universidade
de aveiro

Computer Systems Forensic Analysis
AFSC

Storage Devices

Artur Varanda

School Year 2021-2022

HDD

The importance of hard drives:

- are the primary form of non-volatile data storage
- they are the main source of digital evidence
  - ✓ but progressively replaced by SSD (discussed later on)
  - ✓ SSDs present new challenges to digital investigation

Main topics:

- physical interfaces and their main characteristics
- hidden areas

BIOS versus direct access

Direct access (without BIOS):

- reading and writing data directly through the hard disk controller
  - ✓ the software needs to know how to address the controller and how to issue commands to it
  - ✓ it needs to know the commands code for: read, write, . . .
  - ✓ it needs also how to query the hard disk for details such as type and size
  - ✓ this method is more complex, but also faster
  - ✓ modern OS perform direct accesses to disks

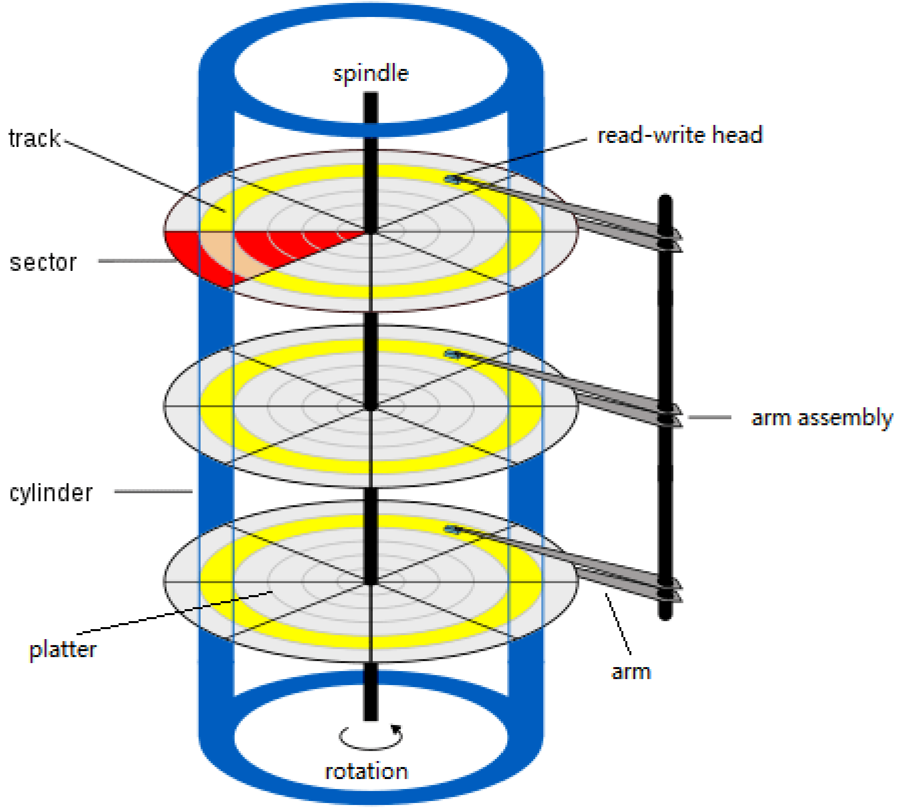
Access with BIOS

- slower than direct access
- but simpler, the BIOS does all the work
- the BIOS provides services to the software to communicate with the hardware
  - ✓ `INT 13h` and `extendedINT 13h`
- nowadays it is only used in the boot process

Hard Disk Geometry

Low level format

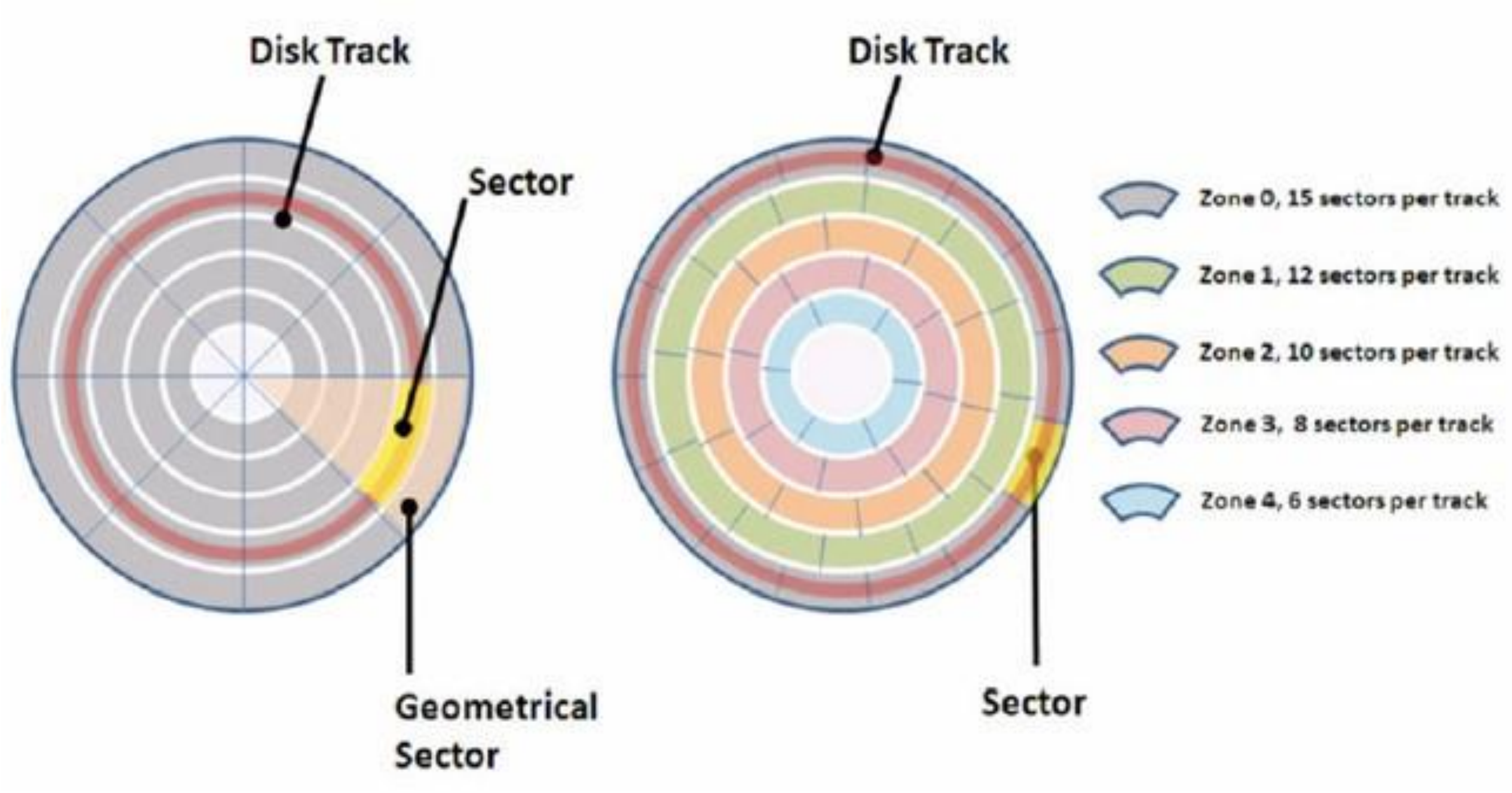
- to create data structures
  - ✓ tracks - addresses start from outside
  - ✓ cylinders - all tracks at a given address on all platters $C \in [0, max_C]$
    - tracks can be addressed by the head number $H \in [0, max_H]$
  - ✓ sectors - subdivision of tracks, typically 512 bytes $S \in [1, max_S]$



Get one sector *CHS*

- Cylinder address (C)
- Head number (H)
- Sector address (S)

Sectors distribution



Sectors addressing schemes

Cylinder, Head, Sector (CHS) – used only on older systems

- maximum addressable capacity 504 MB
- way around the problem with fake geometry
- but this translation was limited to address a maximum of 8,1 GB

Logical Block Address (LBA)

- each sector has a unique address
- the software doesn't need to know the disk geometry
  - ✓ used in some file systems: Linux, BSD, MAC OS, . . .
- still use CHS: FAT, NTFS
- LBA/CHS conversion:
  - ✓ $LBA = (C \times max_H + H) \times max_S + (S - 1)$

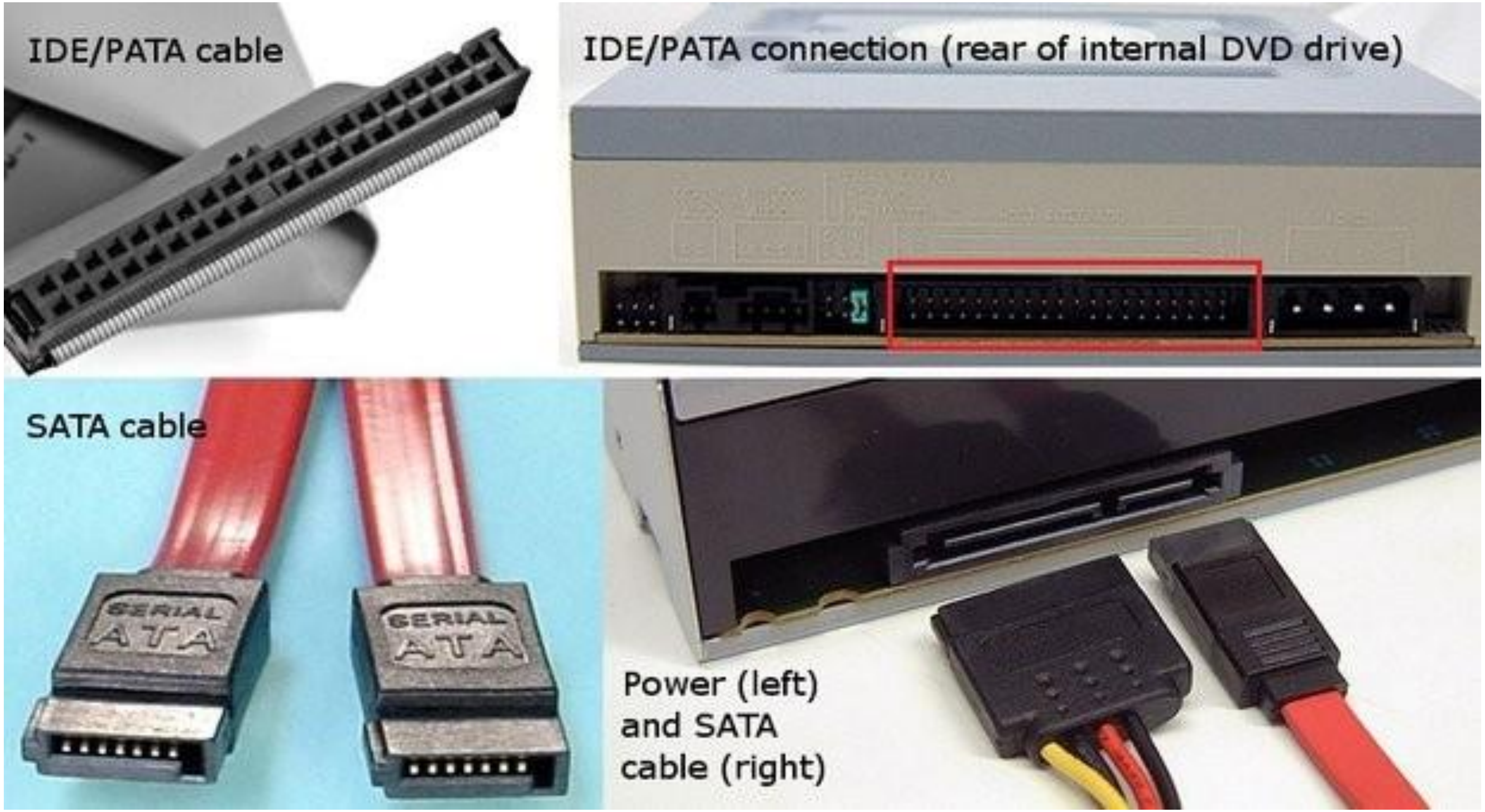
  (http://en.wikipedia.org/wiki/Logical_block_addressing)

ATA Interface
(Advanced Technology Attachment)

Evolution of *Advanced Technology Attachment* (ATA) interface

Name (year)	Synonyms	Max. Mbps	Observations
ATA-1 (1994)	IDE	8,3	cable with 40 wires, 16 bits in parallel
ATA-2 (1996)	EIDE	16,7	2 devices per cable
ATA-3 (1997)		16,7	add SMART and passwords
ATA/ATAPI-4 (1998)		33,3	support for removable devices (CD-ROM, DVD, . . .)
ATA/ATAPI-5 (2000)		66,7	80 wires cable, to lower interferences
ATA/ATAPI-6 (2001)		100	LBA addresses with 48 bits
ATA/ATAPI-7 (2002)		133	
SATA 1.0 (2003)		1 500	serial cable, 1 bit after the other
SATA 2.0 (2004)		3 000	
SATA 3.0 (2009)		6 000	
SATA 3.1 (2011)		6 000	added mini-SATA (for SSD)
SATA 3.2 (2013)	SATA Express	16 000	added PCI Express

IDE and SATA connections

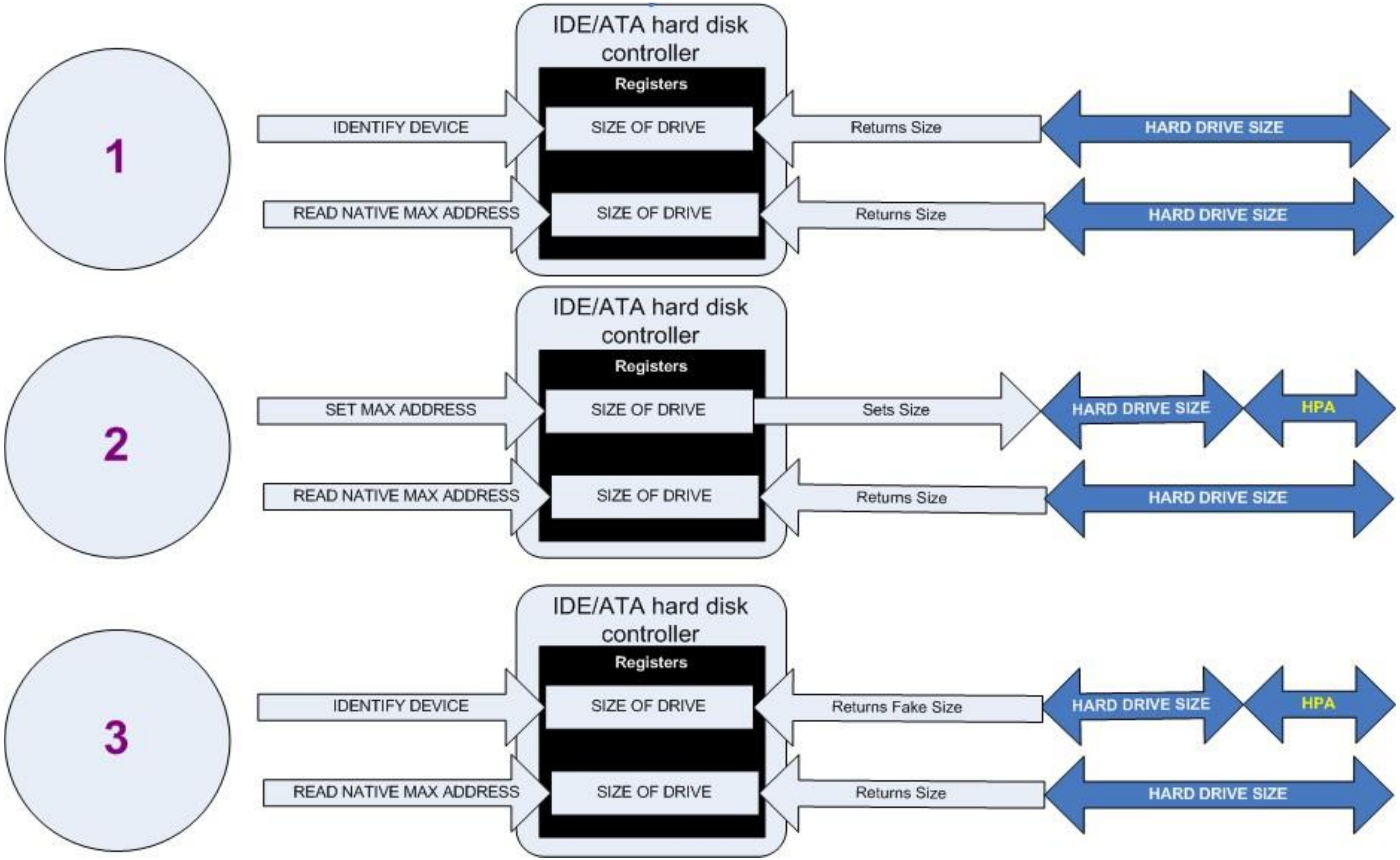


Protected areas

Host Protected Area (HPA)

- added with ATA-4
- special area to store vendors data
  - ✓ size can be zero bytes
  - ✓ guaranteed persistence – it won't be erased with a format
- it is located at the end of the disk
- requires reconfiguration of the disk to be accessible
- it can be used to:
  - ✓ reduce the disk size for the old BIOS to recognize the drive
  - ✓ to store diagnostic applications
  - ✓ pre-loaded OS (*e. g.* dedicated buttons to web OS)
  - ✓ system recovery (*e. g.* IBM, LG, . . .)
  - ✓ anti-theft tools
  - ✓ but, it can also be used to hide illegal files
  - ✓ some rootkits are able to hide themselves to avoid detection by anti-virus
  - ✓ some NSA exploits are known to use HPA to guarantee persistence

How to create and check for HPA



How to identify HPA

On Linux command line:

- at boot time

  dmesg | less
  [...]
  hdb: Host Protected Area detected.
  current capacity is 12000 sectors (6 MB)
  native capacity is 120103200 sectors (61492 MB)

- by comparing size values

  sudo hdparm -N /dev/sdX *# replace X with the device letter, X ∈ {a, b, c, . . .}*
  /dev/sdX:
  max sectors = 976773168/976773168, HPA is disabled

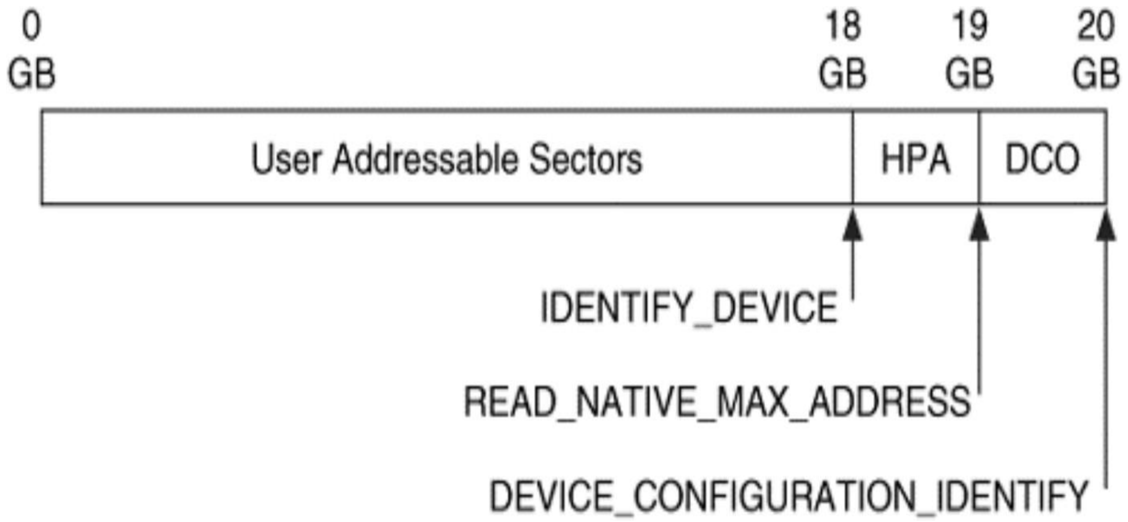
- to create an HPA

  sudo hdparm -N pZZZZZ /dev/sdX *# ZZZZZ is the number of visible sectors*

Linux tools are free, but there are many more:
http://en.wikipedia.org/wiki/Host_protected_area

Device Configuration Overlay (DCO)

- added in ATA-6
- with DCO, both BIOS and OS see the same size
- DCO removable is permanent (HPA remotion can be temporary)
- allows to hide the disks real capacity
  - ✓ PC makers can buy different brands of discs with different sizes and set them to have exactly the same size
- HPA and DCO can coexist on the same disk



How to identify DCO

- on the Linux command line

  hdparm --dco-identify /dev/sdX *# replace X with the device letter, X ∈ {a, b, c, . . .}*

  /dev/sdX:

  DCO Revision: 0x0002

  The following features can be selectively disabled via DCO:

  (...)

  Real max sectors: 976773168 *# DCO can be created*

  (...)

- compare values with

  hdparm -Iv /dev/sdX

  /dev/sdX:

  multcount = 0 (off)

  IO_support = 1 (32-bit)

  readonly = 0 (off)

  readahead = 256 (on)

  geometry = 60801/255/63, sectors = 976773168, start = 0

  (...)

  LBA48 user addressable sectors: 976773168 *# if smaller, there is a DCO area*

  (...)

How to remove a DCO area

Linux command line:

- with `hdparm` tool
- it is possible to remove, but not to create a new one
- **WARNING – it can destroy data permanently**
- to remove DCO and set the disk with the real size

```
hdparm --yes-i-know-what-i-am-doing --dco-restore /dev/sdX
```

Windows tools

TAFT (The ATA Forensics Tool) says it can detect and modify HPA and DCO (old, it mentions floppy disks!!)

https://vidstromlabs.com/freetools/taft/

SAFE-Block says it can detect HPA and DCO and put them back

https://www.softpedia.com/get/Security/Security-Related/SAFE-Block.shtml

more information and tools:

http://www.forensicswiki.org/wiki/DCO_and_HPA

SCSI Interface
(Small Computer Systems Interface)

Small Computer Systems Interface https://en.wikipedia.org/wiki/SCSI

- can connect up to 8 (or 16) devices on the cable
- commands error checking, with parity (SCSI-1, SCSI-2), or CRC32 (SCSI-3)
- are common in servers and high-performance systems
  - ✓ *SCSI-over-Fibre Channel Protocol* (FCP) – NAS systems
  - ✓ *Serial Attached SCSI* (SAS) – sserial cables (allows connection of SATA-2+ devices)
  - ✓ *USB Attached SCSI* (UAS) – external disks
- more expensive than ATA disks
- many kinds of connectors – generates some confusion

SCSI interface evolution

Version	Max. length	Max. throughput	# devices
SCSI-1	6 m	5 MBps	8
Fast SCSI	3 m	10 MBps	8
Fast Wide SCSI	3 m	20 MBps	16
Ultra SCSI	3 m	20 MBps	4
Wide Ultra SCSI	1,5 m	40 MBps	8
Wide Ultra SCSI	3 m	40 MBps	4
Ultra2 SCSI	4 m	40 MBps	8
Wide Ultra2 SCSI	4 m	80 MBps	16
Ultra160 SCSI	4 m	160 MBps	16
Ultra320 SCSI	4 m	320 MBps	16
SAS (2006)	—	3 Gbps	65 535
SAS (2009)	—	6 Gbps	65 535
SAS (2013)	—	12 Gbps	65 535

SCSI Connectors

Main differences between SCSI and ATA

Feature	ATA	SCSI
Devices per cable	up to 2	up to 8 (or 16)
Communication	by controller	direct by bus
Parallel communication	yes, 16 bits	yes, 8 or 16 bits
Wires per cable	40, or 80	50, or 68
Serial communication	> SATA-1	Serial Attached SCSI (SAS)
Availability	common	high availability system
Fault tolerance	—	power supply
Disk size	limited (older versions)	LBA of 32 or 64 bits
Rotations/minute	4,5k 7,2k 10k	10k 15k
Hidden areas	HPA, DCO	—

NAND Flash memory

Hard Disks Drives (HDD)

- few manufactures:
  - ✓ concentration of manufacturers through purchases and mergers over the years
- mature technology, with many aspects in common:
  - ✓ between disks models and sizes
  - ✓ between manufacturers
- digital research in hard drives is almost the same in all models and brands

Solid-State Drives (SSD)

- basic components are the same or very similar
  - ✓ between manufacturers
  - ✓ between flash memory and SSDs
- but there are important differences:
  - ✓ a flash memory requires driver software – uses CPU
  - ✓ SSD has its own processing unit – doesn’t use CPU
  - ✓ firmware between models or manufacturers can be very different

Solid state drives (SSD):

- are mechanically more reliable
  - ✓ have no moving parts and are more resistant to falls
- read speed is independent of the data location (which doesn’t happen with HDD)
- power consumption is lower (1h to 2h of increased battery autonomy on a laptop)
- emits no noise or vibrations
- heat less than HDD – *HDD can reach very high temperatures*
- are lighter – don’t require a metallic structure as HDDs

DRAM

- older solid state disk (they exist for more than 30 years)
- based on volatile DRAM memory
- require battery or other power source to ensure redundancy
- need of a traditional drive to store data permanently
- used in high-performance systems such as banks, stock exchange, military assets, . . .
- the cost of flash memory is falling more than DRAM → the crossing point was reached in 2004

Flash memory

- non-volatile
- there are 2 categories:
  - ✓ NOR gates NAND gates

With NOR gates

- used for small amounts of memory (< 16MB), *e. g.* BIOS
- allows very fast readings, but is slow to write and erase (up to 5 seconds)
- supports fewer write cycles (10× less than NAND gates)
- allows to read or write a single byte at a time
- allows local execution, without having to use RAM
  - ✓ uses a SRAM interface that enables to address all bytes

With NAND gates

- provides large bit density → ideal for replacing HDD
- erase and write faster than NOR (up to 4 ms), but slightly slower readings
- reads and writes are made in large blocks of bytes
- disadvantages:
  - ✓ internal management complexity
  - ✓ serial access to data, wear leveling, garbage collection, . . .

NAND Flash memory – is the most common type of flash

- USB pen drives
- Solid State Drives (SSD)

Management of bad blocks

- all devices have bad blocks
- an initial test to identify bad blacks is required
  - ✓ the cost of creating chips without defects does not pay off
  - ✓ it is preferable to put capacity in excess and then remove the addresses with bad blocks

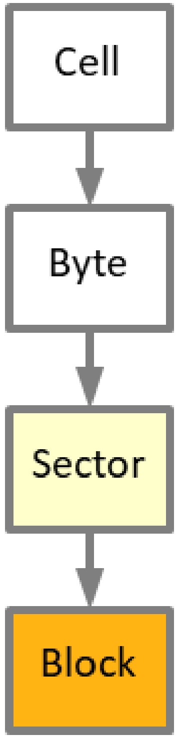
NAND Flash memory

Inner working of a NAND chip

- at rest = 1 (stores the value 1) at load = 0 (stores the value zero)
- to increase density, they can be produced in layers: MLC (multi layer chip)
  - ✓ several bits have to be read/written simultaneously
  - ✓ allows more capacity, but has lower performance than the single layer chip (SLC)
  - ✓ cheaper

Data access

- data access in grid with word lines (16 bits)
- minimum writing unit is a sector with a size multiple of word lines
  - ✓ HDD: 1 sector = 512 Bytes → minimum read and write unit
  - ✓ SSD: 1 sector = [512, 2048] Bytes
    - ✓ depends on several factors, such as a manufacturer and disk capacity
    - ✓ minimum writing unit differs from minimum erasing unit
    - ✓ it is not possible to erase a single sector
    - ✓ data as to be erased by blocks – the electric charge to erase is similar to a photographic flash



1 single bit (initialized to “0”)

8 cells = 1 Byte

Minimum write unit — (512+16) | (2048+64) Bytes = 1 page

Minimum erase unit — (16 | 32 | 64) sectors = 1 block
Erase puts cells back to “0”

What is level wearing?

- SSDs change data location to level the number of writing operations across all cells
- if the location was always the same for changing date, those cells would burn out quickly
  - ✓ each cells stands ≈ 100 000 erasing cycles
- firmware is responsible for doing the level wearing in an automated fashion
- it is also required a garbage collection system to identify freed sectors, that weren't erased yet

How garbage collection works?

- it keeps a list of freed sectors
- one block is erase only when all sectors are marked as free
- even without a data connection, the garbage collection keeps running on its own and restarts in case of a power outage

HDD vs SSD – file writing

On HDD

- it is possible to read and change the information in a specific sector
- reading specific sectors is common in digital investigation

On SSD

- when a file is modified it is not possible to save it in the same sector
- because it is not possible to write into cells before erasing them
- the file is saved into a new empty sector and the original one goes to the garbage collection list
- the old sector is erased only when all sectors in the same block are freed

HDD vs SSD – file erasing

SSD erasing and wear leveling mechanisms consequences:

- when we ask the OS to delete, the data is not actually deleted, it goes to the garbage collection list
- only a few flags change
- it is effectively deleted only when the erase routine is executed by the garbage collection algorithm
  - ✓ for example: 1 block of 64 sectors × 2048 B = 128 kB
- the physical location of a sector changes over time → **this is a problem for data acquisition at the physical level**
- even without an OS commanding, garbage collection operations can happen (power on is enough)
  - ✓ a write blocker does not prevent garbage collection or wear leveling operations
  - ✓ this limits the availability of "deleted" or interesting slack data

Pen USB vs SSD Comparison

Main characteristics:

- doesn’t have its own processor
- so, it requires a mass storage software driver to manage operations:
  - ✓ file system → block device services → mount/read/write/delete virtual sectors
  - ✓ identity/read/write/erase → flash memory
- uses the CPU to:
  - ✓ calculate ECC
  - ✓ bad blocks management
  - ✓ wear leveling, . . .

Main characteristics:

- has its own processor to manage operations:
  - ✓ wear leveling, bad blocks
  - ✓ erasing cycles counts, sectors initial location
  - ✓ error checking code
- so, it doesn’t requires a software driver
- SATA commands are emulated to guarantee compatibility
- garbage collection only needs power to start operations
  - ✓ a write blocker doesn’t stop this operations
- repairing
  - ✓ easy on HDD: you can replace controller cards, heads, . . . and use the same platters
  - ✓ difficult on SSD: too complex, only possible on highly specialized labs

SSD Connectors,
Interfaces and Transfer Protocols

Connectors, Transfer Interfaces and Transfer Protocols

Connectors:

- layer 1 – physical interface to connect devices. Examples: M.2, RJ45, . . .

Link interfaces:

- layer 2 – handles data encoding. Examples: PCIe and SATA

Transport protocols:

- layer 3 – handles data comunication. Examples: NVMe, AHCI and IDE.

M.2 – One connector, several transport protocols:

- M.2 connector = SATA link interface + SATA transport protocol
- M.2 connector = PCIe link interface + AHCI transport protocol
- M.2 connector = PCIe link interface + NVMe transport protocol

M.2 Connector Interface

M.2, formerly known as the Next Generation Form Factor (NGFF)

- specification for internally mounted computer expansion cards
- replaces the mSATA standard
- more flexible physical specification make it more suitable to:
  - ✓ solid-state storage
  - ✓ particularly for the use in small devices such as ultrabooks or tablets

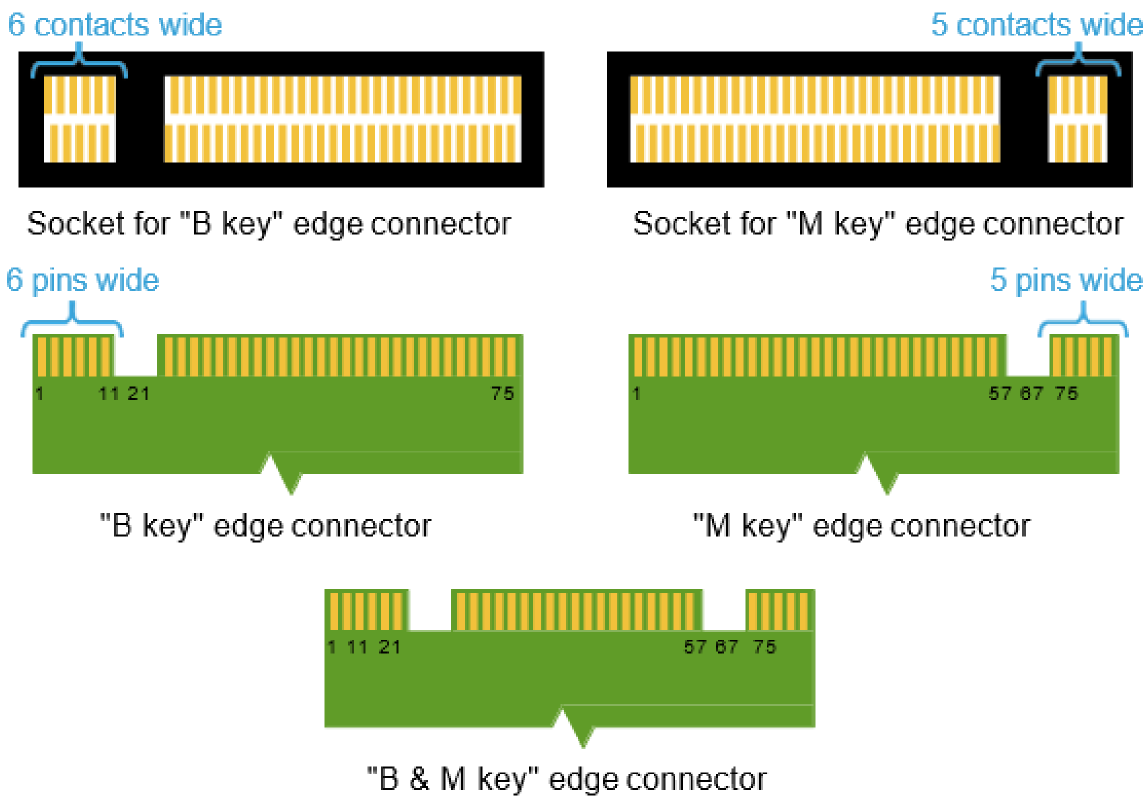
protocols:

- link interface: PCI Express 3.0 (up to four lanes)
- transport protocol: Non-Volatile Memory Express (NVMe) as the logical device interface for

  M.2 PCI Express SSDs
  - ✓ NVMe is designed to fully utilize the capability of high-speed PCIe storage devices to perform many I/O operations in parallel
- Serial ATA 3.0 and USB 3.0 (a single logical port for both)
- the manufacturer selects which interfaces are supported

The M.2 connector has different keying notches:

- to denote various purposes and capabilities of M.2 hosts (SSD, WiFi, 4G modem, . . .)
- to prevent plugging into feature-incompatible host connectors



Source: Wikipedia, https://en.wikipedia.org/wiki/M.2

- PCIe is a high-speed serial transfer interface standard
- M.2 connector supports up to four PCIe channels

Evolution of PCIe

Version	Year	Transfer rate	Throughput (Channel width × transfers/second) × 1	× 2	× 4	× 8	× 16
1.0	2003	2,5 GT/s	250 MB/s	500 MB/s	1,00 GB/s	2,00 GB/s	4,00 GB/s
2.0	2007	5,0 GT/s	500 MB/s	1,00 GB/s	2,00 GB/s	4,00 GB/s	8,00 GB/s
3.0	2010	8,0 GT/s	984,6 MB/s	1,97 GB/s	3,94 GB/s	7,88 GB/s	15,75 GB/s
4.0	2017	16,0 GT/s	1969 MB/s	3,94 GB/s	7,88 GB/s	15,75 GB/s	31,51 GB/s
5.0	2019(?)	32,0 GT/s	3938 MB/s	7,88 GB/s	15,75 GB/s	31,51 GB/s	63,02 GB/s

GT/s = Gigatransfers per second

Apple SSD Proprietary Connectors

Generation	Year	Connector	Interface
G1	2010	6+12	mSATA 3
G2	2011	7+17	mSATA 3
G3	2012	12+16	PCIe 2.0 ×2
G4	2013	12+16	PCIe 3.0 ×4
G5A G5B	2015	22+34 12+16	PCIe 3.0 ×4 NVMe

More info.: https://beetstech.com/blog/apple-proprietary-ssd-ultimate-guide-to-specs-and-upgrades

Some examples of Apple SSD connectors

Do the following exercise:

06-Lab 1 – Add a RAW disk to a virtual machine

Exercise

Do the following exercise:

06-Lab 2 - Smartmontools

QUESTIONS?