

פרויקט בקורס מחשוב מקבילי ומבוזר קיץ 2021

במסמך זה אני אסקור את מבנה התוכנית והאופן שבו בוצע המיקבול, האלגוריתם ומבנה הנתונים ומימוש המיקבול בתהליכים.

מבנה התוכנית והאופן שבו בוצע המיקבול:

התוכנית יוצרת כמות תהליכים כפי שהוגדר לה בשורת ההרצה. תהליך מספר 0 קורא את הנתונים מקובץ טקסט ושולח נתונים רלוונטיים לשאר התהליכים. החלוקה של העבודה בין התהליכים נעשית באופן סטטי, כאשר במידה וקיימת שארית בחלוקה היא תחולק באופן שווה בין מספר התהליכים הראשונים שמספרם קטן משארית החלוקה. כדי לחסוך בזמני תקשורת וריבוי של MPI_Send ו-MPI_Recv, החלטתי להשתמש בפונקציות MPI_Pack עבור תהליך מספר 0 ו-MPI_Unpack עבור שאר התהליכים. לאחר שליחת וקבלת הנתונים תהליך מספר 0 פונה לכרטיס הגרפי ומעביר לו בין השאר את העבודות שלו (מחרוזות מסוג שני) כדי שיבצע חישוב של כל התוצאות. לאחר קבלת התוצאות, התהליך פותח תראדים כדי למצוא את התוצאה המקסימלית, ההיסט ורמת המוטציה עבור כל מחרוזת. בהמשך תהליך 0 אוסף את התוצאות שהתקבלו משאר התהליכים ומבצע הדפסה. במקביל לפניה של תהליך מספר 0 לכרטיס הגרפי, שאר התהליכים ימצאו את התוצאה המקסימלית, ההיסט ורמת המוטציה עבור כל אחת מהעבודות (מחרוזות) שהוקצו להם ע"י פתיחת תראדים שיבצעו את הבדיקה. לאחר מיכן התהליכים שולחים לתהליך 0 את התוצאות.

אלגוריתם ומבני נתונים:

כדי להימנע מחישובים רבים ומיותרים, רשמתי תוכנית נפרדת שיוצרת מטריצה בגודל 26 על 26 של מספרים שלמים בטווח של 0 עד 3 אשר מייצגים את הסימנים. את המטריצה שקיבלתי מתוכנית זו העתקתי לתוכנית הראשית בתור מטריצה של קבועים. באופן זה ע"י מציאת אות במחרוזת הראשונה ואות במחרוזת השנייה ניתן לדעת את הסימן מבלי לבצע חישובים מיותרים. כמו כן למען הסדר הטוב כל תוצאה נשמרה במבנה שכולל שלוש שדות: score, offset, mutant_lvl.

האלגוריתם רץ על המחרוזת הראשונה שבקלט התוכנית למול מחרוזת מסוג שני. בלולאה החיצונית נקבעת רמת המוטציה, כאשר הראשונה היא 1. בלולאה פנימית ברמה ראשונה נקבע ההיסט כאשר הראשון הוא 0. בלולאה פנימית ברמה שניה מתבצעת השוואה של כל זוג תווים והוספה של ערכם לתוצאה הנוכחית. בסיום ההרצה של הלולאה הפנימית ברמה שניה, נקבל את התוצאה עבור רמת מוטציה והיסט מסוים. מתבצעת בדיקה האם התוצאה שהתקבלה גדולה יותר מהמקסימלית הנוכחית, במידה וכן – התוצאה המקסימלית הנוכחית מתעדכנת וגם רמת המוטציה וההיסט שמניבים את התוצאה המקסימלית.

מימוש המיקבול:

- **שאר התהליכים:** לאלגוריתם שתואר לעיל הוספתי לולאה חיצונית נוספת בה נפתחים תראדים שיעבדו במקביל, כל תראד על מחרוזת שונה בהתאם לאינדקס שמוקצה לו מהמערכת. באופן זה העבודה תחולק ותתבצע ע"י מספר תראדים במקום תראד אחד בודד כפי שקורה בגרסה הסדרתית. קיים "תשלום" בזמן של פתיחת תראדים פעם אחת בלבד.
- **תהליך 0:** פונה תחילה לכרטיס הגרפי. עבור כל מחרוזת מסוג שני נפתח grid בטופולוגיה דו-ממדית כאשר ציר x מייצג את מספר ה- offset וציר y מייצג את ה- mutant_lvl + 1. בכל בלוק מחושבת תוצאה של offset ו- mutant_lvl בהתאם לזיהוי שלו ב- grid. מספר התראדים בכל בלוק הוא בהתאם למספר השוואות התווים שיש לבצע. במידה ומספר השוואות שיש לבצע הוא מעל 1024 (מספר מקסימלי של תראדים בתוך בלוק), תראדים מסוימים יבצעו חישוב או מספר חישובים נוסף. כאשר הכרטיס הגרפי מסיים לבצע את החישובים, הוא שולח את התוצאות למעבד. מופעלת פונקציה שרצה באופן מקבילי על התוצאות ושומרת עבור כל מחרוזת את התוצאה המקסימלית, ההיסט ורמת המוטציה המתאימה לה.