Ido Shamir 208119354

הפרויקט ממקבל את הבעיה המתוארת באופן הבא:

בהינתן P תמונות ו-O אובייקטים שנקראים מקובץ הקלט, הפרויקט מריץ לפחות 2 תהליכים, כאשר אחד מהם master והשאר slaves.

ה-master עובד בצורה דינאמית: מחלק לכל slave מנת עבודה התחלתית (תמונה), וכל פעם ש-slave מחזיר ל-master תוצאה, הוא יקבל מנת עבודה חדשה. לאחר שה-master חילק את כל מנות העבודה, הוא ממתין שכל ה-slaves יסיימו את עבודתם ויחזירו אליו את התוצאות של מנות העבודה האחרונות שקיבלו, וכאשר כל התוצאות הנדרשות בידיו, ה-master כותב אותן לקובץ הפלט.

כל slave, לאחר שקיבל את אוסף האובייקטים לחיפוש ותמונה לחפש עליה, מבצע את העיבוד הדרוש ושולח אל ה-master את התוצאה לתמונה שלו (האם בכלל נמצאו בה אובייקטים, ואם נמצאו – איזה אובייקט, והיכן הוא נמצא).

סיבוכיות כללית לפרויקט: O(Log(P)) – עבור P תמונות יורצו לכל היותר P-1 slaves, אך מכיוון שה-slaves רצים במסגרת תהליכים מקביליים, זמן הריצה קצר יותר מזה הדרוש לריצה סריאלית.

מיקבול המשימה בוצע באופן הבא:

* MPI – כל תהליך (מלבד ה-master) מקבל תמונה P ואת אוסף האובייקטים O. כפי שצוין מעלה, ה-master מחלק את המידע בין ה-slaves ואוסף אליו תוצאות, על כן הגישה הנוחה ביותר לעשות זאת היא באמצעות תקשורת בין תהליכים (Multi-Process Interaction).

כמו כן, עיבוד על תמונה אחת אינו מושפע מעיבוד על תמונה אחרת, לכן ניהול החיפוש במסגרת תהליכים הינו פתרון נוח ופשוט למימוש.

* OMP – כל תהליך MPI מפעיל בסה"כ O threads (אחד עבור כל אובייקט באוסף האובייקטים שלו), כאשר תפקיד ה-thread הוא לחפש את האובייקט בתמונה ולהחזיר תוצאה (נמצא/לא נמצא, ואם נמצא – היכן).

חיפוש אובייקט X בתמונה אינו מושפע מחיפוש אובייקט Y באותה התמונה, לכן ניהול החיפוש באותה התמונה באמצעות threads הינו פתרון נוח ויעיל.

* CUDA – כל thread ב-OMP ניגש ל-GPU ומפעיל עליו חיפוש מהיר של האובייקט בתמונה.

השימוש ב-CUDA הכרחי מכיוון שנדרש כוח חישובי חזק כדי להתמודד עם תמונות ואובייקטים ממימדים גדולים, וגם מכיוון שה-GPU מאפשר מקבול אמיתי של חישובים דרך threads.