Virtual Problem Set # 2

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 queuing 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 queuing bottleneck. Therefore, it is the CPU queuing 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 queuing 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 co-
rection for $\rho$ 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 queuing systems, you want to find queueing delays, utilizations, throughputs, and response times. You should know how to find each of these values.

**Problem 9.6**

Our job is to find the value of $T_{\text{user}}$ 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.

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

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

$\lambda_a = \frac{\rho a}{\max(T_u, T_d)}$

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

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

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

$r = 1$

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

$\lambda_a = \frac{.5}{17.5} = 25$ transactions/sec

$\lambda_a T_{\text{user}} = 25 \times 17.5$ 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.
$T_{\text{server}} = 10 \text{ ms}.$
$T_{\text{network}} = 3.6 \text{ ms}.$

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:

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

$T_s = 18.8 \text{ ms}$
$T_u = 981.2 \text{ ms}$
$r = T_u/T_s = 52.2$
$f = T_s/T_c = 0.0188$

By equation 9.6,
$T_w/T_c = 0.0080835$
$T_w \text{ disk} = 8.384 \text{ ms}$
$\lambda_a \text{ disk} = 19.83$
$\rho_a \text{ disk} = 0.3728.$

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

$\rho_a \text{ net} = \lambda_a T_{\text{net}} = 0.0714.$
$T_{\text{w net}} = 0.138 \text{ ms}.$

$\rho_a \text{ server} = \lambda_u T_{\text{server}} = 0.1983$
$T_{\text{w server}} = 1.24 \text{ ms}.$
$T_{\text{w disk}} = 8.384 \text{ ms}.$
$T_{\text{w total}} = 9.762 \text{ ms}.$

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:

$\lambda' q_a = \lambda/(T_w \text{ disk} + T_w \text{ net} + T_w \text{ server} + T_c) = 19.807.$

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:
$\lambda = 20 \text{ requests/sec}.$
$T_c = 1 \text{ sec}.$
$T_w = \text{ sum of } T_w \text{ for each node} = 9.76 \text{ ms}.$

Thus, the response time is $T_w + T_{\text{disk}} + T_{\text{server}} + T_{\text{net}} = 42.14 \text{ ms}.$
Problem 9.12

Rotation speed = 3600 rpm

Seek time = a + 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 \(5.11/16.67\times77 = 23.611\) sectors. The rotational delay for the rotation of the additional 60.39 sectors is 13.07 ms. The transfer takes \(16\times512/3\times10^6 = 2.73\text{ms}\). The total elapsed time is 5.11 + 13.07 + 2.73 = 20.91 ms.

Problem 9.13

The perceived delay is:

\[
\frac{\lambda - \lambda_a T_c}{\lambda_a}
\]

For (a), \(\frac{20-18.9}{61.2-44.5} \times 70 = 4.07\ \text{ms}\)

For (b), \(\frac{61.2-44.5}{44.5} \times 32.5 = 12.42\ \text{ms}\)