

תקשורת נתונים – תרגיל בית 2

1.

page size = 50KB, pin size = 250KB, RTT = 400ms, line speed = 100Mbps

א.

A. HTTP non-persistence:

$$\begin{aligned} T_a &= 2 * RTT + \frac{50KB}{100Mbps} + n * \left(2RTT + \frac{250KB}{100Mbps} \right) = \\ &= 800ms + 0.0039065sec + n(800ms + 0.01953125) = \\ &= 0.8039065sec + 0.81953125n \end{aligned}$$

B. HTTP persistence:

$$T_b = 2RTT + \frac{50KB}{100Mbps} + n \left(\frac{250KB}{100Mbps} + RTT \right) = 0.8039065s + 0.01953125n$$

C. HTTP persistence with pipelining:

$$\begin{aligned} T_c &= 3 * RTT + \frac{50KB}{100Mbps} + n \left(\frac{250KB}{100Mbps} \right) = 1.2sec + 0.00390625 + 0.01953125n = \\ &= 1.20390625 + 0.01953125n \end{aligned}$$

א. גדול .B.

ב. קטן .A.

2. א. נחשב עבור C-S:

$$N = 30, F = 1GB = 2^{33}b, \mu_s = 100MB/sec = 100 * 2^{23} b/sec,$$

$$\min(d_i) = 4 MB/sec = 2^{25} b/sec$$

$$\frac{N * F}{\mu_s} = \frac{30 * 2^{33}}{100 * 2^{23}} = \frac{30 * 1024}{100} = 307.2 sec,$$

$$\frac{F}{\min(d_i)} = \frac{2^{33}}{2^{25}} = 2^8 = 256 sec$$

$$\max\left(\frac{F}{\min(d_i)}, \frac{N * F}{\mu_s}\right) = 307.2 sec$$

נחשב עבור P2P:

$$F = 2^{33}b, \mu_s = 10 MB/sec = 10 * 2^{23} b/sec, \min(d_i) = 2^{25}b/sec, N = 30,$$

$$\sum \mu_i = 20 * 5 + 10 * 2 = 120 MB/sec = 120 * 2^{23} b/sec$$

$$\frac{F}{\mu_s} = \frac{2^{33}}{10 * 2^{23}} = 102.4 sec$$

$$\frac{F}{\min(d_i)} = \frac{2^{33}}{2^{25}} = 256 sec$$

$$\frac{N \cdot F}{\mu_s + \sum \mu_i} = \frac{30 \cdot 2^{33}}{10 \cdot 2^{23} + 120 \cdot 2^{23}} = \frac{30 \cdot 2^{33}}{130 \cdot 2^{23}} = \frac{30 \cdot 1024}{130} = 236.307 \text{ sec}$$

$$\max\left(\frac{F}{\mu_s}, \frac{F}{\min(d_i)}, \frac{N \cdot F}{\mu_s + \sum \mu_i}\right) = 256 \text{ sec}$$

נראה כי P2P מהיר יותר מאשר C-S ולכן עדיף לדון לשתף ב-P2P.

ב. ב C-S הגורם המשפיע הינו ה- μ_s . אם נשפר את מהירות העלאת הנתונים של השרת נוכל לקבל זמן מהיר יותר לשיתוף הקובץ.
ב P2P הגורם המשפיע ביותר הינו ה- $\min(d_i)$. אם נשפר את מהירות הורדת הנתונים של החברים האיטיים ביותר נוכל לקבל זמן מהיר יותר לשיתוף הקבצים.

3. א.

$$D_{wan} = 2s, D_{LAN} = 10 * X_{LAN}^2, X_{LAN} = 100 * \left(\frac{10 \cdot 2MB \cdot 4}{1Gbps \cdot 60}\right) = 100 \left(\frac{5 \cdot 2^{27}b}{15 \cdot 2^{32}bps}\right) = \frac{100}{96} = 1.0416$$

$$D_{LAN} = 1.0416^2 * 10 = 10.8506 \text{ ms}, D_{access} = 10 * X_{access}^2,$$

$$X_{access} = 100 * \left(\frac{10 \cdot 2MB \cdot 4}{50Mbps \cdot 60}\right) = 100 * \left(\frac{640Mb}{3000Mbps}\right) = 21.3333,$$

$$D_{access} = 21.3333^2 * 10 = 4551.1111$$

$$D_{tot} = D_{wan} + D_{LAN} + D_{access} = 2000 + 10.8506 + 4551.1111 = 6561.9617 \text{ ms} = 6.5619 \text{ s}$$

ב.

$$D_{wan} = 0.65 * 2s = 1.3s, X_{access} = 100 * \left(\frac{10 \cdot 2MB \cdot 4}{50Mbps \cdot 60}\right) * 0.65 = 13.8666$$

$$D_{access} = 10 * X_{access}^2 = 10 * 13.8666^2 = 1922.8444 \text{ ms}$$

$$X_{LAN} = 100 * \left(\left(\frac{10 \cdot 2MB \cdot 4}{1Gbps \cdot 60}\right) * 0.65 * 2 + \left(\frac{10 \cdot 2MB \cdot 4}{1Gbps \cdot 60}\right) * 0.35\right) = 1.7187$$

$$D_{LAN} = 10 * X^2 = 29.541 \text{ ms}$$

$$D_{tot} = D_{wan} + D_{LAN} + D_{access} = 1300 + 1922.8444 + 29.541 = 3252.3854 \text{ ms} = 3.2523 \text{ s}$$

ג. אם היינו מגדילים את ההסתברות לפגיעה אז ההשהיה הכוללת הייתה קטנה כיוון ששלוש ההשהיות היו קטנות יותר בעקבות השיפור וכך היינו מקצרים את הזמן.

ד. כן, מכיוון שהדבר יקצר את ההשהיה של האינטרנט (במיוחד במקרים של מרחקים ארוכים) ואת כמות הפניות לספקי אינטרנט אחרים ולכן יחסוך בזמן (ובכסף).

4.

א. הוא ייגש לזיכרון 0, העלות תהייה 1, ולא ימצא את הקובץ.

ב. הוא ייגש לכל הזיכרונות, העלות תהייה $2n+1$, והוא ימצא את הקובץ.

ג. הוא ייגש לזיכרון בו נמצא הקובץ (בוודאות לא 0), העלות תהייה 2 והוא ימצא את הקובץ.

5. א.

$$M \geq p * C + (1 - p) * (C + M)$$

$$p * M \geq C$$

$$p \geq \frac{C}{M}$$

ב. *Bloom Filter* הינו מבנה נתונים מתוחכם שיכול להגיד בוודאות אם קובץ אינו קיים ובסבירות גבוהה אם קיים. מבנה נתונים זה יכול להועיל כך שנחסוך את כל הגישות לקבצים שבוודאות לא קיימים.

ג. כיוון שאם המטמון אינו מעודכן בכל מקרה נאלץ לגשת לאינטרנט ולכן ככל שהמטמון יהיה יותר מעודכן יהיו פחות גישות לאינטרנט.

6. א. ה-*RTT* אינו משפיע על יעילות ה-*conditional get* לעומת *get* רגיל.

ב. גודל הקובץ מוסיף ליעילות של ה-*conditional get* לעומת *get* רגיל כיוון שככל שהגודל יותר גדול יותר יעיל להשתמש ב-*conditional get* אם הוא שמור ב-*cache*.

ג. קצב שינוי הקבצים ב"זיכרון הראשי" בשרת המרוחק יוריד מיעילות ה-*conditional get* לעומת *get* רגיל כיוון שככל שהקבצים משתנים בתדירות גבוהה יותר, כך ה-*cache* רלוונטי פחות ובכל מקרה יאלץ לגשת לאינטרנט.

7. *DPI* זוהי שיטה לעיבוד מידע שעובר ברשת מחשבים ויכול לפקח אחר כל פקטה עד החלקים הפנימיים של הפקטה למשל (ההדורים השונים, מאיפה היא הגיעה וכד').

מקורות: *Wikipedia, youtube*