

## מבוא לרשתות מחשבים אביב תשפ"ה

### תרגיל בית 4 - Load Balancer (Simplified)

תאריך הגשה: יום ב' 15/1/2026 עד השעה 23:58.

האחראי על התרגיל: גילעד

[שאלות והערות – בפורום הקורס במודל](#)

ההגשה בזוגות בלבד והינה אלקטרונית בלבד דרך אתר הקורס.

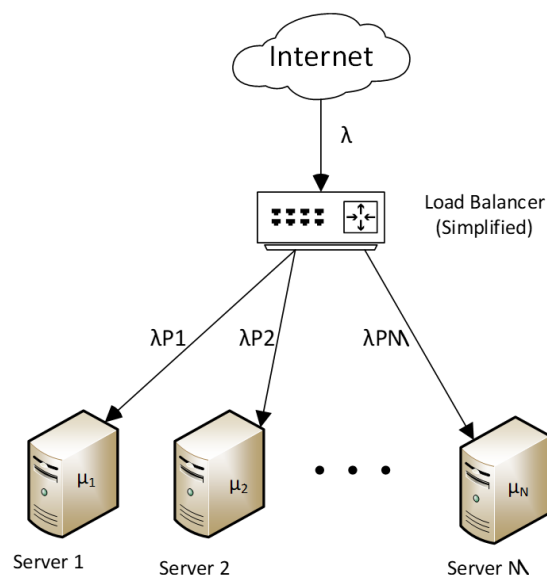
#### מבוא:

בתרגיל זה תממשו מודל מפושט של התקן חלוקת עומס (Load Balancer) בעזרת סימולציה של מערכת תורים.

התרגיל יבנה בשלבים כשבשלב האחרון התוכנית שתכתבו תסמלץ את המערכת הבאה:

#### Load Balancer

להתקן שלנו יש פורט כניסה אחד (Ingress) דרכו נכנסות הודעות בקשה לשירות מהאינטרנט. להתקן יש  $M$  יציאות (Egress) דרכן משדר ה-Load Balancer את ההודעות ל  $M$  שרתים שונים. במודל המפושט שלנו אין תעבורה החוזרת מהשרתים חזרה לאינטרנט.



- בקשות השירות כולן זהות.
- מופע בקשות השירות מהאינטרנט מפולג פואסוניית עם פרמטר  $\lambda$  בקשות ליח' זמן.
- ה-Load Balancer מפנה את בקשות השירות באופן אקראי אל השרתים השונים בהסתברות המתאימה לכושר העיבוד של השרת. לדוגמה: בקשת שירות שמגיעה מהאינטרנט תופנה לקבלת שירות בתחנה  $i$  בהסתברות  $P_i$  ( $0 \leq P_i \leq 1$ ).  $P_i$  יהיו חלק מהקלט לסימולציה. יש לוודא ש  $\sum_{i=1}^M P_i = 1$

- זמן העיבוד של הודעה בתוך ה Load Balancer, זמן השידור וזמן ההתפשטות של ההודעות מה load balancer לשרתים – זניח.
- כל שרת מחזיק בכניסה תור של בקשות לשירות, גודל התור של השרת ה-  $i$  הוא  $Q_i$  בקשות ( $0 \leq Q_i$ ). שימו לב שבמימוש זה גודל התור לא כולל את הבקשה שבשירות.
- אם בקשה מגיעה לשרת והשרת פנוי היא תכנס לשירות מיידית.
- אם בקשה מגיעה לשרת והשרת אינו פנוי היא תכנס להמתנה בתור של השרת אם יש בו מקום.
- אם בקשה מגיעה לשרת כשהשרת אינו פנוי והתור מלא, הבקשה תיזרק ולא תקבל שירות.
- בכל השרתים בקשות השירות מקבלות שירות לפי סדר ההגעה (FIFO).
- קצב השירות של שרת  $i$  מפולג פואסוני עם פרמטר  $\mu_i$  בקשות ליח' זמן.

הנחיות כלליות למימוש:

- ניתן להשתמש בכל שפה שנוחה לכם
- יש לוודא את נכונות הקלט
- מומלץ לתעד את הקוד באופן הבא:
  - (סעיף זה אינו מחייב אך יקל על הבודקים להבין מה עשיתם במקרה שווקטור בדיקה יכשל)
  - בתחילתו הסבר כללי על הרעיון המרכזי במימוש
  - לכל פונקציה הסבר קצר לגבי מטרתה
  - במקומות ספציפיים בהם קיים איזשהו תחכום שהקורא הסביר (=סטודנט אחר בקורס) עשוי להיתקל בקושי בהבנה מומלץ להוסיף הסבר

## 1. מימוש תור M/M/1/N

כתבו סימולציה לתור M/M/1/N (תור M/M/1 בעל N מקומות במערכת - כולל הבקשה שבשירות).

1.1. הקלט לסימולציה יהיה:

1.1.1.  $\lambda$  – קצב ההגעה [הודעות ליח' זמן]

1.1.2.  $\mu$  – קצב השירות [הודעות ליח' זמן]

1.1.3.  $N$  – מספר ההודעות במערכת (כולל זו שמקבלת שירות)

1.1.4.  $T$  – זמן ריצת הסימולציה (באותן יחידות זמן כמו קצב המופע וקצב השירות)

1.2. הפלט יהיה:

1.2.1.  $A$  – מספר הבקשות שקבלו שירות

1.2.2.  $B$  – מספר הבקשות שלא קבלו שירות, בעקבות התקלות בתור מלא או בעקבות תום זמן הסימולציה

הסימולציה תיעצר בזמן  $T$  והודעה שנמצאת בטיפול תחשב כאחת שלא טופלה.

הסימולציה נדרשת להיות מסוג event-driven. סימולציה מסוג זה מכילה תור של אירועים (כגון הגעת לקוח או עזיבת לקוח) שמנוהל באופן דינאמי (טיפול באירוע עשוי להכניס אירועים נוספים לתור או להוציא ממנו אירועים בהתאם לאפיון המערכת)

בסימולציה של התור אתם תגרילו מופע הודעות בעל פילוג פואסוני וזמן שירות בעל פילוג אקספוננציאלי. וודאו שבכל ריצה של הסימולציה מוגרל גרעין (seed) שונה. אחרת, כל הריצות עשויות להיות זהות זו לזו.

הנחיות מפורטות ודוגמה בפייתון ניתן למצוא ב:

<https://www.linkedin.com/pulse/simulating-single-server-queue-python-chirag-subramanian-bpqne/>

1.2. (10 נקודות) האם ניתן לייצר גרסה דומה של תור אינסופי? כיצד תציעו לשנות את המימוש? מה תהיה המשמעות של B במקרה זה?

2. (10 נקודות) עבור  $\lambda = 10$ ,  $\mu = 15$ ,  $N=1000$ ,  $T=5$  הריצו את הסימולציה 5 פעמים רשמו בטבלה את:

2.2. מספר הצרכנים שקיבלו שירות

2.3. זמן ההמתנה הממוצע של צרכנים במערכת (כולל קבלת שירות)

מספר הצרכנים שקיבלו שירות	זמן ההמתנה הממוצע	
		1
		2
		3
		4
		5

3. (20 נקודות) עבור  $\lambda = 10$ ,  $\mu = 15$ ,  $N=1000$  ולאחר התייצבות התור, ענו בהתאם לחומר התאורטי שנלמד בהרצאות ובתרגולים:

3.2. מה ההסתברות שהודעה שמגיעה למערכת תמצא תור מלא ולכן תיזרק? (ניתן להיעזר בכלים נומריים לחישוב סכום הטור)

3.3. חשבו את תוחלת מספר הצרכנים שקיבלו שירות ואת תוחלת זמן ההמתנה הממוצע במערכת (כולל קבלת שירות) בתור  $M/M/1$  (בעל תור אינסופי), ניתן להיעזר בנוסחאות שנלמדו בכיתה.

3.4. הסבירו את הפער (השגיאה) בין התוצאה התיאורטית לבין התוצאה שהתקבלה בפועל בסימולציה, בהתייחס אך ורק לעמודת **מספר הצרכנים שקיבלו שירות** כפי שמופיעה בטבלה שבסעיף 2. מהם הגורמים האפשריים לפער זה?

כמו כן, אם הייתם יכולים לשנות **משתנה קלט אחד** בסימולציה (למשל לשנות את קצב ההגעה מ- $\lambda = 10$  ל- $\lambda = 5$ ), איזה משתנה הייתם בוחרים לשנות, ובאיזו צורה, על מנת להקטין את הפער? אין צורך להריץ מחדש את הסימולציה.

3.5. הדפיסו שני גרפים המציגים את השגיאה היחסית באחוזים\* כפונקציה של זמן ריצת הסימולציה T עבור שני פרמטרים:

- זמן ההמתנה הממוצע במערכת כולל שירות
- מספר הצרכנים שקיבלו שירות

עליכם להריץ את הסימולציה עבור  $T = 10, 20, 30, \dots, 90, 100$  ולחשב את השגיאה הממוצעת על פני 20 הרצות. האם התוצאות שהתקבלו עולות בקנה אחד עם ההסבר שניתן בסעיף 3.3? נמקו.

\*שגיאה יחסית באחוזים בפרמטר x מוגדרת להיות:

$$Error(x) \equiv \frac{|Theoretical\ Value(x) - x|}{Theoretical\ Value(x)} \cdot 100$$

4. (10 נקודות) למדנו בהרצאה שתנאי ליציבות התור הוא  $\lambda < \mu$ , והזכרנו במפורש כי התנאי  $\lambda \leq \mu$  איננו מספק. הציעו ניסוי על התור שמימשתם, שיראה אמפירית כי התור אינו יציב במקרה שבו  $\lambda = \mu$ .
5. הרחיבו את הסימולציה כדי לממש את ה Load Balancer:

אתם נדרשים לממש את ה- Load Balancer המפושט בעזרת סימולציה מבוססת אירועים (Event Driven Simulation).

קלט הסימולטור (לפי הסדר):

- T - זמן הפעולה הכולל של הסימולציה.
- לאחר T יחידות זמן (כולל נקודת הזמן T) לא יגיעו עוד בקשות לשירות לפורט הכניסה של ה Load Balancer אולם הבקשות הקיימות בשרתים מטופלות.
- M - מספר השרתים ( $1 \leq M$ )
- $N - P_1 P_2 \dots P_i \dots P_M$  ערכים מופרדים ברווח ומייצגים את ההסתברות שההתקן יעביר בקשה שנכנסת לשרת i. כל הבקשות שנכנסות ל Load Balancer מועברות לקבלת שירות ( $\sum_{i=1}^N P_i = 1$ )
- $\lambda$  - קצב מופע בקשות השירות מהאינטרנט [בקשות ליח' זמן].
- $M - Q_1 Q_2 \dots Q_i \dots Q_M$  ערכים מופרדים ברווח ומייצגים את גדלי התורים של השרתים (מספר הבקשות המקסימאלי ששרת יכול לשמור לפני קבלת שירות)
- שימו לב שמספר ההודעות הכולל המקסימאלי שיכולות להיות בשרת ה- i הוא  $Q_i + 1$
- $M - \mu_1 \mu_2 \dots \mu_M$  ערכים מופרדים ברווח ומייצגים את קצבי השירות של השרתים [בקשות ליח' זמן]

יש לוודא נכונות הקלט והפקודה אשר תריץ את הסימולציה הינה מהצורה:  
(כל הערכים מופרדים ביניהם ברווח)

> ./simulator T M  $P_1 P_2 \dots P_i \dots P_M$   $\lambda Q_1 Q_2 \dots Q_i \dots Q_M$   $\mu_1 \mu_2 \dots \mu_M$

דוגמה מספרית לקלט כזה הינה:

> ./simulator 5000 2 0.2 0.8 200 2 10 20 190

במקרה זה, הסימולציה תרוץ למשך 5000 יחידות זמן עם שני שרתים. הסתברות שבקשת שירות תנותב לשרת מס' 1 היא 0.2 והסתברות שבקשת שירות תנותב לשרת מס' 2 היא 0.8. קצב הגעת בקשות השירות הוא 200 בקשות ביחידת זמן. אורך התור בשרת הראשון הינו 2 ובשרת השני הינו 10. קצב השירות של שרת מס' 1 הוא 20 בקשות ביח' זמן וקצב השירות של השרת השני הוא 190 בקשות ליחידת זמן.

פלט הסימולטור הינו שורה אחת מופרדת ברווחים ומכילה את הפרמטרים הבאים לפי הסדר:

- A - מספר הבקשות שקיבלו שירות.
- B - מספר הבקשות שנתקלו בתור מלא ונזרקו ללא קבלת שירות.
- $T_{end}$  - זמן סיום הטיפול בהודעה האחרונה
- $\overline{T_w}$  - זמן ההמתנה הממוצע של הודעה במערכת השרתים לפני קבלת שירות (רק עבור הודעות שלא נזרקו)
- $\overline{T_s}$  - זמן השירות הממוצע של הודעה במערכת השרתים

יש לעגל את התוצאות ל 4 ספרות מימין לנקודה העשרונית.

לדוגמה:

עבור קלט הדוגמה הפלט עשוי להיות:

871348 129125 5000.1400 0.0201 0.0135

או

871560 129357 5000.0319 0. 0.0237 0.0101

'בדיקת שפיות' היא ריצה של הסימולציה עבור מקרים שאת תוצאתם ניתן לצפות ובכל לקבל חיזוק מסוים (גם אם מוגבל) לנכונות הסימולציה.

5.2. הריצו 'בדיקת שפיות' לסימולציה שכתבתם עבור המקרים הבאים, מהו פלט הסימולציה והאם הוא תואם את התאוריה?  
1. בדיקת שרת בודד

5.2.1.1. > ./simulator 5000 1 1 20 1000 40

הרחבת המקרה הקודם למספר שרתים אבל עם תוצאה דומה (שעדיין ניתן לצפות)

5.2.1.2. > ./simulator 5000 4 1 0 0 0 20 1000 1000 1000 1000 40 40 40 40

5.2.1.3. > ./simulator 5000 4 0 0 1 0 20 1000 1000 1000 1000 40 40 40 40

5.2.1.4. > ./simulator 5000 4 0.001 0.001 0.997 0.001 20 1000 1000 1000 1000 40 40 40 40

בדיקת מקרה קצה של תור בגודל 0

5.2.1.5. > ./simulator 5000 4 0 0 1 0 20 0 0 0 0 40 40 40 40

בדיקת מקרה של קצב שירות נמוך

5.2.1.6. > ./simulator 5000 4 0.25 0.25 0.25 0.25 20 100 100 100 100 0.5 0.5 0.5 0.5

אופן הבדיקה:

הסימולציה שלכם תורץ בעזרת כלי אוטומטי. זמן הריצה יהיה 10000 יח' זמן וקצב המופע יהיה 200 הודעות ליח' זמן. יורצו 10 וקטורי בדיקה, בדומה לדוגמאות מעלה. השגיאה המותרת: 10% (וקטור תקין הוא וקטור שבו כל הפרמטרים בפלט נמצאים בטווח השגיאה המותר). כל וקטור מזכה ב 5 נקודות.

## הגשה

- יש להגיש אלקטרונית דרך אתר הקורס קובץ zip יחיד בשם <id1>-<id2>.zip (שימו לב רק zip ולא כל כיווץ אחר)
- בתוך קובץ ה- zip יימצא בין השאר קובץ makefile כך שהרצת הפקודה make לאחר פתיחת ה- zip תיצור את קובץ ההרצה בשם simulator שימו לב שהדרישה ל- makefile הינה להקל עליכם לבחור שפת פיתוח (ראו הגבלה בהמשך) ישנן שפות פיתוח שלא דורשות קומפילציה ועבורן יש להגיש makefile ריק שלא מבצע כלום ולהגיש ביחד איתו את סקריפט ההרצה simulator

- בתוך קובץ ה zip -יימצא בנוסף קובץ שייקרא dry.pdf אשר כולל את התשובות לשאלות בגיליון זה.
- את הסימולטור ניתן לכתוב בכל שפת תכנות שתמצאו (פייתון מגרסה 3.10 ומעלה).
- התרגיל יבדק על מכונת Linux 22.04

- הסימולציה שלכם צריכה להסתיים תוך 2 דקות, כל ריצה שלא תסתיים תוך זמן זה תיחשב כריצה תקועה ותגרום להורדת ניקוד. המטרה של מגבלה זו הינה להימנע מלולאות אינסופיות בתוכניות שלכם, ולא לגרום לכם להשקיע זמן באופטימיזציה של הקוד שלכם. כל עוד הקוד שלכם סביר ואין בו לולאות אינסופיות הקוד שלכם אמור להסתיים בזמן זה.
- עקב הבדיקה האוטומטית, הגשות שלא יעמדו בתנאי ההגשה יקבלו ניקוד נמוך, לכן בידקו היטב את תוצאותיכם.

כדי לוודא שכולכם מבינים היטב את התוכנית שכתבתם, חלק מהקבוצות עשויות להתבקש לבצע code review בפגישה אישית עם צוות הקורס. בפגישה זו כל אחד מחברי הקבוצה ידרש להציג הבנה מעמיקה של הקוד שכתבתם.