INF 551 – Fall 2015

Midterm Exam Solution (100 points)

Exam time: Oct. 10, 12pm – Oct. 11, 12pm (i.e., 24 hours)

1. [RAID, 30 points] In this question, you are provided with **4** identical hard disks. The key parameters of each hard disk are as follows.

Capacity	1TB
Rotational speed	10, 000 RPM
Maximum seek time	10ms
Bandwidth	50MB/sec
Block size	4KB

You are asked to configure the 4 disks into RAID 0 (striping), RAID 1 (mirroring), RAID 10 (i.e., mirroring + striping), RAID 4 (striping + block-level parity), and RAID 5 (striping + distributed parity). For each configuration, answer the following questions.

a. What is the **maximum** capacity given by the configuration?
Answer:

RAID 0: 4TB, since stripping does not reduce the capacity, as shown below.

Disk 0	Disk 1	Disk 2	Disk 3
0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

RAID 1: 2TB

Here every disk is required to have a mirror, storing another copy of its data. For example, here shows disk1 is a mirror of disk0, and disk3 is mirror of disk2. Note that blocks are laid out “vertically” in each disk, since no striping here.

Disk 0	Disk 1	Disk2	Disk3
0	0	4	4
1	1	5	5
2	2	6	6
3	3	7	7

RAID 10: still 2TB

As shown below, blocks are first mirrored (disk1 is mirror of disk0, disk3 is mirror of disk2), next striped (disk0 and disk1 store block 0, disk1 and disk2 store block 1, and so on). In other words, if we take disk0 and disk1 as group 1 (of disks), and disk2 and disk3 as group 2. Then inside each group, blocks are mirrored. Across the groups, blocks are striped.

Disk 0	Disk 1	Disk 2	Disk 3
0	0	1	1
2	2	3	3
4	4	5	5
6	6	7	7

RAID 4: 3TB

As shown below, one disk is used to store solely the parity blocks.

Disk 1	Disk 2	Disk 3	Parity
0	1	2	P0
3	4	5	P1
6	7	8	P2
9	10	11	P3

RAID 5: Still 3TB

As shown below, parity blocks are distributed over multiple disks, but they still take up the same space as RAID 3.

Disk 0	Disk 1	Disk 2	Disk 3
0	1	2	P0
3	4	P1	5
6	P2	7	8
P3	9	10	11

b. What are the maximum and minimum time that it takes to **read 2** blocks concurrently from the RAID (**note that concurrent reads/writes occur when two blocks to be read/written are located on different disks**)?

Answer:

First, let's figure out the minimum and maximum time that it takes to read one block from a disk.

The minimum time occurs when the disk head is right above the block. That is, no seek or rotation is necessary. In this case, the time to read the block is solely the transfer time.

Transfer time = 4K/50MB * 1000 ms ~ 4/50 ms = .08 ms

So the minimum time = .08ms

The maximum time occurs when the disk needs to perform a full-swing seek (i.e., from innermost track to outermost, or vice versa) and then performs a full rotation (i.e., the block to be read is just past the head).

Seek time = 10ms (i.e., max. seek time)

Rotational latency = 60,000ms/10,000 = 6ms (i.e., time for a full rotation)

Transfer time = .08ms

So the maximum time = 16.08ms

Next, we note that the concurrent reads occur when two blocks to be read are located on different disks.

In all configurations, reading of two blocks can be completed in parallel. So the minimum/maximum time occurs when it takes minimum/maximum time to complete reading of one block from a disk. In other words,

Min = .08ms
Max = 16.08ms

c. What are the maximum and minimum time that it takes to **write 2** blocks concurrently onto the RAID? For RAID 4 and RAID 5, assume that the subtractive method is used to compute the parity, and you may ignore the time to calculate the parity.

Answer:

In RAID 0, RAID 1, and RAID 10, blocks can be directly overwritten with new contents and two writes can happen in parallel. So

Min = .08ms
Max = 16.08ms

However, there are complications in RAID 4 and RAID 5, where parity blocks also need to be updated. Recall that in subtractive method, contents of block and parity are first read into main memory, and XOR'ed with new content of block to compute new parity.

We assume that there is no latency (caused by seek and rotational latency) between read and write of the same block (data or parity). This is reasonable since they are part of a logical write and disk controller may wait for it to complete before serving other requests. Note that we ignore the time for the head to reposition back to the beginning of block after read, to perform the write.

<u>RAID 4</u>

Parity disk is the bottleneck. Reading and writing parity for two blocks need to be done in sequence. So the maximum time occurs when there is maximum latency for read each parity block.

Maximum:

Read first parity: 16.08ms
Write first parity: .08ms
Read second parity: 16.08ms
Write second parity: .08ms

Total: 32.32ms

The minimum time occurs when there is no latency between updating of two parity blocks.

Minimum:

Read first parity: .08ms
Write first parity: .08ms
Read second parity: .08ms
Write second parity: .08ms

Total: .32ms

RAID 5

Here parity disk is not the bottleneck anymore. Updating of parity blocks, one for each write, can now be done in parallel.

Minimum:

Read one parity: .08ms
Write one parity: .08ms

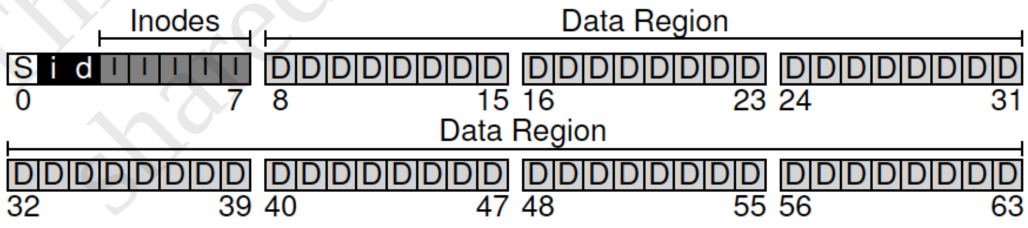
Total: .16ms

Maximum:

Read one parity: 16.08ms
Write one parity: .08ms

Total: 16.16ms

2. [File System, 30 points] Recall that we have seen a small file system stored in a disk of 64 blocks as shown below.



Now let us consider a new disk with 128 blocks, but the organization of file system on the disk remains the same. In other words, the new disk still has the first 8 blocks storing the superblock, two bitmaps (i-map and d-map) for tracking the free slots for inodes (i-map) and free data blocks (d-map). But the new disk now has additional 64 data blocks, numbered 64 to 127.

Suppose that the disk has the following parameters.

Block size	4KB
# of blocks on disk	128
Sector size	512B

Further suppose that the file system has the following parameters.

Inode size	512B
# of inode blocks	5 (blocks #3 to #7)

Please answer the following questions on the new disk and file system.

a. What is the capacity of the disk (note that this is NOT asking how much space is available to store the file content, rather the total storage space of the disk)?

   128*4KB = 512KB

b. How many files can the file system store on the disk?

   Answer:

   This depends on how many inodes the file system has.
   Each block can store 4KB/512B = 8 inodes.
   Since there are 5 inode blocks, so the total number of inodes = 8 * 5 = 40.

   So this file system can store no more than 40 files.

c. How many bits are there in the two bitmaps, i-map and b-map?

   Answer:

   Since there are 40 inodes, i-map has 40 bits.
   Since there are 128 - 8 = 120 data blocks, b-map has 120 bits.

d. If the inumber of a file is n, where is its corresponding inode located on the disk? Assume that the sector numbers start from zero and the blocks have contiguous sector numbers. Which sector is the inode located?

   Offset = 12K + n * 512
   Sector = (12K + n * 512)/512 = 24 + n

e. What is the maximum size of a file that can be stored in this file system?

   The maximum size occurs when the file occupies all data blocks.
   So, 120 blocks * 4KB/block = 480KB
   Note that one byte is sufficient to address a block in this small file system. So an inode will need 120 bytes at most for storing pointers (addresses) to the blocks. Assuming that all are direct pointers. We note that inode has 512 bytes, so this takes up 120 out of 512 bytes.

   Note that in practice, a dynamic scheme is used instead, where a small number of direct pointers are used for small files. For larger files, some indirect pointers are allocated which point to block of pointers instead. See next question.

Note that in this question, we may also accept answers that use a limited number of direct pointers, say 12, and some indirect pointers, say 1, to keep track of locations of blocks storing contents. For example, if one indirect pointer is used, then a data block will need to be used to store direct pointers. This means the max file size is 119 blocks instead.

f. Explain how you can increase the file size by introducing multi-level index.
   Answer: We store in inode not just direct pointers (pointing to data blocks), but also indirect pointers (pointing to a block of pointers). The pointers in the pointer block can point to a data block or point to other pointer blocks. This strategy increases dynamically the number of data blocks a file can occupy (each pointed to by a direct pointer).

g. Suppose the file system stores two files under its root directory: “foo” and “bar”. Each file occupies one block. Suppose that inumber of the root directory is 2. How many blocks need to be read in order to load the content of the files into the memory? Indicate the content of the blocks (e.g., holding content of “foo”, holding some directory, or holding inodes). Assume that superblock is not available in the main memory. Also assume that once a block is read into memory, it can be cached there and looked up by future operations.

   Answer:

   First, we need to read the Superblock which tells us the location of the inode blocks. (one read)

   Next, we read the inode for the root directory according to its inumber. For the inode, we know the location of data block storing the content of the root directory. Read the block. (two reads in this step)

   Next, from the directory content, we know the inumber of “foo” and “bar”. Use the inumber, we read its inode and then follow the pointer in the inode, we read the content of file which has just one block. (four reads, two for each file)
   So total reads: 7 blocks

3. [Network File System, 20 points] In Network File System (NFS), it is possible that a client never receives a response from the server, after it sends a message to the server.
   a. What are the possible reasons for this?
      i. Request may be lost on the way to server
      ii. Server may be down
      iii. Response may be lost on the way to client

b. To address this problem, the client may simply keep resending the message to the server for desired operations until the operations succeed. However, this requires that the operations satisfy certain property. What is the property?

    Idempotent: an operation is idempotent, if the effect of multiple execution of the operation is the same as that of single execution

c. For each of the following operations, indicate whether it has the property as described in b.
    i. Lookup (yes)
    ii. Read (yes)
    iii. Write (yes, assuming request specifies the offset location of data to be written)
    iv. Create (file) (no, will report error after the 1st execution)
    v. Remove (file) (no, error will be reported, file is not there anymore)
    vi. Mkdir (no, can not create the same dir multiple times)
    vii. ReadDir (yes)

4. [ER Modeling, 20 points] Consider building an application for an electronic store chain (i.e., it has many stores in different locations), in order to manage the purchases made at the store.

The store needs to record, for each product, its serial number, brand, model, and price. It also needs to record operating system and hard drive, but for laptops only.

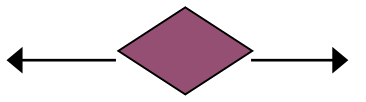
Each store has a unique store ID, name, and city where the store is located.

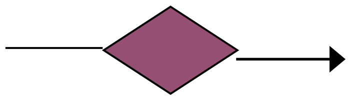
For each purchase, the application needs to record the name of buyer and seller (i.e., sales person), the store where the purchase was made, the products purchased, and the purchase date. Note that there can only be one buyer and one seller involved in a single purchase. Also each purchase can be made at a single store. However, multiple products may be purchased in the same sale.

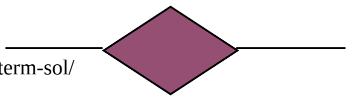
It also records, for buyer and seller, his/her name, phone number, and residence city.

Please depict an ER diagram to model the application. For each entity set, underline its **key attributes**. Properly indicate the multiplicity (i.e., many-one, many-many, or one-one) of each relationship.

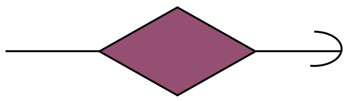
For your reference, here are notations for indicating the multiplicity and type (is-a) of relationships. (Note that hollow arrow represents exactly one, i.e., one and only one, as the example below shows.)


one-to-one

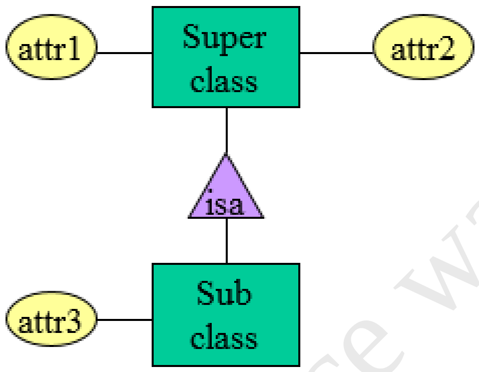

many-to-one



many-to-many

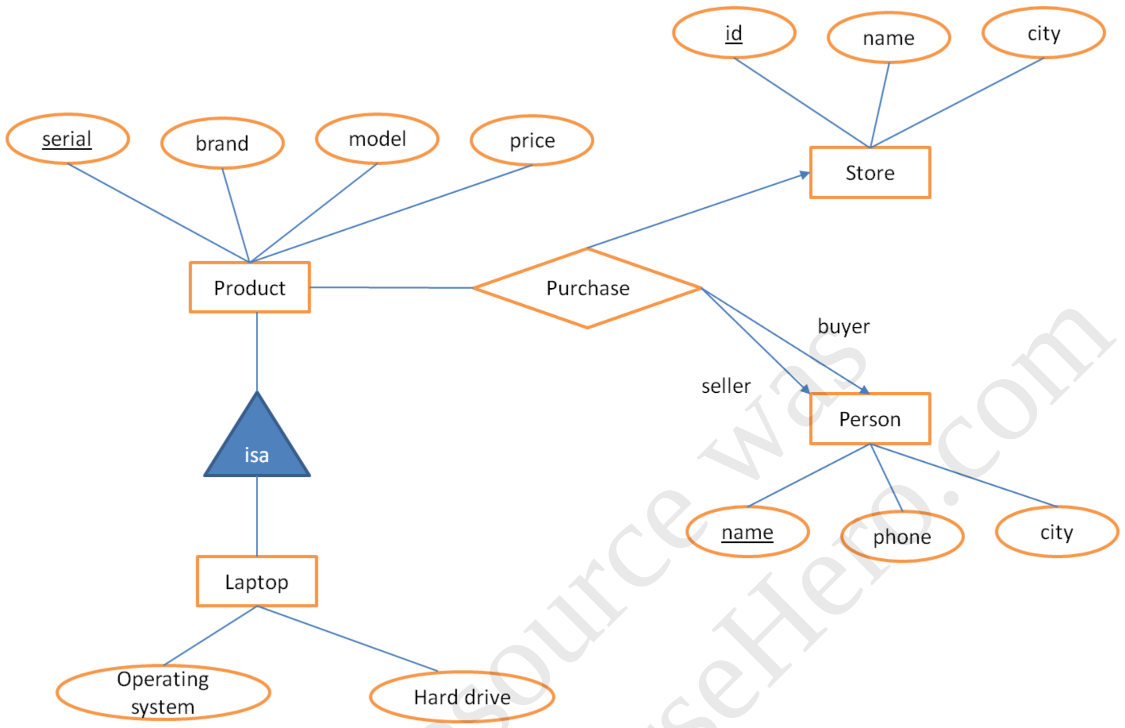


many-to-exactly one



Answer:

Also accept answers where 3 arrows are hollow ones.


id
name
city
serial
brand
model
price
Store
Product
Purchase
buyer
seller
Person
isa
name
phone
city
Laptop
Operating system
Hard drive