סדנה ברשתות תקשורת (67613) - תרגיל 2

נטלי יוסופוב, 319085726 | מור חן, 206407314

5 ביוני 2023

תיאור המימוש

בתרגיל ישנם שני קבצים - הקובץ הראשון הוא bw_template.*cpp* והקובץ השני הוא קובץ *utils.h*. הקובץ הראשון הוא הקובץ שדרכו נריץ את השרת והלקוח והו=קובץ השני מהווה ספריית עזר שהקובץ הראשון מייבא.
פירוט אודות הקבצים השונים:

1. קובץ *utils.h* - הקובץ מכיל קבועים וספריות המשומשים על ידי שני הקובץ האחר. לדוגמה, הוא מכיל מערך של גדלי ההודעות השונים, את מספר הודעות החימום ואת גודל הבאפר שנקצה להודעות.

2. קובץ *bw_template.cpp* - הקובץ מריץ את התוכנית כולה. כדי להריצו מצד השרת, אין לספק ארגומטנים לתוכנית וכדי להריצו מצד הלקוח, יש לספק את כתובת ה-IP של השרת (או את שם ה-$host$ של השרת). הרצת הקובץ מייצרת את התקשורת בין שני הצדדים - יוצרת QP (וממלאת אותו ב-post_receieves), יוצרת CQ מתאים ומקשרת בין השרת ללקוח. לאחר מכן, היא פועלת באופן שונה עבור השרת והלקוח:

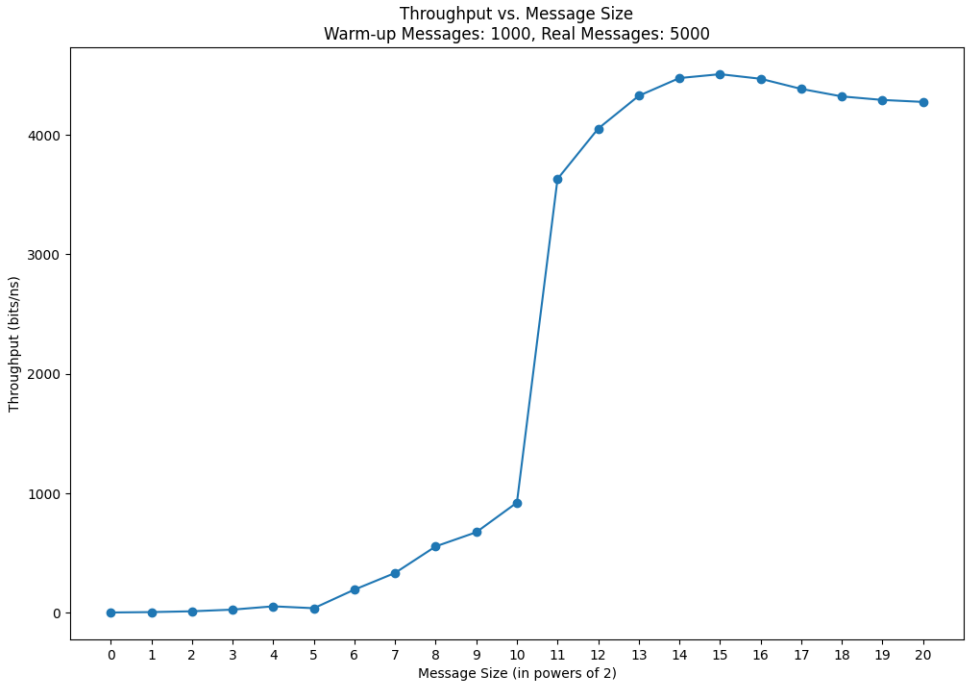
   (א) מצד הלקוח, עבור כל גודל של הודעה (מעבר בלולאה), השרת מתחיל בשליחת הודעות אל השרת (ביצוע post_sends). שליחת ההודעות מבוצעת ב-$Batches$ של עומק ה-send queue (שמסומן בתוכנית על ידי tx_depth) - בכל פעם שבוצעו tx_depth בקשות לשליחה, הלקוח ממתין שהבקשות יסוימו בטרם הוא ממשיך בשליחת הודעות נוספות (המתנה מבוצעת על ידי שימוש ב-pp_wait_completions). כאשר הלקוח מסיים את מספר ההודעות שהוא רוצה לשלוח, הוא מחכה לקבלת הודעה מהשרת (*pp_wait_completions* על בקשת post_receieve שכבר מצויה בתור שכן הוא אותחל עם *post_receieves*). שליחת ההודעות מצד השרת מבוצעת בשני חלקים. תחילה, הוא שולח warmup_messages הודעות חימום. לאחר מכן, הוא פותח טיימר ומתחיל בשליחת real_messages הודעות אמיתיות. לאחר שהוא מסיים לשלוח את כל ההודעות ומקבל את הודעת השרת (על ההודעות האמיתיות), הלקוח סוגר את 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.

   (ב) בדומה לתרגיל 1, השרת תחילה בודק מהו ה-IP של המחשב עליו הוא רץ באמצעות שם ה-$host$ שלו ומדפיס אותו אל המסך (כדי שנוכל להעבירו כארגומנט ללקוח). לאחר מכן, עבור כל גודל של הודעה (מעבר בלולאה), השרת ממתין

לקבלת ההודעות מהלקוח. בדומה ללקוח, הוא ממתין לקבלת הודעות ב־ $Batches$ של עומק ה־receive queue (שמסומן בתוכנית על ידי ctx->rx_depth) ־ בכל פעם הוא ממתין ל־$\min\{rx\_depth, messages\,left\,to\,receive\}$ בקשות לקבלה על ידי שימוש ב־$pp\_wait\_completions$. בתום קבלת כל ההודעות, הוא שולח הודעה לשרת (post_send) וממתין שזו תישלח בהצלחה (שוב, על ידי $pp\_wait\_completions$). גם השרת מבצע תהליך זה בשני חלקים ־ תחילה, עבור warmup_messages הודעות חימום ולאחר מכן, עבור real_messages הודעות אמיתיות.

מספר הודעות החימום שנשלחו ומספר ההודעות האמיתיות שנשלחו נקבעו באמצעות ניסוי ותעייה. 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:



הגרף שהתקבל:

Table With: Warm-up Messages = 1000, Real Messages = 5000

messageSize[Bytes]	Throughput	ThroughputUnits
1	3.469813	BytesPerMicrosecond
2	6.844627	BytesPerMicrosecond
4	13.69863	BytesPerMicrosecond
8	27.605245	BytesPerMicrosecond
16	55.248619	BytesPerMicrosecond
32	39.408867	BytesPerMicrosecond
64	194.884287	BytesPerMicrosecond
128	334.378265	BytesPerMicrosecond
256	556.521739	BytesPerMicrosecond
512	675.283566	BytesPerMicrosecond
1024	923.021453	BytesPerMicrosecond
2048	3631.205674	BytesPerMicrosecond
4096	4052.235853	BytesPerMicrosecond
8192	4327.065286	BytesPerMicrosecond
16384	4475.524476	BytesPerMicrosecond
32768	4507.910304	BytesPerMicrosecond
65536	4469.54197	BytesPerMicrosecond
131072	4384.764122	BytesPerMicrosecond
262144	4321.800574	BytesPerMicrosecond
524288	4292.34107	BytesPerMicrosecond
1048576	4275.880967	BytesPerMicrosecond

<u>הסבר לתוצאות:</u>

כמצופה, ככל שגודל ההודעה גדל כך גם ה־$throughput$ גדלה. זאת למשל כי התקורה של כל הודעה נהפכת לזניחה ביחס לגוד להההודעה 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 $2^{14}bits$ שכל גידול בגודל ההודעה יותר ממנו מוביל לעלייה זניחה יחסית ב־$throughput$ והוא מתקבע סביב ערך של כ־$4400\,bytes/microsecond$.

הדבר ניתן להסבר למשל באופן הבא ־ חלק גדול מבזבוז התפוקה ברשת עבור גדלי הודעות קטנים במיוחד נבע מכך שה־$header$ של הפקטה היה ביחס גודל דומה לזה של ההודעה בפועל ולכן בוזבזו 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.

ראוי לציין כי ישנה עלייה דרסטית בגרף שככל הנראה נובעת מאי־יציבות של הרשת.

כמצופה, התוצאות משמעותית טובות יותר (בערך פי 30) מאשר התוצאות שהונבו בתרגיל 1 עבור שקעי TCP.