

## מבוא לרשותות מחשבים אביב תשפ"ה

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

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

האחראי על התרגיל: גילעד

#### שאלות והערות – בפורום הקורס במודול

ההגשה בזוגות בלבד והינה אלקטרוני בלבד דרך דרכו אתר הקורס.

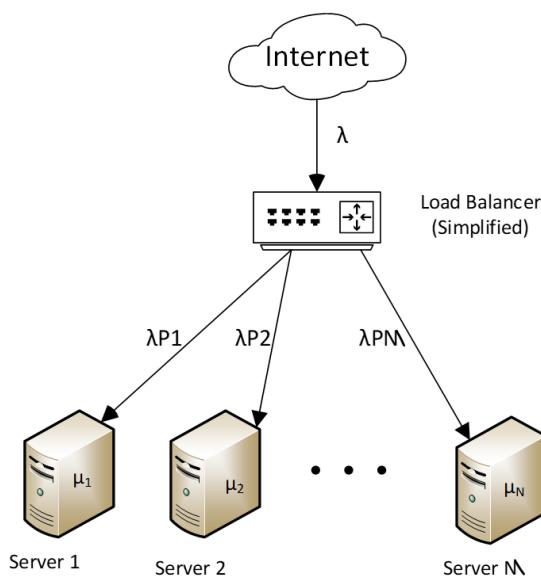
#### מבוא:

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

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

#### Load Balancer

להתקן שלנו יש פורט כניסה אחד (Ingress) דרכו נכנסות הודעות בקשה לשירות מהאינטרנט. להתקן יש M יציאות (Egress) דרכן משדר הרשתות ל M שרתים שונים. במודל המפוסט שלנו אין מעבורה החזרת מהשרתים חוזרת לאינטרנט.



- בקשות השירות כולן זוחות.
- מופע בקשות השירות מהאינטרנט מפולג פואטונית עם פרמטר ג' לבקשת ליח' זמן.
- ה- Load Balancer מפנה את בקשות השירות באופן אקראי אל השירותים השונים בהסתברות המתאימה לכשור העיבוד של השירות. לדוגמה: בבקשת שירות שmagia מהאינטרנט תופנה לקבלת שירות בתנה ? בהסתברות  $P_i \leq 1$  ( $P_i = 0$ ). יהי חלק מהקלט לסימולציה. יש לוודא ש  $\sum_{i=1}^M P_i = 1$ .

- זמן העבודה של הודעה בתוך ה Load Balancer, זמן השידור וזמן ההתפשטות של ההודעות מה load balancer לשירותים – זניח.
- כל שרת מחזיק בכניסה תור של בקשות לשירות, גודל התור של השירות ה-  $i$  הוא  $Q_i$  בקשות ( $Q_i \leq 0$ ). שימוש לב שבמימוש זה גודל התור לא כולל את הבקשה לשירות.
- אם בקשה מגיעה לשירות והשירות אינו פנוי היא תכנס למתנה בתור של השירות אם יש בו מקום.
- אם בקשה מגיעה לשירות והשירות אינו פנוי והتور מלא, הבקשה תיזרק ולא תקבל שירות.
- בכל השירותים בקשות השירות מקבלות שירות לפי סדר ההגעה (FIFO).
- קצב השירות של שירות  $i$  מפוגג פואסונית עם פרמטר  $\mu_i$  בקשות ליח' זמן.

הנחיות כלליות למימוש:

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

1. מימוש תור N/M/M/M/N/M (תור 1/M/M/M בעל N מקומות במערכת - כולל הבקשה לשירותים).
    - 1.1. הקלט לSIMOLCIA יהיה:
      - 1.1.1. ג – קצב ההגעה [הודעות ליח' זמן]
      - 1.1.2. מ – קצב השירותים [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 שקיבלו שירות
      - 1.2.2. B – מספר הבקשות שלא קיבלו שירות, בעקבות התקלות בתור מלא או בעקבות תום זמן הסימולציה

הסימולציה תיעזר בזמן T והודעה שנמצאת בטיפול תחשב כאחת שלא טופלה.
- הסימולציה נדרשת להיות מסוג event-driven. סימולציה מסוג זה מכילה תור של אירועים (כאו הגעת לקוחות או עציבת לקוחות) שמנוהל באופן דינامي (טיפול באירוע עשי להכניס אירועים נוספים לתור או להוציא מהם אירוע בהתאם לאירוע המערכת)
- בסימולציה של התור אתם תగירו מופע הודעה בעל פילוג פואסוני וזמן שירות בעל פילוג אקספוננציאלי. וודאו שככל ריצה של הסימולציה מוגרל גרעין (seed) שונה. אחרת, כל הריצותעשויות להיות זהות זו לזו.
- הנחיות מפורטות ודוגמה בפייטון 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  $T = 5$ ,  $N = 1000$ ,  $\lambda = 15$ ,  $\mu = \mu$ , הריצו את הסימולציה 5 פעמים רשמו בטבלה את:

2.2. מספר הזרים שקיבלו שירות  
2.3. זמן ההמתנה הממוצע של זרים במערכת (כולל קבלת שירות)

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

3. (20 נקודות) עבור  $T = 5$ ,  $N = 1000$ ,  $\lambda = 15$ ,  $\mu = \mu$ , ולאחר התייצבות התור, ענו בהתאם לחומר התאורטי שנלמד בהרצאות ובתרגילים:

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

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

3.4. הסבירו את הפער (השgiaה) בין התוצאה התיאורטית לבין התוצאה שהתקבלה בפועל בסימולציה, בהתייחס אריך ל**עומדת מספר הזרים 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- $\lambda = 5$  ל- $\lambda = 10$ ), איזה משתנה הייתם בוחרים לשנות, ובאיזה צורה, על מנת להקטין את הפער? אין צורך להריץ מחדש את הסימולציה.

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

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

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

\*שגיאה יחסית באחזים 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  $\mu < \lambda$ , והזכרנו במפורש כי התנאי  $\mu \leq \lambda$  איננו מספק. הցינו ניסוי על התור שמיימותם, שיראה אמפירית כי התור אינו יציב במקרה שבו  $\mu = \lambda$ .
5. הרחיבו את הסימולציה כדי למשת את ה Load Balancer :

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

קלט הסימולטור (לפי הסדר):

- T - זמן הפעולה הכללי של הסימולציה.  
לאחר T יחידות זמן (כולל נקודות הזמן T) לא יגיעו עוד בקשות לשירות לפורת הכניסה של ה Load Balancer אולם הביקשות הקיימות בשרתים מטופלות.
- M – מספר השירותים ( $M \leq 1$ )  
 $P_M \dots P_1 P_2 \dots P_i \dots P_N$  – N ערכים מופרדים ברוחם ומיצגים את ההסתברות שהתקן יעביר בקשה שנכנסת לשרת  $i$ . כל הבקשות שנכנסות ל Load Balancer מועברות לקבלת שירות  $(\sum_{i=1}^N P_i = 1)$
- $\lambda$  – קבוע מופיע בקשות השירות מהאינטרנט [בקשות ליח' זמן].
- $Q_M \dots Q_1 \dots Q_2 \dots Q_i \dots Q_N$  – M ערכים מופרדים ברוחם ומיצגים את גודלי התורים של השירותים (מספר הבקשות המקורי לשרת יכול המקסימלי לשירות יכול לשירות) שימו לב שמספר ההודעות הכללי המקורי לשירות יכול לשירות יכול לשירות  $i$  הוא  $1 + Q_i$
- $\mu_1 \dots \mu_M$  – M ערכים מופרדים ברוחם ומיצגים את קבוע השירות של השירותים [בקשות ליח' זמן]

יש לוודא נכונות הקלט והפקודה אשר תרייצ' את הסימולציה הינה מהצורה:  
(כל הערכים מופרדים ביניהם ברוחם)

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

דוגמה מספירת לקלט כזה הינה:

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

במקרה זה, הסימולציה תרוץ במשך 5000 יחידות זמן עם שני שירותים. הסתברות שבקשת שירות תנוטב לשרת מס' 1 היא 0.2 וההסתברות שבקשת שירות תנוטב לשרת מס' 2 היא 0.8. קבוע הגעת בקשות השירות הוא 200 בקשות ביחידת זמן. אורך התור בשרת הראשון הוא 2 ובשרת השני הוא 10. קבוע השירות של שירות מס' 1 הוא 20 בקשות ביח' זמן וקבץ השירות של השירות השני הוא 190 בקשות ליחידת זמן.

פלט הסימולטור הינו שורה אחת מופרדת ברוחים ומכליה 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 האחורונה
- $\bar{T}_w$  – זמן ההמתנה הממוצע של הودעה במערכת השירותים לפני קבלת שירות (רק עבור הודעות שלא נזרקו)
- $\bar{T}_s$  – זמן השירות הממוצע של הודעה במערכת השירותים

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

'בדיקה שפויות' היא ריצה של הסימולציה עבור מקרים שאת תוצאותם ניתן לצפות ובכל מקרה חיזוק מסוים (גם אם מוגבל) לנכונותה הסימולציה.

5.2. הריצו 'בדיקה שפויות' לסימולציה שתכתבם 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 (שעדין ניתן לצפות)

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

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

בדיקה מקירה קצרה של תור בגודל 0

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

בדיקה מקירה 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

אוף הבדיקה:

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

## הגשה

- יש להגיש אלקטронית דרך האתר הקורס קובץ zip יחיד בשם <id1>-<id2>.zip. (שימוש לב רק zip ולא כל כיווץ אחר)
- בתוך קובץ ה- zip נמצא בין השאר קובץ makefile כך שהרצת הפקודה make לאחר פתיחת ה- zip תיצור את קובץ ההרצה בשם simulator
- שימוש לב שהדרישה ל- makefile הינה להקל عليיכם לבחור שפת פיתוח (ראו הגבלת בהמשך) ישנן שפות פיתוח שלא דורשות קומpileציה ובעורן יש להגיש makefile ריק שלא מבצע כלום ולהגיש ביחיד אותו את סק裏יפט ההרצה simulator

- בתוך קובץ ה zip - יימצא בנוסף קובץ שנקרא pdf.dry אשר כולל את התשובות לשאלות בגילון זה.
  - את הסימולטור ניתן לכתוב בכל שפת תכנות שתרצה (פייתון מגרסה 3.10 ומעלה).
  - התרגיל ידק על מכונת Linux 22.04
- הסימולציה שלכם צריכה להסתיים תוך 2 דקות, כל ריצה שלא تستיים תוך זמן זה תיחס כריצה תקינה ותגרום להורדת ניקוד. המטרה של מגבילה זו היא להימנע מלולאות אינסופיות בתוכניות שלכם, ולא לגרום לכם להשקיע זמן באופטימיזציה של הקוד שלכם. כל עוד הקוד שלכם סביר ואינו בו לוולאות אינסופיות הקוד שלכם אמרור להסתיים בזמן זה.
- عقب הבדיקה האוטומטית, הgeshotות שלא יעדכו בתנאי ההגשה יקבלו ניקוד נמוך, لكن בידקו היטב את תוצאותיכם.

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