

# Computer Architecture

*Computer Science & Engineering*

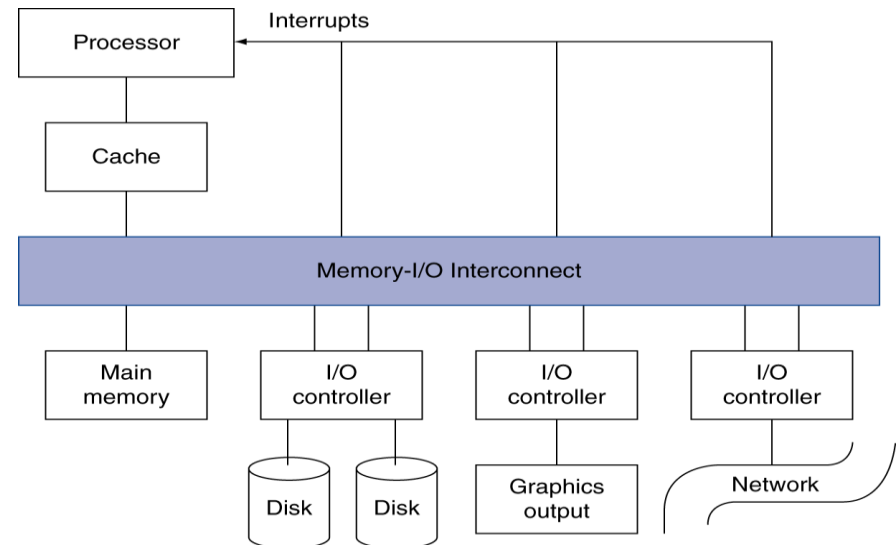
## Chương 6

### Hệ thống lưu trữ và các thiết bị Xuất/Nhập khác



# Dẫn nhập

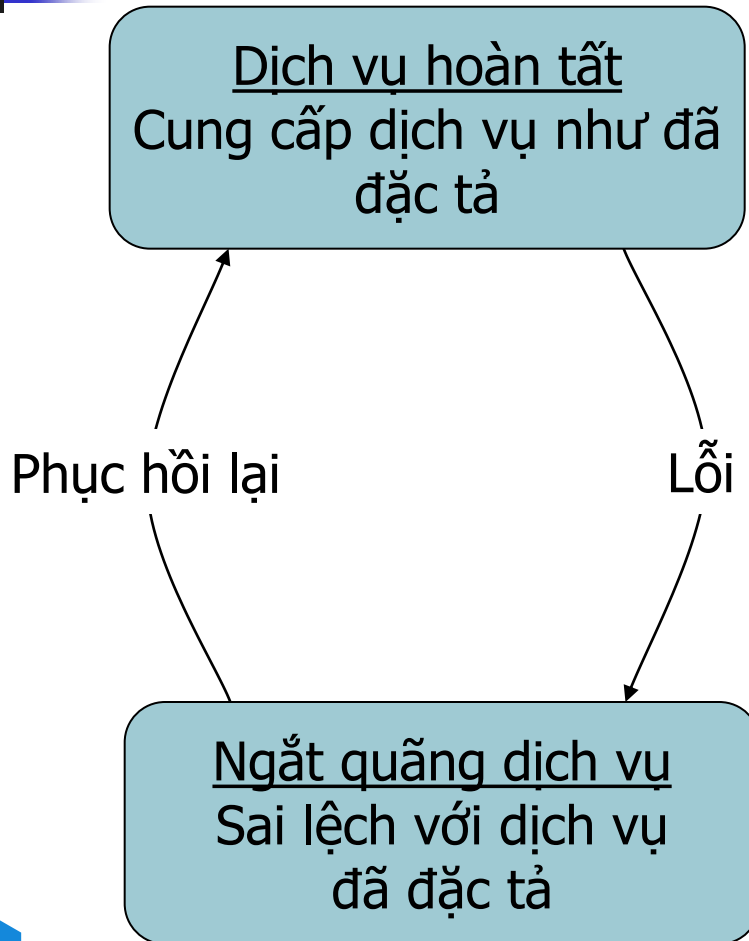
- Đặc tính của các thiết bị ngoại vi thể hiện:
  - Hành vi (chức năng): Nhập (I), Xuất (O), Lưu trữ (storage)
  - Đối tượng tương tác: Người sử dụng hoặc máy
  - Tốc độ truyền: bytes/sec, transfers/sec
- Kết nối tuyến I/O



# Đặc tính của hệ thống I/O

- Tính ổn định = Độ tin cậy (Dependability) rất quan trọng:
  - Đặc biệt các thiết bị lưu trữ
- Đại lượng đo hiệu suất
  - Thời gian đáp ứng (Latency=response time)
  - Hiệu suất đầu ra (Throughput=bandwidth)
  - Hệ thống để bàn & nhúng
    - Quan tâm chủ yếu là thời gian đáp ứng & đa dạng thiết bị
  - Hệ thống máy chủ (Servers)
    - Chủ yếu là hiệu suất đầu ra & khả năng mở rộng

# Độ tin cậy (Dependability)



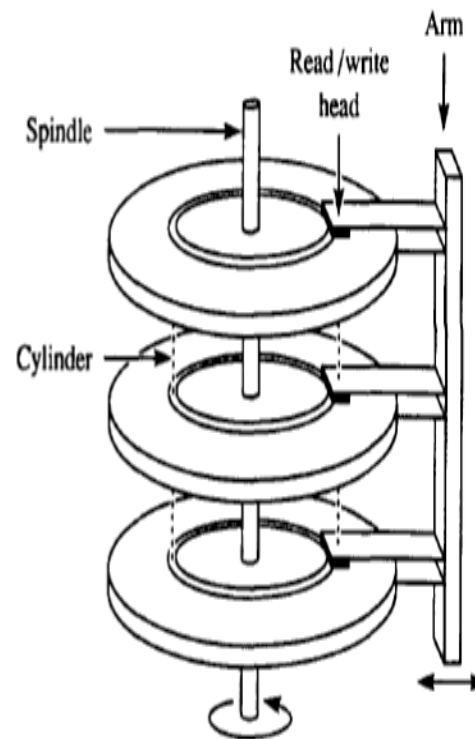
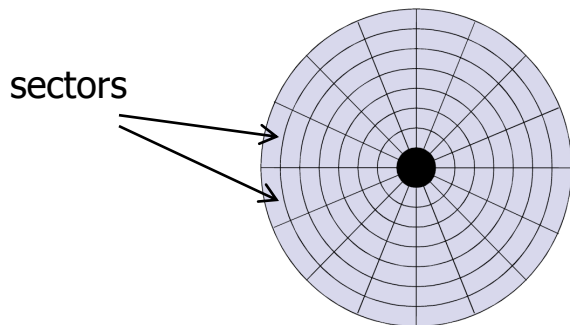
- Lỗi: một bộ phận nào đó sinh lỗi của nó
  - Có & có thể không dẫn đến lỗi hệ thống

# Đo độ tin cậy

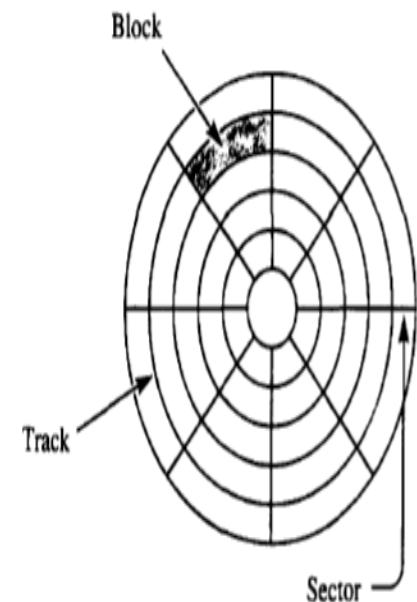
- Mức tin cậy (reliability): thời gian trung bình cho đến khi có lỗi (MTTF=Mean Time To Failure))
- Ngắt dịch vụ: Thời gian trung bình khắc phục lỗi (MTTR= Mean Time to repaire)
- Thời gian trung bình giữa 2 lần lỗi
  - $MTBF = MTTF + MTTR$  (Mean time between failures)
- Tính sẵn sàng (Availability) =
$$\frac{MTTF}{MTTF + MTTR}$$
- Cải thiện tính sẵn sàng
  - Tăng MTTF: tránh lỗi, dự phòng, tiên đoán lỗi
  - Giảm MTTR: cải thiện công cụ & tiến trình tìm và sửa lỗi

# Lưu trữ trên đĩa

- Nonvolatile (không tự biến mất), nhiều đĩa từ tính quay quanh 1 trục



(a) A hard disk drive.



(b) A single disk.



# Sector & Truy cập

- Mỗi sector là đơn vị khối chứa các thông tin
  - Chỉ số nhận dạng Sector
  - Dữ liệu (512 bytes, hướng 4096 bytes per sector)
  - Mã sửa lỗi (ECC)
  - Trường đồng bộ & Khoảng trống phân cách
- Truy cập 1 sector bao gồm:
  - Trễ hàng vì có nhiều yêu cầu đồng thời
  - Tìm rãnh (Seek): Dịch chuyển đầu từ
  - Rotational latency
  - Vận chuyển dữ liệu (Data transfer)
  - Phí tổn mạch điều khiển (Controller overhead)

# Ví dụ: Truy cập đĩa

## ■ Giả sử

- Sector có 512Bytes, tốc độ quay 15,000rpm, thời gian dò tìm 4ms, tốc độ truyền 100MB/s, Phí tổn đ/khiển 0.2ms, idle disk

## ■ Thời gian đọc trung bình

- 4ms dò tìm
  - +  $\frac{1}{2} / (15,000/60) = 2\text{ms}$  rotational latency
  - +  $512 / 100\text{MB/s} = 0.005\text{ms}$  thời gian truyền
  - + 0.2ms trễ do bộ đ/khiển
  - = 6.2ms

## ■ Thời gian thực tế = 25% của nhà sản xuất

- $1\text{ms} + 2\text{ms} + 0.005\text{ms} + 0.2\text{ms} = 3.2\text{ms}$



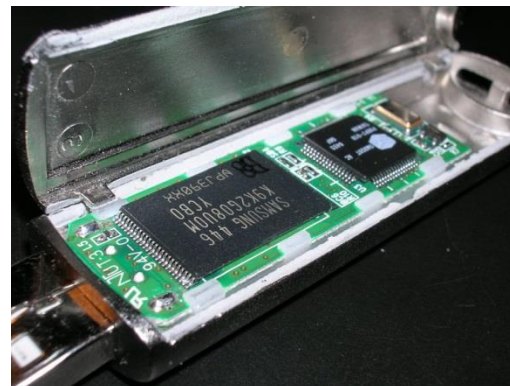


# Các vấn đề Hiệu suất đĩa

- Nhà sản xuất cho biết thời gian dò tìm trung bình
  - Dựa trên mọi trường hợp dò tìm có thể
  - Tính cục bộ & định thời OS sẽ có số liệu thực tế nhỏ hơn
- Mạch điều khiển sẽ xác định vị trí vật lý trên đĩa
  - Máy tính làm việc với giá trị luận lý
  - SCSI, ATA, SATA
- Tăng hiệu suất bằng Cache
  - Truy cập sẵn
  - Tránh dò tìm và trễ vòng quay

# Lưu trữ Flash

- Nonvolatile, lưu trữ bán dẫn
  - 100× – 1000× nhanh hơn đĩa
  - Nhỏ hơn, tổn ít năng lượng tiêu thụ, ổn định hơn
  - Tuy nhiên đắt hơn \$/GB (giữa đĩa và DRAM)





# Các loại bộ nhớ Flash

- NOR flash: bit nhớ giống cổng NOR
  - Truy cập ngẫu nhiên
  - Dùng nhớ lệnh trong hệ tổng nhúng
- NAND flash: bit nhớ giống cổng NAND
  - Mật độ cao (bits/area), truy cập khối mỗi lần
  - Rẻ hơn
  - Dùng trong USB keys, media storage, ...
- Sau khoảng 1000 lần truy xuất: có vấn đề
  - Không thể dùng thay thế RAM hoặc đĩa
  - Khắc phục vấn đề: ánh xạ lại



# Thành phần kết nối

- Cần kết nối giữa các bộ phận như
  - CPU, bộ nhớ, Điều khiển I/O
- Tuyến “Bus”: chia sẻ kênh truyền
  - Bao gồm nhóm các đường dây song song truyền dữ liệu và đồng bộ truyền dữ liệu
  - Hiện tượng cổ chai
- Hiệu suất bị ảnh hưởng bởi các yếu tố vật lý như
  - Độ dài đường truyền, số kết nối
- Phương án hiện nay: kết nối tuần tự tốc độ cao: giống mạng



# Tuyến “Bus” các loại

- Hai tuyến chính
- Tuyến Bus Processor $\leftrightarrow$ Memory
  - Khoảng cách gần (ngắn), tốc độ cao
  - Thiết kế phù hợp với tổ chức bộ nhớ
- Tuyến bus I/O
  - Khoảng cách xa hơn, nhiều điểm tiếp nối
  - Chuẩn hóa để dễ sử dụng
  - Nối với tuyến bus “processor-memory” qua cầu nối (Bridge)



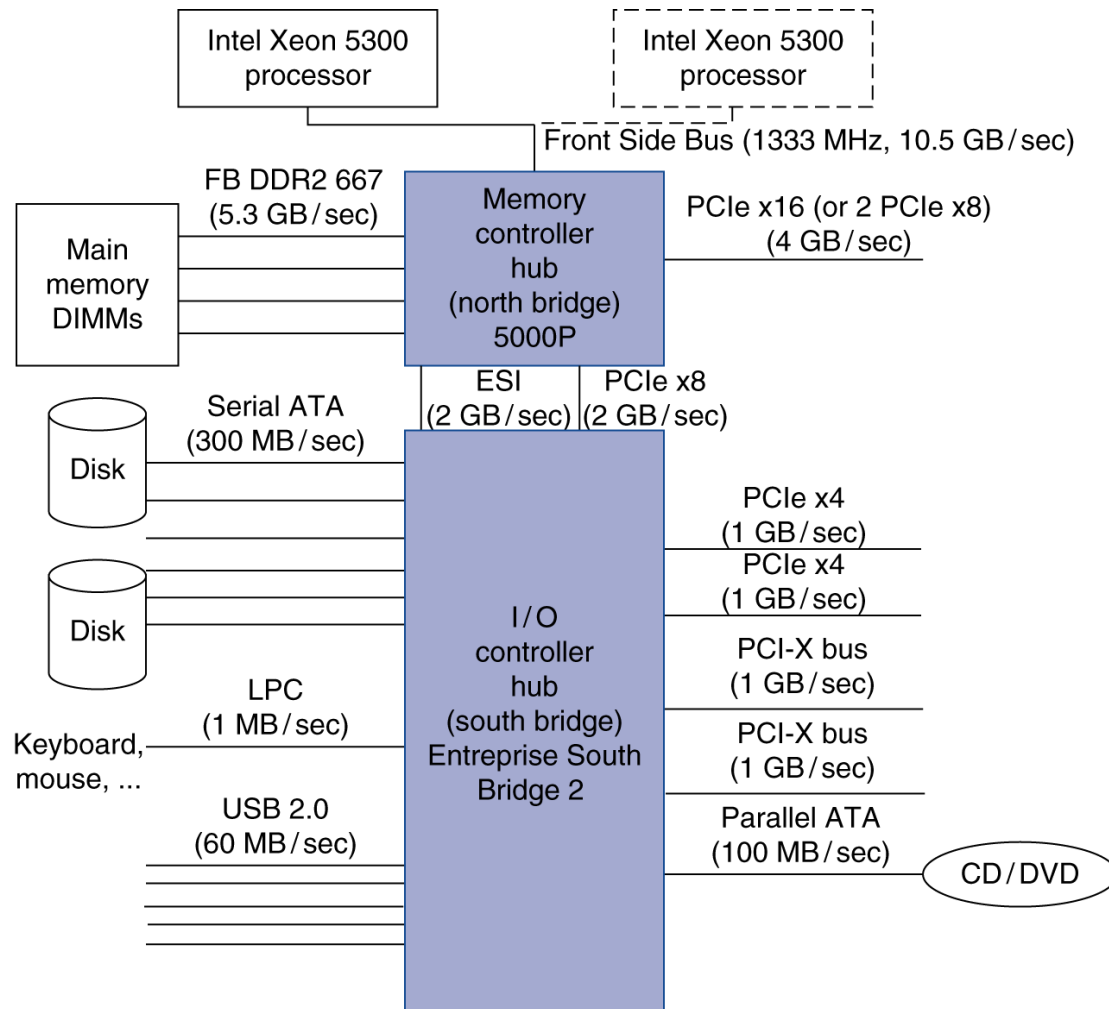
# Tín hiệu và Đồng bộ tuyến Bus

- Đường dữ liệu (Data lines)
  - Địa chỉ & dữ liệu
  - Riêng biệt hoặc trộn lẫn
- Đường điều khiển
  - Thể hiện loại dữ liệu trên đường truyền, đồng bộ các giao dịch
- Đồng bộ
  - Sử dụng đồng hồ tuyến bus (tần số thấp hơn)
- Bất đồng bộ
  - Sử dụng cơ chế bắt tay (request/acknowledge)

# Một số ví dụ Bus I/O chuẩn

	Firewire	USB 2.0	PCI Express	Serial ATA	Serial Attached SCSI
Intended use	External	External	Internal	Internal	External
Devices per channel	63	127	1	1	4
Data width	4	2	2/lane	4	4
Peak bandwidth	50MB/s or 100MB/s	0.2MB/s, 1.5MB/s, or 60MB/s	250MB/s/lane 1×, 2×, 4×, 8×, 16×, 32×	300MB/s	300MB/s
Hot pluggable	Yes	Yes	Depends	Yes	Yes
Max length	4.5m	5m	0.5m	1m	8m
Standard	IEEE 1394	USB Implementers Forum	PCI-SIG	SATA-IO	INCITS TC T10

# Hệ thống x86 PC I/O







# Quản lý I/O

- I/O được quản lý trực tiếp bởi OS
  - Nhiều chương trình đồng thời cùng chia sẻ chung các thiết bị I/O
    - Cần được bảo vệ và định thời
  - I/O tạo ngắt quãng bất đồng bộ
    - Giống cơ chế ngoại lệ
  - Lập trình I/O ít phức tạp (Device Driver)
    - OS tạo các dịch vụ trên I/O để các chương trình gọi các dịch vụ thông qua OS



# Các lệnh I/O

- Thiết bị I/O devices được quản lý bằng phần cứng điều khiển I/O
  - Vận chuyển dữ liệu (từ I/O hay đến I/O)
  - Các tác vụ đồng bộ với phần mềm
- Thanh ghi lệnh (Command registers)
  - Ra lệnh thiết bị thực hiện
- Thanh ghi trạng thái (Status registers)
  - Mô tả trạng thái tức thời của thiết bị
- Thanh ghi dữ liệu (Data registers)
  - Ghi (write): chuyển dữ liệu đến thiết bị
  - Đọc (read): chuyển dữ liệu từ thiết bị



# Truy xuất các thanh ghi I/O

- Ánh xạ như địa chỉ bộ nhớ (Memory mapped)
  - Thanh ghi được địa chỉ hóa như không gian bộ nhớ
  - Giải mã địa chỉ sẽ tự phân biệt
  - OS thực hiện cơ chế chuyển đổi địa chỉ sao cho chỉ có OS mới truy cập được
- Lệnh I/O chuyên biệt
  - Tồn tại các lệnh chuyên biệt để truy xuất các thanh ghi I/O
  - Chỉ thực thi trong (kernel mode)
  - Ví dụ: x86



# Cơ chế Dò quét (polling)

- Kiểm tra thanh ghi trạng thái liên tục
  - Nếu thiết bị sẵn sàng, thực hiện tác vụ I/O
  - Nếu lỗi, thực hiện biện pháp giải quyết
- Thông dụng trong các hệ thống nhỏ hoặc các hệ thống nhúng không đòi hỏi hiệu suất cao, do:
  - Thời gian xử lý dễ tiên đoán trước
  - Giá thành phần cứng thấp
- Trong các hệ thống khác: phí thời gian CPU (busy for waiting)



# Ngắt quãng (interrupts)

- Khi thiết bị sẵn sàng hoặc xuất hiện lỗi
  - Bộ điều khiển thiết bị ngắt quãng CPU
- Ngắt quãng cũng giống một ngoại lệ
  - Nhưng không đồng bộ với lệnh đang thực thi
  - Kích khởi bộ xử lý ngắt quãng tại thời điểm giữa các lệnh
  - Cung cấp thông tin đến thiết bị tương ứng
- Ngắt quãng có thứ tự ưu tiên
  - Khác thiết bị quan trọng có chế độ ưu tiên cao
  - Ngắt quãng có ưu tiên cao hơn có thể ngắt ưu tiên thấp hơn



# Phương thức vận chuyển

- Hoạt động theo cơ chế dò quét & ngắt quãng
  - CPU chuyển dữ liệu giữ bộ nhớ và các thanh ghi dữ liệu của I/O
  - Tốn thời gian cho các thiết bị tốc độ cao
- Truy cập bộ nhớ trực tiếp (DMA)
  - OS cấp địa chỉ bắt đầu trong bộ nhớ
  - Điều khiển I/O controller vận chuyển đến/từ bộ nhớ một cách chủ động
  - Bộ điều khiển I/O ngắt quãng khi hoàn tất hay lỗi xảy ra



# Đo hiệu xuất I/O

- Hiệu xuất I/O phụ thuộc vào:
  - Phần cứng: CPU, bộ nhớ, đ/khiển & buses
  - Phần mềm: Hệ điều hành, Hệ quản trị dữ liệu, ứng dụng
  - Tải: mức độ yêu cầu truy xuất & mẫu
- Khi thiết kế hệ thống I/O system cần hài hòa “thời gian đáp ứng” & hiệu xuất đầu ra

# Hiệu suất giữa I/O & CPU

## Amdahl's Law

- Không thể bỏ qua hiệu suất I/O khi gia tăng hiệu suất tính toán (song song hóa) của CPU
- Ví dụ:
  - Đo đạc cho thấy 90s (CPU time), 10s (I/O time)
  - Số CPU tăng gấp đôi mỗi năm và I/O không đổi

Year	CPU time	I/O time	Elapsed time	% I/O time
now	90s	10s	100s	10%
+2	45s	10s	55s	18%
+4	23s	10s	33s	31%
+6	11s	10s	21s	47%





## RAID=

(Redundant Array of Inexpensive (Independent) Disks)

- Sử dụng nhiều đĩa nhỏ thay vì 1 đĩa thật lớn
- Song song hóa để cải thiện hiệu suất
- Thêm đĩa để tạo thông tin dự trữ (dư thừa)
- Xây dựng hệ thống lưu trữ với an toàn dữ liệu cao
  - Đặc biệt có khả năng thay nóng
- RAID 0
  - Không có thông tin dự thừa ("AID"?)
    - Thông tin chứa liên tiếp theo mảng trên các đĩa
  - Tuy vậy: không tăng hiệu xuất truy cập



# RAID 1 & 2

- RAID 1: Đối xứng “Mirroring”
  - Số đĩa:  $N + N$ , sao chép dữ liệu giống nhau
    - Dữ liệu đồng thời được ghi trên cả 2 đĩa
    - Trong trường hợp lỗi, đọc đĩa đối xứng
- RAID 2: Mã sửa lỗi
  - Số đĩa:  $N + E$  (e.g.,  $10 + 4$ )
  - Tách dữ liệu ở mức bit trên toàn bộ  $N$
  - Tạo E-bit ECC (theo giải thuật)
  - Quá phức tạp → không dùng trong thực tế



# RAID 3: Parity mức bit xen kẽ

- Số đĩa:  $N + 1$ 
  - Dữ liệu phân mảnh, chứa trên toàn bộ  $N$  đĩa ở mức byte
  - Đĩa dư thêm chứa thông tin parity
  - Truy cập (đọc): đọc cùng lúc nhiều đĩa
  - Truy cập (ghi): tạo parity mới tương ứng và ghi cùng lúc trên nhiều đĩa
  - Trường hợp lỗi: dùng thông tin parity để khôi phục dữ liệu bị mất.
- Không thông dụng

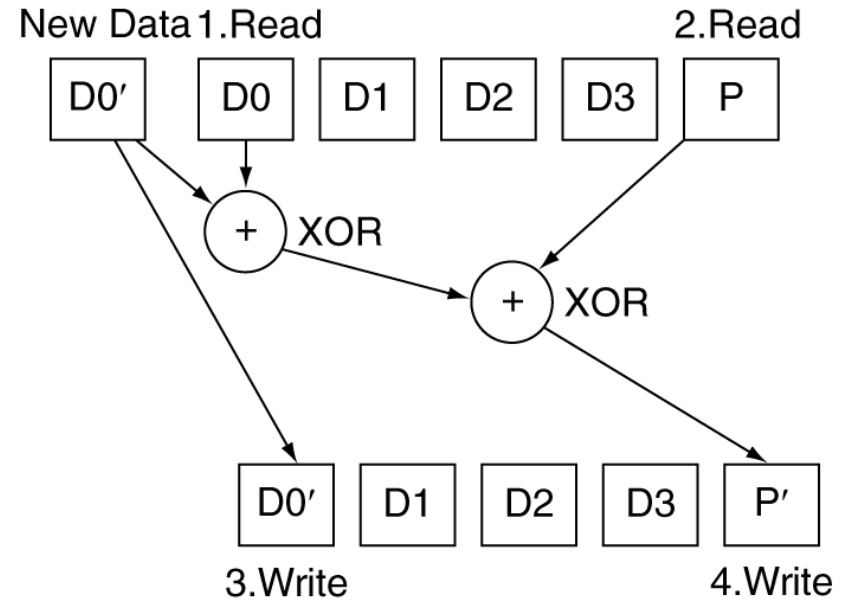
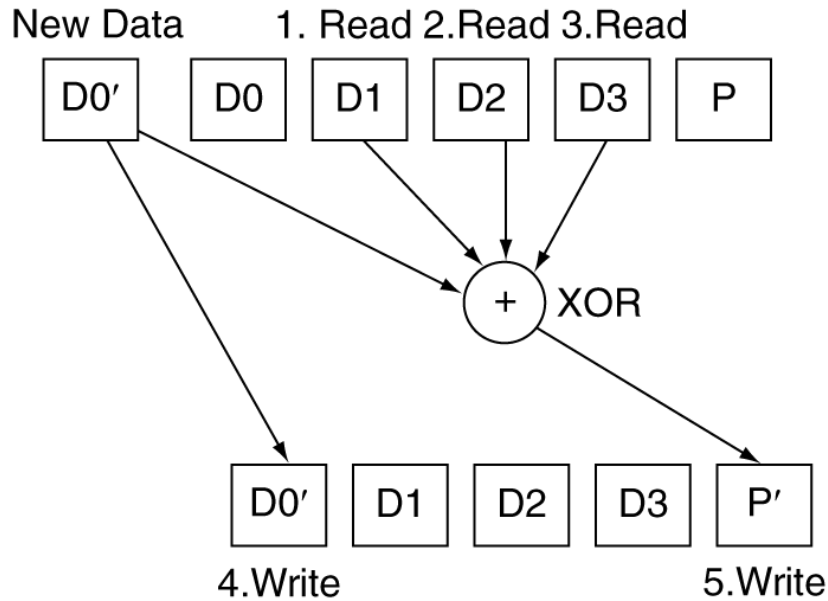


# RAID 4: Parity mức khối xen kẽ

- Số đĩa:  $N + 1$ 
  - Dữ liệu phân mảnh, chứa trên toàn bộ  $N$  đĩa ở mức khối
  - Đĩa dư thêm chứa thông tin parity cho 1 nhóm khối
  - Truy cập (đọc): Chỉ đọc những đĩa chứa khối cần đọc
  - Truy cập (ghi):
    - Đọc đĩa chứa khối bị thay đổi và đĩa parity
    - Tính lại parity mới, cập nhật đĩa chứa dữ liệu và đĩa parity
  - Khi có lỗi
    - Sử dụng parity để khôi phục dữ liệu lỗi

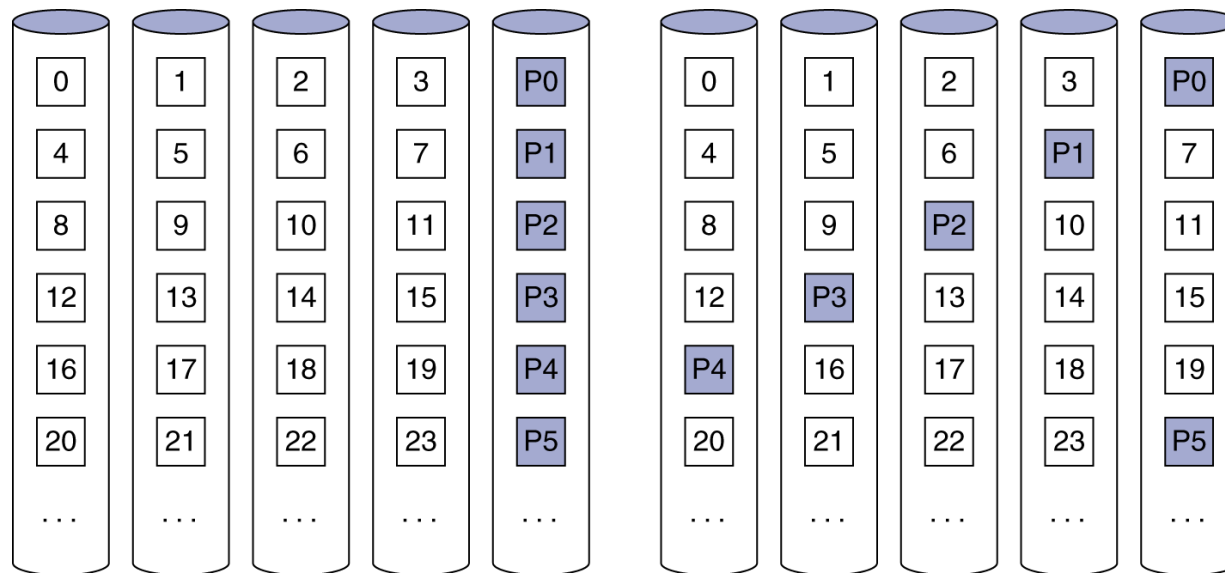
■ Không thông dụng

# So sánh RAID 3 & RAID 4



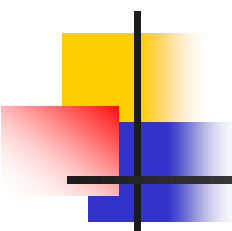
# RAID 5: Parity phân tán

- Số đĩa:  $N + 1$ 
  - Giống RAID 4, nhưng các khối parity phân tán khắp trên các đĩa
    - Tránh hiện tượng "cổ chai" với đĩa parity
- Thông dụng



RAID 4

RAID 5



# RAID 6: P + Q Dư thừa

---

- Số đĩa:  $N + 2$ 
  - Tương tự RAID 5, nhưng 2 đĩa chứa parity
  - Sửa lỗi tốt hơn do có parity dư thừa
- Đa RAID
  - Nhiều hệ thống tân tiến sử dụng phương thức dư thừa thông tin để sửa lỗi tương tự với hiệu suất tốt hơn



# Kết luận về RAID

- RAID cải thiện hiệu suất và tính sẵn sàng
  - Tính sẵn sàng cao đòi hỏi “thay nóng”
- Giả sử lỗi đĩa độc lập, không có mối quan hệ
  - Khả năng phục hồi thấp
- Tham khảo thêm “Hard Disk Performance, Quality and Reliability”
  - <http://www.pcguide.com/ref/hdd/perf/index.htm>





# Tiêu chí thiết kế hệ thống I/O

- Thỏa mãn các yêu cầu thời gian đáp ứng (latency)
  - Cho các tác vụ quan trọng (time-critical)
  - Khi hệ thống không tải (unloaded)
    - Cộng latency của các phần
- Tối đa Hiệu xuất đầu ra (throughput)
  - Phát hiện vị trí “cổ chai” “weakest link”
  - Cấu hình hoạt động ở mức băng thông tối đa
  - Cân bằng toàn bộ hệ thống
- Khi hệ thống có tải, việc phân tích rất phức tạp
  - Cần sử dụng mô hình “xếp hàng” hoặc mô phỏng



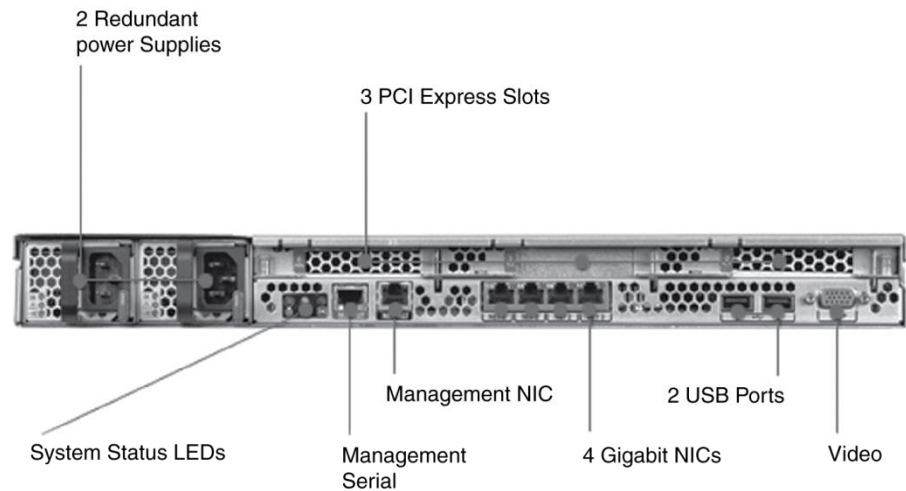
# Máy chủ (Servers)

- Ứng dụng ngày càng được chạy trên máy chủ
  - Web search, office apps, virtual worlds, ...
- Yêu cầu máy chủ làm trung tâm dữ liệu càng lớn
  - Đa xử lý, liên kết mạng, lưu trữ “khủng”
  - Không gian & năng lượng tiêu thụ hạn chế
- Thiết bị xây dựng trên dạng rack 19”
  - Dưới dạng nhiều module 1.75” (1U)

# Rack-Mounted Servers



Sun Fire x4150 1U server



2 Redundant  
power Supplies

3 PCI Express Slots

System Status LEDs

Management NIC

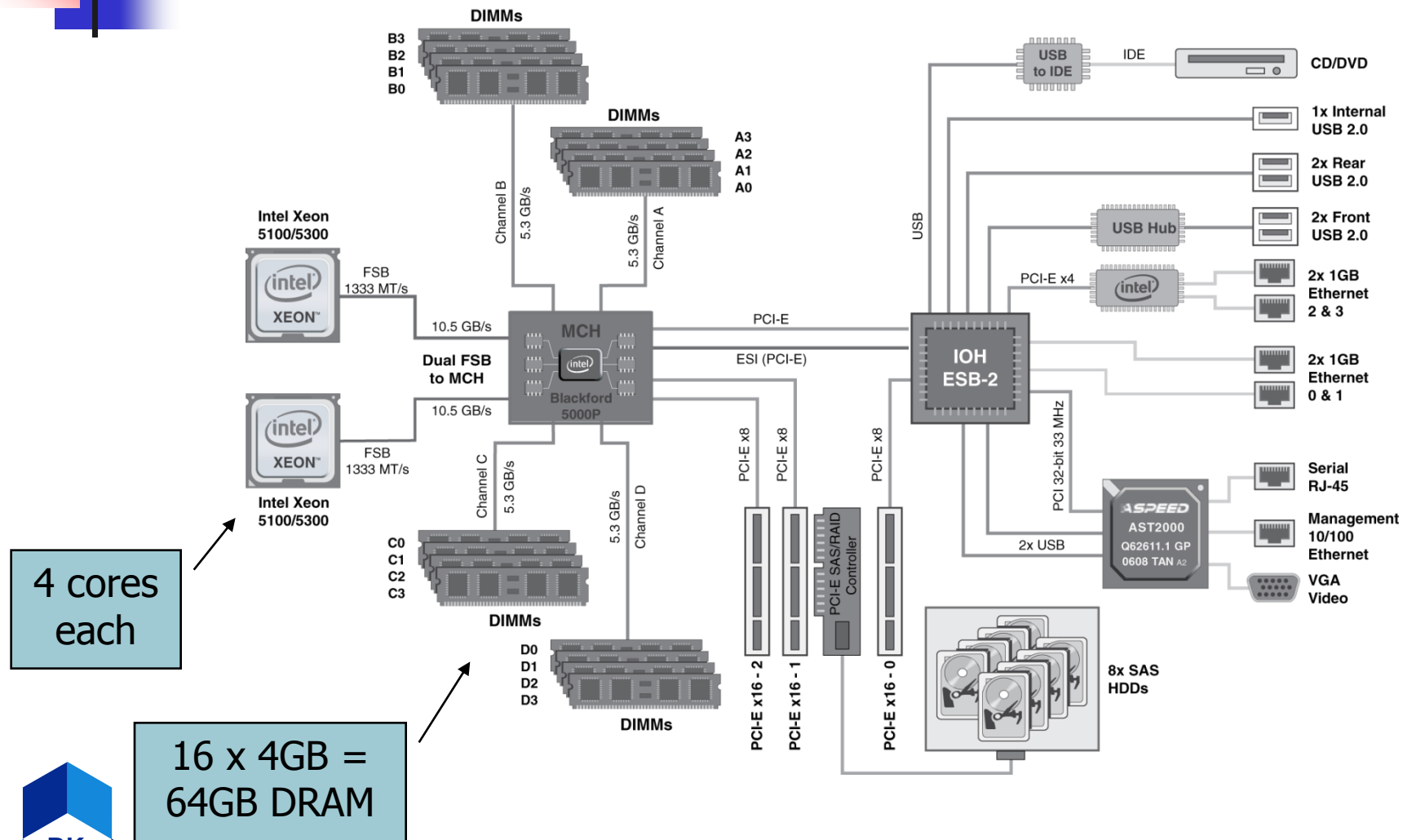
Management  
Serial

2 USB Ports

4 Gigabit NICs

Video

# Sun Fire x4150 1U server





# Ví dụ: Thiết kế hệ thống I/O

- Giả sử hệ thống Sun Fire x4150 với
  - Tải làm việc: đọc các khối đĩa 64KBytes
    - Mỗi tác vụ cần 200,000 lệnh ứng dụng & 100,000 lệnh thuộc OS
  - Mỗi CPU:  $10^9$  lệnh/giây
  - FSB: 10.6 GB/giây tốc độ tối đa
  - DRAM DDR2 667MHz: 5.336 GB/giây
  - PCI-E 8× bus:  $8 \times 250\text{MB/sec} = 2\text{GB/sec}$
  - Đĩa: tốc độ quay 15,000 rpm, thời gian dò 2.9ms, Tốc độ truyền dữ liệu 112MB/giây
- Tốc độ I/O tối đa để đảm bảo yêu cầu trên
  - Đọc random và tuần tự



# Thiết kế hệ thống I/O (tt.)

- Tốc độ I/O với tốc độ xử lý CPUs
  - Mỗi core:  $10^9 / (100,000 + 200,000) = 3,333$  tác vụ
  - 8 cores: 26,667 ops/sec ( $3,333 \times 8$ ) tác vụ/giây
- Đọc ngẫu nhiên, Tốc độ I/O với đĩa
  - Giả sử thời gian dò tìm là 25% theo thông số
  - $\text{Time/op} = \text{seek} + \text{latency} + \text{transfer}$   
 $= 2.9\text{ms}/4 + 4\text{ms}/2 + 64\text{KB}/(112\text{MB/s}) = 3.3\text{ms}$
  - Mỗi giây là 1000ms  $\rightarrow 1000\text{ms}/3.3\text{ms} = 303$  op/s
  - 303 ops/sec per disk, 2424 ops/sec for 8 disks
- Đọc liên tục:  $112\text{MB/s} / 64\text{KB} = 1750$  ops/sec per disk và 14,000 ops/sec for 8 disks



# Thiết kế hệ thống I/O (tt.)

- PCI-E I/O rate
  - $2\text{GB/sec} / 64\text{KB} = 31,250 \text{ ops/sec}$
- DRAM I/O rate
  - $5.336 \text{ GB/sec} / 64\text{KB} = 83,375 \text{ ops/sec}$
- FSB I/O rate
  - Giả sử  $\frac{1}{2}$  peak rate được duy trì
  - $5.3 \text{ GB/sec} / 64\text{KB} = 81,540 \text{ ops/sec per FSB}$
  - $163,080 \text{ ops/sec for 2 FSBs}$
- Nơi yếu nhất (weakest link): chính là đĩa
  - $2424 \text{ ops/sec random, } 14,000 \text{ ops/sec sequential}$
  - Tất cả các bộ phận khác đều thỏa mãn để đáp ứng đòi hỏi truy xuất đĩa

# Ví dụ: Tính độ tin cậy đĩa

- Nếu nhà sản xuất cho biết giá trị MTTF là 1,200,000 giờ (140 năm)
  - Sẽ hiểu rằng nó làm việc cho đến khi đó (140 năm)
- Sai: Đó chỉ là thời gian trung bình đến khi lỗi có thể xảy ra
  - Phân bố lỗi ?
  - Lỗi sẽ ra sao khi có 1000 đĩa?
    - Bao nhiêu lỗi xảy ra trong năm

$$\text{Annual Failure Rate (AFR)} = \frac{1000 \text{ disks} \times 8760 \text{ hrs/disk}}{1200000 \text{ hrs/failure}} = 0.73\%$$





# Tổng kết chương

---

- Đo hiệu xuất thiết bị I/O
  - Throughput, response time
  - Dependability and cost also important
- 2 loại tuyến “Buses” kết nối các thành phần CPU, memory, thiết bị đ/khiển I/O
  - Cơ chế hoạt động: Polling, interrupts, DMA
- Đo đặc hiệu xuất I/O
  - TPC, SPECSFS, SPECWeb
- RAID
  - Cải thiện hiệu xuất và độ tin cậy