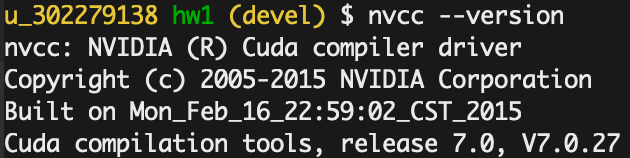


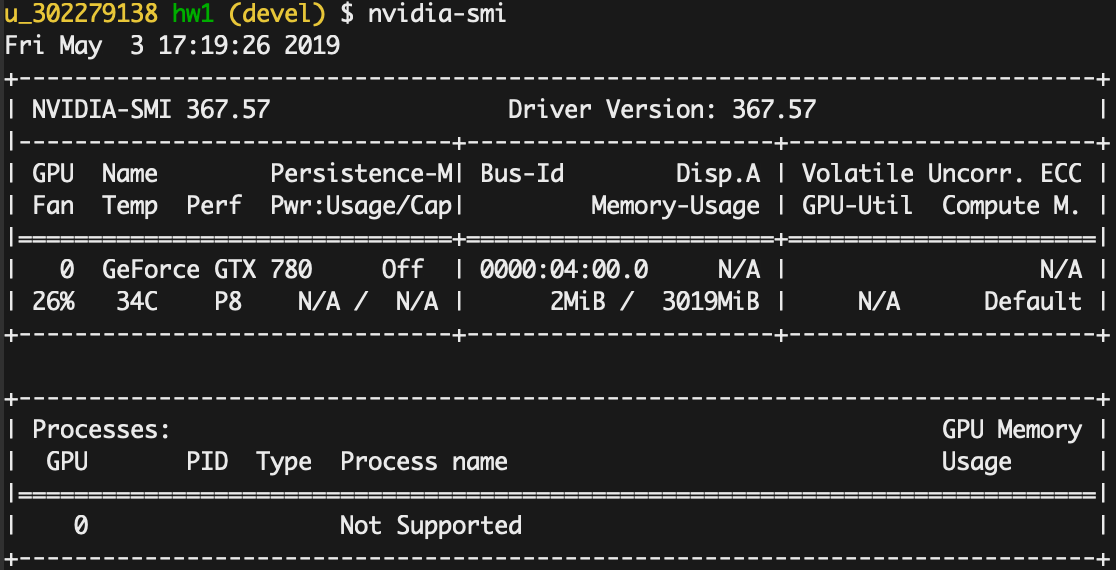
יונתן בתן 302279138

אליה בן אברהם 305104580

* 1. להלן הגרסה של cuda עליה אנחנו עובדים



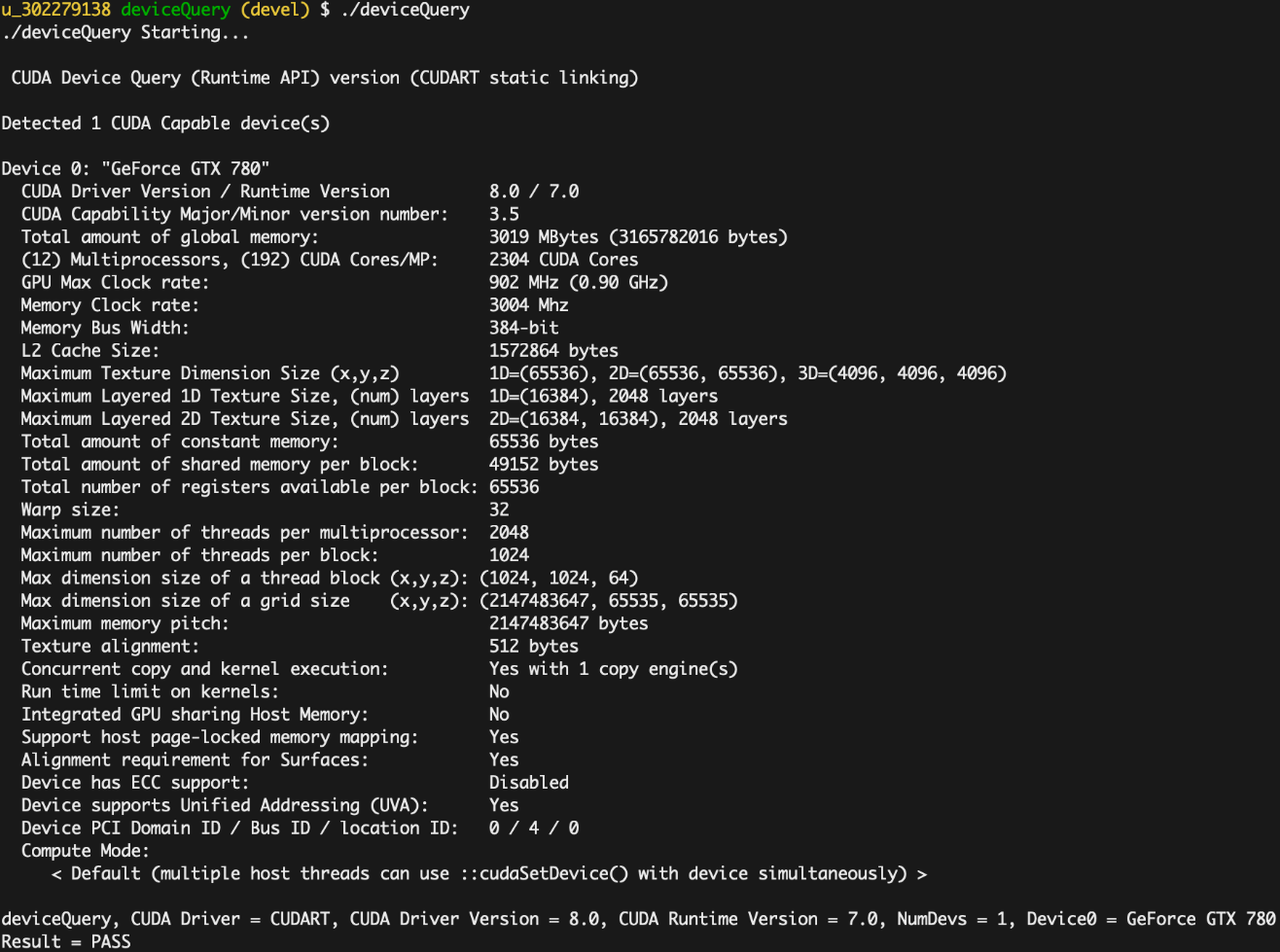
* 1. להלן פרטי ה- GPU על המכונה שהוקצתה לנו



* 1. בתמונה למטה ניתן לראות פרטים חשובים אודות ה- GPU עליו אנחנו עובדים
* מדובר ב- GeForce GTX 789 (לא פסגת הטכנולוגיה 😉 )
* מספר ה- SMs הוא 12

ועוד שלל פרטים חשובים כגון:

* גודל הזיכרון הגלובלי
* מספר החוטים ב- warp וכ"ו



1. בקוד
   1. בקוד
   2. בקוד
   3. כאשר חוט מסוים רוצה לעדכן את ההיסטוגרמה בהתאם ל-byte שבאחריותו בתמונה הוא צריך לקרוא את הערך הנוכחי לתוך רגיסטר, להוסיף לערך זה 1 ולכתוב חזרה את התוצאה בהיסטוגרמה.

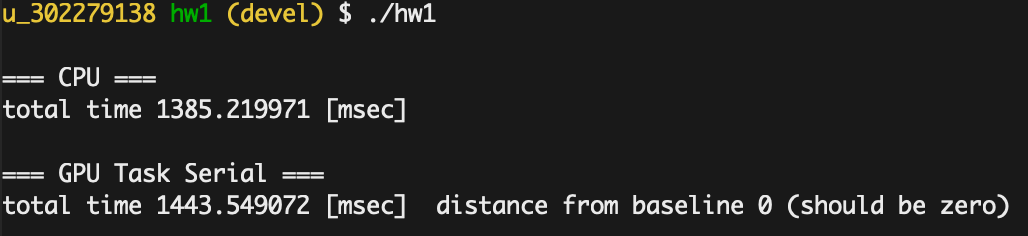
במידה ופעולת ה-increment לא תהיה אטומית סביר שיותר מחוט אחד, אשר רצה הגורל וערך ה-byte שלהם זהה, יקראו את אותו הערך הנוכחי בהיסטוגרמה בו"ז וכאשר יכתבו את התוצאה חזרה הם ידרסו את התוצאה אחד של השני ונקבל תוצאה שגויה.

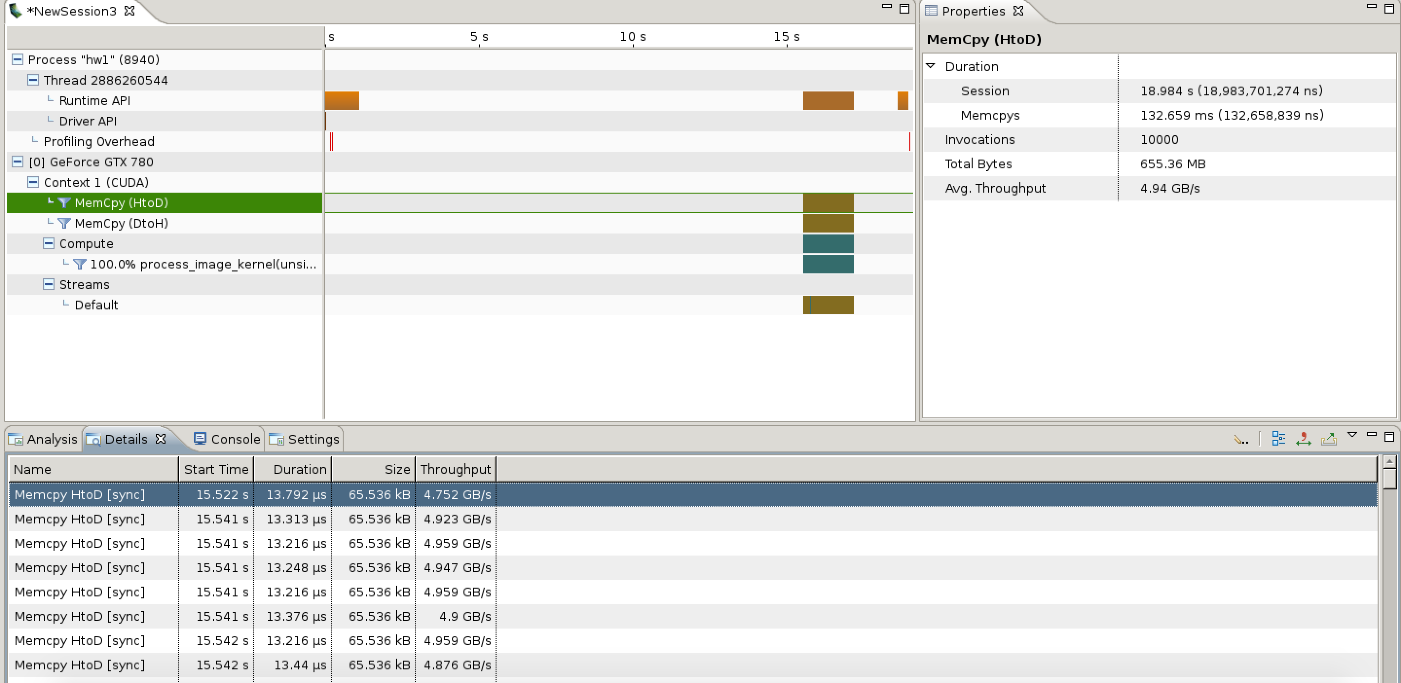
* 1. בקוד
  2. בחרנו להשתמש במספר החוטים המירבי של threadBlock, כלומר 1024 חוטים.

עבור החלוקה שעשינו כל חוט אחראי על בתים בתמונה, באופן עקרוני היינו רוצים יותר חוטים על מנת לייעל את החישוב, במצב זה נצטרך לחלק את העבודה בין threadBlocks שונים כך שכל חוט יטפל באיבר יחיד בתמונה.

כיוון שיש 10,000 תמונות אזי בשלב המקבילי (שלב 4 בתרגיל) גם ככה כל הליבות יהיו עסוקות בחישובים ולכן נקבל ביצועים דומים גם ללא פריסה של תמונה יחידה על יותר מ- threadBlock יחיד ולכן בחרנו באופציה זו כיוון שכך המימוש פשוט יותר.

* 1. כפי שניתן לראות הזמן הכולל הינו עבור 10,000 תמונות ולכן



* 1. התשובה ביחד עם סעיף 3.8
  2.  כפי שניתן לראות אורך פעולת העתקת הזיכרון שבחרנו להציג אורכת

תשובה רלוונטית לסעיף 4

* 1. בחרנו להשתמש ב- חוטים.

כל תמונה הינה בגודל וכיוון שכל threadBlock מכיל לכל היותר חוטים נצטרך סה"כ threadBlocks מלאים על מנת לקבל fine grain מקסימלי.