I/O System Design Took and MIPS Assembly

Colt Uhlenhopp & Jack Utzerath

Grand Canyon University

CST-307

Professor Citro

2/14/2023

I/O System Design Tool and MIPS Assembly

Part 1: Answer the questions below

Consider the default input settings given by the tool. Only change is that all resource utilizations are 100 %.

1. List all the links that decide the performance of the system. What is the bottleneck in the current system?

After changing the resource utilizations to 100% on the given simulator, the links that decide the performance of the system are; the CPU, I/O Bus, Main Memory, Disks, Seek Utilization, IOPS, and the SCSI Buses.

The disk seems to be the bottleneck in the current system. All of the other links appear that they could have a higher performance but the disk is at its max utilization.

	Assuming 100% utilization		Assuming the constraints on Resource utilization			
Resource	**80GB disks,2 strings**	**40GB disks,4 strings**	**80GB disks,2strings**	**40GB disks,4strings**	**80GB disks,2strings**	**40GB disks,4strings**
CPU	6%	12%	6%	12%	6%	12%
Main memory	6%	12%	6%	12%	6%	12%
I/O Bus	10%	20%	10%	20%	10%	20%
SCSI Buses	77%	77%	77%	77%	77%	77%
Disks	100%	100%	100%	100%	100%	100%
Seek utilization	64%	64%	64%	64%	64%	64%
IOPS	3072	6144	3072	6144	3072	6144

2. What can lead to the I/O bus performance to be the bottleneck of the system?

One way to decrease the performance of the I/O Bus, would be to reduce the I/O bus bandwidth. If the bandwidth is decreased enough, the I/O Bus could even become the bottleneck in the system. The amount of data passed through the I/O Bus would be too small to control the amount of data coming in and out.

By decreasing the I/O Bus Bandwidth and decreasing the large disk size, the performance of the I/O Bus increased to 100%. (Meaning the I/O Bus is the bottleneck since no other resource is at maximum utilization)

	Assuming 100% utilization		Assuming the constraints on Resource utilization			
Resource	**50GB disks,3 strings**	**40GB disks,4 strings**	**50GB disks,3strings**	**40GB disks,4strings**	**50GB disks,3strings**	**40GB disks,4strings**
CPU	6%	6%	6%	6%	6%	6%
Main memory	6%	6%	6%	6%	6%	6%
I/O Bus	100%	100%	100%	100%	100%	100%
SCSI Buses	52%	39%	52%	39%	52%	39%
Disks	64%	51%	64%	51%	64%	51%
Seek utilization	41%	33%	41%	33%	41%	33%
IOPS	3125	3125	3125	3125	3125	3125

3. Make necessary change(s) to the system to increase the performance (IOPS). List all the change(s) made and all other possible changes. Give performance measures in each if possible. Also mention if the method is feasible.

Control Table (No values changes)

Input Disk specifications:

Disk size =	*small disk*	40 GB	*large disk*	80 GB
Cost of Disk per GB =	*small disk*	10 $	*large disk*	10 $
Maximum number of disks per package =	*small disk*	12	*large disk*	8
Disk Rotation speed =	*small disk*	15000 RPM	*large disk*	15000 RPM
Average seek time =	*small disk*	5 ms	*large disk*	5 ms
Disk Transfer rate =	*small disk*	40 MB/sec	*large disk*	40 MB/sec
Desired total disk capacity =	1920 GB			
Cost of enclosure =	1500 $			

Constraints on Resource Utilization:

Limit on disk time utilization = 100 % | Limit on utilization of SCSI buses = 100 % | Limit on utilization of I/O bus = 100 %

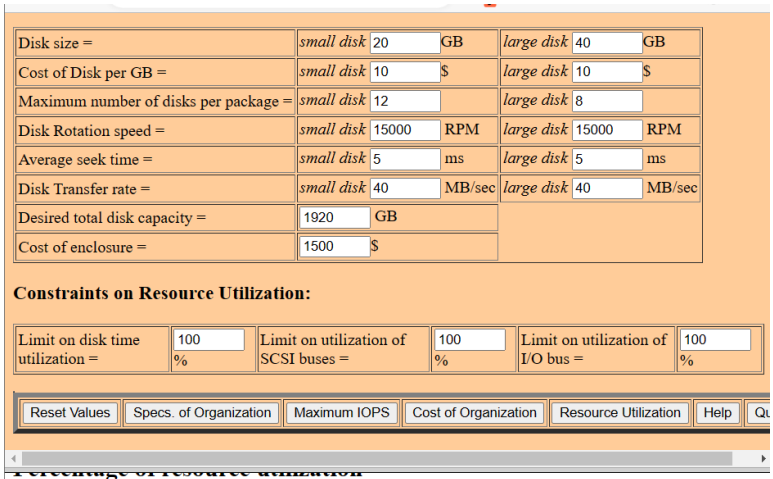
Reset Values | Specs. of Organization | Maximum IOPS | Cost of Organization | Resource Utilization | Help | Quit

Output Results

Percentage of resource utilization

	Assuming 100% utilization		Assuming the constraints on Resource utilization			
Resource	**80GB disks,2 strings**	**40GB disks,4 strings**	**80GB disks,2strings**	**40GB disks,4strings**	**80GB disks,2strings**	**40GB disks,4strings**
CPU	6%	12%	6%	12%	6%	12%
Main memory	6%	12%	6%	12%	6%	12%
I/O Bus	10%	20%	10%	20%	10%	20%
SCSI Buses	77%	77%	77%	77%	77%	77%
Disks	100%	100%	100%	100%	100%	100%
Seek utilization	64%	64%	64%	64%	64%	64%
IOPS	3072	6144	3072	6144	3072	6144

In terms of performance, the disk size is inversely proportional to the IOPS performance. This means that decreasing the disk size will increase the IOPS performance. By decreasing both the small disk and large disk by half, the IOPS doubles.



Disk size =	*small disk*	20	GB	*large disk* 40	GB
Cost of Disk per GB =	*small disk*	10	$	*large disk* 10	$
Maximum number of disks per package =	*small disk*	12		*large disk* 8	
Disk Rotation speed =	*small disk*	15000	RPM	*large disk* 15000	RPM
Average seek time =	*small disk*	5	ms	*large disk* 5	ms
Disk Transfer rate =	*small disk*	40	MB/sec	*large disk* 40	MB/sec
Desired total disk capacity =	1920	GB			
Cost of enclosure =	1500	$			

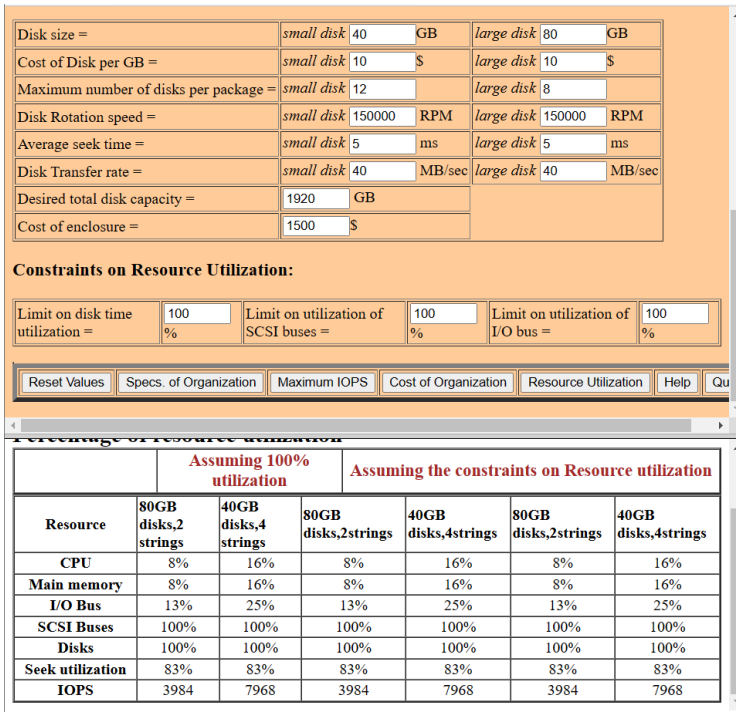
Constraints on Resource Utilization:

Limit on disk time utilization =	100 %	Limit on utilization of SCSI buses =	100 %	Limit on utilization of I/O bus =	100 %

Reset Values | Specs. of Organization | Maximum IOPS | Cost of Organization | Resource Utilization | Help | Qu

	Assuming 100% utilization		Assuming the constraints on Resource utilization			
Resource	**40GB disks,4 strings**	**20GB disks,7 strings**	**40GB disks,4strings**	**20GB disks,7strings**	**40GB disks,4strings**	**20GB disks,7strings**
CPU	12%	25%	12%	25%	12%	25%
Main memory	12%	25%	12%	25%	12%	25%
I/O Bus	20%	39%	20%	39%	20%	39%
SCSI Buses	77%	88%	77%	88%	77%	88%
Disks	100%	100%	100%	100%	100%	100%
Seek utilization	64%	64%	64%	64%	64%	64%
IOPS	6144	12288	6144	12288	6144	12288

Another change that increases performance of the IOPS is the disk rotation speed. As the rotation speed increased from 15000 RPM to 150000 RPM, the IOPS increased by almost a 1000 for the 80 GB disk, and almost 2000 for the 40 GB disk. This option is not a feasible option because changing the disk rotation speed will also affect the disk performance negatively.



Disk size =	*small disk* 40	GB	*large disk* 80	GB	
Cost of Disk per GB =	*small disk* 10	$	*large disk* 10	$	
Maximum number of disks per package =	*small disk* 12		*large disk* 8		
Disk Rotation speed =	*small disk* 150000	RPM	*large disk* 150000	RPM	
Average seek time =	*small disk* 5	ms	*large disk* 5	ms	
Disk Transfer rate =	*small disk* 40	MB/sec	*large disk* 40	MB/sec	
Desired total disk capacity =	1920	GB			
Cost of enclosure =	1500	$			

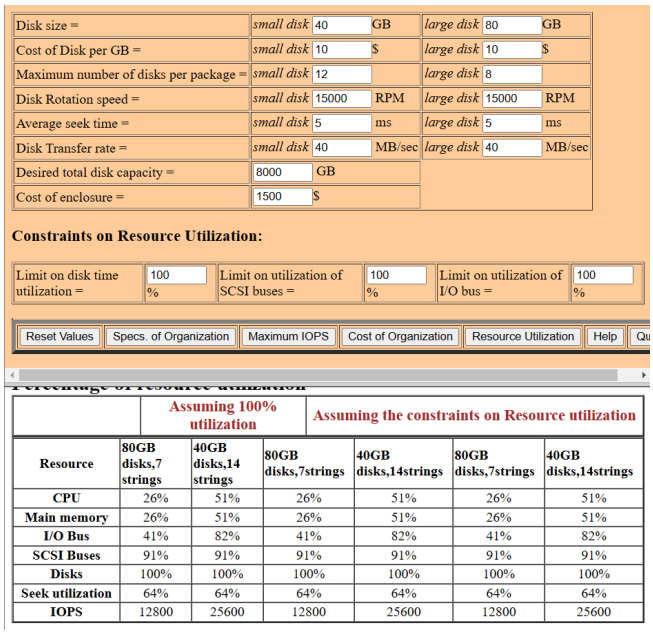
Constraints on Resource Utilization:

Limit on disk time utilization =	100 %	Limit on utilization of SCSI buses =	100 %	Limit on utilization of I/O bus =	100 %

Reset Values | Specs. of Organization | Maximum IOPS | Cost of Organization | Resource Utilization | Help | Qu

	Assuming 100% utilization		Assuming the constraints on Resource utilization			
Resource	**80GB disks,2 strings**	**40GB disks,4 strings**	**80GB disks,2strings**	**40GB disks,4strings**	**80GB disks,2strings**	**40GB disks,4strings**
CPU	8%	16%	8%	16%	8%	16%
Main memory	8%	16%	8%	16%	8%	16%
I/O Bus	13%	25%	13%	25%	13%	25%
SCSI Buses	100%	100%	100%	100%	100%	100%
Disks	100%	100%	100%	100%	100%	100%
Seek utilization	83%	83%	83%	83%	83%	83%
IOPS	3984	7968	3984	7968	3984	7968

The desired total disk capacity also changes the performance of the IOPS. As the total disk capacity increased in GB, the IOPS increased as well. By changing the disk capacity to 8000 GB from 1920 GB, the IOPS rose over four times the value.



Disk size =	*small disk* 40	GB	*large disk* 80	GB	
Cost of Disk per GB =	*small disk* 10	$	*large disk* 10	$	
Maximum number of disks per package =	*small disk* 12		*large disk* 8		
Disk Rotation speed =	*small disk* 15000	RPM	*large disk* 15000	RPM	
Average seek time =	*small disk* 5	ms	*large disk* 5	ms	
Disk Transfer rate =	*small disk* 40	MB/sec	*large disk* 40	MB/sec	
Desired total disk capacity =	8000	GB			
Cost of enclosure =	1500	$			

Constraints on Resource Utilization:

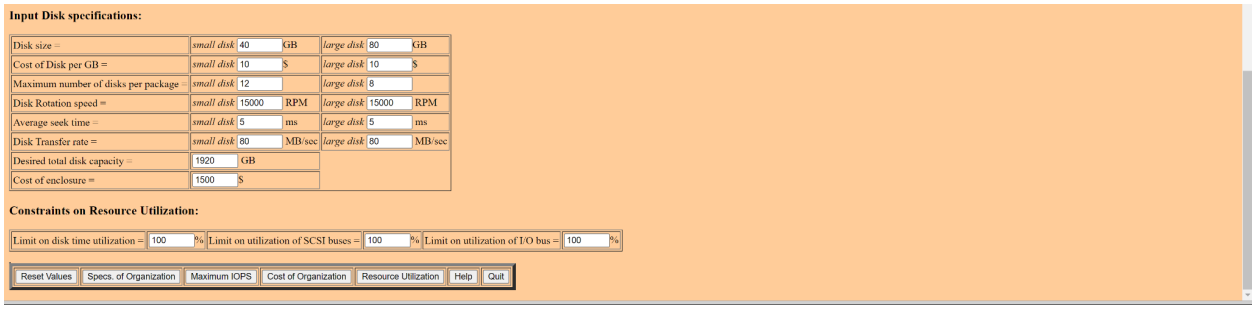
Limit on disk time utilization =	100 %	Limit on utilization of SCSI buses =	100 %	Limit on utilization of I/O bus =	100 %

Reset Values | Specs. of Organization | Maximum IOPS | Cost of Organization | Resource Utilization | Help | Qu

	Assuming 100% utilization		Assuming the constraints on Resource utilization			
Resource	**80GB disks,7 strings**	**40GB disks,14 strings**	**80GB disks,7strings**	**40GB disks,14strings**	**80GB disks,7strings**	**40GB disks,14strings**
CPU	26%	51%	26%	51%	26%	51%
Main memory	26%	51%	26%	51%	26%	51%
I/O Bus	41%	82%	41%	82%	41%	82%
SCSI Buses	91%	91%	91%	91%	91%	91%
Disks	100%	100%	100%	100%	100%	100%
Seek utilization	64%	64%	64%	64%	64%	64%
IOPS	12800	25600	12800	25600	12800	25600

The disk transfer rate will also slightly change the performance of the IOPS. In this image, I doubled the disk transfer rates. The disk transfer rate increases the performance of 40

GB disks, 4 strings by 336 operations per second (Original Value: 6144; New Value: 6480). This made a 5.2% increase in performance.



Input Disk specifications:

Disk size =	*small disk*	40	GB	*large disk*	80	GB
Cost of Disk per GB =	*small disk*	10	$	*large disk*	10	$
Maximum number of disks per package =	*small disk*	12		*large disk*	8	
Disk Rotation speed =	*small disk*	15000	RPM	*large disk*	15000	RPM
Average seek time =	*small disk*	5	ms	*large disk*	5	ms
Disk Transfer rate =	*small disk*	80	MB/sec	*large disk*	80	MB/sec
Desired total disk capacity =	1920	GB				
Cost of enclosure =	1500	$				

Constraints on Resource Utilization:

Limit on disk time utilization =	100	%	Limit on utilization of SCSI buses =	100	%
Limit on utilization of I/O bus =	100	%			

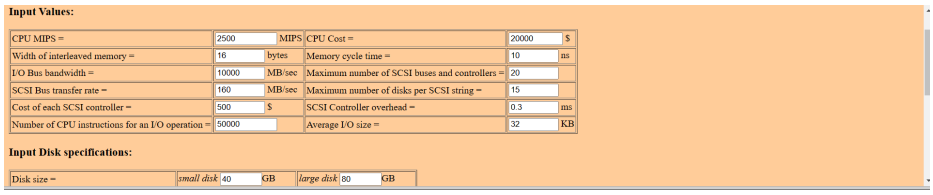
Reset Values | Specs. of Organization | Maximum IOPS | Cost of Organization | Resource Utilization | Help | Quit

Output Results

Percentage of resource utilization

	Assuming 100% utilization		Assuming the constraints on Resource utilization			
Resource	**80GB disks,2 strings**	**40GB disks,4 strings**	**80GB disks,2strings**	**40GB disks,4strings**	**80GB disks,2strings**	**40GB disks,4strings**
CPU	6%	13%	6%	13%	6%	13%
Main memory	6%	13%	6%	13%	6%	13%
I/O Bus	10%	21%	10%	21%	10%	21%
SCSI Buses	81%	81%	81%	81%	81%	81%
Disks	100%	100%	100%	100%	100%	100%
Seek utilization	68%	68%	68%	68%	68%	68%
IOPS	3240	6480	3240	6480	3240	6480

Another thing that would increase the performance of the I/O Bus IOPS is changing the value of the I/O Bandwidth. As the I/O Bandwidth increased, the IOPS performance increased as well. This is a feasible change, as the I/O Bandwidth would not affect any of the other system's performance.



Input Values:

CPU MIPS =	2500	MIPS	CPU Cost =	20000	$
Width of interleaved memory =	16	bytes	Memory cycle time =	10	ns
I/O Bus bandwidth =	10000	MB/sec	Maximum number of SCSI buses and controllers =	20	
SCSI Bus transfer rate =	160	MB/sec	Maximum number of disks per SCSI string =	15	
Cost of each SCSI controller =	500	$	SCSI Controller overhead =	0.3	ms
Number of CPU instructions for an I/O operation =	50000		Average I/O size =	32	KB

Input Disk specifications:

Disk size =	*small disk*	40	GB	*large disk*	80	GB

Output Results

Maximum IOPS

	Assuming 100% utilization		Assuming the constraints on Resource utilization			
Resource	**80GB disks,2 strings**	**40GB disks,4 strings**	**80GB disks,2strings**	**40GB disks,4strings**	**80GB disks,2strings**	**40GB disks,4strings**
CPU	50000	50000	50000	50000	50000	50000
Main memory	50000	50000	50000	50000	50000	50000
I/O Bus	312500	312500	312500	312500	312500	312500
SCSI Buses	4000	8000	4000	8000	4000	8000
Disks	3072	6144	6144	3072	6144	3072

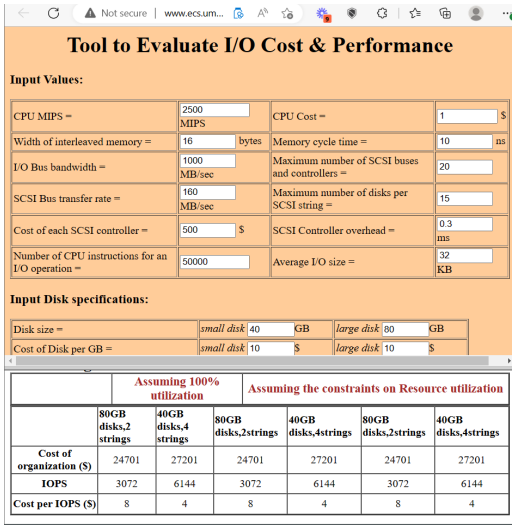
Consider the default input settings given by the tool. Only change is to make all resource utilizations to 100%.

1. What units cost the most in this system?

The unit that costs the most in this system is the CPU. The default setting for the CPU cost in this simulator is $20000. As a whole, the total cost of the organization is $44700.

	Assuming 100% utilization		Assuming the constraints on Resource utilization			
	80GB disks,2 strings	40GB disks,4 strings	80GB disks,2strings	40GB disks,4strings	80GB disks,2strings	40GB disks,4strings
Cost of organization ($)	44700	47200	44700	47200	44700	47200
IOPS	3072	6144	3072	6144	3072	6144
Cost per IOPS ($)	15	8	15	8	15	8

By pretending the CPU cost is $1, we can see how much the original cost of the CPU affects the total cost. As seen below, the price dropped by the entire $19999. This one unit affects the price more than any other part of the computer.



Tool to Evaluate I/O Cost & Performance

Input Values:

CPU MIPS =	2500 MIPS	CPU Cost =	1 $
Width of interleaved memory =	16 bytes	Memory cycle time =	10 ns
I/O Bus bandwidth =	1000 MB/sec	Maximum number of SCSI buses and controllers =	20
SCSI Bus transfer rate =	160 MB/sec	Maximum number of disks per SCSI string =	15
Cost of each SCSI controller =	500 $	SCSI Controller overhead =	0.3 ms
Number of CPU instructions for an I/O operation =	50000	Average I/O size =	32 KB

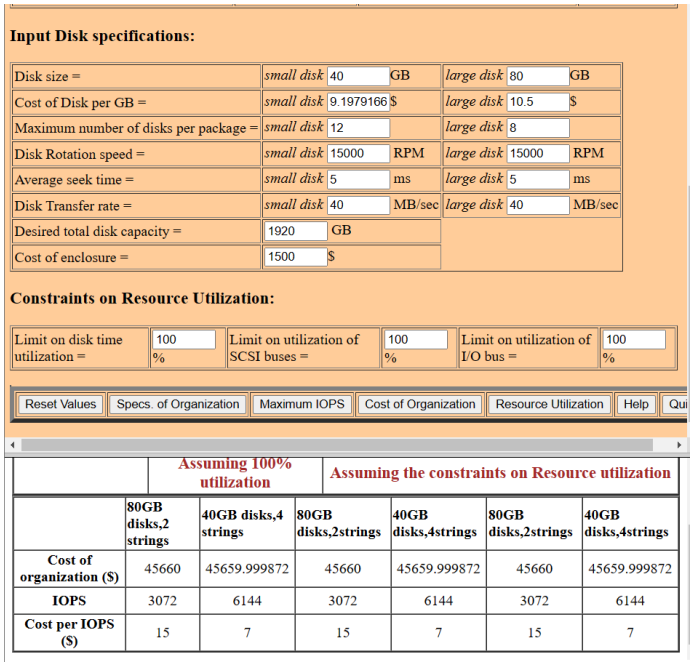
Input Disk specifications:

Disk size =	*small disk* 40	GB	*large disk* 80	GB
Cost of Disk per GB =	*small disk* 10	$	*large disk* 10	$

	Assuming 100% utilization		Assuming the constraints on Resource utilization			
	80GB disks,2 strings	40GB disks,4 strings	80GB disks,2strings	40GB disks,4strings	80GB disks,2strings	40GB disks,4strings
Cost of organization ($)	24701	27201	24701	27201	24701	27201
IOPS	3072	6144	3072	6144	3072	6144
Cost per IOPS ($)	8	4	8	4	8	4

2. What can be done to make the system cost the same for both 80GB and 40GB disks? Can it be done?

One way of making the system cost the same for both the 80GB and the 40GB disks is by changing the cost of disk per GB. Using the default settings, the 40GB disk's total cost is actually more than that of the 80GB disk. When the cost of disk per GB is changed to $9.1979, and the cost of the large disk changes to $10.5, then the total cost of the organization is almost exactly the same. This method is not really feasible since the smallest unit of cost in the American currency is a cent, but there are probably other methods to get the cost around the same.



Input Disk specifications:

Disk size =	*small disk*	40 GB	*large disk*	80	GB
Cost of Disk per GB =	*small disk*	9.1979166 $	*large disk*	10.5	$
Maximum number of disks per package =	*small disk*	12	*large disk*	8	
Disk Rotation speed =	*small disk*	15000 RPM	*large disk*	15000	RPM
Average seek time =	*small disk*	5 ms	*large disk*	5	ms
Disk Transfer rate =	*small disk*	40 MB/sec	*large disk*	40	MB/sec
Desired total disk capacity =	1920 GB				
Cost of enclosure =	1500 $				

Constraints on Resource Utilization:

Limit on disk time utilization =	100 %	Limit on utilization of SCSI buses =	100 %	Limit on utilization of I/O bus =	100 %

Reset Values | Specs. of Organization | Maximum IOPS | Cost of Organization | Resource Utilization | Help | Qui

	Assuming 100% utilization		Assuming the constraints on Resource utilization			
	80GB disks,2 strings	**40GB disks,4 strings**	**80GB disks,2strings**	**40GB disks,4strings**	**80GB disks,2strings**	**40GB disks,4strings**
Cost of organization ($)	45660	45659.999872	45660	45659.999872	45660	45659.999872
IOPS	3072	6144	3072	6144	3072	6144
Cost per IOPS ($)	15	7	15	7	15	7

We have assumed the resource utilizations to be 100% in the above cases. Is it the same in practice? Give reasons for your answer.

No, typically the resource utilizations are not at 100%. This is not even ideal. Having the resources at 100% at all times is like being on a completely full 7 lane highway during rush hour. For example, if a single processor CPU is at 100% utilization, it can only process one thing at a time, that may be ideal for a simple computer like a calculator, but for a larger machine with multi-processors you would not want that. A computer at incredibly high utilization has unpredictable performance.

Give the key function for a transaction processing system and batch processing system.

A key function for the transaction processing system and the batch processing system are their I/O functions. Data input is incredibly important for the processing systems because they handle large quantities of information very quickly, and any error or flaws in the input of data would ruin the entire batch.

Part 2: Palindrome Assembly Code

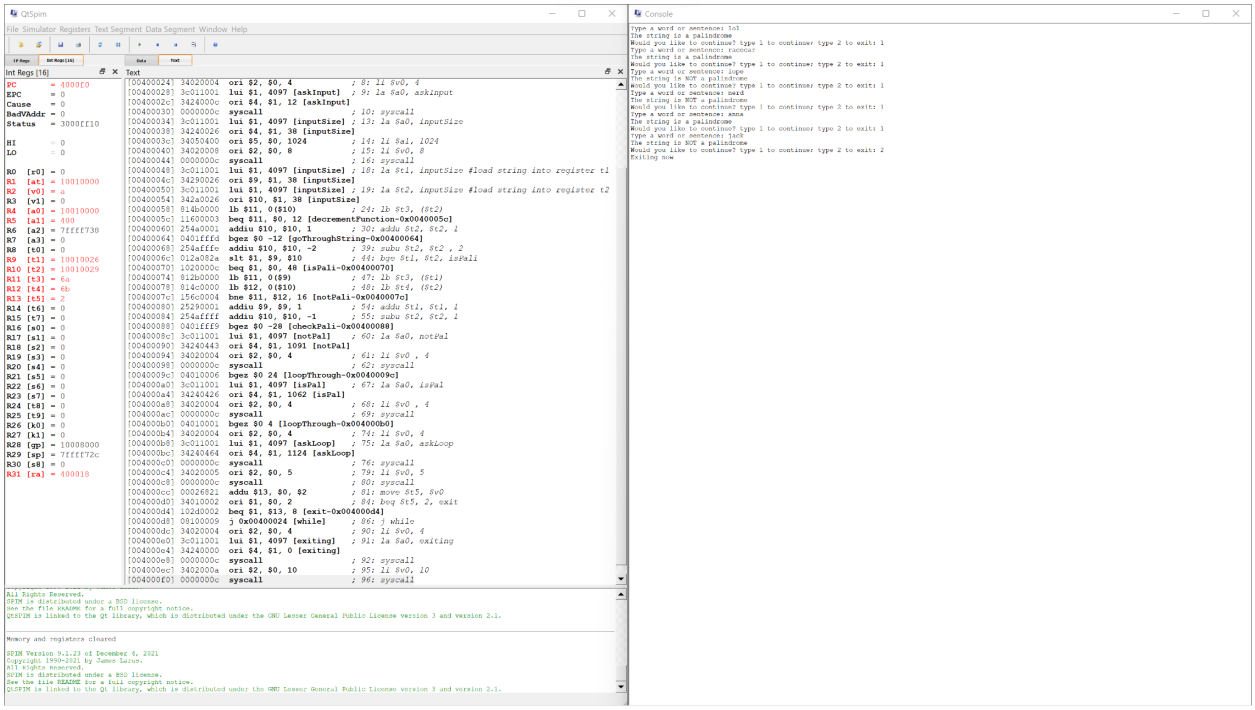
Do Not Copy and Paste This Code. Download .asm to run

```
#Jack Utzerath
#2/14/2023
#Palindrome Assignment

main:
while:
    #print string to console to ask user for input
    li $v0, 4
    la $a0, askInput
    syscall

    #read string from console:
    la $a0, inputSize
    li $a1, 1024
    li $v0, 8
    syscall

    la $t1, inputSize #load string into register t1
    la $t2, inputSize #load string into register t2

    #we need to loop through the string now
    goThroughString:
        #this will load the bytes into t3
        lb $t3, ($t2)

        #if the string is 0 then exit the loop
        beqz $t3, decrementFunction

        #now we increment like a for loop
        addu $t2, $t2, 1

        #continues the loop
        b goThroughString

    decrementFunction:

        #we are going subtract 2 to counteract the other loop
        #this will decrement it
        subu $t2, $t2 , 2
```

```
    checkPali:

        #if it is a palindrome
        bge $t1, $t2, isPali

        #load bytes of start of string into $t3 and end of string into $t4
        lb $t3, ($t1)
        lb $t4, ($t2)

        #if bytes in register 3 and 4 are not the same then its not a
      palindrome
        bne $t3, $t4, notPali

        #now we need to increment t1 and decrement t2 (move along string)
        addu $t1, $t1, 1
        subu $t2, $t2, 1
        b checkPali

    notPali:
        #print notPal to console (it is not a palindrome!)
        la $a0, notPal
        li $v0 , 4
        syscall
        b loopThrough

    isPali:
        #print isPal to console (it is a palindrome!)
        la $a0, isPal
        li $v0 , 4
        syscall
        b loopThrough

loopThrough:
    #ask user if they want to continue
    li $v0, 4
    la $a0, askLoop
    syscall

    #get user input
    li $v0, 5
```

```
    syscall
    move $t5, $v0

    #exit loop if input is 2
    beq $t5, 2, exit

    j while
exit:
    #exiting
    li $v0, 4
    la $a0, exiting
    syscall

    #exit the program
    li $v0, 10
    syscall




.data

#variables
exiting: .asciiz "Exiting now"
askInput: .asciiz "Type a word or sentence: "
inputSize: .space 1024 # setting aside 1024 bytes for string
isPal: .asciiz "The string is a palindrome \n"
notPal: .asciiz "The string is NOT a palindrome \n"
askLoop: .asciiz "Would you like to continue? type 1 to continue; type 2
     to exit: "
```

Example of Run:


QtSpim
File Simulator Registers Text Segment Data Segment Window Help
Int Regs [16]
PC = 4000f0
EPC = 0
Cause = 0
BadVAddr = 0
Status = 3000ff10
HI = 0
LO = 0
R0 [r0] = 0
R1 [at] = 10010000
R2 [v0] = a
R3 [v1] = 0
R4 [a0] = 10010000
R5 [a1] = 400
R6 [a2] = 7fffff38
R7 [a3] = 0
R8 [t0] = 0
R9 [t1] = 10010026
R10 [t2] = 10010029
R11 [t3] = 6a
R12 [t4] = 6b
R13 [t5] = 2
R14 [t6] = 0
R15 [t7] = 0
R16 [s0] = 0
R17 [s1] = 0
R18 [s2] = 0
R19 [s3] = 0
R20 [s4] = 0
R21 [s5] = 0
R22 [s6] = 0
R23 [s7] = 0
R24 [t8] = 0
R25 [t9] = 0
R26 [k0] = 0
R27 [k1] = 0
R28 [gp] = 10008000
R29 [sp] = 7fffff72c
R30 [s8] = 0
R31 [ra] = 400018
Text
[00400024] 34020004 ori $2, $0, 4 ; 8: li $v0, 4
[00400028] 3c011001 lui $1, 4097 [askInput] ; 9: la $a0, askInput
[0040002c] 3424000c ori $4, $1, 12 [askInput]
[00400030] 0000000c syscall ; 10: syscall
[00400034] 3c011001 lui $1, 4097 [inputSize] ; 13: la $a0, inputSize
[00400038] 34240026 ori $4, $1, 38 [inputSize]
[0040003c] 34050400 ori $5, $0, 1024 ; 14: li $a1, 1024
[00400040] 34020008 ori $2, $0, 8 ; 15: li $v0, 8
[00400044] 0000000c syscall ; 16: syscall
[00400048] 3c011001 lui $1, 4097 [inputSize] ; 18: la $t1, inputSize #load string into register t1
[0040004c] 34290026 ori $9, $1, 38 [inputSize]
[00400050] 3c011001 lui $1, 4097 [inputSize] ; 19: la $t2, inputSize #load string into register t2
[00400054] 342a0026 ori $10, $1, 38 [inputSize]
[00400058] 814b0000 lb $11, 0($10) ; 24: lb $t3, ($t2)
[0040005c] 11600003 beq $11, $0, 12 [decrementFunction-0x0040005c]
[00400060] 254a0001 addiu $10, $10, 1 ; 30: addiu $t2, $t2, 1
[00400064] 0401fffd bgez $0 -12 [goThroughString-0x00400064]
[00400068] 254afffc addiu $10, $10, -2 ; 39: subu $t2, $t2 , 2
[0040006c] 012a082a slt $1, $9, $10 ; 44: bgo $t1, $t2, isPali
[00400070] 1020000c beq $1, $0, 48 [isPali-0x00400070]
[00400074] 812b0000 lb $11, 0($9) ; 47: lb $t3, ($t1)
[00400078] 814c0000 lb $12, 0($10) ; 48: lb $t4, ($t2)
[0040007c] 156c0004 bne $11, $12, 16 [notPali-0x0040007c]
[00400080] 25290001 addiu $9, $9, 1 ; 54: addu $t1, $t1, 1
[00400084] 254affff addiu $10, $10, -1 ; 55: subu $t2, $t2, 1
[00400088] 0401fff9 bgez $0 -28 [checkPali-0x00400088]
[0040008c] 3c011001 lui $1, 4097 [notPal] ; 60: la $a0, notPal
[00400090] 34240443 ori $4, $1, 1091 [notPal]
[00400094] 34020004 ori $2, $0, 4 ; 61: li $v0 , 4
[00400098] 0000000c syscall ; 62: syscall
[0040009c] 04010006 bgez $0 24 [loopThrough-0x0040009c]
[004000a0] 3c011001 lui $1, 4097 [isPal] ; 67: la $a0, isPal
[004000a4] 34240426 ori $4, $1, 1062 [isPal]
[004000a8] 34020004 ori $2, $0, 4 ; 68: li $v0 , 4
[004000ac] 0000000c syscall ; 69: syscall
[004000b0] 04010001 bgez $0 4 [loopThrough-0x004000b0]
[004000b4] 34020004 ori $2, $0, 4 ; 74: li $v0, 4
[004000b8] 3c011001 lui $1, 4097 [askLoop] ; 75: la $a0, askLoop
[004000bc] 34240464 ori $4, $1, 1124 [askLoop]
[004000c0] 0000000c syscall ; 76: syscall
[004000c4] 34020005 ori $2, $0, 5 ; 79: li $v0, 5
[004000c8] 0000000c syscall ; 80: syscall
[004000cc] 00026821 addu $13, $0, $2 ; 81: move $t5, $v0
[004000d0] 34010002 ori $1, $0, 2 ; 84: beq $t5, 2, exit
[004000d4] 102d0002 beq $1, $13, 8 [exit-0x004000d4]
[004000d8] 08100009 j 0x00400024 [while] ; 86: j while
[004000dc] 34020004 ori $2, $0, 4 ; 90: li $v0, 4
[004000e0] 3c011001 lui $1, 4097 [exiting] ; 91: la $a0, exiting
[004000e4] 34240000 ori $4, $1, 0 [exiting]
[004000e8] 0000000c syscall ; 92: syscall
[004000ec] 3402000a ori $2, $0, 10 ; 95: li $v0, 10
[004000f0] 0000000c syscall ; 96: syscall
All Rights Reserved.
SPIM is distributed under a BSD license.
See the file README for a full copyright notice.
QtSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version 2.1.
Memory and registers cleared
SPIM Version 9.1.23 of December 4, 2021
Copyright 1990-2021 by James Larus.
All rights reserved.
SPIM is distributed under a BSD license.
See the file README for a full copyright notice.
QtSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version 2.1.
Console
Type a word or sentence: lol
The string is a palindrome
Would you like to continue? type 1 to continue; type 2 to exit: 1
Type a word or sentence: racecar
The string is a palindrome
Would you like to continue? type 1 to continue; type 2 to exit: 1
Type a word or sentence: jape
The string is NOT a palindrome
Would you like to continue? type 1 to continue; type 2 to exit: 1
Type a word or sentence: weird
The string is NOT a palindrome
Would you like to continue? type 1 to continue; type 2 to exit: 1
Type a word or sentence: anna
The string is a palindrome
Would you like to continue? type 1 to continue; type 2 to exit: 1
Type a word or sentence: jack
The string is NOT a palindrome
Would you like to continue? type 1 to continue; type 2 to exit: 2
Exiting now