

UNIVERSITY OF MASSACHUSETTS
 Dept. of Electrical & Computer Engineering

Computer Architecture
 ECE 568

Part 17

I/O Performance

Israel Koren
 Fall 2011

ECE568/Koren Part.17 .1

Copyright 2011 Koren UMass

Example 1: I/O - Compute Overlap Analysis
 Transaction Processing

◆ Assumptions:

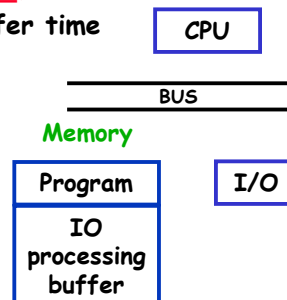
- Blocks of r uniform-size records are read & processed
- Flow of records & computation reached a steady state
- Transfer time for r records: $t_b = t_a + r k_f$

t_a : access time, k_f : one record transfer time

- Process time: $t_p = t_s + r k_p$

t_s : time to initiate processing;
 k_p : processing time per record

- ◆ Assuming I/O transfer and compute are totally independent
- ◆ In practice: $k_p = k'_p + k_i$
 net processing time + interference time
- ◆ Cycle stealing permits concurrency but delays computation



ECE568/Koren Part.17 .2

H. Hellerman, "Digital Computer System Principles," McGraw-Hill, 1967.

Example 1: I/O - Compute Overlap (2)

Sequential system (No IOP)
total time = $t_b + t_p$ ($k_i=0$)

Overlapped system (One IOP)
total time = $\max(t_b, t_p)$ ($k_i \neq 0$)

k_i depends on the organization & bandwidth of the memory (e.g., level of interleaving), & bandwidth of bus

Degree of overlap: $u = t_p/t_b = (t_s + r k_p)/(t_a + r k_f)$
 $u > 1$ $t_p > t_b$ processor - bound, (computation is bottleneck)
 $u < 1$ $t_p < t_b$ I/O - bound
 $u = 1$ $t_p = t_b$ balanced - ideal, (100% utilization)

The only system parameter controlled by the user is r
Two buffers of size r records are needed

Memory

Program
IO buffer 1
IO buffer 2

ECE568/Koren Part.17.3

Copyright 2011 Koren UMass

Example 1: I/O - Compute Overlap (3)

$$r = \lceil (u t_a - t_s)/(k_p - u k_f) \rceil \quad (u \uparrow \Rightarrow r \uparrow)$$

For a balanced system ($u=1$) $r_0 = \lceil (t_a - t_s)/(k_p - k_f) \rceil$

if $k_p \approx k_f$ (and $t_a \neq t_s$) $r_0 \Rightarrow \infty$
a balanced system can not be achieved

Little is gained as r increases beyond a certain point,
only waste memory space: $\text{memory_cost} = 2 r \times \text{record_size}$

processing time per record: $t_p/r = k_p + t_s/r$

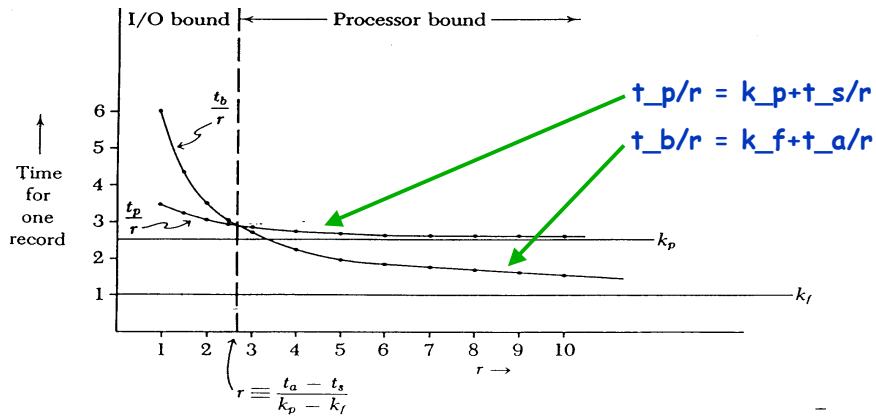
I/O time per record: $t_b/r = k_f + t_a/r$

Total time per record = $\max\{t_p/r; t_b/r\}$

ECE568/Koren Part.17.4

Copyright 2011 Koren UMass

Example 1: I/O - Compute Overlap (4)



If $t_a > t_s$ & $k_f > k_p \Rightarrow$ Always I/O bound ($u < 1$)
 If $t_a < t_s$ & $k_f < k_p \Rightarrow$ Always compute bound ($u > 1$)
 Balanced system is not possible in these two cases

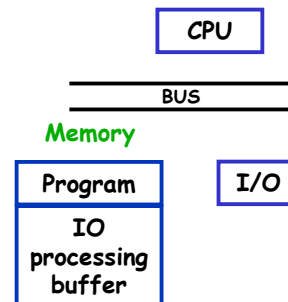
ECE568/Koren Part.17.5

H. Hellerman, "Digital Computer System Principles," McGraw-Hill, 1967.

Example 2: I/O - Compute Overlap

◆ **Given:** Batch processing

- The job has k steps, each step has 3 phases: Input - Compute - Output: $I_j - C_j - O_j$
- Process N data records of fixed size but memory can store only $m < N$ records
- t_{io} : i/o time per record; t_c : compute time per record



◆ Compare three organizations:

- (1) CPU only (no IOP): m records can be processed per step
- (2) One IOP allowing $C_j - I_{j+1}$ and $O_j - C_{j+1}$ overlap; only $m/2$ records can be transferred/computed per step
- (3) Two IOPs allowing $O_{j-1} - C_j - I_{j+1}$ overlap; only $m/3$ records can be transferred/computed per step

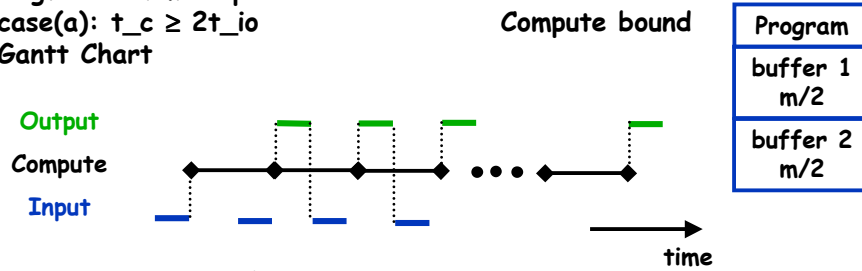
ECE568/Koren Part.17.6

J. L. Baer, "Computer Systems Architecture," Computer Science Press, 1980.

Timing Analysis - Organization 1 & 2

Org. 1: N/m steps,
each step takes $m(t_{io} + t_c + t_{io})$ time units
for a total of $N(2t_{io} + t_c)$

Org. 2: $2N/m$ steps
case(a): $t_c \geq 2t_{io}$
Gantt Chart



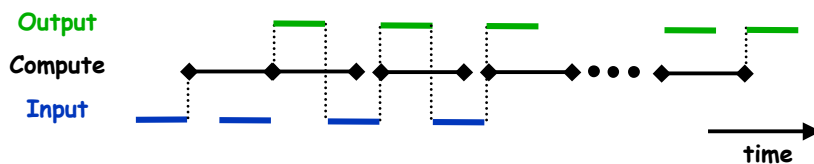
Each step takes $m/2 t_c$ except the 1st & last
Total of $N t_c + 2 m/2 t_{io} = N t_c + m t_{io}$

ECE568/Koren Part.17 .7

Copyright 2011 Koren UMass

Timing Analysis - Organization 2

Case(b): $t_c < 2t_{io}$ and $t_c > t_{io}$ I/O bound



Each step takes $m/2 2 t_{io}$ time units except the 1st & last
Total of $2N/m 2m/2 t_{io} + 2 m/2 (t_c - t_{io}) = 2N t_{io} + m(t_c - t_{io})$

case (c): $t_c < t_{io}$; Total time = $2N t_{io}$

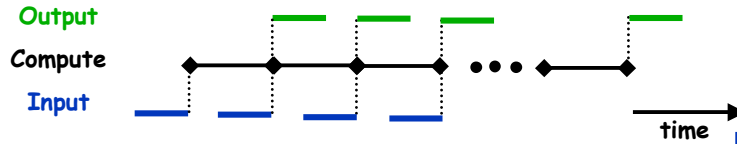
ECE568/Koren Part.17 .8

Copyright 2011 Koren UMass

Timing Analysis - Organization 3

Org. 3 (2 IOPs): $3N/m$ steps

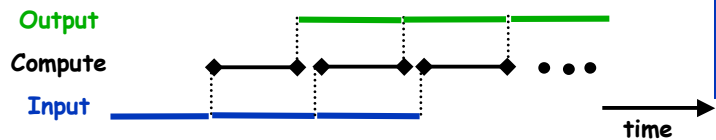
Case (a): $t_c \geq t_{io}$ CPU bound



Each step takes $m/3 t_c$ except the 1st & last

Total of $N t_c + 2 m/3 t_{io}$

Case (b): $t_c < t_{io}$ I/O bound



Each step takes $m/3 t_{io}$ except the 1st (or last). Total of $3N/m m/3 t_{io} + m/3 (t_c + t_{io}) = N t_{io} + m/3 (t_c + t_{io})$

Program
buffer 1 $m/3$
buffer 2 $m/3$
buffer 2 $m/3$

ECE568/Koren Part.17 .9

Copyright 2011 Koren UMass

Timing Analysis - Summary

Org.	Compute bound	I/O bound
CPU only	$N t_c + 2N t_{io}$	$2N t_{io} + N t_c$
1 IOP	$N t_c + m t_{io}$	$2N t_{io} + m(t_c - t_{io}) = (2N - m)t_{io} + m t_c$
2 IOPs	$N t_c + 2/3 m t_{io}$	$(N + m/3)t_{io} + m/3 t_c$

(1) For I/O bound jobs: overlapping with IOPs is always worthwhile

(2) For CPU bound jobs: overlapping with 2 (or more) IOPs is not worthwhile

ECE568/Koren Part.17 .10

Copyright 2011 Koren UMass

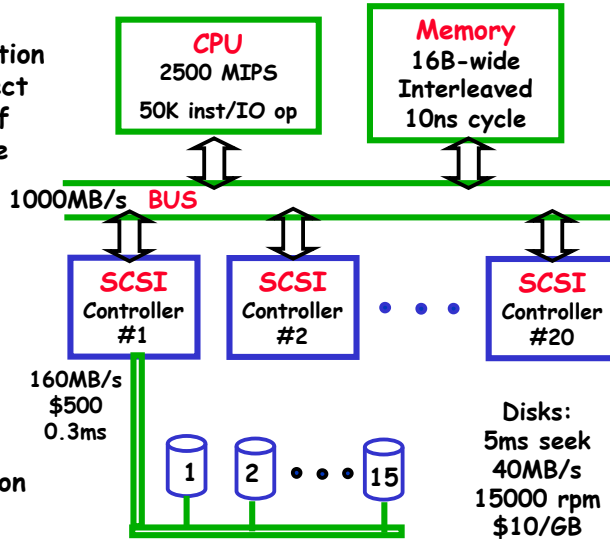
I/O Performance - Transaction Processing (P&H p.744)

Determine preferred I/O system organization (maximum rate): select between two types of disks & decide on the way they should be connected

Total disk capacity required = 1920GB

- (i) 80GB disk - 24
- (ii) 40GB disk - 48

Average I/O operation size = 32KB



ECE568/Koren Part.17 .11

Copyright 2011 Koren UMass

IOPs for each Link

Maximum performance of links in the I/O chain:

CPU: 2500 MIPS/50,000 instructions = 50,000 IOPS

MEM: 16B wide interleaved, 10ns; $(16/10\text{ns})/32\text{KB} = 50,000$ IOPS

I/O Bus: $1000\text{MB/s} / 32\text{KB} = 31,250$ IOPS

SCSI controller & bus: transfer time = $32\text{KB}/160\text{MB/sec} = .2\text{msec}$

+ .3ms controller overhead = .5msec per I/O

Max. of $1/.5\text{msec} = 2000$ IOPS per SCSI controller

but several controllers will be used

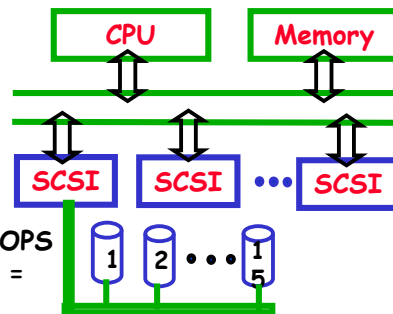
Both **Disks:** 15,000 RPM = 250 RPS; 5ms avg. seek time;

Transfer time of 32Kb = $32\text{KB} / 40\text{MB/sec} = 0.8\text{msec}$

Total time per I/O operation = avg. seek time + avg. rotational delay + avg. data transfer time

Avg. rotational delay = $1/2 \cdot 1/250 \text{ sec} = 2\text{msec}$

Total time = $5\text{ms} + 2\text{ms} + .8\text{msec} = 7.8\text{msec}$



ECE568/Koren Part.17 .12

Copyright 2011 Koren UMass

Disks & SCSI Controllers

Maximum IOPS per disk = $1/7.8 = 128$ IOPS

(1) 24 80GB disks: $24 \times 128 = 3072$ IOPS

(2) 48 40GB disks: $48 \times 128 = 6144$ IOPS

New limit to I/O performance

Determine how many SCSI (busses and controllers) to use (up to 20) and how many disks to connect to each (up to 15).

Minimum number of SCSI controllers Max. IOPS

(1) 24 80GB: $\lceil 24/15 \rceil = 2$; $2 \times 2000 = 4000$ IOPS

(2) 48 40GB: $\lceil 48/15 \rceil = 4$; $4 \times 2000 = 8000$ IOPS

Enclosures (\$1500) supply power and cooling to either 8 80GB or 12 40GB disks per enclosure:

4 enclosures for 4 strings of 12 40GB disks each - OK

but can not use 3 enclosures for 2 strings of 80GB disks - must use 3 strings (controllers)

(1) 24 80GB: 3 SCSI controllers; $3 \times 2000 = 6000$ IOPS

(1) 24 80GB disks: $\text{Min}\{50K; 31250; 3072, 6000\} = 3072$ IOPS

(2) 48 40GB disks: $\text{Min}\{50K, 31250, 6144, 8000\} = 6144$ IOPS

ECE568/Koren Part.17 .13

Copyright 2011 Koren UMass

Cost per IOP

Cost: SCSI controller - \$500, enclosure - \$1.5K,

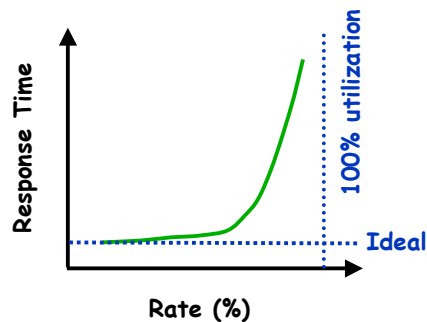
Disk: 1920GB at \$10 per GB = \$19.2K

Cost per IOP:

(1) $[3 \times 500 + 3 \times 1.5K + 19.2K] / 3072 = \$8.2/\text{IOP}$

(2) $[4 \times 500 + 4 \times 1.5K + 19.2K] / 6144 = \$4.4/\text{IOP}$ - better cost/performance ratio (times 1.86)

So far assumed all IOPS are evenly distributed over all controllers and disks. In practice, some resources will be overloaded and the latency (for getting the transaction processed) may increase dramatically. We should not have utilization close to 100%



ECE568/Koren Part.17 .14

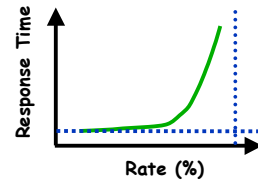
Copyright 2011 Koren UMass

Utilization of Resources

$$\text{Response_Time} = \text{Server_time} \times \left(1 + \frac{\text{Server_utilization}}{\#_of_servers \times (1 - \text{Server_utilization})} \right)$$

80GB disk: Server_time=7.8ms; 24 servers

Server_utilization	80%	90%	95%
Response_time	9.05ms	10.7ms	13.96ms



	Org.1 (80GB)	Org.2 (40GB)
CPU	3072/50000=6.1%	6144/50000=12.3%
Memory	3072/50000=6.1%	6144/50000=12.3%
Sys.IO Bus	3072/31250=9.8%	6144/31250=19.6%
SCSI Bus	3072/6000=51.2%	6144/8000=76.8%
Disks	3072/3072=100%	6144/6144=100%
Seek Util.	5ms/(1/128 IOPS)=64%	5ms/(1/128)=64%

ECE568/Koren Part.17 .15

Copyright 2011 Koren UMass

Empirical rules of thumb for utilization

Bus utilization ≤ 75%; Controller utilization ≤ 40%
 Disk arm seek ≤ 60% of the time; Disk utilization ≤ 80%

Disks: 128 × 80% = 102 IOPS

Seek utilization = 5 ms per seek / (1/102) = 5/9.8 = 51%

I/O bus utilization, in both cases, is below 75%

SCSI controller - currently 12×102/2000=61.2% instead of 40%

For 40% utilization we should expect only 2000×0.40 = 800 IOPS

(1) 80GB disks - 3×800 = 2400 IOPS

(2) 40GB disks - 4×800 = 3200 IOPS

The SCSI controllers became the bottleneck

(1) 24 80GB disks: Min{50K;31250;2448,2400} = 2400 IOPS

(2) 48 40GB disks: Min{50K,31250,4896,3200} = 3200 IOPS

ECE568/Koren Part.17 .16

Copyright 2011 Koren UMass

Increase number of controllers

Disks per controller: $\lfloor 800/102 \rfloor = \lfloor 7.8 \rfloor = 7$

(1) 24 80GB disks = $\lceil 24/7 \rceil = \lceil 3.6 \rceil = 4$ controllers (4 x 6)

(2) 48 40GB disks = $\lceil 48/7 \rceil = \lceil 6.9 \rceil = 7$ controllers (6 x 7 +6)

Enclosure for 8 80GB or 12 40GB: 4 enclosures for (1);

8 controllers for (2) + 4 enclosures

(1) 24 80GB disks: Min{50K,31250,2448,3200} = 2448 IOPS

(2) 48 40GB disks: Min{50K,31250,4896,6400} = 4866 IOPS

Cost per IOP:

(1) $[4 \times 500 + 4 \times 1.5K + 19.2K] / 2448 = \$11/\text{IOP}$

(2) $[8 \times 500 + 4 \times 1.5K + 19.2K] / 4866 = \$6/\text{IOP}$ - better cost/performance ratio (times 1.83)

The cost per IOP increased from \$4.4 to \$6 (times 1.36)

Utilization of each resource

	Org.1 (80GB, 4 SCSI)	Org.2 (40GB, 8 SCSI)
CPU	$2448/50000=5\%$	$4896/50000=10\%$
Memory	$2448/50000=5\%$	$4896/50000=10\%$
Sys.I/O Bus	$2448/31250=8\%$	$4896/31250=16\%$
SCSI Bus	$2448/(4 \times 2000)=31\%$	$4896/(8 \times 2000)=31\%$
Disks	$2448/(24 \times 128)=80\%$	$4896/(48 \times 128)=80\%$
Seek Util.	$5\text{msec}/(1/102)=51\%$	$5\text{msec}/(1/102)=51\%$

90% CPU still available for other applications