Number (LUN) = volumes;
- LUN masking and zoning;
- disk controller, disk array, disk enclosue - redundant architecture;

SAN variations:

- in-band: data + control;
- out-of-band: data || control (Ethernet)

Major SAN types:

- FC-SAN;
- Ethernet-SAN;

Zoning:

- soft - implemented by software;
- hard - implemented by hardware;
- WWN - address based;
- port - port based;

- History began in 1980s with file system sharing protocols: NFS, Novell;
- Network appliances with similar internal structure as SANs (disk pools, storage structures, volumes, etc.);

- Client sees it as a file server;
- Allow file system access: NFS, SMB/CIFS, Novell FS, SFTP, etc.;
- Requires simple IP/Ethernet network;
- Offer some sort of redundancy: power supply, controllers, connections, etc;
- Clustered versions: distributed data and metadata storage on multiple NAS'es, because
- extra processing power can be added;
- device failures will not disrupt the system.
- Can be rack mounted or standalone;

NAS management:

- Similar to that of SAN's: create structures, such as RAID groups, RAIDs;
- Create NAS volumes: user or backup (creating FS structures take longer time);
- Access control:
- users, groups, access lists, shares, (can be taken from auth servers);
- standards based access control (POSIX, Windows ACL);

Performance difference!

Conceptual difference:

- block:
- physical data concepts referring to the organization of data on disk drives, i.e. tracks, sectors;
- shorter access times, shorter data messages;
- files:
- logical data concepts made up of many blocks;
- more random data;
- longer access times, longer data messages;

Transformation is done by file systems!

Variations:

- SAN blocks sit on top of shared devices. Translation is done by storage controller on shared Thin Provisioning Volumes, TPVs.
- NAS gateway attached to SANs.
- Separate controllers for SAN and NAS functions.
- Shared controllers to manage SAN and NAS:
- Ethernet as carrier;
- FCoE, iSCSI, NFS, CIFS protocols;
- STaaS cloud storage, combines on-premises and cloud storage.

Advantages:

- Save space, device, management efforts, network costs;
- leverages utilization between NAS and SAN;
- Conform to STaaS providers;

Disadvantages:

- Performance penalties;

Test#1:

- 2 x SSD in RAID1;
- 2 x SSD in RAID0 cache;
- 6 x HDD in RAID5;
- 1 x hotspare;

Test#2:

- 4 x SSD in RAID5;
- 8 x HDD in RAID6;
- no hotspare + no cache;

General command set:

- EXECUTE DEVICE DIAGNOSTIC
- FLUSH CACHE
- IDENTIFY DEVICE
- READ DMA
- READ MULTIPLE
- READ SECTOR(S)
- READ VERIFY SECTOR(S)
- SET FEATURES
- SET MULTIPLE MODE
- WRITE DMA
- WRITE MULTIPLE
- WRITE SECTOR(S)

- An old PC standard initiated in 1986;
- Made disk controllers independent from drives;
- ATA 0 [3.3 MB/s] - 7 [133 MB/s];
- Integrated Device Electronics, EIDE at Western Digital;
- Originally 16-bits wide, then 22, 28;
- Maximum drive capacity 2^28 x 512 = 128 GBytes;

- 2 drives attached to parallel bus (master, slave, cable select, OS arbitration);
- Traditional 40-pin ribbon cable + 4 for electricity;
- Traditionally serialized command sequence;

Tagged Command Queuing:

- One request at a time to drives;
- As in case of SCSI, sending multiple commands at the same time;
- Due to the high number of interrupts this solution required lot of CPU power and was not really efficient;

PATA commands:

- 8 bit commands, from $00 to $FF;
- IO registers: Command, Control, Status, Data;
- Command registers are used to send ATA commands over the parallel ports;
- Status is received in status registers;
- ATA-8 contains a much broader advanced feature set;
- Protocols:
- PIO read, write;
- DMA transfer;
- Non-data commands.

- It is serialized version of Parallel ATA standard;
- Serialization helps reducing cabling structure and increasing clock speed -> increased transfer speeds: SATA1 [150 MB/s], SATA2 [300 MB/s], SATA3 [600 MB/s], SATA Express, 3.2 [2 GB/s];
- 100% software compatible with PATA standard, enhanced command set;
- Dedicated p2p crossbar connection among the host bus and drives, eliminating the bus (like data network);
- Frame based protocol;
- Hot-plug, hot-swap;
- Layered communication structure;

Benefits:

- lower pin count;
- larger performance, full duplex;
- simple configuration;
- lower voltage;

Link layer:

- 8b/10b encoding;
- converts data streams into frames;

- CRC;
- flow control;

Transport layer:

- managing FIS;

Frame Information Structures:

- Generated in the transport layer;
- Data movement setup;
- Read and write data;
- Used to control IO;
- Carries contents of ATA command registers;

SATA Port Multipliers:

- Up to 15 drives;
- Cost efficient connection of drives;
- Share the HBA bandwidth;

Native Command Queuing:

- Multiple IO commands in parallel (like SCSI);
- Due to mechanical constraints it is not efficient to serve IO requests in the order of their issue = elevator scheduling problem;
- Eliminate unnecessary travels = significantly improve performance;
- MCQ allows out of order IO command completion;
- Reordering of 32 IO commands;
- Command order received vs. command order optimized for shortest completion time;
- minimize rotation and head movement;

- History starts at 1978 (Alan Shugart, SASI), continues at 1986 (Small Computer System Interface, SCSI-1, first standardization);
- Parallel bus standard to ease the communication between the host and the low level disk drives;
- Uses 8/16 data + parity + 9 control signals;
- Originally:
- 8 bit + parity wide;
- 8 or 16 drives could be attached;
- 10 MHz.
- Interface names: SCSI-1 [5 MB/s], Fast SCSI [10 MB/s], Fast Wide SCSI [20 MB/s], Ultra Wide SCSI [40 MB/s], Ultra-320 SCSI [320 MB/s], and Ultra-640 SCSI [640 MB/s];

- SCSI initiator - SCSI target(s);
- Bus used as state machine: bus-free, arbitration, device selection, command, reselection (target disconnect in long operations), data, message, status;

Device types:

- 5-bit field sent in Inquiry;
- broad range: scanners, printers, block disks, CD-ROMS, floppies... $00-$1F

Device identification:

- manually set: 0-7, 0 - bootable disk, 7 - initiator;
- BIOS set;
- slot-set in storage shelves;
- ID discovery for SAS.

Connectors:

- SCSI Parallel Interface;
- Fibre Channel;
- Serial Attached SCSI;
- HD Mini SAS;

SCSI command families:

- non-data commands;
- reading data from target;
- writing data from the initiator to target;
- bidirectional.

Test unit ready

Inquiry

Request sense

Send/Receive diagnostic

Start/Stop unit

Read capacity

Format unit

Read (four variants)

Write (four variants)

Log sense

Mode sense

Mode select

Difference between SCSI and ATA:

- On non-protocol level: Enterprise Storage vs Personal Storage;
- Internal CPU driven IO vs Programmed IO (lack of DMA);
- Variable block size vs Fixed block size;
- Parallel queuing vs Serial command set;
- 15 drives vs 2 x 2 drives;
- larger buffers/speed vs smaller buffers/speed;

- P2P crossbar attached version of SCSI;
- Serializes transport between host adapter expanders, and drives;
- Certain compatibility with SATA;
- SAS1 [300 MB/s], SAS2 [600 MB/s], SAS3 [1.5 GB/s], SAS4 [3 GB/s];
- Identifies ports, HAs, devices with unique WWN;
- narrow and wide ports;

Serial SCSI Protocol:

- Wraps SCSI commands into serialized frames;
- Read Command, Data from target, Response;
- Write Command, XFER_RDY, Data to Target, Response;
- Nondata, Command, Response.

SATA Transport Protocol:

- Wraps SATA commands;
- Tunneling;
- Uses underlying SAS layers for transport;
- FIS.

SCSI Management Protocol:

- management frames to set up and inquire about SAS topology;
- set up and manage SAS routing;
- like RIP, BGP, or OSPF in case of Ethernet;

SAS and SCSI:

- P2P vs multidrop bus;
- no termination vs terminated bus;
- 65535 devices vs 8/16 devices;
- dedicated full bandwidth vs. shared bus bandwidth;

SAS and SATA:

- multiple initiators vs single initiator;
- TCQ vs NCQ;
- SCSI command set vs ATA command set;
- multipath IO vs single path IO;
- 1.6V signaling vs 0.6V signaling;
- up to 10 meters vs 1 meter cabling;

- Internet SCSI;
- Block level access over TCP/IP network;
- Ports 860/3260;
- Purposes:
- storage consolidation;
- disaster recovery;
- iSCSI initiator - iSCSI target;
- Types:
- hardware;
- dedicated;
- offload engines;
- software solutions;
- RFC 3720;

Addressing:

- ISCSI Qualified Name;
- iqn-yyyy-mm-reversed_domain:opt
- iqn-2019-04-hu.uni-obuda:storage1

Security:

- Challenge Handshake Authentication Protocol;
- filtering;
- multipathing;

- logical vs physical isolation;
- SSL over IP;
- no zoning!

Protocol extensions:

- MC/S: Allows multiple TCP/IP connections to set up an iSCSI session;
- Needs additional flow control in the protocol stack!
- Data layer + Basic Header Segments;

Messages:

- RDMA;
- Channel send and receive;
- Transaction;
- Multicast message;
- Atomic operation.

Layers:

- Physical: cables + interfaces;
- Link: Local IDs, Virtual Lanes (VL0-15);
- Network: Global Route Headers between subnets, IPv6 headers are used;
- Transport: in-order delivery, partitioning, channel multiplexing;

- A storage data network optimized for high bandwidth and low latency;
- Infiniband Trade Association including major companies;
- SDR [2Gbps, 5µs], DDR [4Gbps, 2.5µs], QDR [8Gbps, 1.3µs], FDR [14Gpbs, 700ns], EDR [24Gbps, 500ns], HDR [48Gbps, <500ns];
- Ports can be trunked to multiply bandwidth: 1x, 4x, 12x;
- Non-blocking switching;
- Switched network, with Host Channel Adapter, and Target Channel Adapter;
- Physical connections: CX4, SFP, CXP;
- 4k packets;

Virtual Interface Architecture:

- distributed messaging technology;
- it offloads traffic control from the client to dedicated execution queues;
- WQP is assigned to the transmission and reception side;
- Work Queue Entries -> Completion Queue Entries;

Why IB is faster?

- Store and forward switch: stores entire packets and makes decision then;
- Cut through switch: stores only headers, not full data frames -> low latency;

- Delivering SCSI command set over FC network.
- Besides SCSI: HIPPI, ATM, IP, NVMe.
- Bandwidth: 1-32 Gb/s. We have 4 and 8 Gb/s.
- Addressing:
- 3 byte addresses are used
- FFFFFA - fabric manager service
- FFFFFC - fabric name service
- FFFFFD - fabric controller service
- FFFFFE - fabric login service
- Principal switch: responsible for managing and distributing IDs within the domain.
- Flow control: throttling traffic.
- Topologies:
- FC-PP: point-to-point;
- FC-AL: arbitrated loop, ring;
- FC-SW: switched fabric.

Frame structure:

- Exchange: a session;
- Sequence: a set of frames;
- Frame:
- data;
- control (e.g. FSPF).

Protocol exchanges:

- Fabric Login (FLOGI): node enters the fabric;
- Port Login (PLOGI): session between initiator port and targer port;
- Process login (PRI): sending SCSI commands.

- Encapsulates FC data into Ethernet frames;
- Uses multi-functional HBAs and switches;
- Adapters, cables, switches:
- Converged Network Adapter: combines Ethernet and FC functionality in the same device;
- Software FCoE Adapter: uses kernel modules to process FCoE traffic;
- Fibre Channel Forwarder:
- Resides in a combined Ethernet and FC switch;
- Provides FC services: zoning, name resolving, etc.
- The same port types: VN_Port, VF_Port, VE_Port.

FCoE addressing:

- MAC addresses are used;
- jumbo frames: 2112 bytes vs. 1500 bytes;
- SPMA: compute servers provide MAC addresses;
- FPMA: fabrics provide MAC addresses (MAC:FC addresses concatenated);
- Discovery (FIP) - Login - Data forwarding.

Discovery and Login:

- Multicast Frame Initializing Packet messages to find FCF;
- FCFs send FIP Advertisement frames;
- Nodes send FLOGI requests to the closest FCF;

yum install fcoe-utils

yum install targetcli

modprobe libfcoe fcoe tcm_fc

echo ens160 \

>/sys/module/libfcoe/parameters/create

service fcoe start

targetcli

ls

cd backstores

cd block

create teszt1 /dev/sdf

status

ls

cd /tcm_fc

```
create naa.2000001b21551cda

cd naa.2000001b21551cda/luns

create /backstores/block/teszt1

cd ../acls

cat /sys/class/fc_host/host5/port_name

create naa.2000005056bf87e9

cd /etc/fcoe

cp cfg-ethx cfg-ens160

service lldpad start

dcbtool sc ens160 dcb on

dcbtool sc ens160 app:fcoe e:1

fcoeadm -l ens160

fcoeadm -t ens160

targetcli

cd iscsi

create iqn.2019-04.hu.uni-obuda:pelda1

cd iqn.2019-04.hu.uni-obuda:pelda1/tpg1/luns

create /backstores/block/teszt1

cd ..

set attribute authentication=0
```

set attribute generate_node_acls=1

set attribute demo_mode_write_protect=0

ls

exit

yum install iscsi-initiator-utils

vi /etc/iscsi/initiatorname.iscsi

iscsiadm -m discovery -t st -p 10.81.3.17

iscsiadm -m node -T \

iqn.2019-04.hu.uni-obuda:pelda1 -l

dmesg | tail

lsblk

fdisk -l /dev/sdap1

iscsiadm -m node -T \

iqn.2019-04.hu.uni-obuda:pelda1 -u

LE allocation policies:

- contiguous: LEs are adjacent and ordered;
- cling: LEs are on same physical devices;
- normal: indiscriminate allocation;
- anywhere: random allocation.

Operations on LVs:

- Concatenation of PVs;
- Mirroring: map multiple PEs to a single LE;
- Growing and shrinking LVs;

- Growing VGs by adding PVs;
- creating snapshots (copy on write LE mapping);
- Hybrid volumes:
- similar to hierarchical storage systems;
- bcache, dm-cache, Fusion Drive, etc.

- Divides physical space to chunks called Physical Extents (PE);
- Map PEs to Logical Extents (LE);
- Elementary terms:
- Physical Volume(s) (PV): a set of devices formulating the physical space;
- Volume Group (VG): a pool of resources, sum of all PEs serving the same purpose;
- Logical Volume(s) (LV): a set of dynamically allocated LEs;
- PE + LE have fixed size: 4MB.
- Structure: first MegaByte of each PVs stores the same LVM metadata;

- Originally LVM is planned for on host use;
- Clustered LVM:
- PVs are located on different hosts;
- A shared and lock-managed metadata service is used to store shared metadata;
- CLVM: metadata communication flows through the lock manager;
- HA-LVM: uses the default file locking mechanism, and avoids metadata contention, no concurrent accesses, useful in master-slave configs.

- Implements Logical Volume Management in MS Windows environment;
- By Veritas and MS;
- Volume types:
- Basic volumes: data storage is limited to a physical disk + no partitioning;
- Dynamic volumes: flexible volumes on the same physical disk, or on multiple disks:
- Partitionable;
- Striped volumes;
- SPAN volumens;
- Limited to 32 dynamic volumes;
- Uses special partition tables + special partition bounds, i.e. 1MB bound.

Types:

- local: all structures stored on local device;
- networked: structures are stored on remote server.

Directory structure:

- flat: a single root level only;
- hierarchical: arbitrary, yet mimited levels;

Names:

- caps sensitive;
- caps insensitive.

Based on the underlying media:

- disk file systems (optical, HDD, SSD);
- tape file systems (linear, mixed data and metadata);
- RAM file systems;
- database file systems (stored in RDBM);
- device file systems (/proc, /sys);
- nested file systems (disk images);
- etc.

Architecture:

- logical layer: application programming interface to users (open, close, stat, read, write, etc), manages file structures (i.e. open file tables);
- virtual layer: supports multiple instances of physical file systems;
- physical layer: concerns the physical operation of storage media;

Metadata:

- creation, modification time;
- creator;
- size: block allocation vs byte count;
- etc.

- There are plenty of them (>100), it would worth a full semester course.
- Stores the data organized into files (the analogy of paper documents), and directories (the analogy of paper folders).
- Since 1961.
- Classification...
- Metadata...
- Slack space:
- block allocation;
- files rarely end on block limits;
- the average amount of space left between the file end and the last block end;
- on average: block size / 2 / file, file number x block size / 2;
- Fragmentation:
- free space fragmentation;
- file fragmentation;
- File versioning (e.g. VMS);
- File System journaling (e.g. ext3);

FS journaling:

- protecting metadata against data corruption;
- metadata changes first go into a circular buffer, i.e. a journal;
- they are committed to media as atomic operations;
- damaged journal entries cause a roll back into last consistent state;

FS maintenance:

- multiple IOs -> different portion change;
- incomplete write operations -> neglected structural parts (e.g. orphan inodes, allocated blocks, etc.);
- consistency checks, reparations;
- defragmentation;

- One of the oldest FS'es from the 1970s;
- Simple, focuses only on storing data;
- Little endian byte order;
- Still wide-spread due to its simplicity;
- Versions: FAT12, FAT16, FAT32, i.e. the width of an entry in the FAT table;
- Structures and divides space into clusters: i.e. contiguous space of blocks:
- cluster size: 4 kByte as default;
- stores file data in cluster chains;
- that are not necessarily adjacent;
- File Allocation Tables: maps/indices of data region (cluster index or delimiter, 0xFFFF);
- 0: FAT identifier;
- 1: End-of-Chain (EOC) delimiter, 0xFFFFFFF);
- value 0: free cluster;
- value EOC: end of cluster chain;
- value any: cluster index;
- Directory entries:
- 32-byte long entries;
- root FS in a special region;
- original 8.3 file name notation, long names trick;

- Since 1993, as a replacement of FAT; more flexible.
- New features:
- Variable cluster size, default to 4 kBytes, up to 2 MBytes;
- Journaling ($LogFile);
- Hard links;
- Streams: a logical sequence of file records;
- File compression and sparse files;
- Volume Shadow Copy based on Copy on Write;
- ACLs: Directory ACL, Security ACL;
- Encryption;

- Quotas;
- Resizing;
- NTFS structure...

- Metadata: Master File Table (MFT) + MFT mirror;
- Stores ACLs, creation times, size info, data blocks, etc.
- 1 kB entries;
- NTFS metafiles (0-26, 27- regular files);
- MFT structure:
- header (block size, cluster size, type, etc.);
- attribute headers (standard, short name, long name, etc.);
- attributes (length, type, value, etc);
- resident vs. non-resident files/attributes;
- Data stored in either MFT (up to 900 bytes) or clusters.
- Runs: cluster intervals.
- Uses B-trees instead of tables and lists.

- Multiple variations: UFS, EXT2, EXT3, ReiserFS, etc.
- Represents a different principle:
- No letter tagged devices;
- single tree-structure;
- each partition, device is mounted under the / tree;
- Metadata stored in i-nodes.
- Directory entries are text files containing file names and i-node assignments.
- Multiple references to same i-nodes are hard links.
- "Everything is a file:"
- directories and files;
- device files, sockets, pipes, links;

UFS structure:

- boot block: serves for OS boot;
- superblock: describe FS geometry, version, parameters;
- cylinder groups: superblock backup, cylinder group header, number of i-nodes, and data blocks;
- i-node bitmap: allocation of i-nodes;
- data bitmap: allocation of data nodes;
- i-node table;
- data blocks;

Linux Ext4 file system features:

- journaling;
- data stored in extents, i.e. contiguous blocks, like cluster runs in NTFS;
- dynamic resizing;
- B-tree like directory structure;

- compression, encryption;

- block storage: data on physical device;
- file storage: data in human readable structures;
- object storage: some sort of associative storage over HTTP;
- Design principles:
- simplicity;
- robustness;
- partial associativity;

Terms:

- Bucket: A data storage pool, e.g. a device;
- Objects: Data units to be stored, like files, but have properties:
- GID;
- partially associative GID matching;
- any parameters that help indexing;
- optimized for large amounts of data: large number of large files;

Péter Stefán

Virtual Storage Architectures

## Practice #1

ssh root@10.10.10.10

tar -xvf iozone.tar; cd iozone/src; make linux

## Storage media

## Storage protocols

## that are of practical relevance here

## Storage architectures

## Storage structures

./iozone -a

./iozone -a -b output.xls

./iozone -i 0

./iozone -a -i 0 -i 1

./iozone -a -r 32 -s 1024

## Introduction

https://www.thegeekstuff.com/2011/05/iozone-examples/13/

## Storage media is the hardware in which information is physically stored.

compound storage devices built upon simple storage media

the way we construct network of elementary storage devices and storage structures

## Disk partitioning

## ATA over Ethernet

## Serial ATA

## Devices

Parallel Advanced Technology Attachment (PATA)

## The Course

## Practice #2

ls -l /dev/sdb

ls -l /dev/sdc

## smartctl -s on /dev/sda

## smartctl -a /dev/sdb

## smartctl --scan

## smartctl -i /dev/sdc

## smartctl -s on -t short /dev/sdd

## smartctl -i -A -f brief /dev/disk0

## Storage Area Networks (SAN)

## Direct Attached Storage (DAS)

## Practice#4: Take a look at SAN devices

the way we communicate over the storage interconnects

lspci -v

cat /proc/partitions

lsblk

## Hard Disk Drives

## Network Attached Storage (NAS)

[6] https://en.wikipedia.org/wiki/S.M.A.R.T.

- Simplicity attracts! :)
- Encapsulates ATA commands into Ethernet carrier;
- Carries ATA commands information to be put into registers;
- Data is sent in 8 KBytes = 16 blocks;
- Replaces traditional ATA transport layer (bus and cabling) to Ethernet;
- No need for multipathing;
- Lightweight protocol stack;
- Non-routing protocol;
- AoE initiator - AoE target;

Cylinder

Head

Sectors

LBA

Workload: 1 presentations + 1 tutorial a week;

Objective: give an overview on the world of contemporary data storage systems;

Assessment: midterm presentation + midterm test;

file level data storage

a network that provides access to block level devices

## Practice #5: Running AoE devices

aoe-tools + vblade installation

modprobe aoe aoe_deadsecs=10

lsmod

modinfo aoe

vblade 0 0 lo /dev/sdb &

aoe-discover

aoe-stat

fdisk /dev/etherd/e0.0

mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/etherd/e0.0 /dev/etherd/e0.1

Components

- Storage components appear as devices on operating system level.
- Corresponding drivers (= interface between the kernel and the devices) take care of them.
- Linux kernel: /dev files
- major numbers: kernel, SCSI, sd, 8, 65, etc.
- minor numbers: used by the driver, for sd it indicates disk and partition (0 disk, 1, part, 2 part, 16 disk, etc);

- Is a process of slicing the disk into individual units of variable sizes and purposes.
- Partitions are stored in partition tables written on the device:
- Master Boot Record (MBR) scheme, 16 byte entries;
- GUID Partition Table (GPT) scheme, 512 byte entries;
- In some cases it can be resized.
- Important to take care of partition table!
- Formatting: creating structures over the devices.

## Magnetic tapes

Data storage

http://www.t13.org/documents/uploadeddocuments/docs2006/d1699r3f-ata8-acs.pdf

ftp://ftp.seagate.com/acrobat/reference/111-1c.pdf

https://www.linuxjournal.com/content/mastering-ata-over-ethernet

http://storagegaga.com/aoe-all-about-ethernet/

http://download.alyseo.com:81/pub/partners/Coraid/Docs/AoE/AoEDescription.pdf

+ Management software

e.g.

SMI

https://www.mindshare.com/files/ebooks/SATA%20Storage%20Technology.pdf

https://en.wikipedia.org/wiki/Parallel_ATA

- A challenge in every age: from the first stone carvings to the digital revolution.
- Principles: safety, validity, integrity.
- Data: piece of information.
- Digital data: bits, bytes, k, M, G, T, P, E ...

- Not a network!
- Storage device directly connected to and exclusively used by the host;
- Most of the protocols are used (SCSI, SAS, ATA, SATA, FC, USB, etc.);
- It is not necessary to be a small device! E.g. SUN 10000 + DAS;
- External or internal;
- Smallest latency, closest to server, but limited complexity.

[4] https://en.wikipedia.org/wiki/Tape_drive

- HBAs;
- fabrics;
- disks + structures;

creating logical groups of devices

Practice #6: SCSI commands

[5] https://www.oreilly.com/library/view/linux-device-drivers/0596000081/ch03s02.html

- structure functionality;
- load balancing;

- route data in and out;
- optimize traffic;

[6] https://en.wikipedia.org/wiki/Disk_partitioning

## SCSI Protocol

lsscsi -l -L -s -c

sg_inq /dev/sdb

sg_raw -r 1k /dev/sdb 12 00 00 00 60 00

sg_read if=/dev/sdb bs=512 count=1

sg_readcap /dev/sdb

sg_raw -r 512 /dev/sdx 28 00 00 00 1f ff 00 00 01 00

-r = request -s = send;

1k + 02 = data size;

1f ff = LBA;

28 = SCSI READ command;

dd if=/dev/urandom of=infile.bin bs=512 count=1

od -x -t x1z -v infile.bin

sg_raw -s 512 -i infile.bin /dev/sdb \

2A 00 00 00 1f ff 00 00 01 00

sg_raw -r 512 /dev/sdb 28 00 00 00 1f ff 00 00 01 00

## iSCSI Protocol

## Serial Attached SCSI

http://sg.danny.cz/sg/sg3_utils.html

NAS caches: metadata + larger + write-back on sync requests.

- isolates physical storage elements from hosts;

- Volatile vs non volatile;
- Random vs sequential;
- Mechanical vs electronic;

## Virtualization methods

Logical Unit Number (LUN): individually addressable R/W device, volume

- logical disk units;
- tapes;

Controller, Target, Disk, Slice nomenclature;

## Virtualization

- Magnetic drives:
- HDD's
- magnetic tapes
- Flash drives:
- USB sticks
- SSD's
- Optical drives:
- CD's
- DVD's

- A way of decoupling data storage from its physical appearance.
- At any levels of data storage hierarchy.
- Not a magic word, but a set of real technical solutions: e.g. volume management, device mapping, clustered file systems.
- Layers hide details beneath, offer services above.

Block device based: it creates a mapping between the provider (storage) and user of data (host) on block level

- Host based (LVM, LDM);
- Storage device based: RAID controllers;
- Network based: iSCSI, FC;

File system based: it offers generalized access to different file systems in a uniform way, e.g. VFS.

https://arkit.co.in/san-switch-basic/

https://en.wikipedia.org/wiki/Storage_area_network

https://slideplayer.com/slide/1517247/

SAN

[1] https://en.wikipedia.org/wiki/Storage_virtualization]

- About this course
- Storage?
- Virtualization?
- Layers and protocols
- Data storage standards
- Fundamental terms

## Solid State Drives

Command Descriptor Block

## Practice #3: Creating RAID

[4] https://en.wikipedia.org/wiki/Solid-state_drive

## Just a Bunch of Disks (JBoD)

http://wiki.indie-it.com/wiki/Direct_Attached_Storage

## RAID

yum install mdadm

mdadm --create --help

= disk expansion units

http://pages.cs.wisc.edu/~remzi/Classes/838/Fall2001/Papers/scsi-ata.pdf

https://en.wikipedia.org/wiki/SCSI

http://fundasbykrishna.blogspot.com/2013/04/scsi-fundamentals-2.html

https://www.snia.org/sites/default/education/tutorials/2011/spring/networking/HufferdJohn-IP_Storage_Protocols-iSCSI.pdf

https://www.also.com/pub/pdf/hp_serial_attached_scsi.pdf

https://www.mindshare.com/files/ebooks/SAS%20Storage%20Architecture.pdf

## Unified Storage

## Standardization bodies

offers FCoE, iSCSI, NFS, CIFS in the same box

## Case study: measuring

## Eternus DX200

## Thin Provisioning Volumes

- International Committee for Information Technology Standards (INCITS)
- Internet Engineering Task Force (IETF)
- Institute of Electrical and Electronics Engineers (IEEE)
- American National Standards Institute (ANSI)
- International Organization for Standardization (ISO)

- Storage Networking Industry Association (SNIA)
- Fibre Channel Industry Association (FCIA)
- Linear Tape Open consortium (LTO)

- storage virtualisation technology;
- "Normally" 20-30% of capacity is not used;
- helps to improve storage capacity utilization;
- gives the appearance that there is more physical resource as there is;

- dynamic allocation of resources;
- allocates data blocks as they are written, not in advance (i.e. thick provisioning) at initial formatting;
- intelligent mapping: e.g. if zeroes come in, no space is allocated;

## FC over Ethernet Protocol

## Fibre Channel Protocol

## Infiniband

The configuration...

The results...

- It offers very simple disk organizing features;
- No significant added value, individual disks are connected through intelligent expansion units.
- These units are built up of as follows:
- basic enclosure;
- backplane with one or more expander chips;
- redundant power supply;
- proper connection, like SAS or SATA;
- Simple operations may apply:
- concatenation = spanning;
- mirroring.

mdadm --zero-superblock /dev/sdd

mdadm --assemble --scan

mdadm --zero-superblock /dev/sdb /dev/sdc

mdadm --create /dev/md2 --level 5 \

--raid-devices 3 /dev/sdb /dev/sdc /dev/sdd

mdadm --detail /dev/md2

mdadm --stop /dev/md2

mdadm --examine --scan

```
mdadm --add /dev/md2 /dev/sde

mdadm --grow /dev/md2 --raid-devices 4

mdadm --detail /dev/md2

mdadm --fail /dev/md2 /dev/sdd /dev/sdc ...

mdadm --create /dev/md2 --level 5 \

--raid-devices 3 /dev/sdb /dev/sdc /dev/sdd

mdadm --add /dev/md2 /dev/sde

mdadm --grow /dev/md2 --level 6

fdisk /dev/md2

mdadm /dev/md2 --fail /dev/sdd \

--remove /dev/sdd

mdadm /dev/md2 --add /dev/sdd

mdadm --create /dev/md1 --level=1 \

--raid-devices=2 /dev/sdb /dev/sdc

cat /proc/mdstat

mdadm --detail /dev/md1

ls -la /dev/md1

lsblk

fdisk -l /dev/md1

mdadm --fail /dev/md1 /dev/sdc
```

```
mdadm --remove /dev/md1 /dev/sdc

mdadm --add /dev/md1 /dev/sdc

mdadm --add-spare /dev/md1 /dev/sdd

mdadm --stop /dev/md1

mdadm --assemble --scan

mdadm --build /dev/md2 --level=1 \

--raid-devices=2 /dev/sde /dev/sdf

mdadm --stop /dev/md1 /dev/md2

mdadm --assemble --scan

show enclosure-status -type all

show hardware-information

show fru-ce -type cm0

show disks

show raid-groups

show eco-mode

show volumes

set volume -volume-name OEVOL01

show fc-parameters

show iscsi-parameters

show host-wwn-names
```

show host-sas-addresses

test iscsi-ping -port 10 -ip 192.168.2.10 -count 2

## Fundamental terms

show network

show smi-s

show storage-system-name

show raid-tuning

show cache-parameters

show extreme-cache

https://raid.wiki.kernel.org/index.php/RAID_superblock_formats

[7] https://docs.fedoraproject.org/en-US/Fedora/14/html/Storage_Administration_Guide/index.html

[8] https://media.techtarget.com/searchNetworking/Downloads/IncidentResponseChapter10.pdf

https://www.fujitsu.com/cn/Images/dx60-dx80-cli.pdf

https://sp.ts.fujitsu.com/dmsp/Publications/public/dx_s3_Extreme_Cache_HighPerformanceSolution_en.pdf

http://www.fujitsu.com/global/products/computing/storage/disk/eternus-dx/feature/component-redundancy.html

https://www.computerweekly.com/feature/Unified-Storage-FAQ

http://www.eternus-dx.com/eternus-overview/unified-storage/

https://en.wikipedia.org/wiki/Cut-through_switching

Practice #7: FCoE and iSCSI target creation

https://www.mellanox.com/pdf/whitepapers/IB_Intro_WP_190.pdf

https://www.arista.com/assets/data/pdf/Infiniband-vs-ethernet.pdf

http://www.tsmtutorials.com/2016/08/fc-san-protocols.html

https://en.wikipedia.org/wiki/Fibre_Channel

https://en.wikibooks.org/wiki/Introduction_to_Computer_Information_Systems/Storage

https://en.wikipedia.org/wiki/Computer_data_storage

## Practice #8: NTFS Forensics

yum install ntfs-3g ntfsprogs hexaedit

fdisk /dev/sdb

mkfs.ntfs /dev/sdb1

mount /dev/sdb1 /mnt0

vi /mnt0/egyikfile.txt (txt in hexa: 0x7400780074)

dd if=/dev/uradnom of=/mnt0/masikfile.txt\

bs=1k count=4

umount /mnt0

ntfsls -sal /dev/sdb1

hexedit /dev/sdb1

0x08 x 0x200 x 0x0004 = 0x4000 (16384)

ctrl + G 4000

ctrl + G 4400, 4800, 4b00, etc.

/ 740078007400

/ 494E4458

ctrl +G 189000

## File Systems

## Case study: Our SAN

## Storage operations

## Volume management

## High level structures

http://www.c-jump.com/bcc/t256t/Week04NtfsReview/Week04NtfsReview.html#W01_0180_mft_zone

https://flatcap.org/linux-ntfs/ntfs/concepts/data_runs.html

the operations over storage systems

create flexible volumes and mappings on nodes

create human-usable structures, i.e. files and directories

## Logical Volume Management

## Logical Disk Manager

## a more flexible way of allocating block device resources than in traditional way, i.e. partitioning

- Databases
- Cloud storage

## Practice #8: Using LVM

mkfs.ext4 /dev/tesztvg/bubuka

mount /dev/tesztvg/bubuka /mnt0/

```
vi /mnt0/stefan.txt

lvcreate -L 2M -s -n bubuka_snap \

/dev/tesztvg/bubuka

lvs

ls -l /dev/tesztvg/

mount /dev/tesztvg/bubuka_snap /mnt1/

vi /mnt0/stefan2.txt

ls /mnt1

yum install lvm2

pvcreate /dev/sdd

pvcreate /dev/sde

pvdisplay; pvs

vgcreate -s 1M /dev/tesztvg /dev/sdd

vgextend /dev/tesztvg /dev/sde

vgdisplay; vgs

lvcreate -L +1M --name bubuka tesztvg

ls -l /dev/tesztvg/bubuka

lvdisplay; lvs

lvextend -L +1M /dev/tesztvg/bubuka

pvmove /dev/sdd /dev/sdak
```

lvremove bubuka

vgscan

vgremove tesztvg

pvscan

pvremove /dev/sdd

- Multipathing
- Device mapping, persistent naming
- Deduplication
- Tiering
- Snapshot
- Device retention

https://www.howtoforge.com/linux_lvm_p2

http://strugglers.net/~andy/blog/2017/07/19/bcache-and-lvmcache/

https://en.wikipedia.org/wiki/Logical_Disk_Manager

https://en.wikipedia.org/wiki/Logical_volume_management

https://en.wikipedia.org/wiki/Logical_Volume_Manager_(Linux)

http://www.cyberphoton.com/questions/question/what-is-the-difference-between-lvm-and-raid

New Technologies File System (NTFS)

## Object Storage

## Unix File Systems (UFS)

## Distributed, Network and Global File Systems

## File Allocation Table (FAT) File System

Device layout structure:

Cluster chains:

- Huge areas on their own;
- Eliminate single points of failure;
- Allow multiple clients to access the same file data structure;
- Metadata protection, lock and sharing;
- Virtual and shared name spaces across local file systems;
- Variants:
- shared disk file systems: sharing a common area on disks (NFS, Lustre);
- distributed file systems: use network protocols to share objects (GFS, GPFS);

Directory entries:

https://social.technet.microsoft.com/wiki/contents/articles/6771.the-fat-file-system.aspx

https://en.wikipedia.org/wiki/Design_of_the_FAT_file_system#FAT

http://www.ntfs.com/ntfs_basics.htm

https://en.wikipedia.org/wiki/NTFS

https://www.yumpu.com/en/document/read/11722944/ntfs-cheat-sheet-writeblocked

https://www.nongnu.org/ext2-doc/ext2.html