# Analytic Models

Assignment 2

*Mina Mousavifar - 11279515 - sem311*

Question 1.

**a)**

Based on Little's law so far we have: $N = X(R + Z)$
Where N = number of users in system, X = access rate to webpage, R = mean residence time, Z = think time.
In addition access rate to disk is 16 $\frac{access}{second}$ and each webpage access needs 2 accesses to disk. So the access rate for a webpage is: $X_{webpage} = 16/2 = 8$ $\frac{access}{second}$
So:

$$36 = 8 * (1.5 + Z) \rightarrow 4.5 = 1.5 + Z \rightarrow Z = 3 seconds$$

**b)**

Based on Little's law and previous part: $N = X * Z$
Where N = average number of users thinking, X = access rate to webpage, Z = think time.

$$N = 3 * 8 = 24$$

---

Question 2.

**a)**

$$answer!$$

**b)**

$$answer!$$

---

Question 3.

**a)**

Based on Forced flow law we have: $D_k = \frac{B_k}{C}$ where D is service demand, k is each service part, B is resource busy time and C is number of requests completed.

$$D_{processor} = \frac{B_{processor}}{C} = \frac{400}{5000} = 0.08 seconds$$

$$D_{disk1} = \frac{B_{disk1}}{C} = \frac{500}{5000} = 0.1 seconds$$

$$D_{disk2} = \frac{B_{disk2}}{C} = \frac{600}{5000} = 0.12 seconds$$

So the system service demand would be:

$$D = \sum D_k = 0.08 + 0.1 + 0.12 = 0.3 seconds$$

Therefor we find that disk2 is our bottleneck with the most service demand.

**b)**

Processor is made 10x faster, so it's service demand has been changed: $D_{processor} = \frac{D_{processor}}{10} = \frac{0.08}{10} = 0.008 seconds$
and the system service demand has been changed: $D_{proc10x} = \sum D_k = 0.008 + 0.1 + 0.12 = 0.228 seconds$
Then we calculate $N^*$ which is the intersection of heavy load and light load.

$$N^* = \frac{D + Z}{D_{max}} = \frac{0.3 + 2}{0.12} = \frac{2.3}{0.12} = 19.1\overline{6}$$

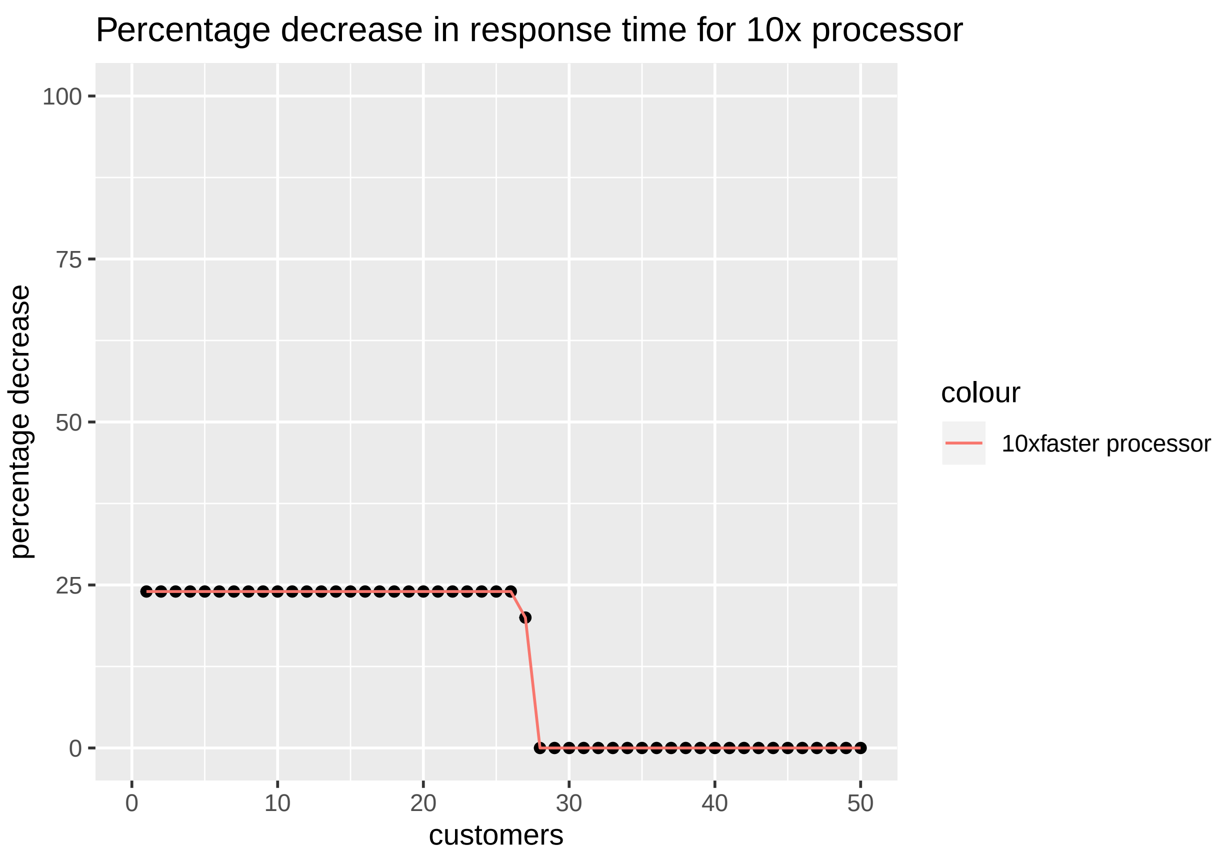
$$N^*_{proc10x} = \frac{D_{proc10x} + Z}{D_{max}} = \frac{0.228 + 2}{0.12} = \frac{2.228}{0.12} = 18.5\overline{6}$$

For bounds on throughput we have:

$$max(D, ND_{max} - Z) \leq R(N) \leq ND \rightarrow max(0.3, 0.12N - 2) \leq R(N) \leq 0.3N$$

$$max(D_{proc10x}, ND_{max} - Z) \leq R_{proc10x}(N) \leq ND_{proc10x} \rightarrow max(0.228, 0.12N - 2) \leq R(N) \leq 0.228N$$

```r
library(dplyr)
# calculate lower bounds for original system
Rs <- data.frame("N"=1:50, "R"=rep(0,50), stringsAsFactors = FALSE)

for(i in 1:50){
  Rs[i,2] = max(0.3, (0.12*Rs[i,1]) - 3)
}

# calculate lower bounds for 10x faster processor
Rs10x <- data.frame("N"=1:50, "R"=rep(0,50), stringsAsFactors = FALSE)

for(i in 1:50){
  Rs10x[i,2] = max(0.228, (0.12*Rs10x[i,1]) - 3)
}

# calculate percentage decrease
des <- data.frame("N"=1:50, "change"=rep(0,50), stringsAsFactors = FALSE)

for(i in 1:50){
  des[i,2] = ((abs(Rs10x[i,2]-Rs[i,2]))*100)/Rs[i,2]
}
despre <- des

# plotting
library(ggplot2)
p <- ggplot(data = des, mapping = aes(x = N, y = change)) +
  xlim(0,50) + ylim(0, 100) +
  ggtitle("Percentage decrease in response time for 10x processor") +
  xlab("customers") + ylab("percentage decrease") +
  geom_point() +
  geom_line(data = des, aes(x = N, y = change, color = "10xfaster processor"))
p
```




**c)**

First we should calculate the new service demand time. Number of visits still stays the same and we have the same $S_k$ because of our identical disks. Also, because of new load balancing new $D_k$ would be the same too. So the demand would be:

$$V_1 + V_2 = \frac{D_1}{S} + \frac{D_2}{S} \rightarrow \frac{V_1 S}{S} + \frac{V_2 S}{S} + \frac{V_3 S}{S} = \frac{1}{S}(0.1 + 0.12) \rightarrow \frac{D_1 S}{S} + \frac{D_2 S}{S} + \frac{D_3 S}{S} = \frac{1}{S}(0.22)$$

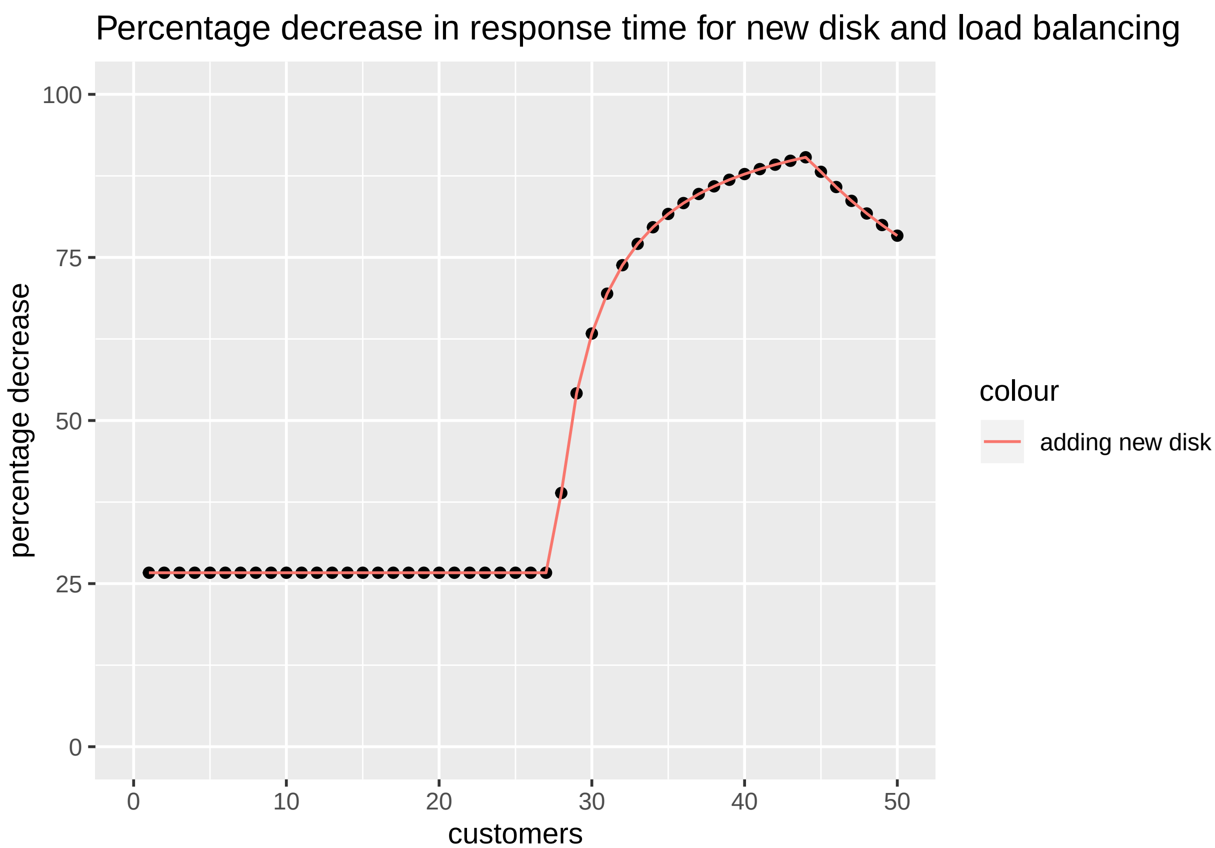
$$D_1 = D_2 = D_3 \rightarrow D_1(\frac{1}{S} + \frac{1}{S} + \frac{1}{S}) = \frac{1}{S}(0.22) \rightarrow D_1 * 3 = 0.22 \rightarrow D_1 = 0.07\overline{3}$$

Therefore the new bounderies would be: $D = \sum D_k = 0.22$ and $D_{max} = 0.07\overline{3}$

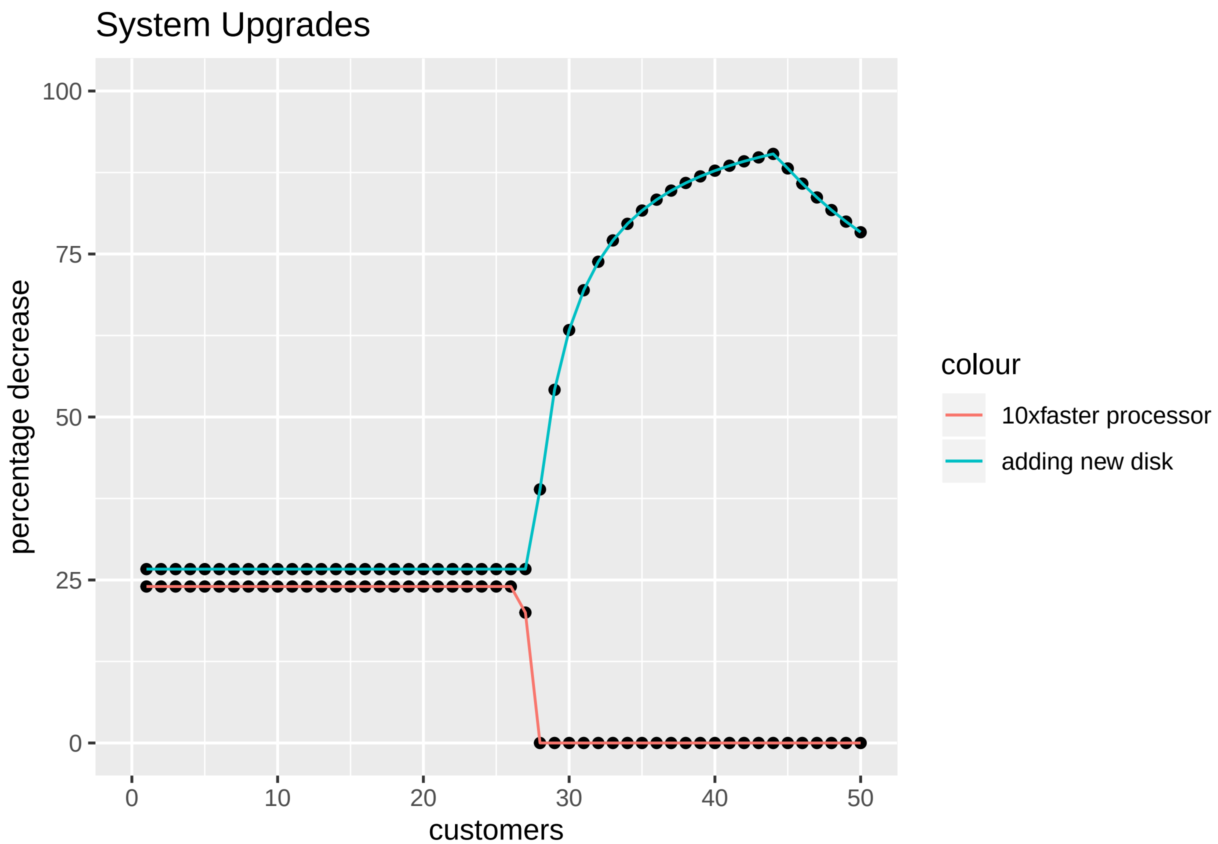
$$max(D_{newdisk}, ND_{max} - Z) \leq R_{newdisk}(N) \leq ND_{newdisk} \rightarrow max(0.22, 0.07\overline{3}N - 2) \leq R(N) \leq 0.07\overline{3}N$$

```
# calculate lower bounds for new disk
Rsd3 <- data.frame("N"=1:50, "R"=rep(0,50), stringsAsFactors = FALSE)

for(i in 1:50){
  Rsd3[i,2] = max(0.22, (0.073*Rsd3[i,1]) - 3)
}

# calculate percentage decrease
des <- data.frame("N"=1:50, "change"=rep(0,50), stringsAsFactors = FALSE)

for(i in 1:50){
  des[i,2] = ((abs(Rsd3[i,2]-Rs[i,2]))*100)/Rs[i,2]
}
```

```r
# plotting
library(ggplot2)
p <- ggplot(data = des, mapping = aes(x = N, y = change)) +
  xlim(0,50) + ylim(0, 100) +
  ggtitle("Percentage decrease in response time for new disk and load balancing") +
  xlab("customers") + ylab("percentage decrease") +
  geom_point() +
  geom_line(data = des, aes(x = N, y = change, color = "adding new disk"))
p
```




```r
p <- ggplot(data = des, mapping = aes(x = N, y = change)) +
  xlim(0,50) + ylim(0, 100) +
  ggtitle("System Upgrades") +
  xlab("customers") + ylab("percentage decrease") +
  geom_point() +
  geom_line(data = des, aes(x = N, y = change, color = "adding new disk")) +
  geom_point(data = despre, mapping = aes(x = N, y = change)) +
  geom_line(data = despre, aes(x = N, y = change, color = "10xfaster processor"))
p
```




So we can see that the change in the processor has minimal effect in light load which disappears in the heavy load. However, the change in the disks lead to a significant improvement, especially in heavy load condition. Because our bottleneck was the disk performance.

---

Question 4.

**a)**

$$answer!$$

**b)**

$$answer!$$

**c)**

$$answer!$$

---

Question 5.

**a)**

$$answer!$$

**b)**

$$answer!$$