Ch 7 [H]

Modification Analysis: "What-If" for Closed Systems

Preview

- asymptotic bounds
- modification analysis for closed systems
- closed vs. open networks

Review (7.1 [H])

Little's Law for an Open Systems

$$E[N] = \lambda \cdot E[T]$$

$$(E[N_Q] = \lambda \cdot E[T_Q])$$

$$(E[N_{red}] = \lambda \cdot E[T_{red}])$$

Little's Law for a Closed Batch System

$$N = X \cdot E[T].$$

(This actually holds for any closed system, but for interactive closed system, the following also holds.)

subsystem (system of servers & queues)

Little's Law for a Closed Interactive System

$$E[R] = \frac{N}{X} - E[Z] \qquad \left(\text{note: } E[T] = \frac{N}{X}\right)$$

Utilization Law

$$\rho_i = \frac{\lambda_i}{\mu_i} = \lambda_i E[S_i] = X_i E[S_i]$$

Forced Flow Law

$$X_i = E[V_i] \cdot X$$

Bottleneck Law

$\rho_i = X \cdot E[D_i]$, where D_i is the total service demand on device i for all visits of a single job.

7.2 [H] Asymptotic Bounds for Closed Systems

Let m be the number of devices in the system. $E[D_i]$ is as defined before, i.e., the expected total service demand on device i by a single job.

Let $D = \sum_{i=1}^{m} E[D_i]$

Let $D_{max} = \max_{i \in [1..m]} \{E[D_i]\}$

Theorem 7.1 *For any closed interactive system with N terminals,*

$$X \leq \min\left(\frac{N}{D + \mathbf{E}[Z]}, \frac{1}{D_{\max}}\right).$$

$$\mathbf{E}[R] \geq \max\left(D, N \cdot D_{\max} - \mathbf{E}[Z]\right).$$

Importantly, the first term in each clause ($\frac{N}{D+\mathbf{E}[Z]}$ or D) is an asymptote for small N, and the second term ($\frac{1}{D_{\max}}$ or $N \cdot D_{\max} - \mathbf{E}[Z]$) is an asymptote for large N.

Proof.

First, consider the large N (multiprogramming level) case

1. $\forall i \quad X \left[\frac{\text{jobs}}{\text{sec}}\right] \cdot \mathbb{E}[D_i]\ [\text{sec}] = \rho_i\ [\text{jobs}]\ (\text{utilization}) \leq 1$

(Remember that ρ_i is the expected number of jobs in server i.)

$$\Rightarrow \forall i \quad X = \frac{\rho_i}{E[D_i]} \leq \frac{1}{E[D_i]} \Rightarrow X \leq \frac{1}{D_{max}}.$$

2, $E[R] =$ (Little's Law for interactive closed systems) $= \frac{N}{X} - E[Z] \geq N \cdot D_{max} - E[Z]$.

Note that for large N, the D_{max} server is always busy ($\rho_{max} \simeq 1$), so $X = \frac{1}{D_{max}}$ is an asymptote for large N.

Second, consider the small N asymptote.

1. Let $E[R(N)]$ be the mean response time when the multiprogramming level is N. Then,

$$E[R(N)] \geq E[R(1)] = D \left(= \sum_{i=1}^{m} E[D_i]\right)$$

For low N, there is no "congestion," so the bound is tight.

2. $X = \dfrac{N}{E[R]+E[Z]} \leq \dfrac{N}{D+E[Z]}$.

$\square$

A Simple Example of Bounds

$E[Z] = 18$ sec

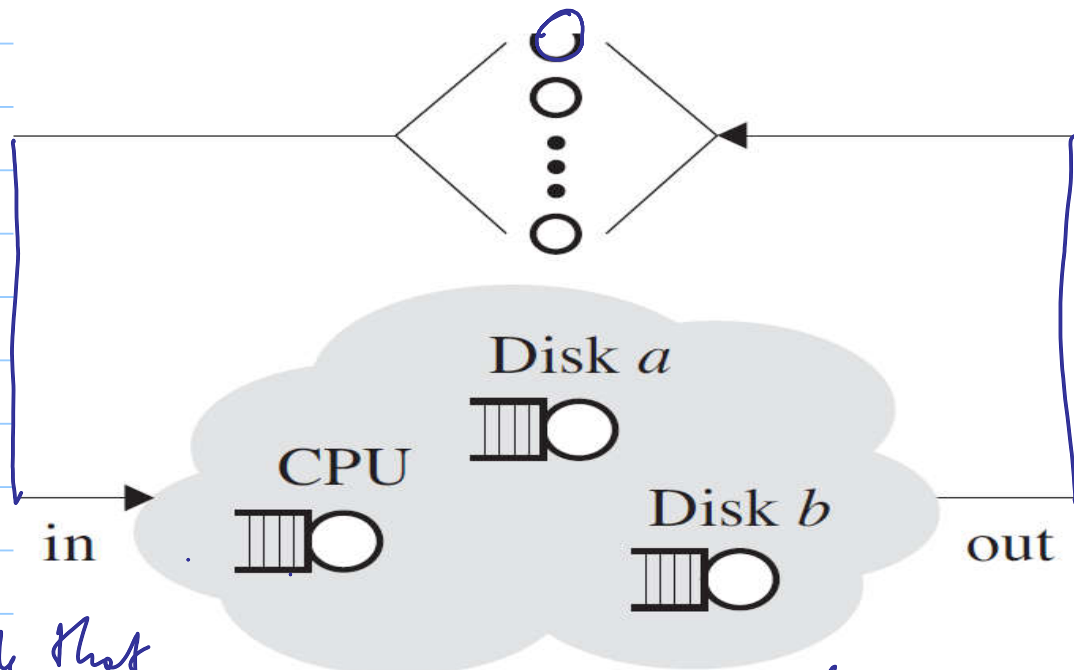
$E[D_{CPU}] = 5$ sec

$E[D_{disk\ a}] = 4$ sec

$E[D_{disk\ b}] = 3$ sec

So, $D = 5+4+3 = 12$ sec

$D_{max} = 5$ (the CPU is the bottleneck device)

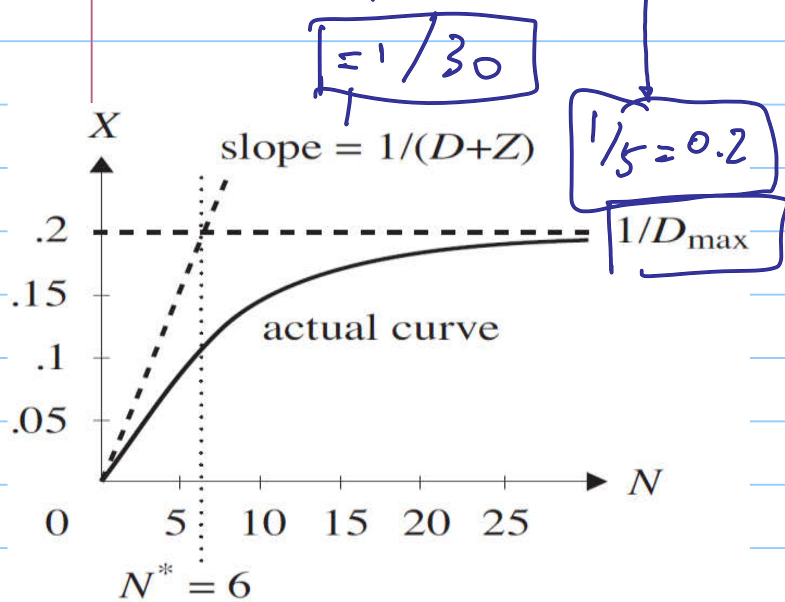


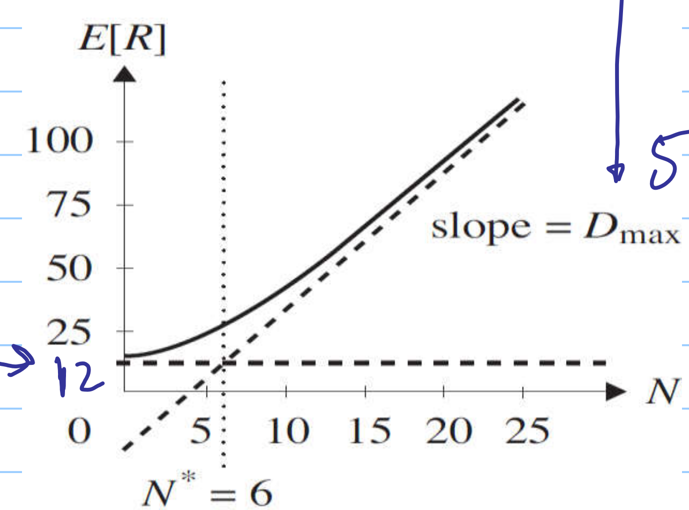
Central subsystem (Fig 7.1 [H])

Thm. 7.1 allows us to conclude that

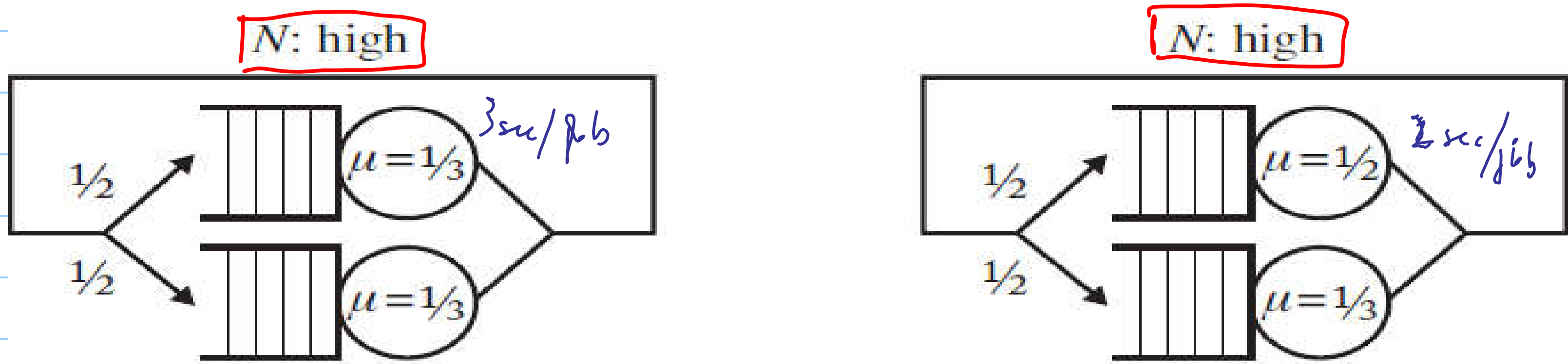
$X \le \min\left\{\frac{N}{12+18}, \frac{1}{5}\right\}$, and

$\boxed{E[R] \ge \max\{12, 5N - 18\}}$


= 1/30
slope = 1/(D+Z)
1/5 = 0.2
X
.2
.15
.1
.05
0
1/D_{max}
actual curve
5
10
15
20
25
N
$N^* = 6$


E[R]
100
75
50
25
12
0
5
slope = D_{max}
5
10
15
20
25
N
$N^* = 6$

7.3 [H] Modification Analysis for Closed Systems



(a) Original (b) "Improved"

The throughput is bounded above, for high N, by $1/D_{max}$, and D_{max}, the service demand on the slower disk $(\mu = \frac{1}{3})$ is unchanged!

Important Observations

The bounds for X and $E[R]$ meet at $N^* = \dfrac{D + E[Z]}{D_{max}}$.

- N^* represents the point beyond which there must be some queueing in the system. ($E[R] > D$).

- For fixed $N > N^*$, to increase throughput or lower response time, one must reduce D_{max}. Other changes will be largely ineffective.

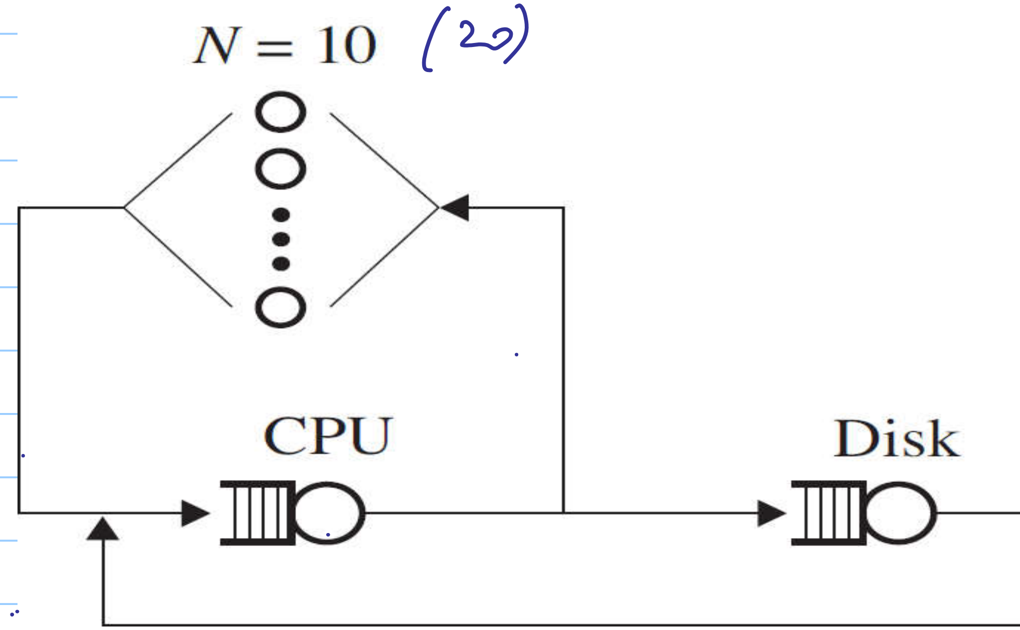
– If $E[Z] = 0$ (batch case), N^* decreases, i.e., the domination of D_{max} occurs with fewer jobs in the system

7.4 (H) More Modification Analysis Examples

Simple example

1. System A: $D_{CPU} = 4.6$, $D_{disk} = 4.0$ $\quad E[Z] = 5$

2. System B: $D_{CPU} = 4.9$, $D_{disk} = 1.9$

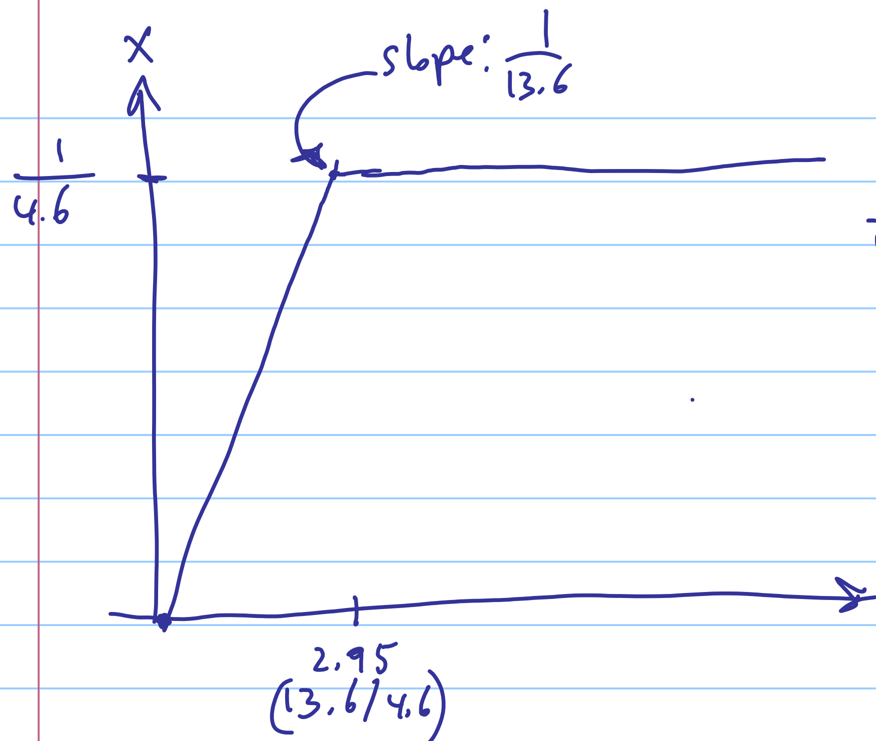


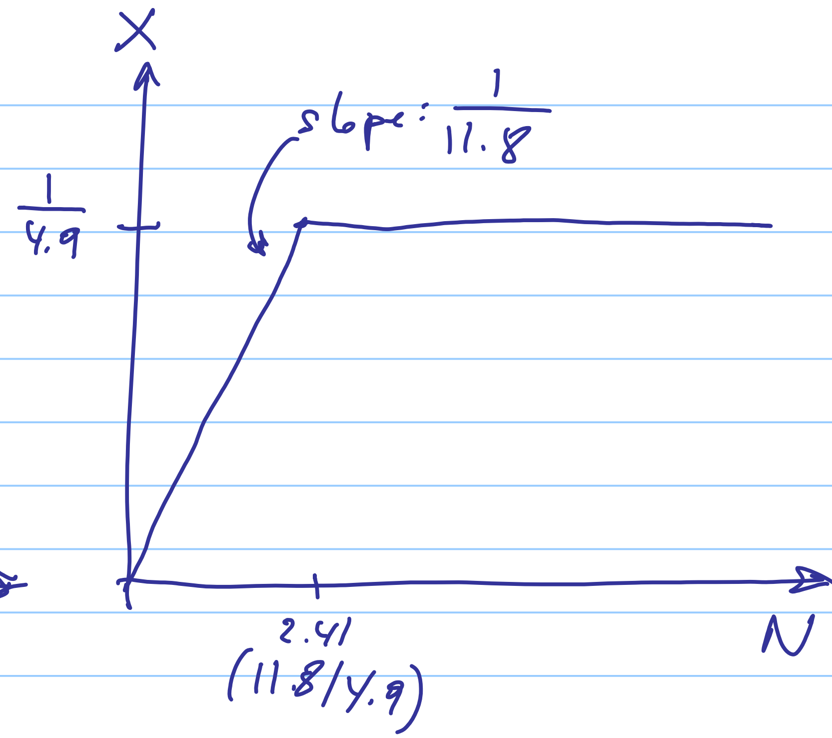
Which system has higher throughput?

$$N_A^* = \frac{D + E[Z]}{D_{max}} = \frac{8.6 + 5}{4.6} < 3$$

$$N_B^* = \frac{D + E[Z]}{D_{max}} = \frac{6.8 + 5}{4.9} < 3$$

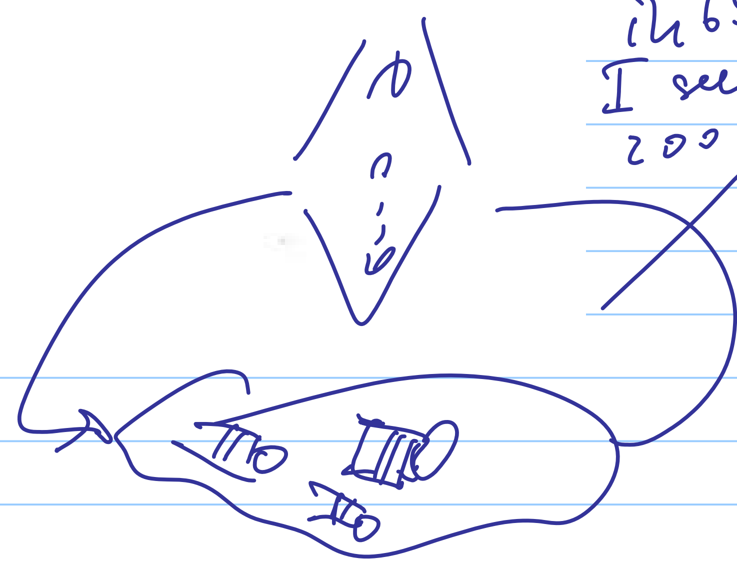
Since $N \gg N^*$ for both systems, $X \approx 1/D_{max}$, and therefore System A wins.


x
slope: 1/13.6
1/4.6
2.95
(13.6/4.6)


x
slope: 1/11.8
1/4.9
2.41
(11.8/4.9)
N

The following measurements were obtained for an interactive system[2]:

- $T = 650$ seconds (the length of the observation interval)
- $B_{\text{cpu}} = 400$ seconds
- $B_{\text{slowdisk}} = 100$ seconds
- $B_{\text{fastdisk}} = 600$ seconds
- $C = C_{\text{cpu}} = 200$ jobs
- $C_{\text{slowdisk}} = 2{,}000$ jobs
- $C_{\text{fastdisk}} = 20{,}000$ jobs
- $\mathbf{E}[Z] = 15$ seconds
- $N = 20$ users



in 650 sec, I see 200 jobs through here

In this example, we examine four possible improvements (modifications) – hence the name "modification analysis."

1. **Faster CPU:** Replace the CPU with one that is twice as fast.
2. **Balancing slow and fast disks:** Shift some files from the fast disk to the slow disk, balancing their demand.
3. **Second fast disk:** Buy a second fast disk to handle half the load of the busier existing fast disk.
4. **Balancing among three disks plus faster CPU:** Make all three improvements together: Buy a second fast disk, balance the load across all three disks, and also replace the CPU with a faster one.

Derived quantities - some expectation signs are dropped.

- $D_{cpu} = B_{cpu}/C = 400$ sec$/200$ jobs $= 2.0$ sec/job
- $D_{slowdisk} = B_{slowdisk}/C = 100$ sec$/200$ jobs $= 0.5$ sec/job
- $D_{fastdisk} = B_{fastdisk}/C = 600$ sec$/200$ jobs $= 3.0$ sec/job
- $\mathbf{E}[V_{cpu}] = C_{cpu}/C = 200$ visits$/200$ jobs $= 1$ visit/job
- $\mathbf{E}[V_{slowdisk}] = C_{slowdisk}/C = 2{,}000$ visits$/200$ job $= 10$ visits/job
- $\mathbf{E}[V_{fastdisk}] = C_{fastdisk}/C = 20{,}000$ visits$/200$ job $= 100$ visits/job
- $\mathbf{E}[S_{cpu}] = B_{cpu}/C_{cpu} = 400$ sec$/200$ visits $= 2.0$ sec/visit
- $\mathbf{E}[S_{slowdisk}] = B_{slowdisk}/C_{slowdisk} = 100$ sec$/2{,}000$ visits $= .05$ sec/visit
- $\mathbf{E}[S_{fastdisk}] = B_{fastdisk}/C_{fastdisk} = 600$ sec$/20{,}000$ visits $= .03$ sec/visit

$D_{max} = \max\{2, 0.5, 3\} = 3$

$D = D_{cpu} + D_{slowdisk} + D_{fastdisk} = 2 + 0.5 + 3 = 5.5$ $\quad$ $D + E[Z] = 5.5 + 15$

1. Faster CPU: Originally, $D_{\max} = 3$ sec/job, $D = 5.5$, $N^* = \frac{20.5}{3} \approx 7 \ll N$. $D_{\text{cpu}} \to 1$ sec/job does not change $D_{\max} = 3$ sec/job. Notice that N^* hardly changes at all. The fast disk is the bottleneck. We can never get more than 1 job done every 3 seconds on average.

2. Balancing slow and fast disks: Shift some files from the fast disk to the slow disk, balancing their demand. To do this we need that

$$V_{slow} + V_{fast} = 110 \text{ as originally}$$

$$\text{but } S_{slow} \cdot V_{slow} = S_{fast} \cdot V_{fast} \text{ because we are balancing the demand.}$$

Solving this system of linear equations yields the new demands $D_{slow} = D_{fast} = 2.06$. Now, $D_{max} = 2.06$ sec/job, although D increases slightly because some files have been moved from the fast disk to the slow disk.

$$N^* = \frac{2 + 2.06 + 2.06 + 15}{2.06} = \frac{21.12}{2.06} \approx 10.25$$

3. **Second fast disk:** We keep $D_{slow} = 0.5$, the same as before. However, we buy a second fast disk to handle half the load of the original fast disk. So now

$$D_{fast1} = D_{fast2} = 1.5 \text{ sec/job}.$$

Thus our new D_{max} is 2.0 sec/job (the CPU becomes the bottleneck).

$D_{CPU} = 2.0$

$D_{slowdisk} = 0.5$

$D_{fast_1} = D_{fast_2} = 1.5$

$D_{max} = 1.5$ sec/job

$D = \underline{5.5}$ sec/job

$$N^* = \frac{5.5 + 15}{1.5} = \frac{20.5}{1.5} = 13.6$$

4. Balancing among three disks plus faster CPU: We now make the CPU faster *and* balance load across all three disks, so

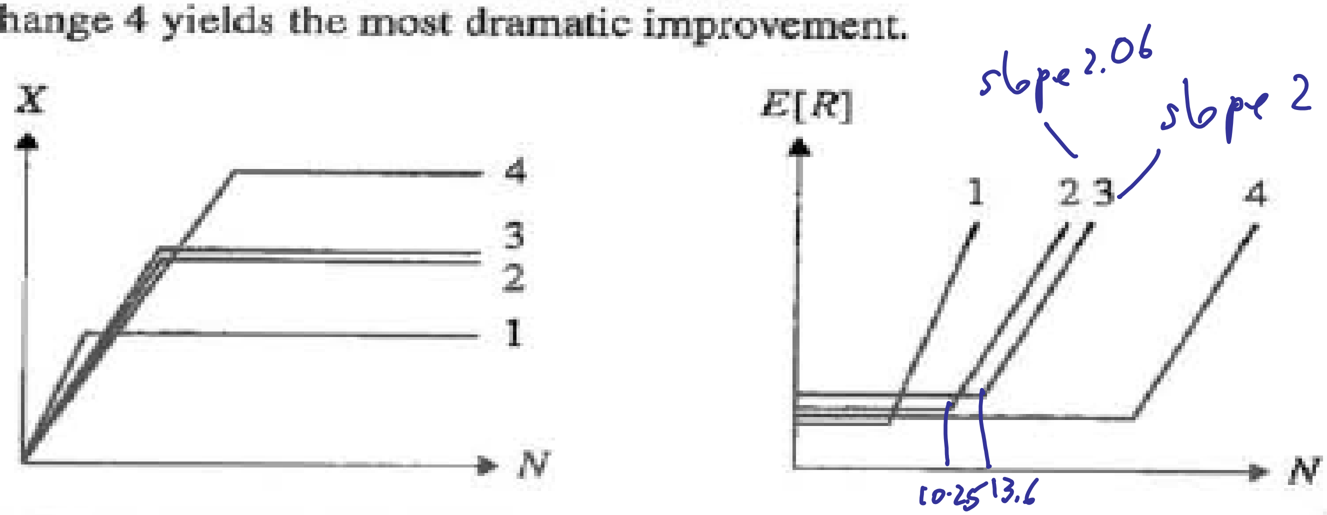
$$V_{slow} + V_{fast1} + V_{fast2} = 110.$$

$$S_{slow} \cdot V_{slow} = S_{fast1} \cdot V_{fast1} = S_{fast2} \cdot V_{fast2}.$$

Solving these simultaneous equations yields: $D_{disk1} = D_{disk2} = D_{disk3} = 1.27$. So $D_{max} = 1.27$, since we cut D_{cpu} to 1 already.

A graph of the results is shown in Figure 7.5. Assuming N is not too small, we conclude the following:

- Change 1 is insignificant.
- Changes 2 and 3 are about the same, which is interesting because change 2 was achieved without any hardware expense.
- Change 4 yields the most dramatic improvement.



The 'knee' is N^*.

Figure 7.5. Throughput and response time versus N, showing the effects of four possible improvements from the harder example, where the improvements are labeled 1, 2, 3, and 4.

Why does modification analysis not apply to open systems?

See 7.5 [H].