CMPT-454 Fall 2009
Instructor: Martin Ester
TA: Yi Liu

Solution Assignment 1

Total marks: 200 (20 % of the assignments)
Due date: September 30, 2009

Problem 1.1 (50 marks)

Consider a disk with a sector size of 512 bytes, 1,000 tracks per surface, 100 sectors per track, 5 double-sided platters and a block size of 2,048 bytes. Suppose that the average seek time is 5 msec, the average rotational delay is 5 msec, and the transfer rate is 100 MB per second. Suppose that a file containing 1,000,000 records of 100 bytes each is to be stored on such a disk and that no record is allowed to span two blocks.

a) How many records fit onto a block?
2048/100 = 20. We can have at most 20 records in a block.

b) How many blocks are required to store the entire file? If the file is arranged sequentially (according to the ‘next block concept’) on disk, how many cylinders are needed?
1,000,000/20=50,000 blocks are required to store the entire file.
A track has 25 blocks, a cylinder has 25*10=250 blocks. Therefore, we need 50,000/250=200 cylinders to store the file sequentially.

c) How many records of 100 bytes each can be stored using this disk?
The disk has 1000 cylinders with 250 blocks each, i.e. it has 250,000 blocks. A block contains 20 records. Thus, the disk can store 5,000,000 records.

d) If blocks of the file are stored on disk according to the next block concept, with the first block on block 1 of track 1, what is the number of the block stored on block 1 of track 1 on the next disk surface?

There are 25 blocks in each track. It is block 26 on block 1 of track 1 on the next disk surface.

e) What is the time required to read the file sequentially?

We need to read 200 cylinders. In order to read one cylinder, we have 1 seek, no rotational delays and the transfer of 250 blocks = 500KB. Therefore, the time to read one cylinder is 5msec

+ 500K/100M sec = 5msec + 5 msec = 10 msec. The time to read the entire file is 200*10msec = 2sec.

f) What is the time required to read the file in random order? Note that in order to read a record, the block containing the record has to be fetched from disk.

In random access, the read of every block requires an average seek time of 5 msec, an average rotational delay of 5 msec and a transfer time of 2K/100M sec = 0.02 msec, i.e. 10.02 msec. Therefore, the time to read the entire file is 50,000*10.02msec ~ 500sec.

Problem 1.2 (50 marks)

Suppose a disk with 1,000 tracks per surface. Assume that the current track of the disk heads and the tracks of read / write requests are uniformly distributed over the range [1..1000].

a) With first-come-first-served scheduling, what is the average number of tracks that the disk heads need to travel to serve the next request?

   Every track has the same probability of being the current track *i* and being the track *j* of the next request. Therefore, the average number of tracks *avgt* to be traveled is

$$avgt = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}|i-j| = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{\substack{j=1\\ j<i}}^{n}(i-j) + \frac{1}{n^2}\sum_{i=1}^{n}\sum_{\substack{j=1\\ j>i}}^{n}(j-i)$$

   Here is the *n* x *n* matrix of numbers of tracks traveled for all combinations of current and destination tracks:

		Destination track				
		1	2	3	4	n
Current Track	1	0	1	2	3	n-1
	2	1	0	1	2	n-2
	3	2	1	0	1	n-3
	n	n-1	n-2	n-3		0

Note that this matrix is symmetric with respect to its diagonal and the sum of all entries is two times the sum of either of the two triangular half matrices.

$$avgt = \frac{2}{n^2}\sum_{i=1}^{n}\sum_{\substack{j=1\\ j>i}}^{n}(j-i) = \frac{2}{n^2}\sum_{k=1}^{n-1}\frac{(n-k)(n-k+1)}{2} =$$

$$\frac{1}{n^2}\sum_{k=1}^{n-1}[(n-k)^2+(n-k)] = \frac{1}{n^2}\sum_{k=1}^{n-1}(k^2+k) = \frac{1}{n^2}[\frac{(n-1)n(2n-1)}{6}+\frac{(n-1)n}{2}] \approx \frac{n}{3}$$

We are using

$$\sum_{k=1}^{n} k = \frac{(n+1)n}{2} \qquad \text{and}$$

$$\sum_{k=1}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}$$

For n =1000: $avgt$ = 333.

b) Assuming that the disk head is currently on track 500, provide a sequence of 10 read / write requests (track numbers) that lead to the worst case performance of first-come-first-served scheduling.

1000, 1, 1000, 1, 1000, 1, 1000, 1, 1000, 1

Every request (except the first one) requires the disk head to travel 999 tracks, i.e. the total number of tracks traveled is 9491.

c) Suppose that the queue at any given time contains 10 read / write requests. With elevator algorithm scheduling, what is the average number of tracks that the disk heads need to travel to serve the next request? How would your answer change if the queue size was 20?

The 10 requests divide the sequence of all tracks into 11 intervals. Given a uniform distribution of the requested tracks, the average width of these intervals is 1000/11 ~ 90. The disk head is currently in one of these intervals at a random position within that interval. An analysis similar to that of a) shows that the average number of tracks traveled is, with n = 90,

$$avgt = \frac{n}{3} = 30\,.$$

With a queue of size 20, n = 1000/21= 48, and the average number of tracks traveled is 16.

d) Suppose that the queue at any given time contains 10 read / write requests. Assuming that the disk head is currently on track 500, provide a sequence (queue) of 10 read / write requests (track numbers) that lead to the worst case performance of elevator algorithm scheduling.

1, 100, 300, 530, 540, 750, 800, 850, 950, 1000

The total number of tracks traveled is 499+999 = 1498.

Problem 1.3 (50 marks)

Suppose that a disk has probability F of failing in a given year, and it takes H hours to replace a disk. Assume independence of failures of different disks and uniform distribution of disk failures over time.

a) If we use mirrored disks, which events lead to data loss? What is the mean time to data loss, as a function of F and H?

Data loss occurs if one of the two disks crashes and the other disk crashes within the H hours during which the crashed disk is being replaced. We assume independence of crashes of the disks and uniform distribution of the crashes over time. There are two disjoint events that lead to data loss:

- Event 1: disk 1 crashes first, and then disk 2 crashes within H hours,
- Event 2: disk 2 crashes first, and then disk 1 crashes within H hours.

The probability of event1 (event 2) within a given year is $\frac{F \cdot F \cdot H}{24 \cdot 365}$, and the probability P(*loss*) of event 1 or event 2 within a given year is the sum of the probabilities of the two events, i.e. $P(loss) = 2\frac{F \cdot F \cdot H}{24 \cdot 365} = \frac{F^2 H}{4380}$.

The mean time to data loss is therefore $\frac{1}{2P(loss)} = \frac{24 \cdot 365}{4F \cdot F \cdot H} = \frac{2190}{F^2 H}$ years.

b) If we use a RAID level 4 or 5 scheme, with N disks, which events lead to data loss? What is the mean time to data loss, as a function of F, H, and N?

Data loss occurs if one of the disks crashes and at least one of the other N-1 disks crashes within the H hours during which the crashed disk is being replaced. We assume independence of crashes of the disks and uniform distribution of the crashes over time.

There are N disjoint events that lead to data loss:

- Event 1: disk 1 crashes first, and then at least one of the other disks crashes within H hours,
- Event 2: disk 2 crashes first, and then at least one of the other disks crashes within H hours, . . .
- Event N: disk N crashes first, and then at least one of the other disks crashes within H hours.

Let $P = \frac{F \cdot H}{24 \cdot 365}$ denote the probability of any single disk crashing during the repair. The number of disks among the other N-1 disks that crash within the H hours during which the first disk is being replaced follows a binomial distribution with parameter P. The probability that this number is at least one is $= \sum_{i=1}^{N-1} Bin(i \mid N-1, P) = \sum_{i=1}^{N-1} \binom{N-1}{i} P^i (1-P)^{N-1-i}$.

Thus, the probability that any of the above N events happens within a given year is $F \cdot \sum_{i=1}^{N-1} Bin(i \mid N-1, P) = F \cdot \sum_{i=1}^{N-1} \binom{N-1}{i} P^i (1-P)^{N-1-i}$. The probability P(*loss*) that at least one of these events (event 1 or event 2 or . . . or event N) happens within a given year is the sum of the probabilities of the individual events $P(loss) = N \cdot F \cdot \sum_{i=1}^{N-1} \binom{N-1}{i} P^i (1-P)^{N-1-i}$. The mean time to data loss is then $\frac{1}{2P(loss)}$ years.

c) How does the mean time to data loss in cases a) and b) roughly compare to each other?

P(loss) in case b) is roughly $N(N-1)FP$ compared to 2 F P in case a), i.e. roughly $\frac{1}{2}N(N-1) \approx N^2$ times as high as in case a). Therefore, the mean time to data loss is roughly N^2 times shorter.

Problem 1.4 (50 marks)

Suppose that a file has r = 100,000 STUDENT records with the following fields:

- NAME (30 bytes),
- SSN (9 bytes),
- ADDRESS (40 bytes),
- PHONE (9 bytes),
- BIRTHDATE (8 bytes),
- SEX (1 byte),
- MAJORDEPTCODE (4 bytes),

- MINORDEPTCODE (4 bytes),
- CLASSCODE (4 bytes, integer), and
- DEGREEPROGRAM (3 bytes).

The fields are of fixed-length.

Suppose only 75% of the STUDENT records have a value for PHONE, 80% for MAJORDEPTCODE, 15% for MINORDEPTCODE, and 95% for DEGREEPROGRAM, and 100% of the STUDENT records have a value for the other fields. We use a variable-length record format. Each record has a 2-byte field type for each field occurring in the record, plus the 1-byte deletion marker and a 1-byte end-of-record marker. Suppose we use a spanned record organization, where each block has a 5-byte pointer to the next block (this space is not used for record storage). Each block contains 1,024 bytes.

a) Calculate the average record length R in bytes.

Assuming that every field has a 2-byte field type, and that the fields not mentioned above (NAME, SSN, ADDRESS, BIRTHDATE, SEX, CLASSCODE) have values in every record, we need the following number of bytes for these fields in each record, plus 1 byte for the deletion marker, and 1 byte for the end-of-record marker:

R fixed

= (30+2) + (9+2) + (40+2) + (8+2) + (1+2) + (4+2) +1+1

= 106 bytes

For the other fields (PHONE, MAJORDEPTCODE, MINORDEPTCODE DEGREEPROGRAM), the average number of bytes per record is:

R variable

= ((9+2)*0.75)+((4+2)*0.80)+((4+2)*0.15)+((3+2)*0.95)

= 8.25+4.8+0.9+4.75

= 18.7 bytes

The average record size R = R fixed + R variable = 106 + 18.7 = 124.7 bytes

The total number of bytes needed for the whole file is r * R = 100,000 * 124.7 = 12,470,000 bytes.

b) Calculate the number of blocks needed for the file.

Using a spanned record organization with a 5-byte pointer at the end of each block, the number of bytes available in each block is B = 1024 - 5 = 1019 bytes.

The number of blocks b needed for the file is:

b = ceiling((r * R) / B)

= ceiling(12470000 / 1019)

= 12,238 blocks