

סדנה ברשתות תקשורת (67613) - תרגיל 1

נטלי יוסופוב, 319085726 | מור חן, 206407314

1 במאי 2023

תיאור המימוש

בתרגיל ישנם שלושה קבצים - הקובץ הראשון הוא *server.cpp*, הקובץ השני הוא *client.cpp* והקובץ השלישי הוא קובץ *utils.h*. כשמים הם, הקובץ הראשון דואג לפונקציות של השרת, הקובץ השני לפונקציות של הלקוח והקובץ השלישי ספריית עזר ששני הקבצים מייבאים. פירוט אודות הקבצים השונים:

1. קובץ *utils.h* - הקובץ מכיל קבועים וספריות המשומשים על ידי שני הקבצים האחרים. לדוגמה, הוא מכיל מערך של גדלי הודעות השונים, את הפורט הייעודי לתקשורת ואת הודעות ה-*reply* (שנקבעה מראש) שתישלח.

2. קובץ *server.cpp* - הקובץ בודק מהו ה-*IP* של המחשב עליו הוא רץ באמצעות שם ה-*host* שלו ומדפיס אותו אל המסך (כדי שנוכל להעבירו כארגומנט ללקוח, כמתואר בתרגיל). לאחר מכן, הוא פותח *socket* ומקשרו לכתובת ה-*IP* ולפורט ייעודי שנקבע מראש וממתין להתחברות מצד הלקוח. בכל פעם שלקוח מבקש להתחבר (עבור כל גודל של הודעות, נפתח מחדש *socket* של הלקוח), השרת מעבד את ההודעות הנשלחות ממנו (בלולאה מגודל מספר ההודעות המצופה) ולאחר שסיים לקבל את כל ההודעות, הוא שולח הודעת *reply* שפירושה שהוא סיים את עיבוד ההודעות והלקוח יכול להמשיך הלאה. עבור כל גודל הודעה, הודעת *Reply* נשלחת הן לאחר סיום עיבוד הודעות החימום והן לאחר סיום עיבוד ההודעות האמיתיות. בסוף התהליך, הוא סוגר את ה-*socket* של השרת.

3. קובץ *client.cpp* - הקובץ בודק תחילה כי קיבל את מספר הארגומנטים הרצוי (שם התוכנית וה-*IP*). לאחר מכן, הוא פותח קובץ פלט שיכיל את תוצאות התוכנית. עבור כל גודל של הודעה (מעבר בלולאה), נפתח *socket* ייעודי שמתחבר אל ה-*socket* של השרת (על ידי ה-*IP* שהועבר כארגומנט והפורט הייעודי) ומתחיל בשליחת ההודעות. הוא מבצע תחילה שליחה של הודעות החימום וכשמסיים (מזוהה על ידי קבלת הודעת *Reply* מהשרת), פותח *timer* ומתחיל בשליחת ההודעות האמיתיות. לאחר שהוא מסיים לשלוח את כל ההודעות, הוא ממתיך לקבלת הודעת ה-*reply* של השרת. כאשר הודעת ה-*reply* מתקבלת, הלקוח סוגר את ה-*socket* ואת ה-*timer* ומחשב את ה-*throughput* בהתאם למספר ההודעות שנשלחו ולזמן שעבר. את התוצאות הללו הוא מכניס לקובץ הפלט ואז ממשיך בלולאה לגודל ההודעה הבא. בסוף התהליך, נסגר קובץ הפלט והתוכנית מסתיימת.

מספר הודעות החימום שנשלחו ומספר ההודעות האמיתיות שנשלחו נקבעו באמצעות ניסוי ותעיה. התחלנו ממספר מועט של הודעות מכל סוג. בכל פעם, קיבענו את מספר הודעות החימום והגדלנו את מספר ההודעות האמיתיות עד שהדבר הפסיק לשפר את הביצועים.

ביצענו תהליך דומה עבור מספר הולך וגובר של הודעות חימום ועצרנו כאשר הביצועים היו הטובים ביותר. אלו הם הביצועים המוצגים בתוצאות. הביצועים נבחנו במונחי תפוקה ממוצעת של התוכנית (על פני כל גדלי ההודעות), תפוקה מירבית של התוכנית (מבין כל גדלי ההודעות) והגרף שנתקבל.

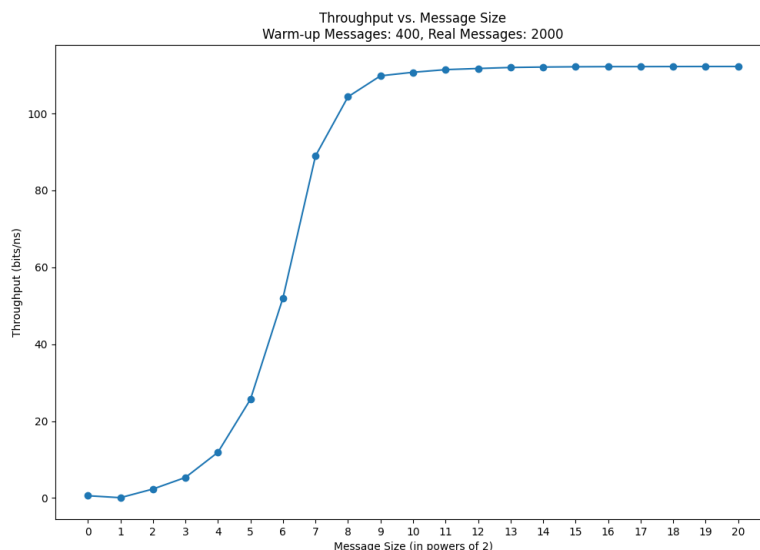
הרצנו את *server.cpp* באמצעות השרת הרביעי שסופק (*mlx-stud-04*) ואת *client.cpp* באמצעות השרת השלישי שסופק (*mlx-stud-03*).

טבלת התוצאות שהתקבלה:

Table With: Warm-up Messages = 400, Real Messages = 2000

messageSize[Bytes]	Throughput	ThroughputUnits
1	0.597971	MB/S(MegaBytesPerSecond)
2	0.0883732	MB/S(MegaBytesPerSecond)
4	2.33612	MB/S(MegaBytesPerSecond)
8	5.34591	MB/S(MegaBytesPerSecond)
16	11.8962	MB/S(MegaBytesPerSecond)
32	25.6718	MB/S(MegaBytesPerSecond)
64	52.0429	MB/S(MegaBytesPerSecond)
128	88.963	MB/S(MegaBytesPerSecond)
256	104.346	MB/S(MegaBytesPerSecond)
512	109.801	MB/S(MegaBytesPerSecond)
1024	110.723	MB/S(MegaBytesPerSecond)
2048	111.403	MB/S(MegaBytesPerSecond)
4096	111.708	MB/S(MegaBytesPerSecond)
8192	111.967	MB/S(MegaBytesPerSecond)
16384	112.084	MB/S(MegaBytesPerSecond)
32768	112.152	MB/S(MegaBytesPerSecond)
65536	112.191	MB/S(MegaBytesPerSecond)
131072	112.195	MB/S(MegaBytesPerSecond)
262144	112.215	MB/S(MegaBytesPerSecond)
524288	112.226	MB/S(MegaBytesPerSecond)
1048576	112.228	MB/S(MegaBytesPerSecond)

הגרף שהתקבל:



הסבר לתוצאות:

כמצופה, ככל שגודל ההודעה גדל כך גם ה-*throughput* גדלה. זאת למשל כי התקורה של כל הודעה נהפכת לזניחה ביחס לגודל ההודעה כולו (למשל ב-*header* של הפקטות). עם זאת, בשלב מסוים, ההשפעה של הגדלת ההודעה מתחילה לקטון באופן משמעותי ולהוביל לגידול מזערי ב-*throughput*. הדבר מתרחש כנראה כי גודל ההודעה העצום מוביל לעומס רב על הרשת וכתוצאה מכך הביצועים נפגעים. הדבר מתואר בגרף החל מגודל הודעת של 2^7 bits שכל גידול בגודל ההודעה יותר ממנו מוביל לעלייה זניחה יחסית ב-*throughput* והוא מתקבע סביב ערך של כ- 112.228 MB/sec .

הדבר ניתן להסבר למשל באופן הבא - חלק גדול מבזבז התפוקה ברשת עבור גדלי הודעות קטנים במיוחד נבע מכך שה-*header* של הפקטה היה ביחס גודל דומה לזה של ההודעה בפועל ולכן בוזבז משאבים על העברתו. הגדלת גודל הפקטה גרם לכך שחלק ה-*header* יהיה יותר ויותר זניח ביחס לגודל ההודעה עצמה ולכן התפוקה השתפרה. עם זאת, עבור גודל הודעה רב מספיק, ה-*header* כבר זניח וגידול נוסף בגודל הפקטה לא תורם באותו אופן.

הערה:

התוצאות לעיל הן התוצאות הטובות ביותר מבחינת ה-*throughput* אך תוצאות טובות כמעט באותה מידה (הגיעו ל-*throughput* מירבי של כ- 112.224 MB/sec) התקבלו גם כאשר נשלח מספר מועט יותר של הודעות (ולכן הזמן שנדרש היה משמעותית פחות יותר). כך למשל, עבור שליחה של 250 הודעות חימום ו-1000 הודעות אמיתיות, התקבלו התוצאות:

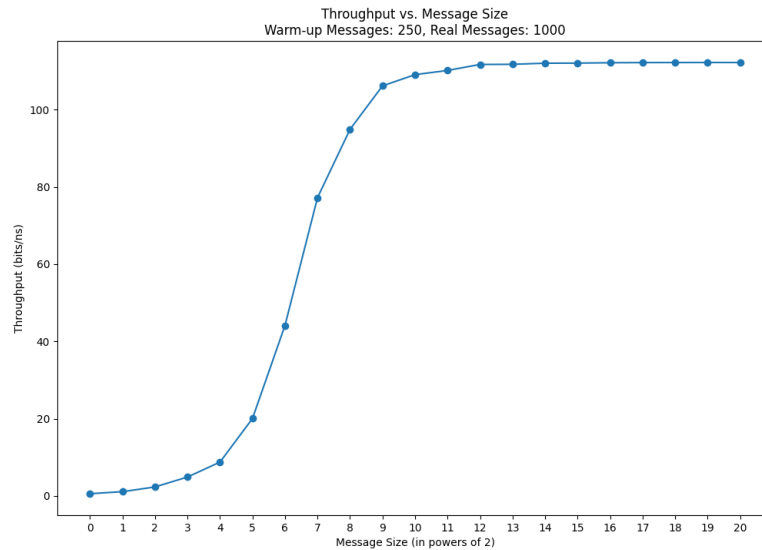


Table With: Warm-up Messages = 250, Real Messages = 1000

messageSize[Bytes]	Throughput	ThroughputUnits
1	0.591697	MB/S(MegaBytesPerSecond)
2	1.11135	MB/S(MegaBytesPerSecond)
4	2.32832	MB/S(MegaBytesPerSecond)
8	4.89992	MB/S(MegaBytesPerSecond)
16	8.76028	MB/S(MegaBytesPerSecond)
32	20.0433	MB/S(MegaBytesPerSecond)
64	44.135	MB/S(MegaBytesPerSecond)
128	77.208	MB/S(MegaBytesPerSecond)
256	94.9054	MB/S(MegaBytesPerSecond)
512	106.17	MB/S(MegaBytesPerSecond)
1024	109.072	MB/S(MegaBytesPerSecond)
2048	110.166	MB/S(MegaBytesPerSecond)
4096	111.717	MB/S(MegaBytesPerSecond)
8192	111.756	MB/S(MegaBytesPerSecond)
16384	112.029	MB/S(MegaBytesPerSecond)
32768	112.066	MB/S(MegaBytesPerSecond)
65536	112.158	MB/S(MegaBytesPerSecond)
131072	112.204	MB/S(MegaBytesPerSecond)
262144	112.213	MB/S(MegaBytesPerSecond)
524288	112.225	MB/S(MegaBytesPerSecond)
1048576	112.221	MB/S(MegaBytesPerSecond)

ואכן ניתן לראות כי התפוקה המירבית (112.225 MB/sec) היא כמעט זהה לזו עבור 400 הודעות חימום ו-2000 הודעות אמיתיות (112.228 MB/sec).