עבודת הגשה 1 עיבוד מקבילי

מגישים: אלעד אלמקיאס 208195370
ניצן עוז 206433609

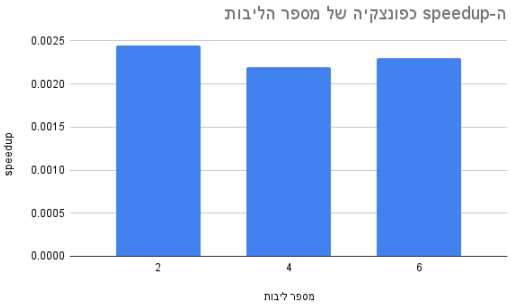
הוראות הפעלה: את הקובץ של העיבוד המקבילי נריץ כרגיל על ידי `mpirun -np N task1` כאשר N זהו מספר הליבות.

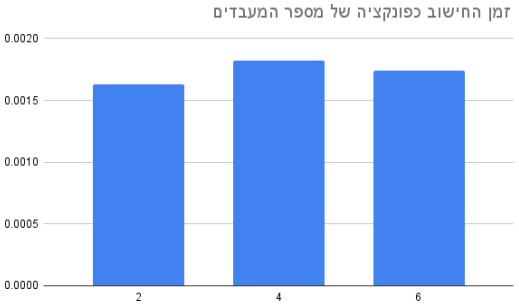
הערות:
לגבי בדיקת הזמנים בדקנו את הזמנים רק עבור הביצוע של חישוב הprefix sum ולא עבור משהו אחר, כלומר בחישוב הזמנים התעלמנו מה-init (כי הוא לא תלוי בגודל המערך) והתעלמנו מזמן היצירה של המערך.
ביצענו את חישוב הזמנים על מחשב עם 6 ליבות, ועבור המכונה הוירטואלית במציאת צילומי המסך בjumpshot ו-scalasca ביצענו על מכונה עם 2 ליבות (בעקבות קשיים טכניים) אך עדיין ניתן להסיק את המסקנות לגבי שימוש בחישוב מקבילי.
בתיקייה extra יש מספר קבצים נוספים מהמכונה הוירטואלית שם יש את כל קבצי ההרצה של scalasca ו-jumpshot
באלגוריתם השתמשנו בExscan כדי לבצע את ה-shift

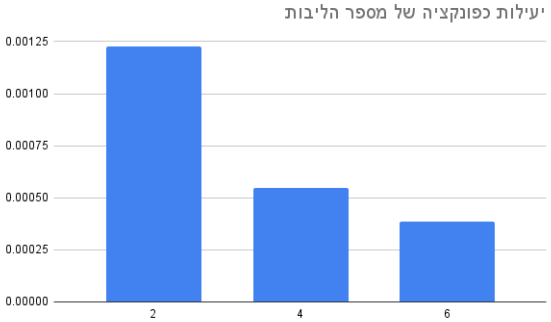
האלגוריתם:
האלגוריתם הסיראלי הוא כזה, לאחר יצירה של שני מערכים מתבצעת סכימה של האיבר הנוכחי וסכום האיברים עד עכשיו עד שעברנו על כל המערך.
האלגוריתם המקבילי עובד באופן הבא, לאחר יצירת המערך (כאשר רק process 0 מבצע את חישוב המערך), process 0 מפזר את המערך לכל התהליכים כולל לעצמו, וכל תהליך מחשב את הprefix sum החלקי לתת המערך שלו,
אחר כך כל תהליך משדר לתהליך הבא את כמות ההיסט שהוא צריך לבצע
ואז כל תהליך מגדיל את איברי המערך בהיסט המתאים ושולח בחזרה לתהליך 0 שמאחד את כל תתי המערכים למערך אחד.
יש גם דוגמא בסוף הקובץ.

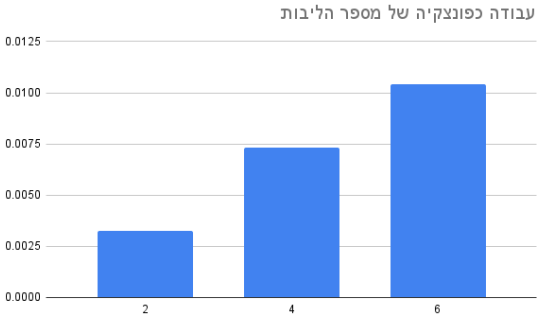
זמני הרצה:
עבור 600,000 ערכים קיבלנו את התוצאות הבאות:

מספר ליבות	זמן	speedup	work	efficiency
1 (סיראלי)	0.000004			
2	0.001633	0.002449479486	0.003266	0.001224739743
4	0.001825	0.002191780822	0.0073	0.0005479452055
6	0.001737	0.002302820956	0.010422	0.0003838034926



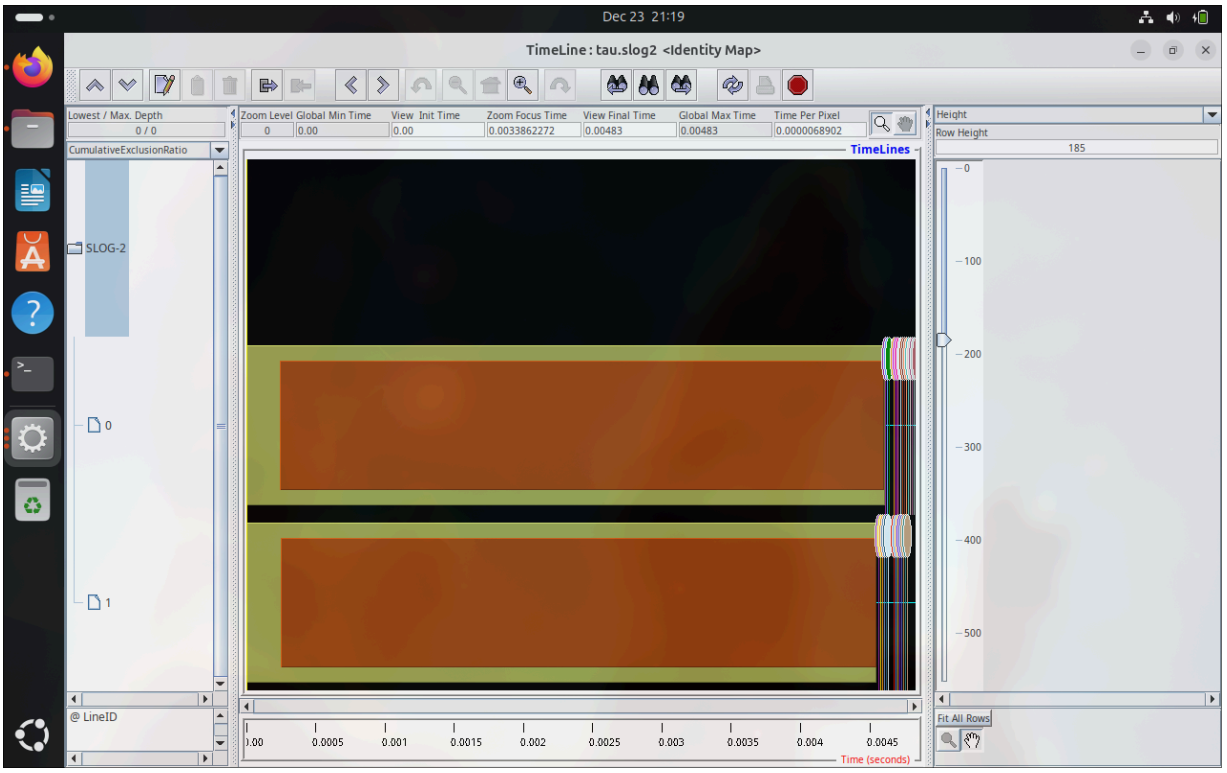




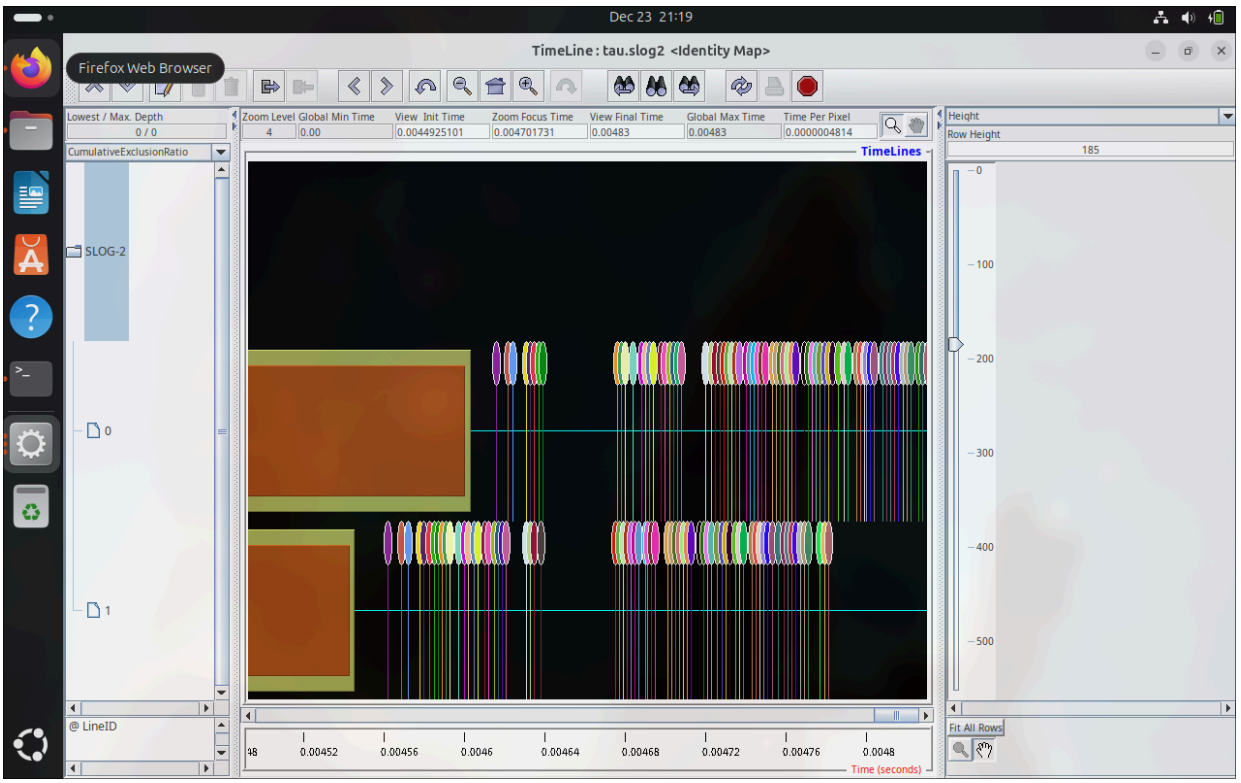


ניתן לראות כי עבור האלגוריתם שלנו היעילות יורדת כשמספר הליבות עולה, זאת מכיוון שאנחנו מבזבזים הרבה מאוד זמן בשליחת המערך עצמו, פתרון יותר טוב היה להשתמש בזכרון משותף כאשר כל התהליכים כותבים לאותו מקום בזכרון.

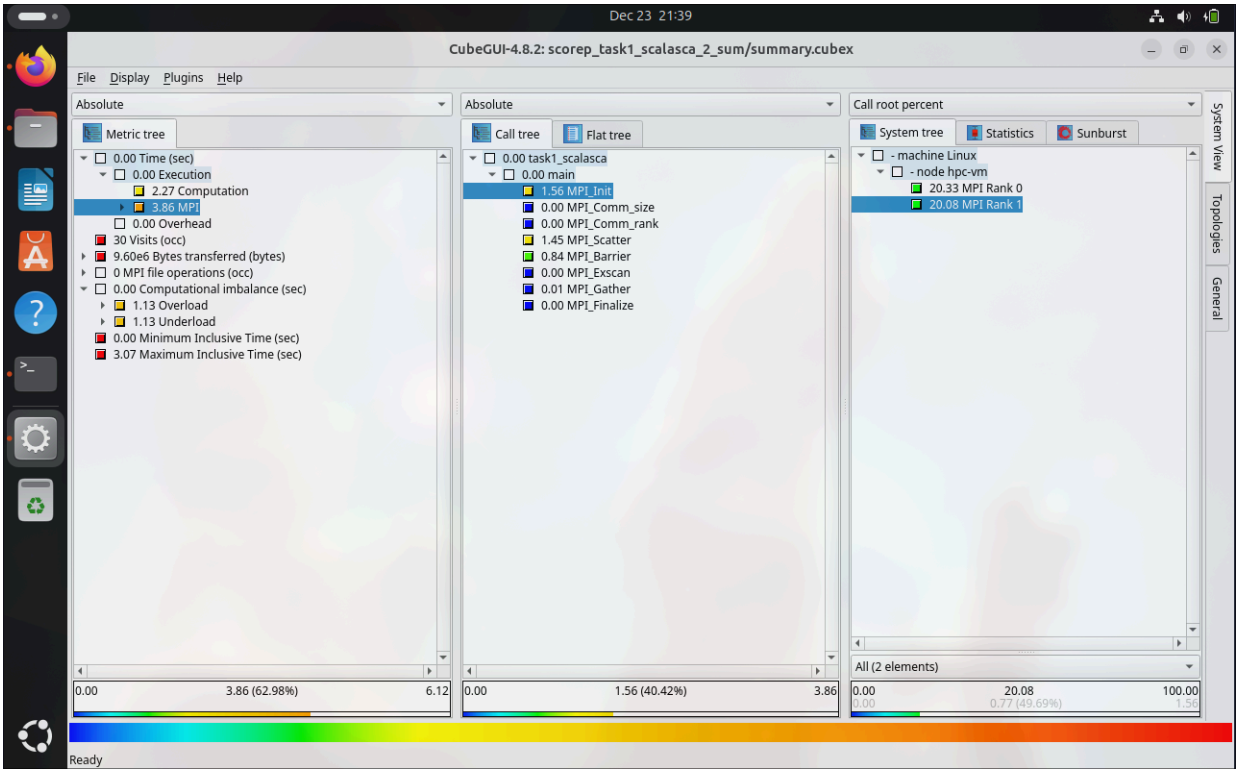
בעבור הjumpshot וscalasca קיבלנו:



ב



בjumpsho ניתן לראות זמן רב שמבוזבז על הinit ורק לאחר מכן יש חישוב בפועל.



ניתן להסיק מהתמונות כי רוב הזמן של הריצה מתבזבז על הinit לכן היעילות עוד יותר נמוכה ממה שראינו.
כלומר כל עוד לא נשתמש בזכרון משותף עדיף לנו לא לבצע חישוב מקבילי על התוכנית הזו.

דוגמא:
נניח יש 4 מעבדים ונוצר המערך 1,2,3,4,5,6,7,8,9,10,11,12
אז תהליך 0 יחשב 1,3,6
תהליך 1: 4,9,15
תהליך 2: 7,15,24
תהליך 3: 10,21,33

לכן תהליך 0 יבצע שיפט של 0 (תמיד) תהליך 1 יבצע שיפט של 6, תהליך 2 של 15+6=21 ותהליך 3 יבצע שיפט של 24+15+6=45 ואז נקבל
תהליך 0: 1,3,6
תהליך 1: 10,15,21
תהליך 2: 28,36,45
תהליך 3: 55,66,78
ולאחר האיחוד נקבל את המערך הסופי:
1,3,6,10,15,21,28,36,45,55,66,78