

README

פרטי מגשים :

ים חורין ת.ז. 318853256

הסבר על תוכנית

זרימת התוכנית :

מאתחל את סביבת MPI.

תהליך השורש קורא את גרף הקלט ומשדר נתונים חיוניים לתהליכים אחרים.

חישוב הנתיב הקצר ביותר מתחלק בין תהליכים.

תהליך השורש אוסף תוצאות מתהליכי עובד, מחליט על הקודקוד הבא ומשדר אותו.

תהליכי עובדים מחשבים את המרחקים המינימליים המקומיים שלהם ומדווחים לתהליך השורש.

האלגוריתם ממשיך עד שנמצאים כל הנתיבים הקצרים ביותר או הגעה ליעד הרצוי.

כל התהליכים מסיימים את סביבת MPI והתוכנית מסתיימת

מבנה התוכנית :

סוגי נתונים ומבנים :

enum tags : מונה תגים שונים להעברת הודעות, כגון WORK, STOP, FIND_MIN_VERTEX וכו'.

struct קודקוד : מייצג קודקוד עם המזהה שלו והמרחק שלו מקודקוד המקור.

enum goal : מציינ אם המטרה היא למצוא את המרחק ליעד בודד או לכל הקודקודים.

משתנים גלובליים :

NV : מספר קודקודים.

done : מערך המציין אילו קודקודים גמורים.

קצוות : מאחסן את הגרף כמטריצת סמיכות.

מרחק : מערך המאחסן את המרחק הקצר ביותר של כל קודקוד מהמקור.

יעד : מציינ אם התוכנית צריכה למצוא מרחק לקודקוד בודד או לכל הקודקודים.

אתחול התוכנית :

הפונקציה `init()` מגדירה את התנאים ההתחלתיים על סמך ארגומנטים של שורת הפקודה וקוראת את הגרף.

`readGraph()`: קורא את משקלי הגרף מהקלט הסטנדרטי.

מקביליות עם MPI:

לאחר אתחול MPI, התוכנית קובעת את הדרגה שלה. אם הוא מדורג 0, הוא פועל כמאסטר, אחרת הוא פועל.

מאסטר שולח חלקים מהגרף לעובדים ומתאם את תהליך מציאת הנתיבים הקצרים ביותר. עובדים מחשבים את הנתיבים הקצרים ביותר המקומיים ומעבירים תוצאות בחזרה למאסטר.

פונקציות מפתח:

`find_vertex_with_minimum_distance_MASTER` ו-`find_vertex_with_minimum_distance_WORKER`: מאסטר ופונקציות פועל כדי למצוא את הקודקוד עם המרחק המינימלי שעדיין לא סוכם.

`update_distances_MASTER` ו-`update_distances_WORKER`: עדכן את הנתיבים הקצרים ביותר לאחר סיום קודקוד חדש.

`printResult()`: הדפס את התוצאה של חישוב הנתיב הקצר ביותר.

`printGraph()`: לצורך איתור באגים, מדפיס את מטריצת הסמיכות של הגרף.

סוגי נתונים מותאמים אישית של MPI:

התוכנית מגדירה MPI datatype מותאם אישית `mpi_vertex_type` כדי לשלוח ולקבל את מבנה הקודקוד.

ביצוע:

התוכנית משתמשת בגרסה הרציפה של האלגוריתם של דיקסטר, אך מפיצה את העבודה של מציאת הנתיבים הקצרים ביותר לקודקודים שונים על פני תהליכים.

אם צוין רק יעד אחד, התוכנית נעצרת ברגע שנמצא הנתיב הקצר ביותר ליעד זה. אחרת, הוא מחשב את הנתיבים הקצרים ביותר לכל הקודקודים.

תוצאות, כולל נתיבים ומרחקים, מודפסות בסוף.

שימוש ב-OPENMP

1. Parallelizing the Loop for Distance Update:

לאחר מציאת הקודקוד עם המרחק המינימלי שלא הוגדר סופית, התוכנית מעדכנת את המרחקים OpenMP. לולאה זו יכולה להיות מקבילה באמצעות. לקודקודים אחרים בהתבסס על קודקוד זה

2. Using Reduction for Finding Minimum Distance:

כדי למצוא את הקודקוד עם המרחק המינימלי שלא הושלם, ניתן להשתמש בפעולת Reduction

הערה חשובה:

נכון שנהוג בכל תוכנית של MPI לסיים בפקודה `MPI_Finalize()` אבל התוכנית הוציאה איזה שגיאה מוזרה שלא אני ולא גדי יודע עליה

```
1:1  
[WARNING] yaksa: 1 leaked handle pool objects  
[WARNING] yaksa: 1 leaked handle pool objects  
[WARNING] yaksa: 1 leaked handle pool objects  
[WARNING] yaksa: 1 leaked handle pool objects  
(base) ~$
```