Introduction to Databases
《数据库引论》

Lecture 8: Physical Storage Systems & Data Storage Structures
第8讲：物理存储系统与数据存储结构

周水庚 / Shuigeng Zhou

邮件: sgzhou@fudan.edu.cn 网址：admis.fudan.edu.cn/sgzhou

复旦大学计算机科学技术学院

Content of the Course

- **Part 0: Overview**
  - Lect. 0/1 (Feb. 20) - Ch1: Introduction
- **Part 1 Relational Databases**
  - Lect. 2 (Feb. 27) - Ch2: Relational model (data model, relational algebra)
  - Lect. 3 (Mar. 6) - Ch3: SQL (Introduction)
  - Lect. 4 (Mar. 13) – Ch4 & 5: Intermediate & Advanced SQL
- **Part 2 Database Design**
  - Lect. 5 (Mar. 20) - Ch6: Database design based on E-R model
  - Lect. 6 (Mar. 27) - Ch7: Relational database design (Part I)
  - Lect. 7 (Apr. 3) - Ch7: Relational database design (Part II)
- **Midterm exam: Apr. 10**
- **Part 3 Data Storage & Indexing**
  - Lect. 8 (Apr. 17) - Ch12/13: Storage systems & structures
  - Lect. 9 (Apr. 24) - Ch14: Indexing
- **Part 4 Query Processing & Optimization**
  - May 1, holiday, no classes
  - Lect. 10 (May 8) - Ch15: Query processing
  - Lect. 11 (May 15) - Ch16: Query optimization
- **Part 5 Transaction Management**
  - Lect. 12 (May 22) - Ch17: Transactions
  - Lect. 13 (May 29) - Ch18: Concurrency control
  - Lect. 14 (Jun. 5) - Ch19: Recovery system
  - Lect. 15 (Jun. 5) – Course review

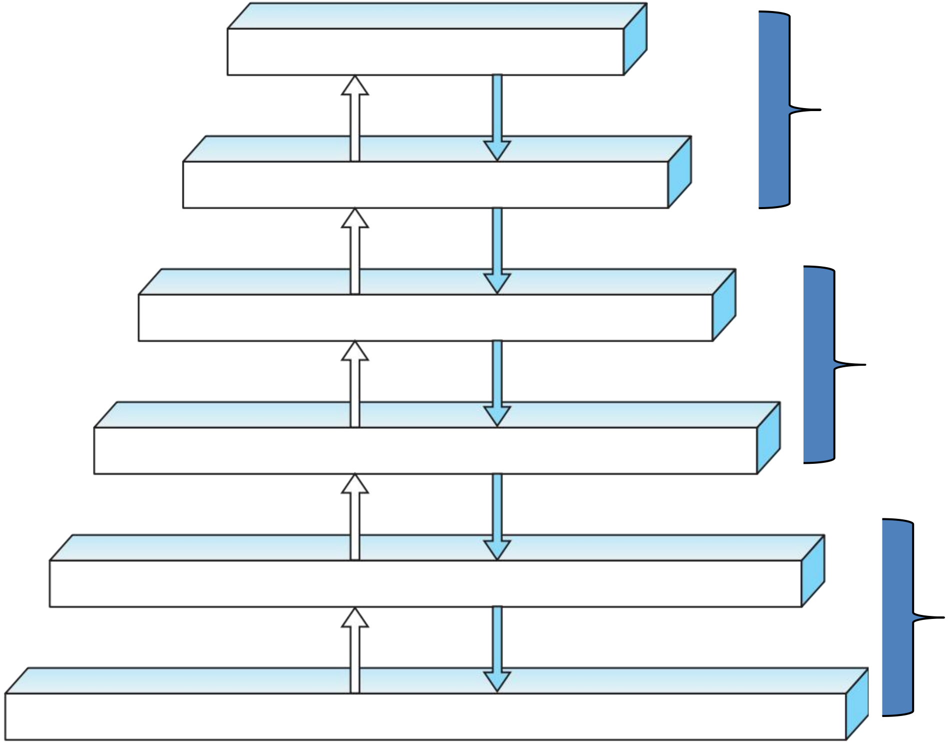
Final exam: 13:00-15:00, Jun. 18

Outline

☞ **Overview of Physical Storage Media**

- Magnetic Disks
- RAID
- Tertiary Storage
- Storage Access
- File Organization
- Organization of Records in Files
- Data-Dictionary Storage

Storage Hierarchy



primary storage
volatile

secondary storage
non-volatile

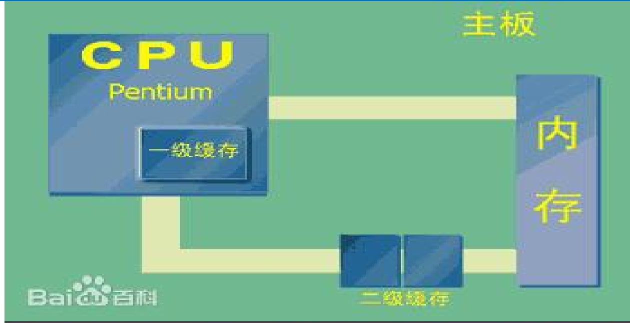
tertiary storage
non-volatile

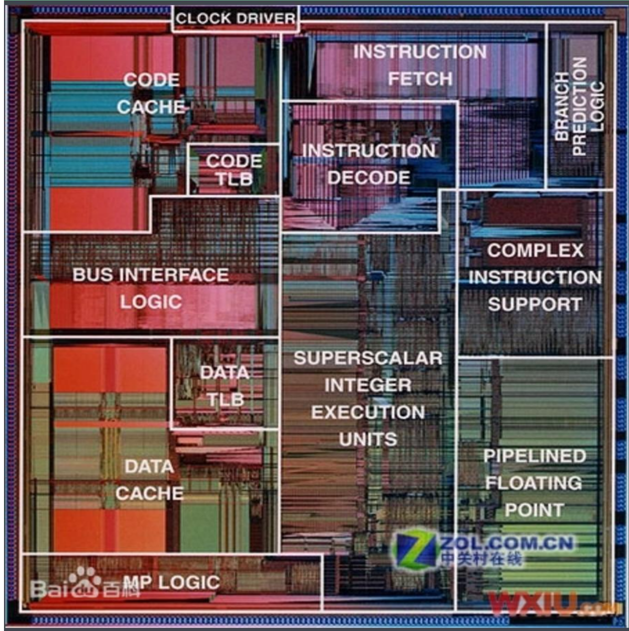
Classification of Physical Storage Media

- Speed
  - The speed with which data can be accessed
- Cost
  - The cost of per unit of data
- Reliability
  - volatile storage (易失性存储）：lose contents when power is switched off
  - non-volatile storage (非易失性存储）：contents persist when power is switched off
    - secondary （第二级）and tertiary（第三级）storage
    - battery-backed up main-memory

Cache & Main Memory

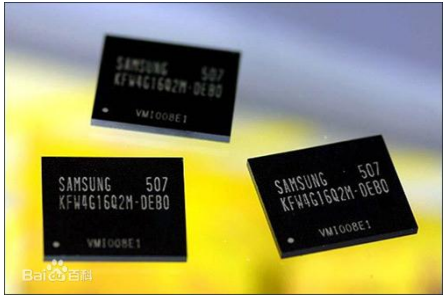
- **Cache (缓存)**
  - fastest and most costly form of storage
  - volatile
- **Main memory (主存/内存)**
  - fast access (about 100 nanoseconds, 1 nanosecond = 10^{-9} seconds)
  - generally too small (or too expensive) to store the entire database
    - capacities of up to hundreds of Gigabytes widely used currently
    - capacities have gone up and per-byte cost has decreased steadily
  - Volatile, contents of main memory are lost if a power failure or system crash occurs





Flash Memory

- **Flash memory (闪存)**
  - Data survives power failure
  - Combining the advantages of RAM & ROM, while falling between RAM and magnetic disk
  - Data can be written at a location only once, but the location can be erased and written again
    - support 10K – 1M write/erase times
  - Reads are roughly as fast as main memory, but writes are slow (few microseconds)
  - Cost per unit of storage roughly similar to main memory
  - Widely used in embedded devices such as digital cameras
  - USB, SSD (Solid State Drives, 固态硬盘)

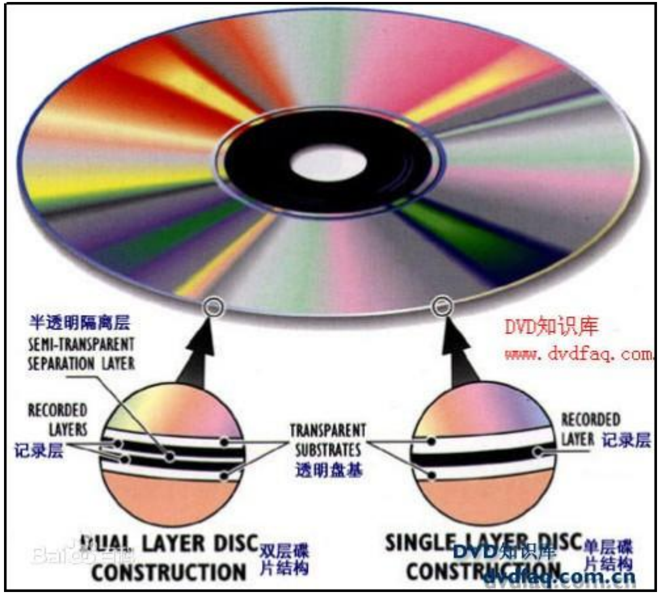


Magnetic Disk

- **Magnetic disk (磁盘)**
  - Primary medium for the long-term storage of data
    - Typically stores the entire database
  - Data must be moved from disk to main memory for access, and written back for storage (I/O)
    - Much slower access than main memory
  - Capacities range up to roughly several hundreds of TBs currently
    - Much larger capacity and lower cost/byte than main/flash memory
    - Growing constantly and rapidly with technology improvements (factor of 2 to 3 every 2 years)
  - Survives power failures and system crashes
    - disk failure can destroy data, but is rare

Optical Storage

- **Optical storage (光学存储器)**
  - Non-volatile, data is read optically from a spinning disk using a laser
  - CD-ROM (640 MB) and DVD (4.7 to 17 GB) are the most popular forms
  - Write-one, read-many (WORM) optical disks used for archival storag
    - CD-R, DVD-R, DVD+R
  - Multiple write versions also available
    - CD-RW, DVD-RW, DVD+RW, DVD-RAM
  - Reads and writes are slower than magnetic disk
  - 光盘1965年由美国詹姆斯·拉塞尔(James Russell)发明
  - 光盘分五层：基板、记录层、反射层、保护层、印刷层



Optical Storage (续)

- Optical storage (光学存储器)
  - 2010年5月，日本东京大学化学教授**大越慎一**(Shin-ichi Ohkoshi)研究团队发现一种材料，可以用来制造更便宜、容量大得多的超级光盘，可储存容量是目前**一般DVD的5千倍**
  - 这种材料是一种透明的新型氧化钛，平常是能导电的黑色金属状态，在受到光的点击后会转变成棕色的半导体。在室温下受到光的照射，能够任意在金属和半导体之间转变，因而产生储存数据功能。这种材料所制成的新光碟，容量是蓝光光盘的1千倍，而蓝光光盘的容量则是一般DVD的5倍。一般一张DVD为4.7G，一张蓝光光盘为25G。最新的蓝光协会表示，新蓝光光盘容量可达128G。东京大学的最新光盘容量将达25000G，即25T
  - 2012年富士胶片开发利用双光子吸收热量的新型光盘记录方式，可实现每层25GB的记录密度，与蓝光光盘相同，且该技术有可能实现多达20层的多层化。富士新技术单张盘片容量可达15TB
  - 2021年报道来自上海理工大学、墨尔本理工大学、新加坡国立大学的联合团队使用了一种新的纳米复合材料，将石墨烯氧化物薄片和上转换纳米粒子(UCNPs)结合起来，以达到前所未有的数据密度，实验室阶段已经在**12公分光盘中存储下了700TB的数据，相当于28000张25GB的蓝光盘**

Tape Storage

- **Tape storage (磁带存储器)**
  - **Non-volatile**, used primarily for backup (to recover from disk failure), and for archival data
    - IBM、EMC、Dell…
  - **Sequential-access** – much slower than disk
  - **Very high capacity** (40 to 300 GB tapes available)
  - Can be **removed** from drive
  - Storage cost is much **cheaper** than disk, but the drive is **expensive**
  - 磁带是一种古老存储数据方式，1928年诞生

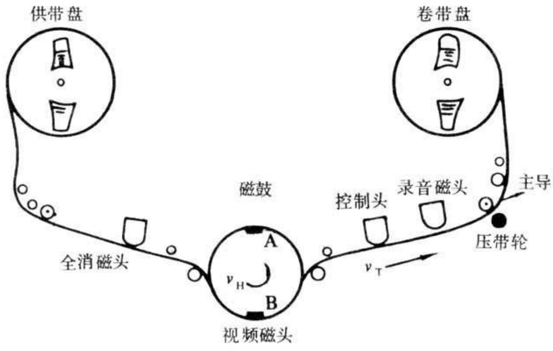


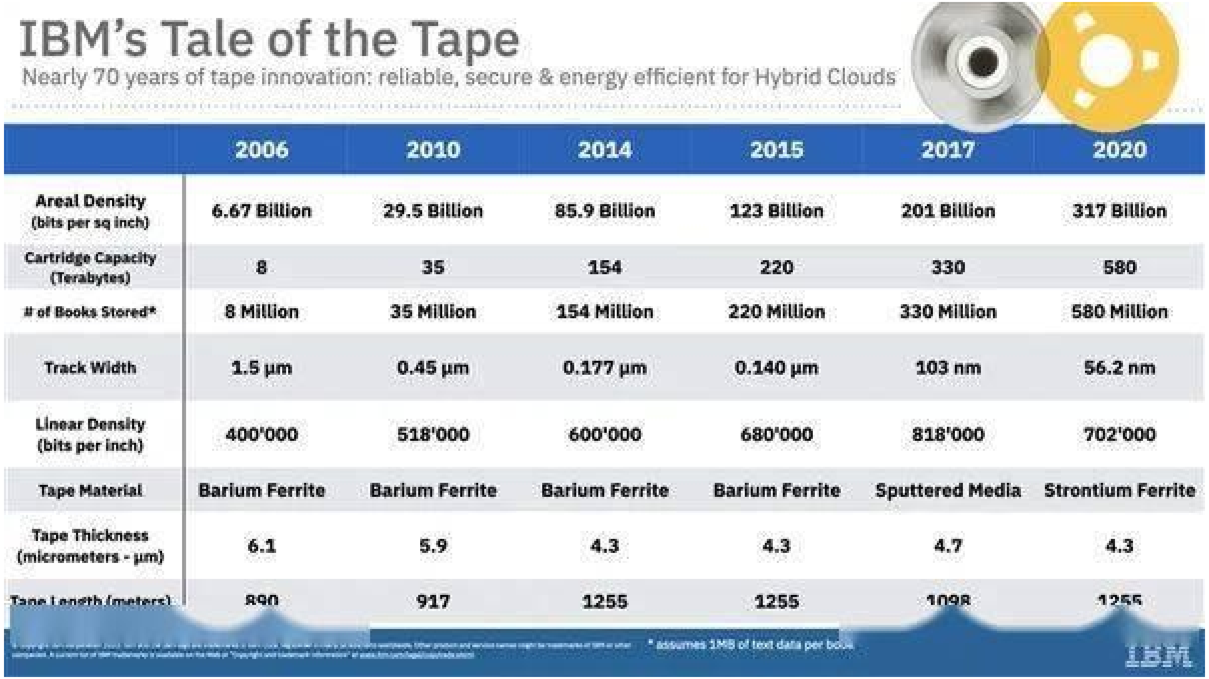
图 1 磁头旋转相对于磁带运动

Tape Storage（续）

- 随着人类数据量爆炸式增长，磁带设备焕发新生，IBM大中华区CTO谢东提出，磁带在未来仍将是主流存储媒介，其容量将会每年增长30%，而传统硬盘增幅只有10%。人类需要保存的很多数据都是冷数据，是不常用的数据，大部分是为了备份、保存，需求量还在不断增长，磁带优势正是存储容量超大、成本低廉、保存时间长等。
- 富士LTO系列磁带，Ultrium 8系列磁带采用BaFe（钡铁氧体磁性材料），并有富士专利的纳米超薄涂层技术，磁带长度960m，宽度12.65mm，厚度5.6um。LTO Ultrium 8系列磁带容量高达30TB（未压缩时是12TB），是前代产品的两倍，而且速度可达750MB/s（未压缩时是360MB/s），性能及容量都要比HDD硬盘要有优势，适合长期保存重要数据。
- LTO 8系列磁带有两种类型，一种是可以重复擦写数据的，另外一种是WROM，写入一次，多次读取类型的，这种类型的可以防止数据被篡改或者意外删除，提高了安全性。

Tape Storage（续）

- 2020年12月16日，IBM与富士胶片(Fujifilm)联合开发了新的锶铁氧体(SrFe)材料，用于LTO-8级磁带，存储密度达每平方英寸317Gb，密度虽不大，但磁带可非常长，这次就达到1255米，因此总容量达到了惊人的**580TB**，假设每本书容量1MB，那就可以保存5.8亿本书。这种新型磁带的厚度为4.3微米，数据轨道宽度缩窄到了56.2纳米。



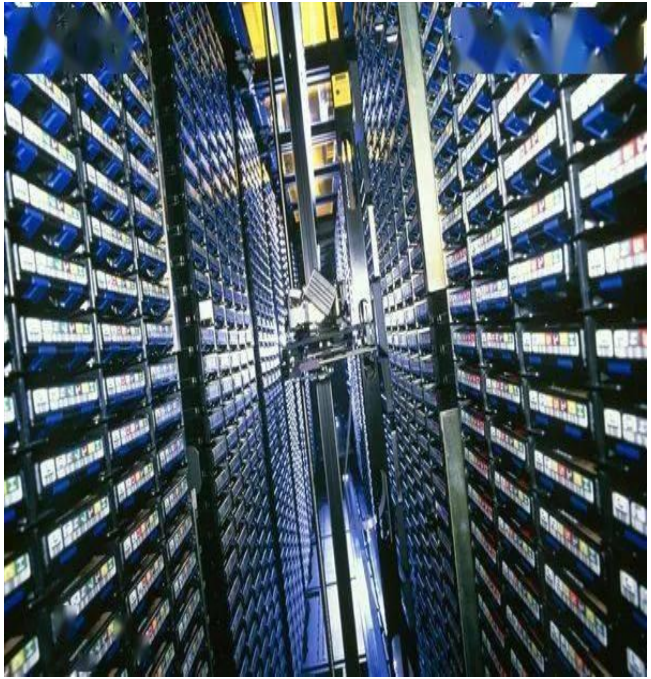
IBM's Tale of the Tape

Nearly 70 years of tape innovation: reliable, secure & energy efficient for Hybrid Clouds

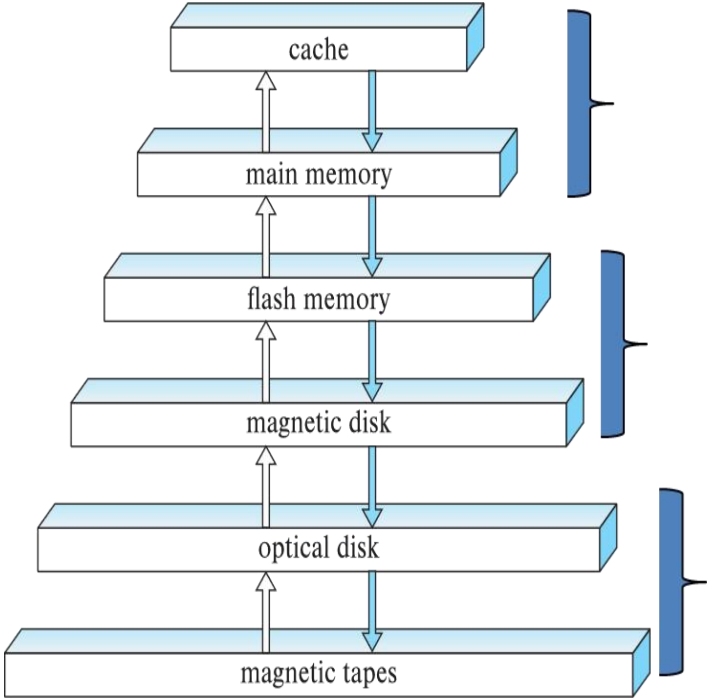
	2006	2010	2014	2015	2017	2020
Areal Density (bits per sq inch)	6.67 Billion	29.5 Billion	85.9 Billion	123 Billion	201 Billion	317 Billion
Cartridge Capacity (Terabytes)	8	35	154	220	330	580
# of Books Stored*	8 Million	35 Million	154 Million	220 Million	330 Million	580 Million
Track Width	1.5 µm	0.45 µm	0.177 µm	0.140 µm	103 nm	56.2 nm
Linear Density (bits per inch)	400'000	518'000	600'000	680'000	818'000	702'000
Tape Material	Barium Ferrite	Barium Ferrite	Barium Ferrite	Barium Ferrite	Sputtered Media	Strontium Ferrite
Tape Thickness (micrometers - µm)	6.1	5.9	4.3	4.3	4.7	4.3
Tape Length (meters)	890	917	1255	1255	1098	1255

[illegible]

* assumes 1MB of text data per book



Storage Hierarchy

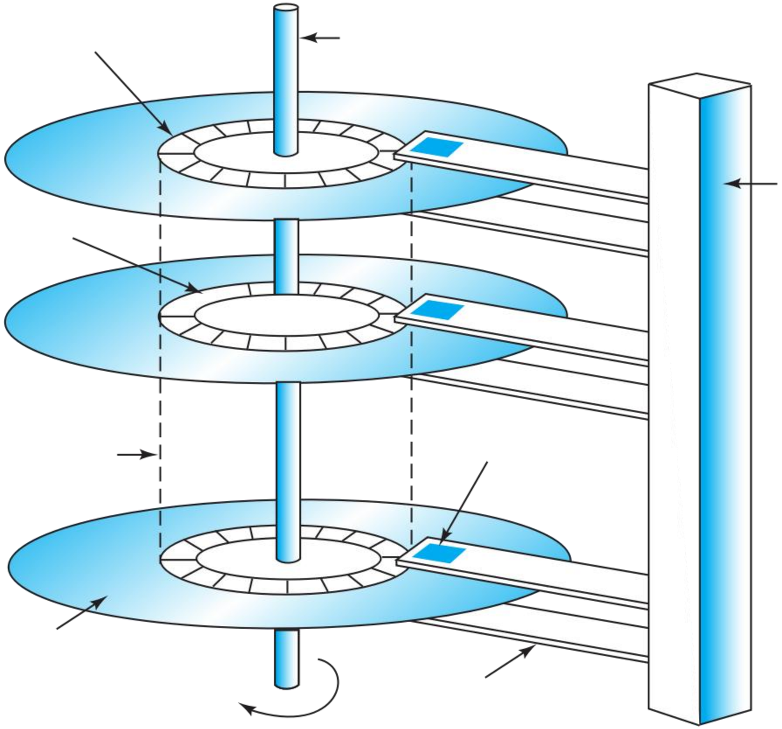


- **Primary storage**
  - Fastest media but volatile (cache, main memory)
- **Secondary storage**
  - Non-volatile, moderately fast access time
  - Also called on-line storage, e.g., flash memory, magnetic disks
- **Tertiary storage**
  - Non-volatile, slow access time
  - Also called off-line storage, e.g., magnetic tape, optical storage

Outline

- Overview of Physical Storage Media
- ☞ **Magnetic Disks**
- RAID
- Tertiary Storage
- Storage Access
- File Organization
- Organization of Records in Files
- Data-Dictionary Storage

Magnetic Hard Disk Mechanism



Schematic diagram of magnetic disk drive

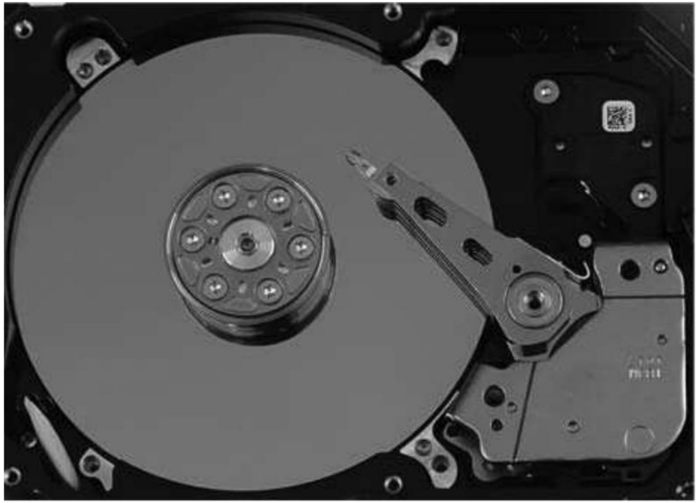


Photo of magnetic disk drive

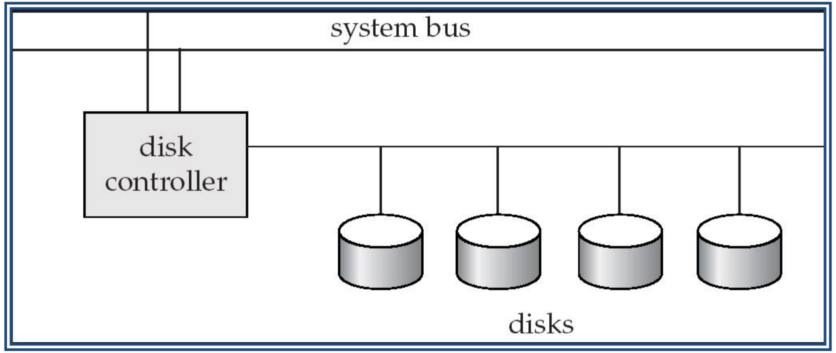
Magnetic Disks

- Read-write head (读写头)
  - positioned very close to the platter surface (almost touching it)
  - reads or writes magnetically encoded information
- Surface of platter is divided into circular tracks (磁道)
  - over 50K-100K tracks per platter on typical hard disks
- Each track is divided into sectors (扇区)
  - a sector is the smallest unit of data that can be read or written
  - sector size is typically **512 bytes**
  - typical sectors per track: 500-1000 (on inner tracks), 1000-2000 (on outer tracks)
- To read/write a sector
  - disk arm swings to position head on right track
  - platter spins continually; data is read/written as sector passes under head
- Head-disk assemblies (磁头-磁盘装置)
  - multiple disk platters on a single spindle (1 to 5 usually)
  - one head per platter, mounted on a common arm.
- Cylinder(柱面) i consists of the i-th tracks of all the platters(盘片)

Magnetic Disks (Cont.)

- **Earlier generation disks were susceptible to head-crashes**
  - Surface of earlier generation disks had metal-oxide coatings (金属氧化物涂层) which would disintegrate on head crash and damage all data on disk
  - Current disks are less susceptible to such disastrous failures
- **Disk controller**(磁盘控制器) – interfaces between the computer system and the disk drive hardware
  - Accept high-level commands to read or write a sector
  - Initiate actions such as moving the disk arm to the right track and actually reading or writing the data
  - Compute and attach checksums(校验和) to each sector to verify that data is read back correctly.
  - Performs remapping of bad sectors(坏扇区重映射)

Disk Subsystem



- **Multiple disks connected to a computer system through a controller**
  - Controller functionality (checksum, bad sector remapping) is often carried out by individual disks to reduce the load on controller
- **Disk interface standards families**
  - ATA (AT adaptor/attachment) range of standards (1994-2002,7 standards)
  - SATA (Serial ATA)、PATA (Parallel ATA)
  - SCSI (Small Computer System Interconnect) range of standards
  - Several variants of each standard (different speeds and capabilities)

Performance Measures of Disks

- **Access time (访问时间)**
  - the time it takes from when a read or write request is issued to when data transfer begins, including
    - **Seek time(寻道时间)** – time it takes to reposition the arm over the correct track.
      - Average seek time is about 1/2 the worst case seek time
      - 4 to 10 milliseconds 毫秒 on typical disks
    - **Rotational latency(旋转等待)** – time it takes for the sector to be accessed to appear under the head
      - Average latency is 1/2 of the worst case latency.
      - 4 to 11 milliseconds on typical disks (5400 to 15000 rpm)

Performance Measures (Cont.)

- **Data-transfer rate** (数据传输率)
  - the rate at which data can be retrieved from or stored to the disk
  - Max rate: 25 to 100 MB per second, lower for inner tracks
  - Multiple disks may share a controller, so rate that controller can handle is also important
    - E.g., ATA-5: 66 MB/sec, SATA: 150 MB/sec, Ultra 320 SCSI: 320 MB/s
    - Fiber Channel (FC2Gb): 256 MB/s

Performance Measures (Cont.)

- **Mean time to failure** (平均故障时间, MTTF)
  - the average time the disk is expected to run continuously without any failure, typically 3 to 5 years
  - Probability of failure of new disks is quite low, corresponding to a "theoretical MTTF" of 500,000 (≈57 years) to 1,200,000 (≈137 years) hours for a new disk
    - E.g., an MTTF of 1,200,000 hours for a new disk means that given 1000 new disks, one will fail every 1200 hours on the average
    - 1/ (1- (1-1/1,200,000)^1000)≈1,200,000/1000=1200
  - MTTF decreases as disk ages

Optimization of Disk-block Access

- **Block** – **a contiguous sequence of sectors from a single track**
  - data is transferred between disk and main memory in blocks
  - sizes range from 512 bytes to several kilobytes
    - small blocks: more transfers from disk
    - large blocks: more space wasted due to partially filled blocks
    - typical block sizes range from **4 to 16 kilobytes**
- **Disk-arm-scheduling** (磁盘臂调度) algorithms order pending accesses to tracks so that disk arm movement is minimized
  - **elevator algorithm (电梯算法)** : move disk arm in one direction (from outer to inner tracks or vice versa), processing next request in that direction, till no more requests in that direction, then reverse direction and repeat

Optimization of Disk Block Access (续)

- **File organization** – optimize block access time by organizing the blocks according to the way of data access
  - E.g., store related information on the same or nearby cylinders
  - Files may get fragmented(碎片化) over time
    - data is inserted to/deleted from the file
    - free blocks on disk are scattered, and newly created file has its blocks scattered over the disk
    - sequential access to a fragmented file results in increased disk arm movement
  - Some systems have utilities to defragment the file system (去除文件碎片), in order to speed up file access

Optimization of Disk Block Access (Cont.)

- **Non-volatile write buffers** speed up disk writes by writing blocks to a non-volatile RAM buffer immediately
  - Non-volatile RAM: battery backed up RAM or flash memory
    - Even if power fails, the data is safe and will be written to disk when power returns
  - Controller writes to disk when the disk has no other requests or the non-volatile RAM is full
  - Database operations can continue without waiting for data to be written to disk
  - Writes can be reordered to minimize disk arm movement

Optimization of Disk Block Access (Cont.)

- **Log disk (日志磁盘)**
  - A disk devoted to writing a sequential log of block updates
  - Used exactly like nonvolatile RAM
    - Write to log disk is very fast since seeking is not required
    - No need for special hardware (NV-RAM)

- File systems typically reorder writes to disk to improve performance
  - Journaling file systems (日志文件系统) write data in safe order to NV-RAM or log disk
  - Reordering without journaling: risk of corruption of file system data

Outline

- Overview of Physical Storage Media
- Magnetic Disks

☞ **RAID**

- Tertiary Storage
- Storage Access
- File Organization
- Organization of Records in Files
- Data-Dictionary Storage

RAID（独立冗余磁盘阵列）

- **RAID: Redundant Arrays of Independent Disks**
  - Disk organization techniques that manage a large numbers of disks, providing a view of a single disk of
    - High capacity and high speed by using multiple disks in parallel
    - High reliability by storing data redundantly. Data can be recovered even if a disk fails

RAID (Cont.)

- The chance that some disk out of a set of N disks will fail is much higher than the chance that a specific single disk will fail
  - E.g., a system with 100 disks, each with MTTF of 100,000 hours (approx. 11 years), will have a system MTTF of 1000 hours (around 41 days)
  - 1/(1-(1-1/100,000)^100)
  - Techniques for using redundancy to avoid data loss are critical with large numbers of disks

- **Originally a cost-effective alternative to large, expensive disks**
  - I in RAID originally stood for "inexpensive"
  - Now the "I" is interpreted as independent
  - **RAID(独立冗余磁盘阵列)**

Improvement of Reliability via Redundancy

- **Redundancy** – store extra information that can be used to rebuild information lost in a disk failure
  - **Mirroring (镜像):** Duplicate every disk, and each logical disk consists of two physical disks
  - Every write is carried out on both disks. Reads can take place from either disk
  - If one disk in a pair fails, data still available in the other
    - Data loss would occur only if a disk fails, and its mirror disk also fails before the system is repaired
    - Probability of combined event is very small except for dependent failure modes such as fire or building collapse or electrical power surges
- **Mean time to data loss (平均数据丢失时间)** depends on **mean time to failure (平均故障时间)** and **mean time to repair (平均修复时间)**
  - MTDL=MTF^2/(2*MTR) (why?)

Improvement in Performance via Parallelism

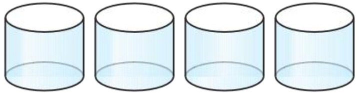
- Two main goals of **parallelism** in a disk system
  - Load balance multiple small accesses to increase throughput
  - Parallelize large accesses to reduce response time
- **Bit-level striping (位级拆分):** split the bits of each byte across multiple disks
  - In an array of eight disks, write bit i of each byte to disk i
  - Each access can read data at eight times the rate of a single disk
- **Block-level striping (块级拆分):** with n disks, block i of a file goes to disk $(i \bmod n) + 1$
  - Requests for different blocks can run in parallel if the blocks reside on different disks
  - A request for a long sequence of blocks can utilize all disks in parallel

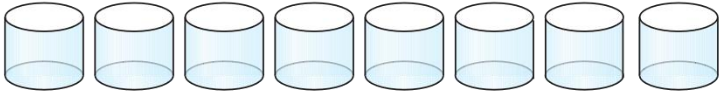
RAID Levels

- Schemes to provide redundancy at lower cost by using **disk striping (磁盘拆分)** combined with **parity bits(奇偶校验位)**
  - Different RAID organizations, or RAID levels, have differing cost, performance and reliability characteristics

- RAID Level 0: Block striping, non-redundant. (无冗余拆分)
  - **Used in high-performance applications where data lose is not critical**

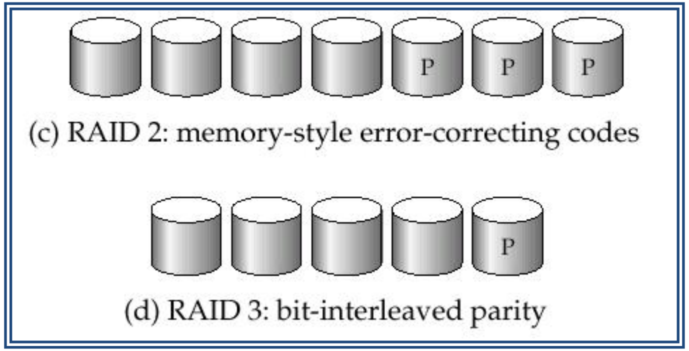
- RAID Level 1: Mirrored disks with block striping (镜像磁盘)
  - **Offers best write performance.**
  - **Popular for applications such as storing log files in a database system**





RAID Levels (Cont.)

- RAID Level 2: Memory-Style Error-Correcting-Codes (ECC) with bit striping. (内存风格的纠错码)
- RAID Level 3: Bit-Interleaved Parity (位交叉的奇偶校验)
  - a single parity bit is enough for error correction, not just detection, since we know which disk has failed
    - When writing data, corresponding parity bits must also be computed and written to a parity bit disk
    - To recover data in a damaged disk, compute XOR of bits from other disks (including parity bit disk)

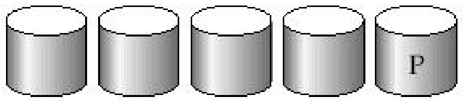


(c) RAID 2: memory-style error-correcting codes

(d) RAID 3: bit-interleaved parity

RAID Levels (Cont.)

- **RAID Level 3 (Cont.):** Bit-Interleaved Parity (位交叉的奇偶校验)
  - Faster data transfer than with a single disk, but fewer I/Os per second since every disk has to participate in every I/O.
  - Subsumes Level 2 (provides all its benefits, at lower cost).
- **RAID Level 4:** Block-Interleaved Parity(块交叉的奇偶校验)
  - uses block-level striping, and keeps a parity block on a separate disk for corresponding blocks from N other disks.
  - When writing data block, corresponding block of parity bits must also be computed and written to parity disk
  - To find value of a damaged block, compute XOR of bits from corresponding blocks (including parity block) from other disks.



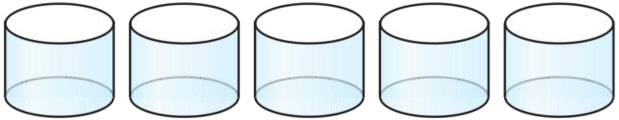
(e) RAID 4: block-interleaved parity

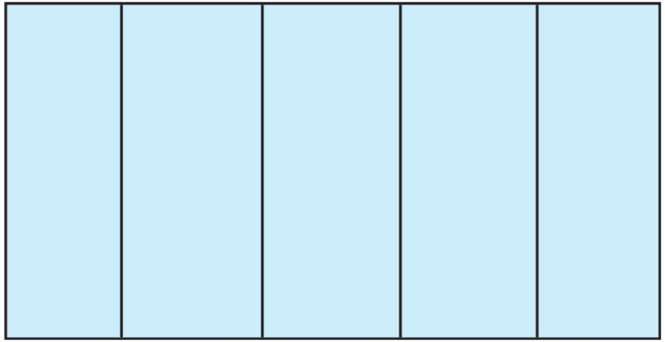
RAID Levels (Cont.)

- RAID Level 4 (Cont.)：Block-Interleaved Parity(块交叉的奇偶校验)
  - Provides higher I/O rates for independent block reads than Level 3
    - block read goes to a single disk, so blocks stored on different disks can be read in parallel
  - Provides high transfer rates for reads of multiple blocks than no-striping
  - Before writing a block, parity data must be computed
    - Can be done by using old parity block, old value of current block and new value of current block (2 block reads + 2 block writes)
      - More efficient for writing large amounts of data sequentially
  - Parity block becomes a bottleneck for independent block writes since every block write also writes to parity disk

RAID Levels (Cont.)

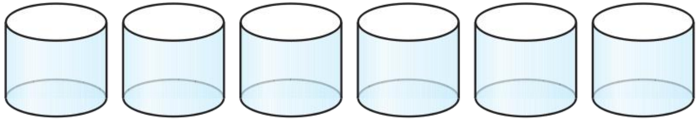
- **RAID Level 5**: Block-Interleaved Distributed Parity(块交叉的分布奇偶校验)
  - partitions data and parity among all N + 1 disks, rather than storing data in N disks and parity in 1 disk.
  - E.g., with 5 disks, parity block for nth set of blocks is stored on disk (n mod 5) + 1, with the data blocks stored on the other 4 disks.



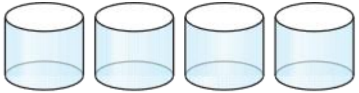


RAID Levels (Cont.)

- **RAID Level 5 (Cont.)** Block-Interleaved Distributed Parity(块交叉的分布奇偶校验)
  - Higher I/O rates than Level 4.
    - Block writes occur in parallel if the blocks and their parity blocks are on different disks.
  - Subsumes Level 4: provides same benefits, but avoids bottleneck of parity disk.
- **RAID Level 6:** P+Q Redundancy scheme
  - similar to Level 5, but stores extra redundant information to guard against multiple disk failures.
  - Better reliability than Level 5 at a higher cost; not used as widely.

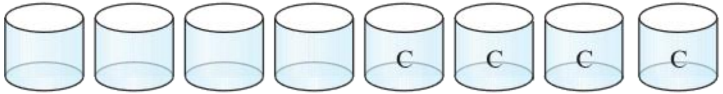


A Comparison of Different Levels



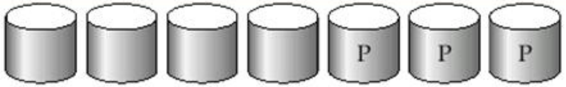
RAID 0: nonredundant striping

Block-stripping

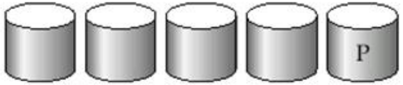


RAID 1: mirrored disks

Block-stripping

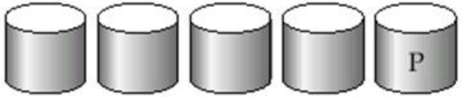


RAID 2: memory-style error-correcting codes

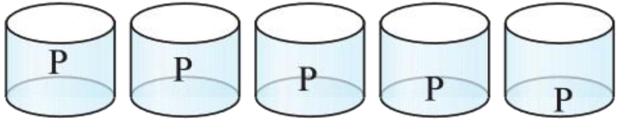


RAID 3: bit-interleaved parity

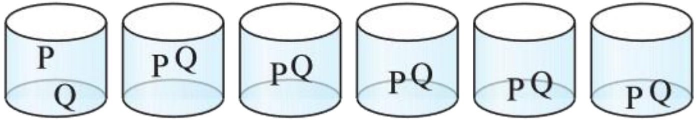
Bit-stripping



RAID 4: block-interleaved parity



RAID 5: block-interleaved distributed parity



RAID 6: P + Q redundancy

Choice of RAID Level

- Factors in choosing RAID level
  - Monetary cost
  - Performance: Number of I/O operations per second, and bandwidth during normal operation
  - Performance during failure
  - Performance during rebuild of failed disk
    - Including time taken to rebuild failed disk
- RAID 0 is used only when data safety is not important
  - E.g. data can be recovered quickly from other sources
- Level 2 and 4 never used since they are subsumed by 3 and 5
- Level 3 is not used anymore since bit-striping forces single block reads to access all disks, wasting disk arm movement, which **block striping (level 5)** avoids
- Level 6 is rarely used since levels **1** and **5** offer adequate safety for almost all applications
- So competition is **between 1 and 5** only

Choice of RAID Level (Cont.)

- **Level 1** provides much better write performance than **level 5**
  - Level 5 requires at least 2 block reads and 2 block writes to write a single block, whereas Level 1 only requires 2 block writes
  - Level 1 preferred for high update environments such as log disks

- **Level 1** had higher storage cost than **level 5**
  - Disk drive capacities increasing rapidly (50%/year) whereas disk access times have decreased much less (x 3 in 10 years)
  - I/O requirements have increased greatly, e.g. for Web servers
  - When enough disks have been bought to satisfy required rate of I/O, they often have spare storage capacity, so there is often no extra monetary cost for Level 1!

- **Level 5** is preferred for applications with **low update rate**, and **large amounts of data**
- **Level 1** is preferred for all other applications

Hardware Issues

- **Software RAID:** RAID implementations done entirely in software, with no special hardware support
- **Hardware RAID:** RAID implementations with special hardware
  - Use non-volatile RAM to record writes that are being executed
  - Beware: power failure during write can result in corrupted disk
    - E.g., failure after writing one block but before writing the second in a mirrored system
    - Such corrupted data must be detected when power is restored
      - Recovery from corruption is similar to recovery from failed disk
      - NV-RAM helps to efficiently detect potentially corrupted blocks
        - » Otherwise all blocks of disk must be read and compared with mirror/parity block

Hardware Issues (Cont.)

- **Hot swapping(热交换):** replacement of disk while system is running, without power down
  - Supported by some hardware RAID systems
  - reduces time to recovery, and improves availability (可用性) greatly
- Many systems maintain **spare disks** which are kept online, and used as replacements for failed disks immediately on detection of failure
  - Reduces time to recovery greatly
- Many **hardware RAID systems** ensure that a single point of failure will not stop the functioning of the system by using
  - Redundant power supplies with battery backup
  - Multiple controllers and multiple interconnections to guard against controller/ interconnection failures

Outline

- Overview of Physical Storage Media
- Magnetic Disks
- RAID

☞ **Tertiary Storage**

- Storage Access
- File Organization
- Organization of Records in Files
- Data-Dictionary Storage

Optical Disks

- **Compact disk-read only memory (CD-ROM)**
  - Removable disks, 640 MB per disk
  - Seek time about 100 msec (optical read head is heavier and slower)
  - Higher latency (3000 RPM) and lower data-transfer rates (3-6 MB/s) compared to magnetic disks
- **Digital Video Disk (DVD)**
  - DVD-5 holds 4.7 GB , and DVD-9 holds 8.5 GB
  - DVD-10 and DVD-18 are double sided formats with capacities of 9.4 GB and 17 GB
  - Slow seek time, for same reasons as CD-ROM
- **Record once versions (CD-R,DVD-R)** are popular
  - Data can only be written once, and cannot be erased
  - High capacity and long lifetime; used for archival storage
- **Multi-write versions (CD-RW,DVD-RW,DVD+RW,DVD-RAM)** also available

Magnetic Tapes

- Hold **large volumes of data** and provide **high transfer rates**
  - Few GB for DAT (Digital Audio Tape) format
  - 10-40 GB with DLT (Digital Linear Tape) format
  - 100 GB+ with Ultrium format
  - 330 GB with Ampex helical scan format
  - Transfer rates from few to 10s of MB/s

- Currently the **cheapest storage medium**
  - Tapes are cheap, but cost of drives is very high

Magnetic Tapes (Cont.)

- Very slow access time in comparison to magnetic disks and optical disks
  - Limited to sequential access.
  - Some formats provide faster seek (10s of seconds) at cost of lower capacity

- Used mainly for backup, for storage of infrequently used information, and as an off-line medium for transferring information from one system to another

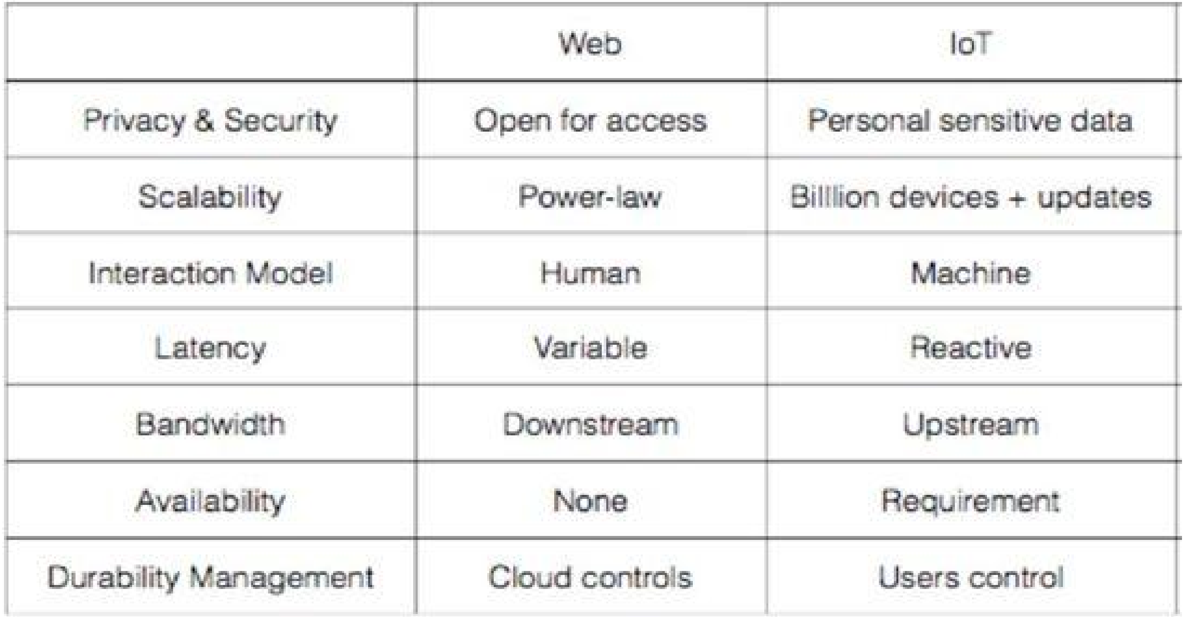
- **Tape jukeboxes(自动磁带机)** used for very large capacity storage
  - terabyte (10^{12} bytes) to petabye (10^{15} bytes)

未来存储技术

- 分布式、云化、闪存化、智能等存储不断发展，随着人工智能、物联网、区块链、人体增强 (Human Augmentation) 等技术快速发展，存储也将迎来新的形态，如：
  - **边缘存储**
  - **长期存储**
  - **生物存储/基因存储**
  - **区块链存储**
  - **量子存储**

边缘存储

- 加州大学伯克利分校的研究团队在2015年发表 “The Cloud is Not Enough: Saving IoT from the Cloud” 文章，指出物联网与互联网的七个不同之处：**隐私和安全；可伸缩性；交互模型；延迟；带宽；可用性；持久性管理**。**边缘存储应该具有高可用性、超低延迟、高安全性、高隐私性、弱一致性、功耗低、空间小的特点**，边缘存储仍处于发展早期，还有很多需研究探索



	Web	IoT
Privacy & Security	Open for access	Personal sensitive data
Scalability	Power-law	Billlion devices + updates
Interaction Model	Human	Machine
Latency	Variable	Reactive
Bandwidth	Downstream	Upstream
Availability	None	Requirement
Durability Management	Cloud controls	Users control

长期存储 - Long Data

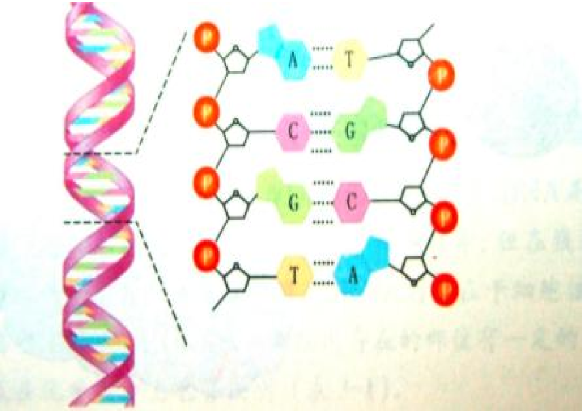
- **蓝光光盘**
  - 2019中国数据与存储峰会，“Long Data”挑战国际上研究课题How to preserve information for 100 years? 基于蓝光技术可提供解决办法，蓝光光盘具超过50年寿命，极低功耗，高安全性，较高密度，较低成本，对环境要求低。总体而言，光存储技术尤其适合大量冷数据的长期安全存储。
  - 针对于需要保存长达数十年的行业或场景，基于蓝光光盘的存储技术是一个选择，如电子档案、医疗影像、金融影像、备份归档等。一个标准机柜可保存1.22 万片光盘，如果采用500GB 光盘，裸容量可高达6PB 。市场上一种M-Disc 的产品，号称千年光盘

- **玻璃光盘**
  - 正在研究的玻璃光盘容量是蓝光光盘的几千倍，能承受超过几百度的高温，寿命能高达上万年甚至百亿年。如微软的玻璃光盘项目ProjectSilica

- **全息光存储**
  - 蓝光之后下一代变革性光存储技术包括两种：第一种是同轴多维全息光存储技术，列入国家重点研发计划。第二种是2014年得了诺贝尔奖的突破光的衍射极限项目，澳大利亚科学家把这个技术用到光上，把光斑从300纳米理论上可以减少到九个纳米，容量上得到巨大提高，至少可达每盘15TB，理想上可实现PB级。

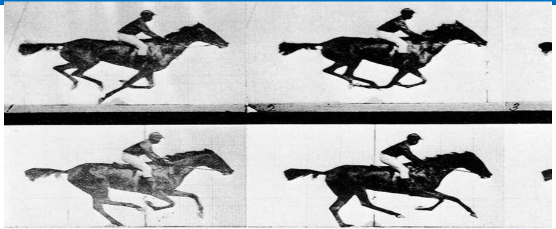
生物存储或基因存储

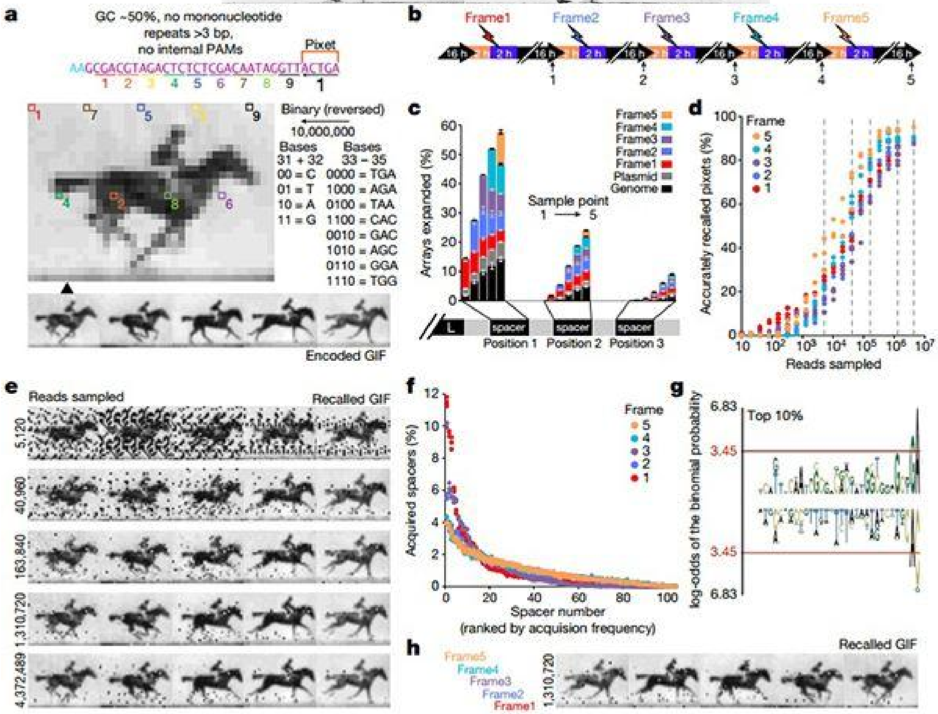
- **生物存储或基因存储**
  - 人类历史上，已经有过历经千百年的存储介质，如青铜器铭文、竹简、纸书、石碑等。不过，这类手段存储密度极低。**《自私的基因》**一书观点：包括人在内的动植物不过是DNA繁衍的躯壳，你也可以看成是DNA 的存储器。
  - 我们可以反向让DNA 片段，即基因成为人类存放信息的存储器。基因存储将0 、1 数据通过一定的编码方法转换成**DNA 中的A 、T 、C 、G 四种碱基**，通过合成含有这些碱基序列的DNA 即可实现数据信息存储。 (DNA是由四种碱基组成的螺旋结构, DNA(脱氧核糖核酸)分子由两条很长的糖链构成骨架，通过碱基对结合在一起，象梯子一样, 整个分子环绕自身中轴形成一个双螺旋。4种碱基, 根据它们英文名称首字母分别为A(*ADENINE* 腺嘌呤)、T(*THYMINE* 胸腺嘧啶)、G(*GUANINE* 鸟嘌呤)、C(*CYTOSINE* 胞嘧啶)。每种碱基分别与另一种碱基的化学性质完全互补，A总与T配对，G总与C配对, 转录时A与U(URACIL尿嘧啶)配,.(RNA核糖核酸: AUGC; DNA: ATCG)



生物存储或基因存储

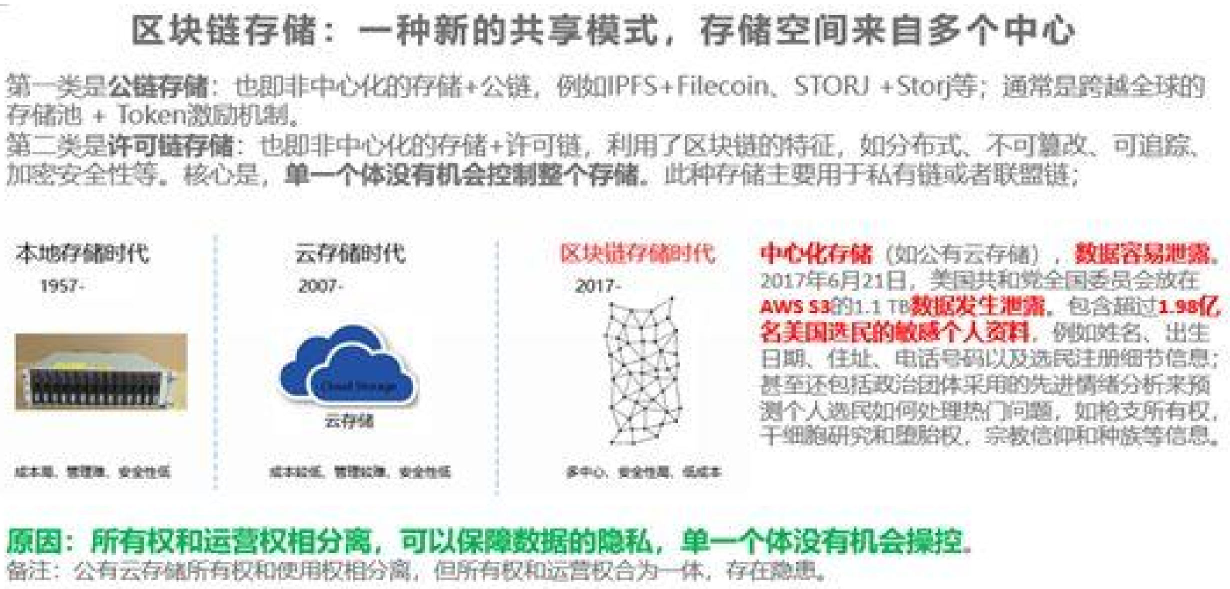
- **生物存储或基因存储**
  - 2017年7月，哈佛大学医学院利用CRISPR DNA编辑技术，将赛马视频录入大肠杆菌的基因组中，并以超过90%的准确率读取出来。研究人员在一群大肠杆菌中存储了一张gif图：一个人骑马狂奔, 其编码、存储和读取信息的过程如右图。
  - 2019年7月，美国布朗大学研究人员用基于生物小分子的存储系统累计存储了超过10万比特的数字图像信息，从中获得图像的准确率可达98% 以上。
  - 2020年2月，华尔街日报报道，波士顿初创公司Catalog Technologies 正在开发一种独特的方式来存储大量数据，将维基百科中14GB 数据存储在DNA 分子中，像“在试管中滴几滴水”。用分子生物学仪器“打印”合成分子序列，以DNA形式存储和表示数字信息，使用DNA测序仪和专有软件的来读取信息，软件可以将分子翻译成文本、图片甚至视频。





区块链存储

- 如何在激励数据提供方分享数据的同时，又能保护隐私？**区块链和存储的结合**
    - **区块链**解决了数据确权、激励分享、数据资产交易和流转等问题。但区块链要发展，区块链基础设施要先行，目前区块链基础设施还处于非常早期，使得区块链应用数据的存储，具有很大的问题。无论是公有链还是联盟链，数据都是存放在中心化的存储上，前者可能选择公有云存储，后者选择类似NAS存储，都存在安全隐患、隐私泄露的可能性。实际上，去中心化的应用DApp （Distributed App）需要的是端到端的去中心化的基础设施，包括去中心化方式组织的存储。



区块链存储：一种新的共享模式，存储空间来自多个中心

第一类是**公链存储**：也即非中心化的存储+公链，例如IPFS+Filecoin、STORJ +Storj等；通常是跨越全球的存储池 + Token激励机制。

第二类是**许可链存储**：也即非中心化的存储+许可链，利用了区块链的特征，如分布式、不可篡改、可追踪、加密安全性等。核心是，**单一个体没有机会控制整个存储**。此种存储主要用于私有链或者联盟链；

本地存储时代
1957-
成本高、管理难、安全性低
云存储时代
2007-
Cloud Storage
云存储
成本较低、管理较难、安全性低
区块链存储时代
2017-
多中心、安全性高、低成本

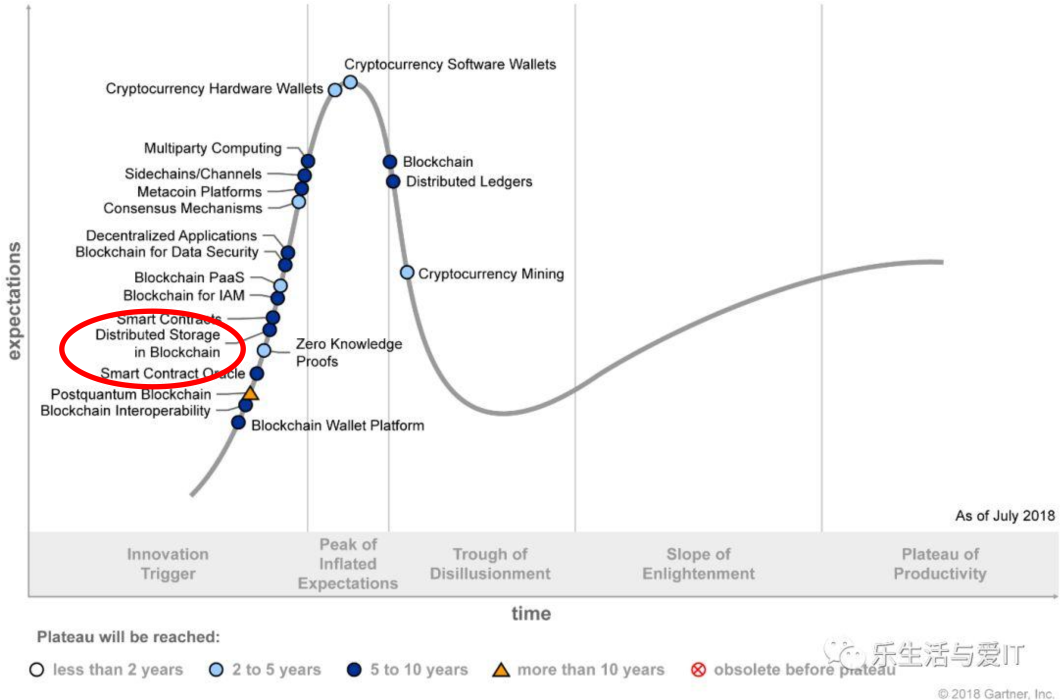
中心化存储（如公有云存储），**数据容易泄露**。2017年6月21日，美国共和党全国委员会放在AWS S3的1.1 TB**数据发生泄露**，包含超过**1.98亿名美国选民的敏感个人资料**，例如姓名、出生日期、住址、电话号码以及选民注册细节信息；甚至还包括政治团体采用的先进情绪分析来预测个人选民如何处理热门问题，如枪支所有权，干细胞研究和堕胎权，宗教信仰和种族等信息。

原因：所有权和运营权相分离，可以保障数据的隐私，单一个体没有机会操控。

备注：公有云存储所有权和使用权相分离，但所有权和运营权合为一体，存在隐患。

区块链存储

- **区块链和存储的结合**
  - 2018年7月，Gartner技术成熟度曲线中区块链存储(Distributed Storage in Blockchain)列入科技诞生促动期(Technology Trigger)，预计2023~2028年进入到成熟应用的技术阶段，将有大量主流用户开始接纳



Outline

- Overview of Physical Storage Media
- Magnetic Disks
- RAID
- Tertiary Storage

☞ **Storage Access**

- File Organization
- Organization of Records in Files
- Data-Dictionary Storage

Storage Access

- **Storage of database file**
  - partitioned into fixed-length storage units called **blocks**. Blocks are units of both storage allocation and data transfer
  - Database system seeks to minimize the number of block transfers between the disk and memory. We can reduce the number of disk accesses by keeping as many blocks as possible in main memory

- **Buffer（缓存)**
  - The portion of main memory available to store copies of disk blocks
  - **Buffer manager:** subsystem responsible for allocating buffer space in main memory

Buffer Manager

- Programs call the **buffer manager** when they need a block from disk
  - If the block is already in the buffer, buffer manager returns the address of the block in main memory
  - If the block is not in the buffer, the buffer manager
    - **Allocates space** in the buffer for the block
      - **Replacing** (throwing out) some other block, if required, to make space for the new block.
      - Replaced block is **written back** to disk only if it was modified since the most recent time that it was written to/fetched from the disk
    - **Reads the block** from the disk to the buffer, and **returns the address** of the block in main memory to requester

Buffer-Replacement Policies

- **LRU (Least Recently Used) 最近最少使用**
  - Most operating systems **replace the block least recently used**
  - Idea behind LRU – use past patterns of block references as a predictor of future references
  - LRU can be a bad strategy for certain access patterns involving repeated scans of data
    - For example: when computing the join of 2 relations ***r*** and ***s*** by a nested loops

```
for each tuple tr of r do
    for each tuple ts of s do
        if the tuples tr and ts match …
```

Buffer-Replacement Policies (Cont.)

- **Pinned block(被钉住的块)**
  - memory block that is **not allowed to be written back** to disk

- **Toss-immediate (立即丢弃) strategy**
  - free the space occupied by a block **as soon as** the final tuple of that block has been processed

- **Most recently used (MRU) (最近最常使用) strategy**
  - system must **pin the block** currently being processed. After the final tuple of that block has been processed, the block is **unpinned**, and it becomes the most recently used block

Outline

- Overview of Physical Storage Media
- Magnetic Disks
- RAID
- Tertiary Storage
- Storage Access

☞ **File Organization**

- Organization of Records in Files
- Data-Dictionary Storage

File Organization

- **The database is stored as a collection of files.**
- **Each file is a sequence of records.**
- **A record is a sequence of fields.**

- **One approach:**
  - Assume that the record size is fixed
  - Each file has records of one particular type only
  - Different files are used for different relations

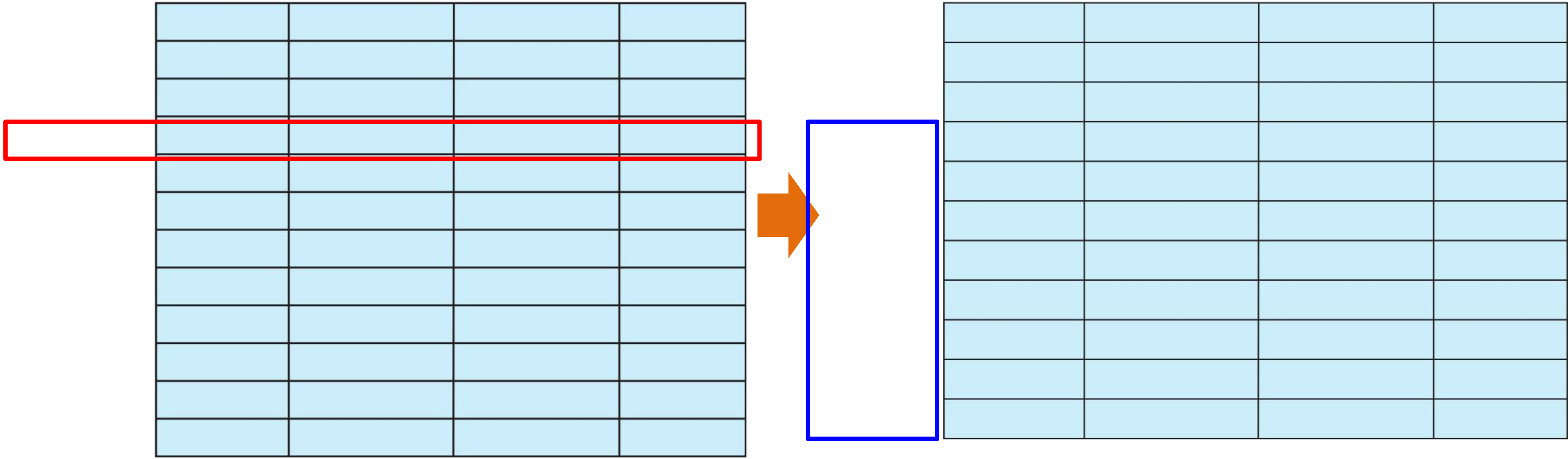
- **Note**: this case is easiest to implement; will consider variable length records later

Fixed-Length Records

- **Simple approach**
  - Store record i starting from byte $n \cdot (i-1)$, where n is the size of each record
  - Record access is simple but records may cross blocks
    - Modification: don't allow records to cross block boundaries

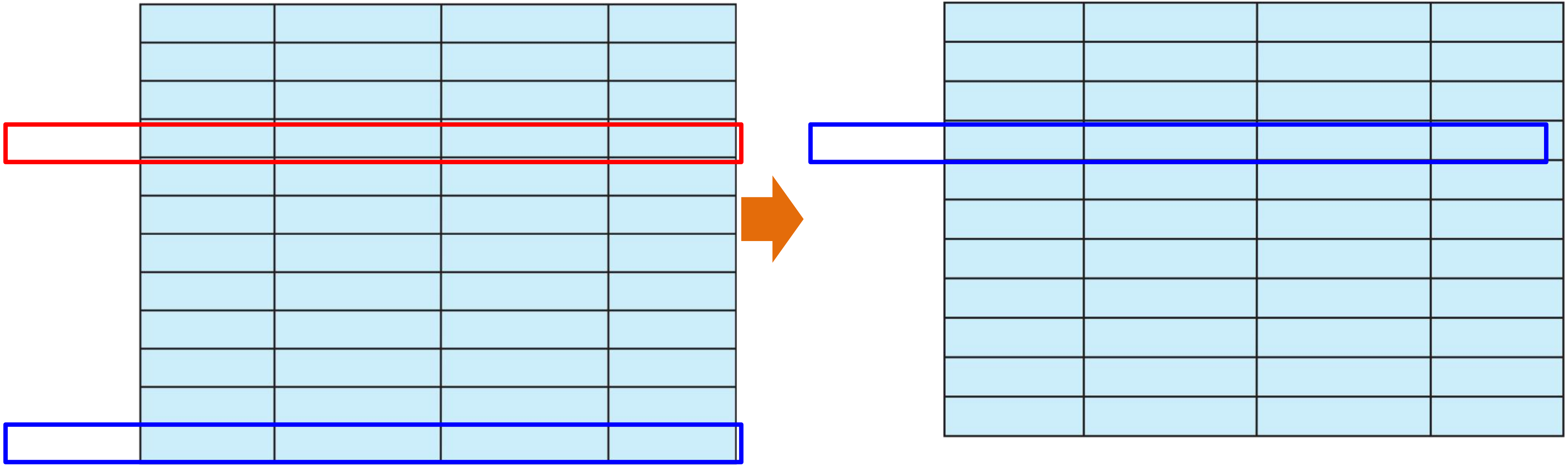
- **Alternative methods for deleting record i**
  - move records $i+1, \ . \ . \ ., n$ to $\mathbf{i}, \ . \ . \ . \ , \mathbf{n-1}$
  - move record n to i
  - do not move records, but link all free records on a free list

With Record 3 Deleted and All Records Moved



move records $i+1, \ . \ . \ ., n$ to i, ... , n − 1

With Record 3 Deleted and Records Moved

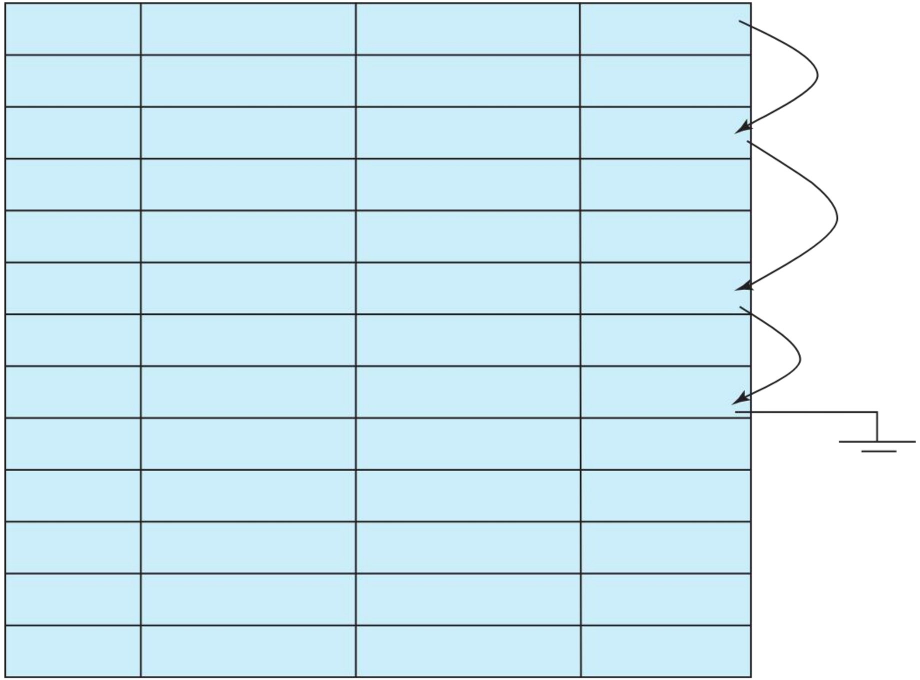


move record n to i

Free Lists (空闲列表)

- Store the **address** of the first deleted record in the **file header**
- Use the first record to store the address of the second deleted record, and so on
- These stored addresses are **pointers** since they “point” to the location of a record

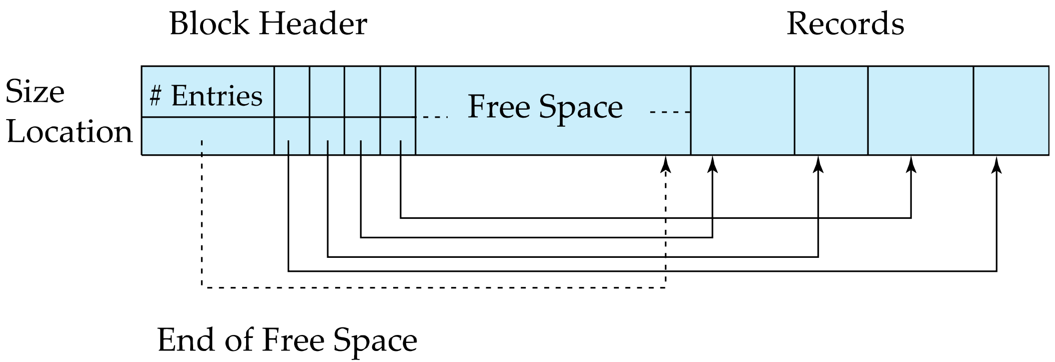
do not move records, but link all free records on a free list



Variable-Length Records (变长记录)

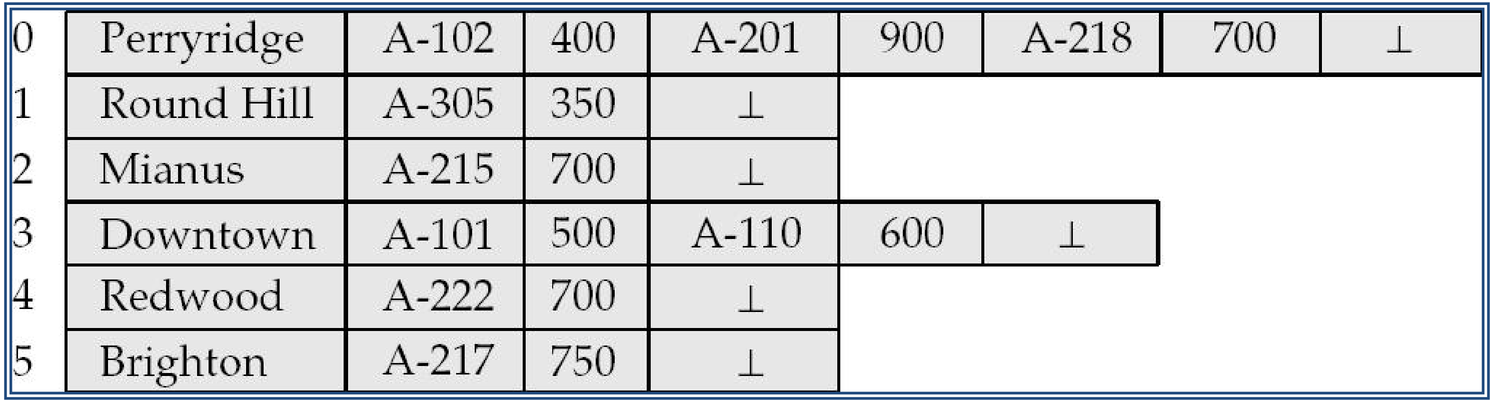
- **Variable-length records**
  - Storage of multiple record types in a file
  - Record types that allow **variable lengths** for one or more fields
  - Record types that allow repeating fields, e.g., array and multiset (used in some old data models)

Variable-Length Records: Slotted Page Structure



- **Slotted page (分槽的页) header** contains
  - **number of record entries**
  - **end of free space in the block**
  - **location and size of each record**
- Records can be moved around within a page to keep them contiguous with no empty space between them; entry in the header must be updated
- **Pointers** should not point directly to record — instead they should point to the entry for the record in header

Byte String Representation of Variable-Length Records



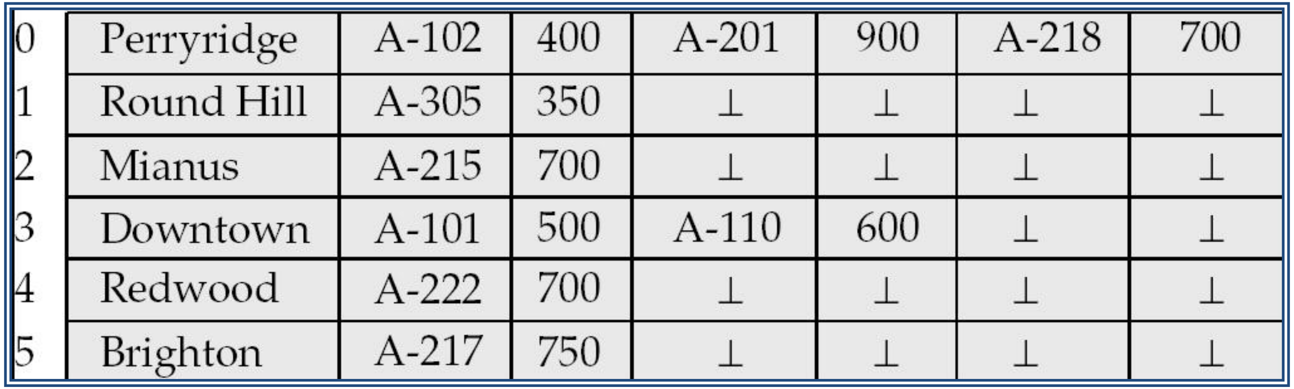
0	Perryridge	A-102	400	A-201	900	A-218	700	⊥
1	Round Hill	A-305	350	⊥				
2	Mianus	A-215	700	⊥				
3	Downtown	A-101	500	A-110	600	⊥		
4	Redwood	A-222	700	⊥				
5	Brighton	A-217	750	⊥				

Byte string representation

- Attach an *end-of-record* (⊥) control character to the end of each record
- Difficulty with deletion
- Difficulty with growth

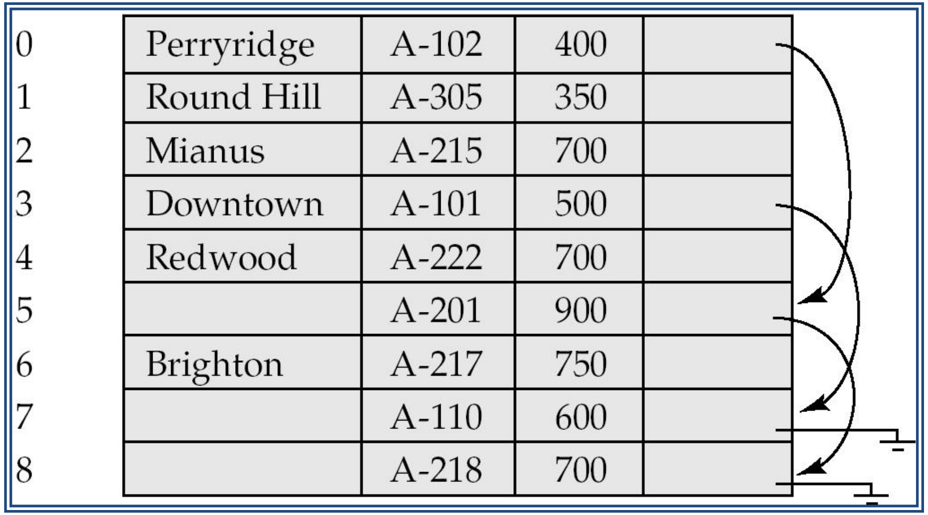
Fixed-Length Representation

- Use one or more **fixed length records**
  - **reserved space**
  - **pointers**
- **Reserved space** – can use fixed-length records of a known maximum length; unused space in shorter records filled with a null or end-of-record symbol.



0	Perryridge	A-102	400	A-201	900	A-218	700
1	Round Hill	A-305	350	⊥	⊥	⊥	⊥
2	Mianus	A-215	700	⊥	⊥	⊥	⊥
3	Downtown	A-101	500	A-110	600	⊥	⊥
4	Redwood	A-222	700	⊥	⊥	⊥	⊥
5	Brighton	A-217	750	⊥	⊥	⊥	⊥

Pointer Method

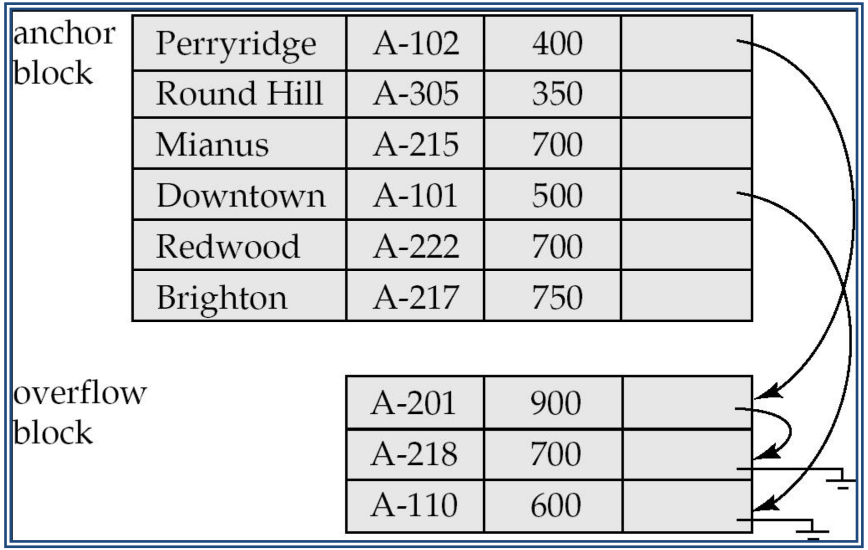


0	Perryridge	A-102	400	
1	Round Hill	A-305	350	
2	Mianus	A-215	700	
3	Downtown	A-101	500	
4	Redwood	A-222	700	
5		A-201	900	
6	Brighton	A-217	750	
7		A-110	600	
8		A-218	700	

- **Pointer method**
  - **A variable-length record** is represented by a list of fixed-length records, chained together via pointers
  - Can be used even if the maximum record length is not known

Pointer Method (Cont.)

- **Disadvantage** to pointer structure; space is wasted in all records except the first in a chain
- Solution is to allow two kinds of block in file:
  - **Anchor block:** contains the first records of chain
  - **Overflow block:** contains records other than those that are the first records of chains



anchor block

Perryridge	A-102	400	
Round Hill	A-305	350	
Mianus	A-215	700	
Downtown	A-101	500	
Redwood	A-222	700	
Brighton	A-217	750	

overflow block

A-201	900	
A-218	700	
A-110	600	

Outline

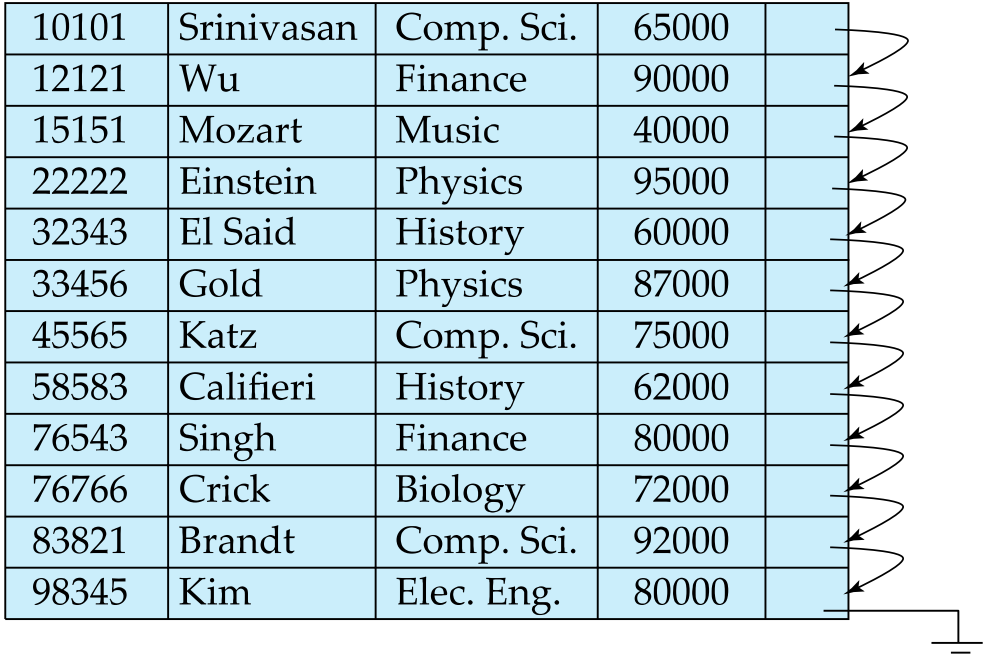
- Overview of Physical Storage Media
- Magnetic Disks
- RAID
- Tertiary Storage
- Storage Access
- File Organization
- ☞ **Organization of Records in Files**
- Data-Dictionary Storage

Organization of Records in Files

- **Heap (堆)**
  - a record can be placed anywhere in the file where there is space
- **Sequential (顺序)**
  - store records in sequential order, based on the value of the search key of each record
- **Hashing (散列)**
  - a hash function computed on some attribute of each record; the result specifies in which block of the file the record should be placed

Sequential File Organization

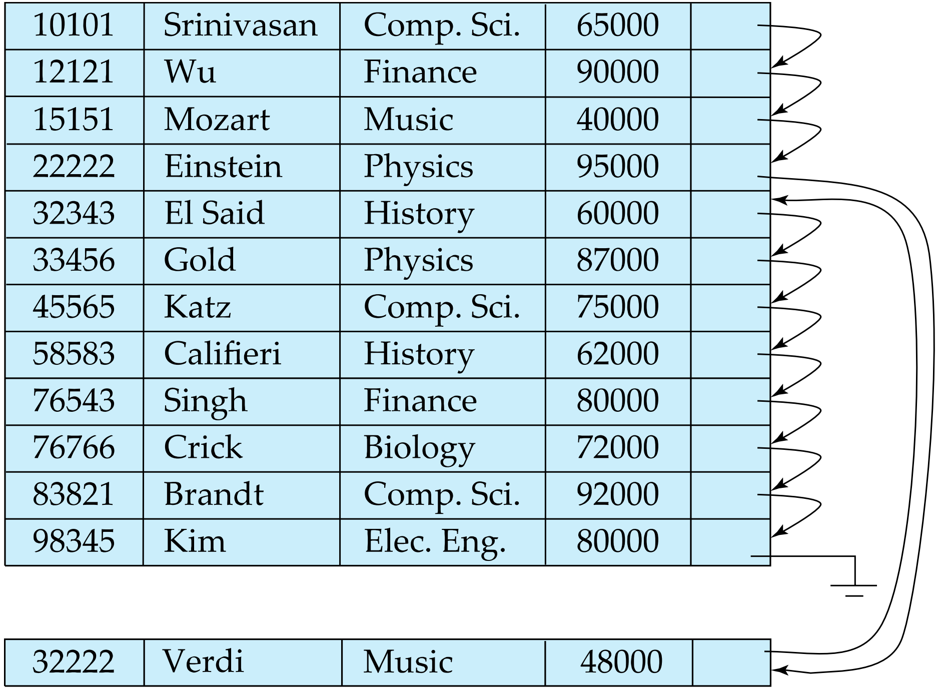
- Suitable for applications that require sequential processing of the entire file
- The records in the file are **ordered by a search-key**



10101	Srinivasan	Comp. Sci.	65000	
12121	Wu	Finance	90000	
15151	Mozart	Music	40000	
22222	Einstein	Physics	95000	
32343	El Said	History	60000	
33456	Gold	Physics	87000	
45565	Katz	Comp. Sci.	75000	
58583	Califieri	History	62000	
76543	Singh	Finance	80000	
76766	Crick	Biology	72000	
83821	Brandt	Comp. Sci.	92000	
98345	Kim	Elec. Eng.	80000	

Sequential File Organization (Cont.)

- **Deletion** – use pointer chains
- **Insertion** –locate the position where the record is to be inserted
  - if there is free space insert there
  - if no free space, insert the record in an overflow block
  - In either case, pointer chain must be updated
- Need to reorganize the file from time to time to restore sequential order



10101	Srinivasan	Comp. Sci.	65000	
12121	Wu	Finance	90000	
15151	Mozart	Music	40000	
22222	Einstein	Physics	95000	
32343	El Said	History	60000	
33456	Gold	Physics	87000	
45565	Katz	Comp. Sci.	75000	
58583	Califieri	History	62000	
76543	Singh	Finance	80000	
76766	Crick	Biology	72000	
83821	Brandt	Comp. Sci.	92000	
98345	Kim	Elec. Eng.	80000	

32222	Verdi	Music	48000	

Multi-table Clustering File Organization

- Store several relations in one file using a multi-table clustering file organization

department

–		

instructor

		–	

multitable clustering of *department* and *instructor*

Multi-table Clustering File Organization (Cont.)

- **good** for queries involving *department* $\bowtie$ *instructor*, and for queries involving one single department and its instructors
- **bad** for queries involving only *department*
- results in variable size records
- Can add pointer chains to link records of a particular relation

Outline

- Overview of Physical Storage Media
- Magnetic Disks
- RAID
- Tertiary Storage
- Storage Access
- File Organization
- Organization of Records in Files

☞ **Data-Dictionary Storage**

Data Dictionary Storage

Data dictionary (also called **system catalog**) stores **metadata,** i.e., data about data, such as

- **Information about relations**
  - names of relations
  - names and types of attributes of each relation
  - names and definitions of views
  - integrity constraints
- **User and accounting information, including passwords**
  - Statistical and descriptive data, number of tuples in each relation
- **Physical file organization information**
  - How relation is stored (sequential/hash/…)
  - Physical location of relation
- **Information about indices (Chapter 14)**

Data Dictionary Storage (Cont.)

- **Catalog structure**
  - Relational representation on disk
  - Specialized data structures designed for efficient access
- A possible catalog representation:

Relation_metadata = *(relation_name, number_of_attributes,*
storage_organization, location)

Attribute_metadata = *(attribute_name, relation_name, domain_type, position, length)*

User_metadata = *(user_name, encrypted_password, group)*

Index_metadata = *(index_name, relation_name, index_type, index_attributes)*

View_metadata = *(view_name, definition)*

数据库及存储技术（补充）

主存数据库

- **内存数据库**
  - 顾名思义就是将数据放在内存中直接操作的数据库。相对于磁盘，内存的数据读写速度要高出几个数量级，将数据保存在内存中相比从磁盘上访问能够极大地提高应用的性能同时，内存数据库抛弃了磁盘数据管理的传统方式，基于全部数据都在内存中重新设计了体系结构，并且在数据缓存、快速算法、并行操作方面也进行了相应的改进，所以数据处理速度比传统数据库的数据处理速度要快很多，一般都在10倍以上。**内存数据库的最大特点是其“主拷贝”或“工作版本”常驻内存，即活动事务只与实时内存数据库的内存拷贝打交道**。
- **定义**
  - 设有数据库系统DBS，DB为DBS中的数据库，DB(t)为在时刻t，DB在内存的数据集，DB(t)属于DB。TS为DBS中所有可能的事务构成的集合。AT(t)为在时刻t处于活动状态的事务集，AT(t)属于TS。Dt(T)为事务T在时刻t所操作的数据集，
  - Dt(T)属于DB。若在任意时刻t，均有: 任意T属于AT(t)， Dt(T)属于DB(t)成立，则称DBS为一个内存数据库系统，简称**MMDBS**; DB为一个内存数据库，简称**MMDB**
  - 常见的例子有MySQL的MEMORY存储引擎、eXtremeDB、TT、FastDB、SQLite、Microsoft SQL Server Compact等

主存数据库与传统数据库

- 传统的数据库系统是关系型数据库，开发这种数据库的目的，是处理永久、稳定的数据。关系数据库强调维护数据的完整性、一致性，但很难顾及有关数据及其处理的定时限制，不能满足工业生产管理实时应用的需要，因为实时事务要求系统能较准确地预报事务的运行时间。对磁盘数据库而言，由于磁盘存取、内外存的数据传递、缓冲区管理、排队等待及锁的延迟等使得事务实际平均执行时间与估算的最好情况执行时间相差很大，如果将整个数据库或其主要的“工作”部分放入内存，使每个事务在执行过程中没有I/O，则为系统较准确估算和安排事务的运行时间，使之具有较好的动态可预报性提供了有力的支持，同时也为实现事务的定时限制打下了基础。这就是内存数据库出现的主要原因。
- **内存数据库**所处理的数据通常是“短暂”的，即有一定的有效时间，过时则有新的数据产生，而当前的决策推导变成无效。所以，实际应用中采用内存数据库来处理实时性强的业务逻辑处理数据。而传统数据库旨在处理永久、稳定的数据，其性能目标是高的系统吞吐量和低的代价，处理数据的实时性就要考虑得相对少一些。实际应用中利用传统数据库这一特性存放相对实时性要求不高的数据。
- 在实际应用中这**两种数据库常常结合使用**，而不是以内存数据库替代传统数据库。

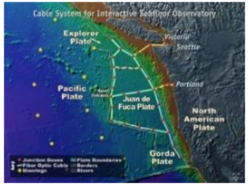
闪存数据库

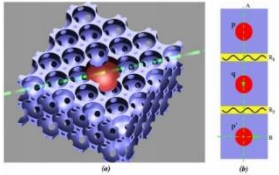
- **闪存（Flash Memory)** 是一种长寿命的**非易失性**（在断电情况下仍能保持所存储的数据信息）的存储器，数据删除不是以单个的字节为单位而是以固定的区块为单位，区块大小一般为256KB到20MB。闪存是电子可擦除只读存储器（EEPROM）的变种，EEPROM与闪存不同的是，它能在字节水平上进行删除和重写而不是整个芯片擦写，这样闪存就比EEPROM的更新速度快。由于其断电时仍能保存数据，闪存通常被用来保存设置信息，如在电脑的BIOS（基本输入输出程序）、PDA（个人数字助理）、数码相机中保存资料等。另一方面，闪存不像RAM（随机存取存储器）一样以字节为单位改写数据，因此不能取代RAM。
- **闪存卡（Flash Card)** 是利用闪存（Flash Memory）技术达到存储电子信息的存储器，一般应用在数码相机，掌上电脑，MP3等小型数码产品中作为存储介质，所以样子小巧，有如一张卡片，所以称之为闪存卡。根据不同的生产厂商和不同的应用，闪存卡大概有SmartMedia（SM卡）、Compact Flash（CF卡）、MultiMediaCard（MMC卡）、Secure Digital（SD卡）、Memory Stick（记忆棒）、XD-Picture Card（XD卡）和微硬盘（MICRODRIVE）这些闪存卡虽然外观、规格不同，但是技术原理都是相同的。

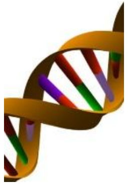
大数据及云存储

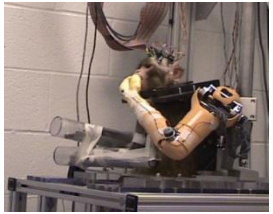
The Future is Full of Opportunity

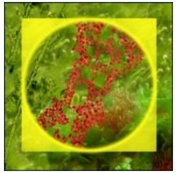
- Designing a next Internet – GENI, Starlink
  - Driving advances in all fields of science and engineering
- Wreckless driving
- Personalized education
- Predictive, preventive, personalized medicine
- Quantum computing (量子计算)
- Personalized health monitoring => quality of life
- Data-intensive supercomputing
- Neurobotics (神经学机器人)
- Synthetic biology (合成生物学)
- The algorithmic lens => Cyber-enabled Discovery and Innovation



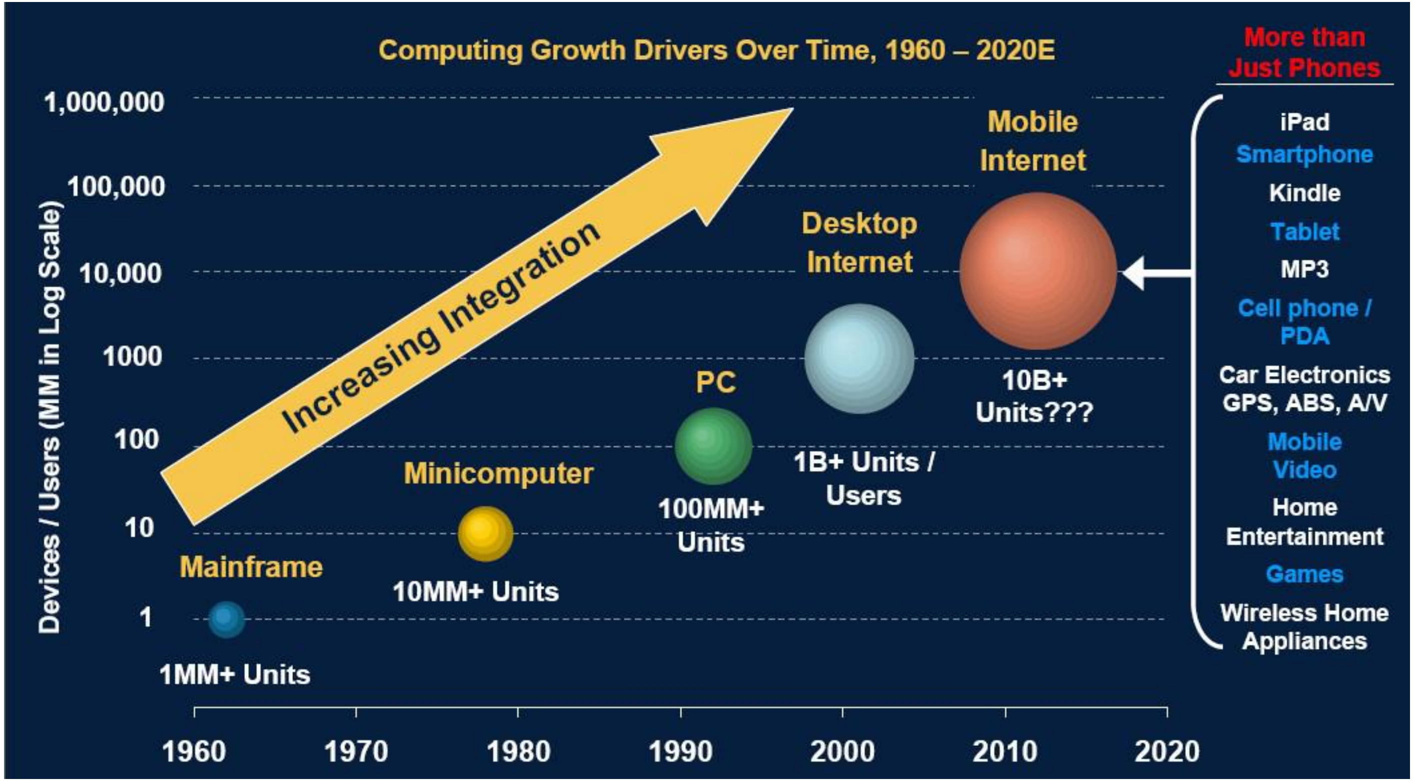






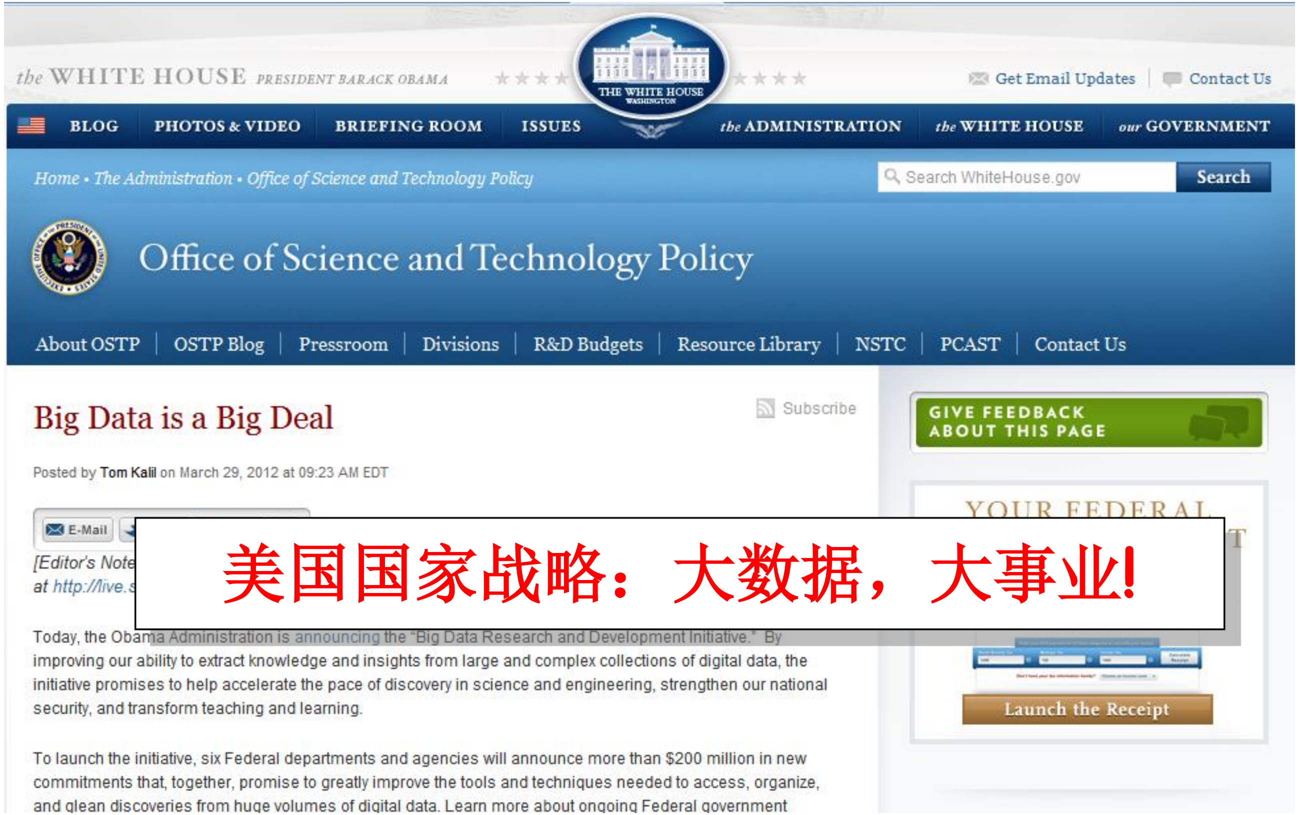


大数据时代的背景



Morgan Stanley, Internet Trends, June 7, 2010

大数据时代的背景



Big Data is a Big Deal

Posted by Tom Kalil on March 29, 2012 at 09:23 AM EDT

[Editor's Note

at http://live.s

Today, the Obama Administration is announcing the "Big Data Research and Development Initiative." By improving our ability to extract knowledge and insights from large and complex collections of digital data, the initiative promises to help accelerate the pace of discovery in science and engineering, strengthen our national security, and transform teaching and learning.

To launch the initiative, six Federal departments and agencies will announce more than $200 million in new commitments that, together, promise to greatly improve the tools and techniques needed to access, organize, and glean discoveries from huge volumes of digital data. Learn more about ongoing Federal government

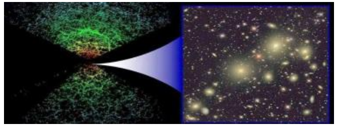
美国国家战略：大数据，大事业!

大数据的困难

- **容量大**

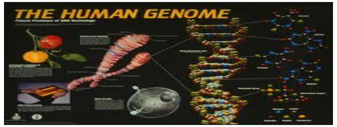
美国国会图书馆存档信息量：约 80TB

科学计算



新墨西哥州的天文望远镜每年产生80TB的图像信息

生物信息



第一个中国人的全基因组图谱，1177亿碱基对

电子商务

淘宝网
Taobao.com

每月交易21亿笔，产生300TB交易日志信息

网络生活

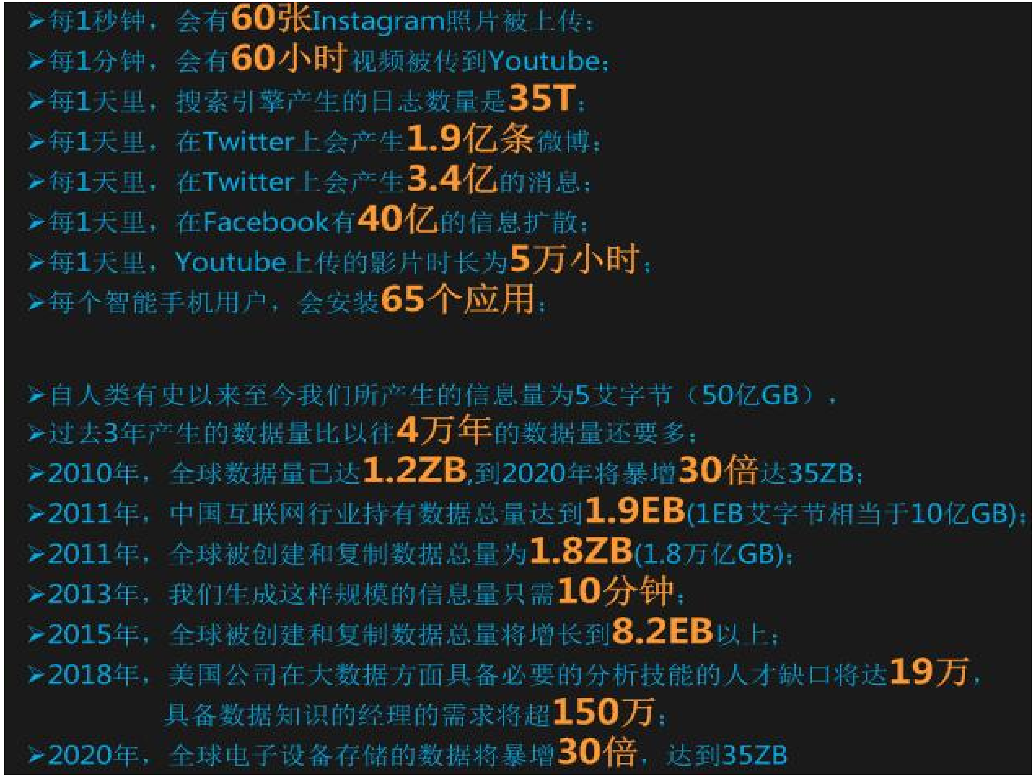
facebook

7亿用户、400亿张照片，总容量超过1500TB

Kilo
Mega
Giga
Tera
Peta
Exa
Zeta
Yotta

大数据的困难

- **容量大**



➢每1秒钟，会有60张Instagram照片被上传；

➢每1分钟，会有60小时视频被传到Youtube；

➢每1天里，搜索引擎产生的日志数量是35T；

➢每1天里，在Twitter上会产生1.9亿条微博；

➢每1天里，在Twitter上会产生3.4亿的消息；

➢每1天里，在Facebook有40亿的信息扩散；

➢每1天里，Youtube上传的影片时长为5万小时；

➢每个智能手机用户，会安装65个应用；

➢自人类有史以来至今我们所产生的信息量为5艾字节（50亿GB），

➢过去3年产生的数据量比以往4万年的数据量还要多；

➢2010年，全球数据量已达1.2ZB,到2020年将暴增30倍达35ZB；

➢2011年，中国互联网行业持有数据总量达到1.9EB(1EB艾字节相当于10亿GB)；

➢2011年，全球被创建和复制数据总量为1.8ZB(1.8万亿GB)；

➢2013年，我们生成这样规模的信息量只需10分钟；

➢2015年，全球被创建和复制数据总量将增长到8.2EB以上；

➢2018年，美国公司在大数据方面具备必要的分析技能的人才缺口将达19万，
具备数据知识的经理的需求将超150万；

➢2020年，全球电子设备存储的数据将暴增30倍，达到35ZB

Kilo
Mega
Giga
Tera
Peta
Exa
Zeta
Yotta

大数据的困难

- **非结构化**
  - 非结构化=没有找到
  - 共性特征：没有
  - 多层结构

大数据的困难

- 仅有MapReduce

MapReduce=Big Data?

MapReduce不是万能的

MapReduce只顾埋头干活、从不抬头看路

MapReduce不是万能的

大数据的困难

- **数据中心—看上去，规模庞大**

脸谱(Facebook) 数据中心

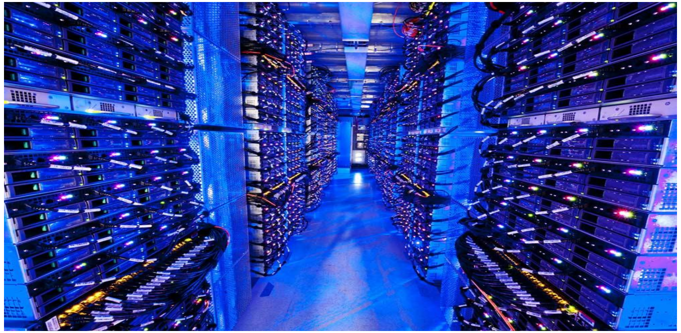
微软(Microsoft)数据中心

苹果(Apple) 数据中心

谷歌(Google) 数据中心

大数据的困难

- 数据中心—走进去，结构复杂



新技术

- **针对存储**
  - **石英玻璃数据存储技术**：可保存数据数亿年，电脑硬盘驱动器的数据保存时间就只有10年，使用较频繁的闪存盘数据保存时间至多5年，而磁带的数据保存时限则达15至30年。
  - **双写入存储技术**：提升至10TB/平方英寸，目前硬盘中所使用的垂直记录技术的存储密度极限大约是每平方英寸数百GB。 两种新写入方式名为“热辅助磁记录技术”(Thermally-Assisted Magnetic Recording：TAR）,和 “位式记录技术” （bit-patterned recording：BPR)

新技术

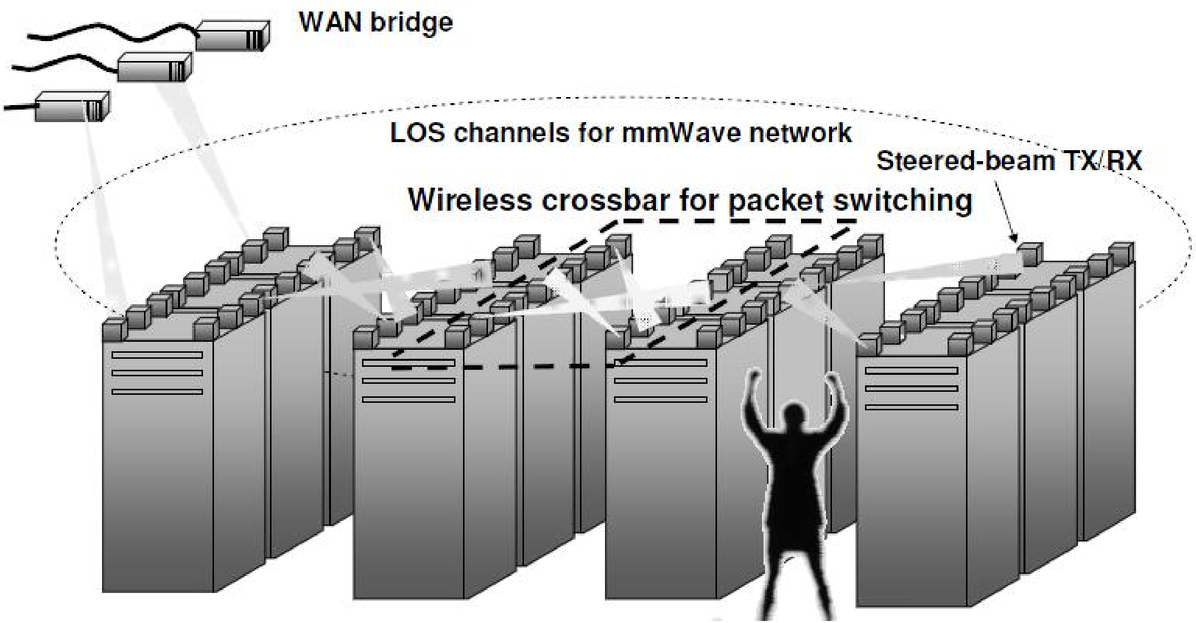
- **针对传输**
  - **量子网络瞬间通信不延迟**：一个粒子可以传递有限的信息，而亿万个粒子联手就形成了量子网络，而这种信息传递没有任何的延迟。通过量子网络相互连接的量子计算机和量子服务器将应用量子纠缠实现无缝通讯
  - **光纤**：一根光纤可供2.1亿对人同时通话，一根普通单模光纤中C波段168路，每路103Gb/s的超大容量超密集波分复用传输2240公里，传输总容量达到17.32Tb/s，相当于2.1亿对人在一根光纤上同时通话
  - **HMC技术**：传输率可达1TB/s，Hybrid Memory Cube技术相比于当今主流的DDR3在能源效率上有至少7倍以上优势。Hybrid Memory Cube技术使用堆叠技术将内存芯片压缩成一个紧凑的“立方体”，并配有新的高速传输接口。新的数据传输接口传输率可达到1TB/s。

新技术

- **针对Hadoop**
  - 交互式数据分析系统Dremel
    - 可以在几秒钟内处理PB级别的数据，并能轻松应对即时查询
  - 谷歌的图数据计算框架Pregel，微软的Trinity
    - 针对大规模图算法（如图遍历（BFS）、PageRank，最短路径（SSSP）等）。处理一个有着几十亿节点、上万亿条边的图，只需数分钟即可完成，其执行时间随着图的大小呈线性增长。

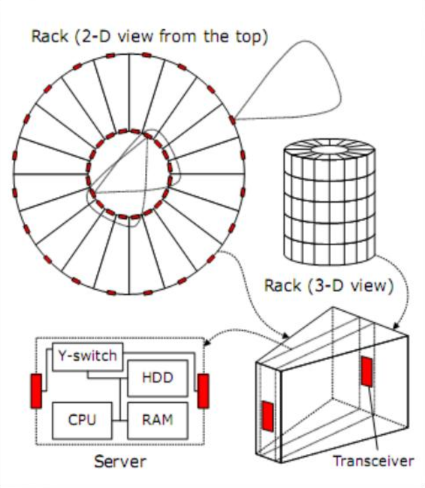
新技术

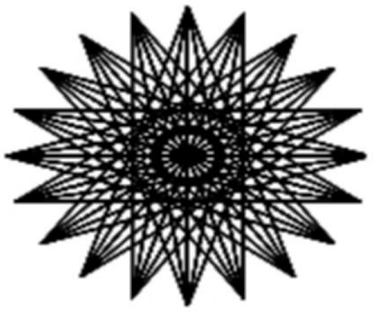
- 针对数据中心


WAN bridge
LOS channels for mmWave network
Steered-beam TX/RX
Wireless crossbar for packet switching

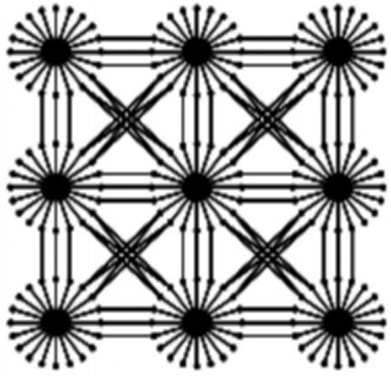
新技术

- **针对数据中心**
  - 扇形机架 + 柱形机箱
  - 机箱内: 冗余链接、强连通
  - 机箱间: 2维类Mesh拓扑





(a) Intra-rack



(b) Inter-rack

新技术

- 针对数据中心
  - 微型数据中心
  - 台式数据中心

MDC
RZ-Products

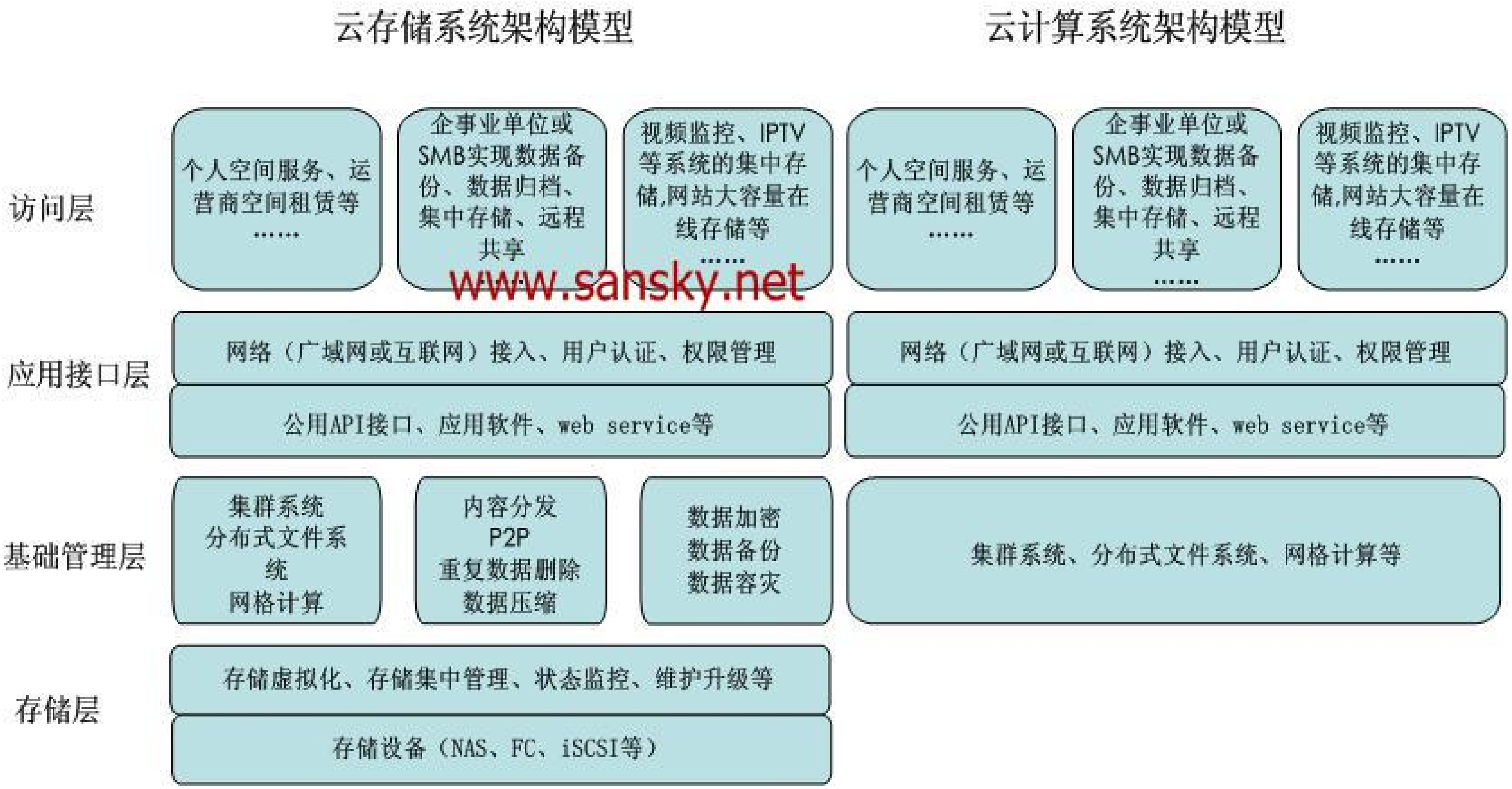
云存储（cloud storage）

- **云存储**是云计算的存储部分，即虚拟化的、易于扩展的存储资源池。用户通过云计算使用存储资源池，但不是所有的云计算的存储部分都是可以分离的。
- 云存储意味着存储可以作为一种服务，通过网络提供给用户。用户可通过若干种方式使用存储，按使用（时间、空间或两者结合）付费：
  - 通过互联网开放接口（如REST），使得第三方网站可以通过云存储提供的服务为用户提供完整的Web服务；
  - 用户直接使用存储相关的在线服务，比如网络硬盘，在线存储，在线备份，或在线归档等服务；
  - 用户传送文件、或者服务商发布内容时的缓冲

云存储（cloud storage）

- 云存储在云计算 (cloud computing)概念上延伸和发展出来的概念
  - 云计算是分布式处理(Distributed Computing)、并行处理(Parallel Computing)和网格计算(Grid Computing)的发展，是透过网络将庞大的计算处理程序自动分拆成无数个较小的子程序，再交由多部服务器所组成的庞大系统经计算分析之后将处理结果回传给用户。通过云计算技术，网络服务提供者可以在数秒之内，处理数以千万计甚至亿计的信息，达到和”超级计算机”同样强大的网络服务
  - 云存储的概念与云计算类似，它是指通过集群应用、网格技术或分布式文件系统等功能，将网络中大量各种不同类型的存储设备通过应用软件集合起来协同工作，共同对外提供数据存储和业务访问功能的一个系统
  - 云存储对使用者来讲，不是指某一个具体设备，而是指一个由许许多多个存储设备和服务器所构成的集合体。使用者使用云存储，并不是使用某一个存储设备，而是使用整个云存储系统带来的一种数据访问服务。所以严格来讲，云存储不是存储，而是一种服务！

云存储（cloud storage）



	云存储系统架构模型			云计算系统架构模型		
访问层	个人空间服务、运营商空间租赁等……	企事业单位或SMB实现数据备份、数据归档、集中存储、远程共享	视频监控、IPTV等系统的集中存储,网站大容量在线存储等……	个人空间服务、运营商空间租赁等……	企事业单位或SMB实现数据备份、数据归档、集中存储、远程共享……	视频监控、IPTV等系统的集中存储,网站大容量在线存储等……
应用接口层	网络（广域网或互联网）接入、用户认证、权限管理 公用API接口、应用软件、web service等			网络（广域网或互联网）接入、用户认证、权限管理 公用API接口、应用软件、web service等		
基础管理层	集群系统 分布式文件系统 网格计算	内容分发 P2P 重复数据删除 数据压缩	数据加密 数据备份 数据容灾	集群系统、分布式文件系统、网格计算等		
存储层	存储虚拟化、存储集中管理、状态监控、维护升级等 存储设备（NAS、FC、iSCSI等）					

云存储（cloud storage）

- 与传统存储设备相比，云存储不仅仅是一个硬件，而是一个网络设备、存储设备、服务器、应用软件、公用访问接口、接入网、和客户端程序等多个部分组成的复杂系统。各部分以存储设备为核心，通过应用软件来对外提供数据存储和业务访问服务
- **云存储系统**的结构模型由 4层组成
  - **存储层是云存储最基础的部分**
  - **基础管理层是云存储最核心部分，也是最难实现的部分**
  - **应用接口层是云存储最灵活多变的部分**
  - **访问层**

云存储（cloud storage）

- **云存储系统的结构模型由 4层组成：存储层、基础管理层、应用接口层、访问层**
- **存储层**
  - 存储层是云存储最基础的部分。存储设备可以是FC光纤通道存储设备，可以是NAS和iSCSI等IP存储设备，也可以是 SCSI或SAS等 DAS存储设备。云存储中的存储设备往往数量庞大且分布多不同地域，彼此之间通过广域网、互联网或者 FC光纤通道网络连接在一起
  - 存储设备之上是一个统一存储设备管理系统，可以实现存储设备的逻辑虚拟化管理、多链路冗余管理，以及硬件设备的状态监控和故障维护
- **基础管理层**
  - 基础管理层是云存储最核心的部分，也是云存储中最难以实现的部分。基础管理层通过集群、分布式文件系统和网格计算等技术，实现云存储中多个存储设备之间的协同工作，使多个的存储设备可以对外提供同一种服务，并提供更大更强更好的数据访问性能
  - CDN内容分发系统、数据加密技术保证云存储中的数据不会被未授权的用户所访问，同时，通过各种数据备份和容灾技术和措施可以保证云存储中的数据不会丢失，保证云存储自身的安全和稳定

云存储（cloud storage）

- **应用接口层**
  - 应用接口层是云存储最灵活多变的部分。不同的云存储运营单位可根据实际业务类型，开发不同的应用服务接口，提供不同的应用服务。比如视频监控应用平台、IPTV和视频点播应用平台、网络硬盘应用平台，远程数据备份应用平台等
- **访问层**
  - 任何一个授权用户都可以通过标准的公用应用接口来登录云存储系统，享受云存储服务。云存储运营单位不同，云存储提供的访问类型和访问手段也不同

云存储（cloud storage）

- **云存储系统**是多设备、多应用、多服务协同工作集合体，其技术前提：
  - 云存储系统是多设备、多应用、多服务协同工作的集合体
  - WEB2.0技术
  - 应用存储的发展
  - 集群技术、网格技术和分布式文件系统
  - CDN内容分发、P2P技术、数据压缩技术、重复数据删除技术、数据加密技术
  - 存储虚拟化技术、存储网络化管理技术

云存储（cloud storage）

- **云存储（cloud storage）**概念提出，得到众多厂商的支持和关注
  - **Amazon推出的Elastic Compute Cloud（EC2：弹性计算云）**云存储产品，旨在为用户提供互联网服务形式同时提供更强的存储和计算功能
  - **内容分发网络服务提供商CDNetworks和业界著名的云存储平台服务商 Nirvanix**发布了一项合作，并宣布结成战略伙伴关系，以提供业界的云存储和内容传送服务集成平台
  - **微软**推出提供**网络移动硬盘服务的Windows Live SkyDrive**
  - **EMC加入道里可信基础架构项目**，致力于云计算环境下关于信任和可靠度保证的全球研究协作
  - **IBM**将**云计算标准**作为全球备份中心的3亿美元扩展方案的一部分

RAID (补充)

- **磁盘阵列（Redundant Arrays of Independent Disks，RAID）**，有“独立磁盘构成的具有冗余能力的阵列”之意。原理是利用数组方式来作磁盘组，配合数据分散排列的设计，提升数据的安全性。
- 磁盘阵列是由很多价格较便宜的磁盘，组合成一个容量巨大的磁盘组，利用个别磁盘提供数据所产生加成效果提升整个磁盘系统效能。利用这项技术，将数据切割成许多区段，分别存放在各个硬盘上。
- 磁盘阵列还能利用同位检查（Parity Check）的观念，在数组中任一个硬盘故障时，仍可读出数据，在数据重构时，将数据经计算后重新置入新硬盘中。

RAID（补充）

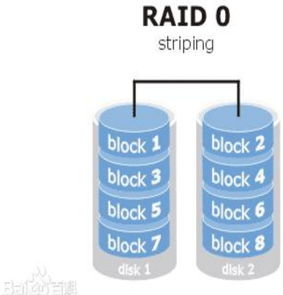
- 由加利福尼亚大学伯克利分校（University of California-Berkeley）在1987年，发表的文章：“A Case for Redundant Arrays of Inexpensive Disks”。文章中，谈到了RAID这个词汇，而且定义了RAID的5层级。伯克利大学研究目的是反应当时CPU快速的性能。CPU效能每年大约成长30～50%，而硬磁机只能成长约7%。研究小组希望能找出一种新的技术，在短期内，立即提升效能来平衡计算机的运算能力，主要研究目的是效能与成本。
- 研究小组也设计出容错（fault-tolerance），逻辑数据备份（logical data redundancy），而**产生了RAID理论**。研究初期，便宜（Inexpensive）的磁盘也是主要的重点，但后来发现，大量便宜磁盘组合并不能适用于现实的生产环境，后来Inexpensive被改为independent，许多独立的磁盘组。
- **独立磁盘冗余阵列（RAID，redundant array of independent disks）**是把相同的数据存储在多个硬盘的不同的地方（冗余地）。通过把数据放在多个硬盘上，输入输出操作能以平衡的方式交叠，改良性能。因为多个硬盘增加了平均故障间隔时间（MTBF），储存冗余数据也增加了容错。

RAID（补充）

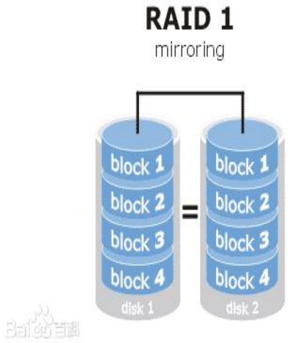
- **磁盘阵列其样式有三种**
  - 外接式磁盘阵列柜：最常被使用大型服务器上，具可热交换（Hot Swap）的特性，不过这类产品的价格都很贵。
  - 内接式磁盘阵列卡：因为价格便宜，但需要较高的安装技术，适合技术人员使用操作。
  - 利用软件仿真：由于会拖累机器速度，不适合大数据流量的服务器。
- 磁盘阵列作为独立系统在主机外直连或通过网络与主机相连。磁盘阵列有多个端口可以被不同主机或不同端口连接。一个主机连接阵列的不同端口可提升传输速度。
- 在磁盘阵列内部为加快与主机交互速度，都带有一定量的缓冲存储器。主机与磁盘阵列的缓存交互，缓存与具体的磁盘交互数据。
- 在应用中，有部分常用的数据是需要经常读取的，磁盘阵列根据内部的算法，查找出这些经常读取的数据，存储在缓存中，加快主机读取这些数据的速度，而对于其他缓存中没有的数据，主机要读取，则由阵列从磁盘上直接读取传输给主机。对于主机写入的数据，只写在缓存中，主机可以立即完成写操作。然后由缓存再慢慢写入磁盘。

RAID（补充）

- **RAID 0**
  - 最早出现的RAID模式，即Data Stripping数据分条技术。RAID 0是组建磁盘阵列中最简单的一种形式，只需要2块以上的硬盘即可，成本低，可以提高整个磁盘的性能和吞吐量。RAID 0没有提供冗余或错误修复能力，但实现成本是最低的

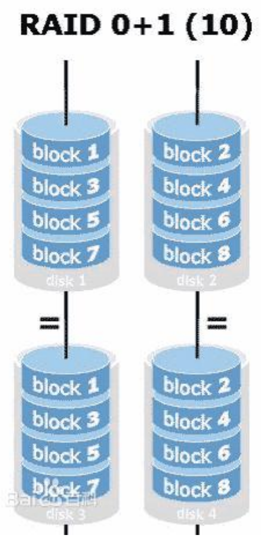


- **RAID 1**
  - 称为磁盘镜像，是把一个磁盘的数据镜像到另一个磁盘上，数据在写入一块磁盘的同时，会在另一块闲置的磁盘上生成镜像文件，在不影响性能情况下最大限度保证系统的可靠性和可修复性，只要系统中任何一对镜像盘中至少有一块磁盘可以使用，甚至可以在一半数量的硬盘出现问题时系统都可以正常运行。当一块硬盘失效时，系统会忽略该硬盘，转而使用剩余的镜像盘读写数据，具备很好的磁盘冗余能力。对数据来讲绝对安全，但是成本明显增加，磁盘利用率仅为50%。出现硬盘故障的RAID系统不再可靠，应及时更换坏的硬盘，否则剩余镜像盘也出现问题，整个系统就会崩溃。更换新盘后原有数据会需要长时间同步镜像，外界对数据的访问不会受影响，只是这时整个系统性能有所下降。RAID 1多用在保存关键性的重要数据的场合。



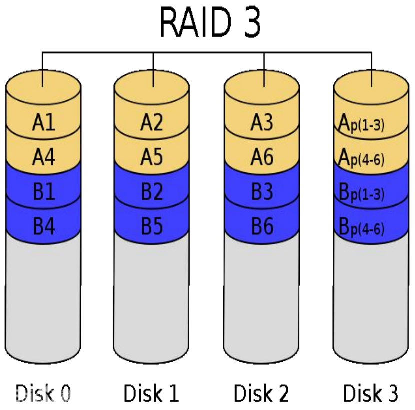
RAID（补充）

- **RAID 0+1**
  - 是RAID0与RAID1的结合体。在我们单独使用RAID 1也会出现类似单独使用RAID 0那样的问题，即在同一时间内只能向一块磁盘写入数据，不能充分利用所有的资源，为此在磁盘镜像中建立带区集。把RAID0和RAID1技术结合起来，数据除分布在多个盘上外，每个盘都有其物理镜像盘，提供全冗余能力，允许一个以下磁盘故障，而不影响数据可用性，并具有快速读/写能力。RAID0+1要在磁盘镜像中建立带区集至少4个硬盘
- **RAID2**
  - 带海明码校验。RAID 2同RAID 3类似，都是将数据条块化分布于不同硬盘上，条块单位为位或字节。RAID 2 使用一定编码技术提供错误检查及恢复，需要多个磁盘存放检查及恢复信息，RAID 2技术实施更复杂。在商业环境中很少使用。由一个数据不同的位运算得到的海明校验码可以保存另一组磁盘上，以保证输出正确。数据传送速率相当高，如果希望达到比较理想的速度，那最好提高保存校验码ECC码的硬盘，对于控制器的设计来说，它又比RAID3，4或5要简单。要利用海明码，必须要付出数据冗余的代价。输出数据的速率与驱动器组中速度最慢的相等。



RAID（补充）

- **RAID3**
  - 带奇偶校验码的并行传送。这种校验码与RAID2不同， RAID2只能查错不能纠错。它访问数据时一次处理一个带区，可以提高读取和写入速度。校验码在写入数据时产生并保存在另一个磁盘上。需要实现时用户必须要有三个以上的驱动器，写入速率与读出速率都很高，因为校验位比较少，因此计算时间相对而言比较少。用软件实现RAID控制十分困难，控制器实现也不容易，主要用于图形（动画）等要求吞吐率比较高的场合。不同于RAID 2，RAID 3使用单块磁盘存放奇偶校验信息。如果一块磁盘失效，奇偶盘及其他数据盘可重新产生数据。如果奇偶盘失效，不影响数据使用。RAID 3对于大量的连续数据可提供很好传输率，但对于随机数据，奇偶盘会成为写操作的瓶颈。



- **RAID4**
  - 带奇偶校验码的独立磁盘结构。RAID4和RAID3很象，不同的是，它对数据的访问是按数据块进行的，也就是按磁盘进行的，每次是一个盘。RAID3是一次一横条，而RAID4一次一竖条。它的特点和RAID3也挺象，不过在失败恢复时，难度要比RAID3大得多，控制器的设计难度也要大许多，而且访问数据的效率不好。

RAID（补充）

- **RAID5**
  - 分布式奇偶校验的独立磁盘结构。奇偶校验码存在于所有磁盘上，其中p0代表第0带区的奇偶校验值，其它类似。RAID5的读出效率很高，写入效率一般，块式的集体访问效率不错。奇偶校验码在不同的磁盘上，提高了可靠性。但是对数据传输的并行性解决不好，控制器的设计也相当困难。RAID 3 与RAID 5相比，重要区别在于RAID 3每进行一次数据传输，需涉及到所有阵列盘。而对于RAID 5，大部分数据传输只对一块磁盘操作，可进行并行操作。在RAID 5中有“写损失”，即每一次写操作，将产生四个实际读/写操作，其中两次读旧的数据及奇偶信息，两次写新的数据及奇偶信息。
- **RAID6**
  - 带有两种分布存储的奇偶校验码的独立磁盘结构。是对RAID5的扩展，用于要求数据绝对不能出错场合。引入了第二种奇偶校验值，需要N+2个磁盘，对控制器的设计变得十分复杂，写入速度也不好，用于计算奇偶校验值和验证数据正确性所花费的时间比较多，造成了不必须的负载
- **RAID7**
  - 优化的高速数据传送磁盘结构。RAID7所有的I/O传送均是同步进行的，可以分别控制，提高了系统的并行性，提高系统访问数据的速度；每个磁盘都带有高速缓冲存储器，实时操作系统可以使用任何实时操作芯片，达到不同实时系统的需要。允许使用SNMP协议进行管理和监视，可以对校验区指定独立的传送信道以提高效率。可以连接多台主机，因为加入高速缓冲存储器，当多用户访问系统时，访问时间几乎接近于0。由于采用并行结构，因此数据访问效率大大提高。引入了一个高速缓冲存储器，有利有弊，因为一旦系统断电，在高速缓冲存储器内的数据就会全部丢失，因此需要和UPS一起工作。当然了，这么快的东西，价格也非常昂贵。

RAID (补充)

- RAID 5E (RAID 5 Enhancement)
  - RAID 5E是在RAID 5级别基础上的改进，与RAID 5类似，数据的校验信息均匀分布在各硬盘上，但是，在每个硬盘上都保留了一部分未使用的空间，这部分空间没有进行条带化，最多允许两块物理硬盘出现故障。看起来，RAID 5E和RAID 5加一块热备盘好像差不多，其实由于RAID 5E是把数据分布在所有的硬盘上，性能会比RAID5 加一块热备盘要好。当一块硬盘出现故障时，有故障硬盘上的数据会被压缩到其它硬盘上未使用的空间，逻辑盘保持RAID 5级别。
- RAID 5EE
  - 与RAID 5E相比，RAID 5EE的数据分布更有效率，每个硬盘的一部分空间被用作分布的热备盘，它们是阵列的一部分，当阵列中一个物理硬盘出现故障时，数据重建的速度会更快。
- RAID 50
  - RAID50是RAID5与RAID0的结合。此配置在RAID5的子磁盘组的每个磁盘上进行包括奇偶信息在内的数据的剥离。每个RAID5子磁盘组要求三个硬盘。RAID50具备更高的容错能力，因为它允许某个组内有一个磁盘出现故障，而不会造成数据丢失。而且因为奇偶位分部于RAID5子磁盘组上，故重建速度有很大提高。优势：更高的容错能力，具备更快数据读取速率的潜力。需要注意的是：磁盘故障会影响吞吐量。故障后重建信息的时间比镜像配置情况下要长。

End of Lecture 7