

תיעוד הסבר אלגוריתם ומקבול התוכנית

מגיש : דמיטרי אינקה 336228317

(1) המקבול

בתוכנית יש תהליך master שמספרו הוא 0. התהליך קולט את הנתונים מקובץ. התהליך מחלק את העבודה באופן סטטי בינו לבין שאר התהליכים. לפני שתהליך ה-master שולח את הנתונים לשאר התהליכים הוא אורז אותם יחדיו. באופן זה אני חוסך שליחות מיותרות. לבסוף שאר התהליכים שולחים לתהליך מספר 0 את התוצאות שהם חישובו והוא מדפיס.

CUDA – עבור כל seq2 שנשלח יפתח GRID דו-ממד בו לכל בלוק בציר x האינדקס מייצג את ה- offset ובציר y האינדקס + 1 מייצג את ה-k. כל בלוק יבצע חישוב על תוצאה אחת וירשום אותה למערך כל התוצאות של כל seq2. לבסוף הוא שולח את התוצאות ל-CPU. בעזרת OPENMP מוצאים את התוצאה הכי טובה עבור כל seq2. מכיוון שיש תלות בין החישובים למציאת התוצאה הכי טובה בחרתי לבצע אותם בשימוש ב- OPENMP ולא CUDA. כך זה מתבצע מהר יותר.

OPENMP – מריץ את האלגוריתם שיתואר בהמשך כך שכל טראד מחשב את התוצאה הטובה ביותר עבור מחרוזת מסוימת.

בשלב זה רשמתי קוד עבור שני המימושים הבאים :

- ❖ המימוש הראשון (יעיל להרצה על שני מחשבים) : תהליך מספר 0 מבצע 50% מהעבודה על CUDA ו- 50% על OPENMP. במקביל אליו תהליך מספר 1 מבצע אותו הדבר. כל תהליך הוא בעצם מחשב ולכן יש לו GPU ו-CPU משל עצמו. לכן אין תחרות על המשאבים.
- ❖ המימוש השני (יעיל להרצה על מחשב אחד) : תהליך מספר 0 מריץ CUDA. במקביל אליו תהליך מספר 1 מריץ OPENMP. באופן כזה אין תחרות על המשאבים.

מכיוון שההרצה על 2 מחשבים לפחות מאז פרסום המטלה לא עובדת ב-VLAB ובמודל קיימת הודעה בה נכתב שיש להריץ על מחשב 1 - **בחרתי להגיש את המימוש השני (יעיל להרצה על מחשב אחד)**. מימוש זה עובד מהר יותר על מחשב אחד מאשר המימוש הראשון (בדקתי), מכיוון שבמימוש הראשון מתקיימת תחרות על המשאבים (שני התהליכים פונים ל-GPU למרות שיש GPU אחד בלבד וכן אפשרות ליצירת כמות טראדים גדולה יותר מליבות ה-CPU).

(2) מבנה הנתונים של האלגוריתם

מבנה הנתונים שבחרתי להשתמש בו כדי ליעל את עבודת האלגוריתם הוא מטריצה בגודל של 26 על 26. בכל תא מופיע המשקל המתאים בהתאם לסימנים (\$, %, #, "). הרעיון הוא שכל אינדקס של שורה ועמודה במטריצה מייצג אות באנגלית. שילוב של שני האינדקסים היא תוצאת ההשוואה. באופן זה חסכתי חישובים רבים שאינם רלוונטיים.

(3) האלגוריתם

1. הרץ לולאה ראשונה שמתחילה ב- $k = 1$ עד אורך מחרוזת seq2 כולל. k מייצג את גודל המוטציה).
2. הרץ לולאה שנייה (בתוך לולאה ראשונה) שמתחילה ב- $n = 0$ עד אורך מחרוזת seq1 פחות אורך מחרוזת seq2. n מייצג את ה- offset).
3. הרץ לולאה שלישית (בתוך לולאה שנייה) שמתחילה ב- $i = 0$ עד אורך מחרוזת seq2. אם i קטן מ- k אז יש להוסיף לתוצאה נוכחית את הערך המתאים לפי מבנה הנתונים בשילוב עם המשקולות. אחרת יש לדלג על אות אחת ב- seq1 כדי לא להשוות למקף, אלא לאות שאחריו. i מייצג את מיקום האותיות במחרוזת seq2).
4. בסוף ההרצה של הלולאה השנייה מקבלים תוצאה של k ו- n מסוימים. בודקים אם התוצאה שיצאה גדולה או שווה לתוצאה הכי גבוהה עד כה. אם כן, מעדכנים את התוצאה הכי גבוהה עד כה כתוצאה שהתקבלה. כמו כן מעדכנים את ה- n ו- k בהתאם.