Chapter 9. I/O and the Storage Hierarchy

Problem 9.1

A request is made to the Hitachi DK516 drive outlined in Table 9.4. The new track is 30 tracks distant from the current position. The request is to move 20 sections into a buffer (the transfer rate is 3 Mbytes). The starting block location is unknown but assumed to be uniformly distributed over possible locations on the track. What is the total time to access and transfer the requested data?

Seek Time:

From Table 9.5, $a = 3.0$, $b = 0.45$. The seek distance is 30 tracks. By the equation, seek time = 5.46 ms.

Rotation:

Uniformly distributed over [0, 16.67 ms]. Expected rotation delay = 8.33 ms.

Transfer:

20 blocks at 512 bytes at 3 Mbytes/sec = 3.41 ms.

Total = $5.46 + 8.33 + 3.41$ ms = 17.2 ms.

Problem 9.2

$\lambda = \frac{1}{33\text{ms}} = 30.3$ requests/sec

$\mu = 50$ requests/sec; in other words, $T_s = 20$ ms

$\rho = \frac{30.3}{50} = .606$

Using M/G/1,

a. Total response time
   $T_w = \frac{\rho}{2(1-\rho)}(1 + c^2)T_s = \frac{.606}{2(1-.606)}(1 + .5) \times 20$ ms = 23.07 ms
   $T_r = T_w + T_s = 23.07 + 20.0$ ms = 43.07 ms

b. Average number of requests queued at the disk awaiting service
   $Q = \frac{\rho^2}{2(1-\rho)} \times 1.5 = \frac{.606^2}{2(1-.606)} \times 1.5 = .699$

Problem 9.4

Repeat the first example in study 9.1 with the following changes: $c^2 = 0.5$, $T_{\text{user}} = 20$ ms (i.e., 200K user-state instructions before an I/O request), and $n = 3$.

First, look at the disk to determine how applicable the open-queue model really is. $\lambda = 1$ request per 30 ms (time for user process to generate an I/O and system to process request) = 33.3 requests/sec. Thus, $\rho = 2/3$ and $T_w = 30$ ms. So with the open-queue model, the CPU generates an I/O request; 30 ms later (on average) the disk begins service on the request; 20 ms later the request is complete.

The 50 ms it takes to perform the disk operation for the first task is less than the time the CPU will spend processing the other two jobs in memory (60 ms), thus as an approximation, the open-queue model will apply.

Since we are dealing with statistical behavior, there will be some instances in which the disk queueing time and disk service time will exceed the time the CPU is processing the two other tasks. We will use the closed-queue asymptotic model to better estimate the delays.

$T_u = 30$ ms.

$T_s = 20$ ms.

$T_c = 50$ ms.

$n = 3$.

Now, the rate of requests to the disk = rate of requests serviced by the CPU = $\lambda = \min(1/T_u, n/T_c) =$ 33.3 requests/sec.

Since $T_u > T_s$, we need to use the inverted server model. What this really means is that the CPU, not the disk, is the queueing bottleneck. Therefore, it is the CPU queueing delays which will ultimately lower the peak system throughput. Thus,

$T'_u = 20$ ms.

$T'_s = 30$ ms.

$T_c = 50$ ms.

$n = 3$.

Notice, however, that the service rates of the disk and CPU are closer than they were in example 9.1. We should therefore expect the relative queueing delays at the disk to be potentially higher and our model to be less accurate.

We compute $r = T'_u/T'_s = 2/3$. Since we have a small n, we use the limited population correction for the CPU utilization. Notice that the correction for ρ in section 9.4.2 depends on the distribution of service time being exponential. Since the service time we are concerned with is the CPU's service time, we need to assume that the CPU's $c^2 = 1.0$. With that assumption, $\rho_a = 0.975$ (by applying the equation for $n = 3$). Finally, $\lambda_a = 32.5$ requests/second.

With this we can derive the utilizations and waiting times for the CPU and the disk. In most queueing systems, you want to find queueing delays, utilizations, throughputs, and response times. You should know how to find each of these values.

Problem 9.5

$T_{\text{user}} = \frac{200K}{40 \text{ MIPS}} = 5$ ms

$T_{\text{sys}} = 2.5$ ms

$n = 3$

$T_s = 20$ ms

Using a noninverted model,

$T_u = 7.5$ ms

$T_s = 20$ ms

$r = .375$

$\rho_a = \frac{1+r+\frac{r^2}{2}}{1+r+\frac{r^2}{2}+\frac{r^3}{6}} = .94$

$\lambda_a = \frac{.94}{.020} = 47.2$ instead of 50

Problem 9.6

Our job is to find the value of T_{user} at $n = 1$ that has the same user computation time per second ($\lambda_a T_{\text{user}}$) as $n = 2$.

At $n = 2$, user computation time ($\lambda_a T_{\text{user}}$) = 445 ms.

$T_u = T_{\text{user}} + T_{\text{system}}$

$\rho_a = \frac{1}{1+r}, r = \frac{T_u}{T_s}$ (inverted service case)

$\lambda_a = \frac{\rho_a}{max(T_u, T_s)}$

We can find T_{user} by taking an initial approximation and iterating several times.

Let's pick a T_{user} that makes $T_u = 20$ ms.

$T_u - T_{\text{system}} = 20 \text{ ms} - 2.5 \text{ ms} = 17.5$ ms.

$r = 1$

$\rho_a = \frac{1}{1+1} = .5$

$\lambda_a = \frac{.5}{.02} = 25$ transactions/sec

$\lambda_a T_{\text{user}} = 25 \times 17.5 \text{ ms} = 437.5$ ms

One more iteration:

Try $T_{\text{user}} = 18$ ms

$\lambda_a T_{\text{user}} = 24.7 \times 18 = 444.3$ ms

This is a close enough value.

In conclusion, for approximately $T_{\text{user}} = 18$ ms at $n = 1$ we have the same performance as $n = 2$.

Problem 9.7

In study 9.2, suppose we increase the server processor performance by a factor of 4, but all other system parameters remain the same. Find the disk utilization and user response time for $n = 20$ (assume $c^2 = 0.5$ for the disk).

We have the following expected service times:

$T_{\text{disk}} = 18.8$ ms.

$T_{\text{server}} = 10$ ms.

$T_{\text{network}} = 3.6$ ms.

Thus, the disk will be the bottleneck in this case. We will use the asymptotic model without the low population correction. We have the following parameters for our model:

$T_c = 1$ sec. Remember, the workstation user (if not slowed down by the rest of the system) will generate a disk operation every second.

$T_s = 18.8$ ms

$T_u = 981.2$ ms

$r = T_u/T_s = 52.2$

$f = T_s/T_c = 0.0188$

By equation 9.6,

$T_w/T_c = 0.0080835$

$T_{w\ \text{disk}} = 8.384$ ms

$\lambda_{a\ \text{disk}} = 19.83$

$\rho_{a\ \text{disk}} = 0.3728$.

Now, we can compute the expected waiting times and utilizations of the other nodes in the system using the open queue model:

$\rho_{a\ \text{net}} = \lambda_a T_{snet} = 0.0714$.

$T_{w\ \text{net}} = 0.138$ ms.

$\rho_{a\ \text{server}} = \lambda_a T_{s\ \text{server}} = 0.1983$

$T_{w\ \text{server}} = 1.24$ ms.

$T_{w\ \text{disk}} = 8.384$ ms.

$T_{w\ \text{total}} = 9.762$ ms.

Notice that although the service times of the server and the disk differed by less than 50%, the waiting times are nearly 50x different. Comparing the waiting time of the total system with the waiting time of the disk, the disk is responsible for about 85% of the waiting time. We should determine if the waiting time of the net and server impact the request rate:

$\lambda'_a = \lambda/(T_{w\ \text{disk}} + T_{w\ \text{net}} + T_{w\ \text{server}} + T_c) = 19.807$.

With this value, we should recompute the utilizations and waiting times. Since, however, this is very close to our initial estimate of the achieved request rate, it will not alter our results significantly.

Now for the workstation:

$\lambda = 20$ requests/sec.

$T_c = 1$ sec.

$T_w =$ sum of T_w for each node $= 9.76$ ms.

Thus, the response time is $T_w + T_{sdisk} + T_{sserver} + T_{snet} = 42.14$ ms.

Problem 9.12

Rotation speed = 3600 rpm

$$\text{Seek time} = \text{a} + \text{b}\sqrt{\text{seek tracks}} = 3.0 + 0.45\sqrt{23-1} = 5.11\,\text{ms}.$$

When the seek is complete, the head has moved $5.11/16.67 * 77 = 23.611$ sectors. The rotational delay for the rotation of the additional 60.39 sectors is 13.07 ms. The transfer takes $16*512/3*10^6 = 2.73ms$. The total elapsed time is $5.11 + 13.07 + 2.73 = 20.91$ ms.

Problem 9.13

The perceived delay is:

$$\frac{\lambda - \lambda_a}{\lambda_a} T_c$$

For (a), $\frac{20-18.9}{18.9} \times 70 = 4.07$ ms

For (b), $\frac{61.5-44.5}{44.5} \times 32.5 = 12.42$ ms

Problem 9.18

Without disk cache, $T_{\text{user}} = 40$ ms, $T_{\text{sys}} = 10$ ms.

With disk cache, $T_{\text{user}} = \frac{T_{\text{user (no disk cache)}}}{.3} = \frac{40\text{ ms}}{.3} = 133.3$ ms.

Then

$$\begin{aligned}
T_u &= 133.3\text{ ms} + 10\text{ ms} = 143.3\text{ ms} \\
T_s &= 20\text{ ms} \\
n &= 2 \\
r &= \frac{T_s}{T_u} = \frac{20}{143.3} = .14 \\
\rho_a &= \frac{1+r}{1+r+\frac{r^2}{2}} = .99 \\
\lambda_a &= \frac{.99}{.143} = 6.92 \text{ requests per second}
\end{aligned}$$

The maximum capability of the processor is 1 request each 143.3 ms. We achieved 99% of this capacity—i.e., $\frac{1}{6.92} = 144.5$ ms.