

עלון

אישהח"מ

עלון האיגוד הישראלי לשיטות חישוביות במכניקה

מספר 7

מרץ 2002

עורך: דן גבעולי, הפקולטה להנדסת אוירונאוטיקה וחלל, טכניון, חיפה 32000, טל. 8293814 (04), פקס 8231848 (04), דואר אלקטרוני: givolid@aerodyne.technion.ac.il
חברי ועד אישהח"מ: עמנואל אור, מיכאל אנגלמן, פנחס בר-יוסף (נשיא), מישל ברקובייד, דן גבעולי (נציג ב-IACM), יצחק הררי (מנהל אתר האינטרנט), זוהר יוסיבש (מזכיר-גזבר).

אתר אישהח"מ (IACMM) באינטרנט: <http://www.iacmm.org.il>
רישום לחברות באגודה ופרטים נוספים: באתר האגודה הנ"ל, או פנו למזכיר-גזבר, ד"ר זוהר יוסיבש, המחלקה להנדסת מכונות, אוניברסיטת בן-גוריון בנגב, באר שבע 84105, טל. 6477103 (07), פקס 6472813 (07), דואר אלקטרוני: zohary@pversion.bgu.ac.il

הערות העורך:

נא שלחו לכתובת המערכת (בדואר אלקטרוני או רגיל) חומר לפרסום בעלון. ניתן ורצוי לצרף ציורים ותמונות. לידיעת חברות - ניתן גם לפרסם חומר מסחרי- פרסומי בתשלום. לפרטים נא לפנות למערכת. גירסה צבעונית של עלון זה מופיעה באתר האגודה (ראה לעיל).

בגליון זה

המאמר המופיע להלן נכתב ביזמתו של המחבר בעקבות הלומדה שהוצגה ב- ISCM-10 (ע"ד"ר משה גולדברג) בנושא חישוב מקבילי.

מחשבים וקטוריים מקביליים

יהונתן טל
חברת קריי ישראל
jtal@cray.com

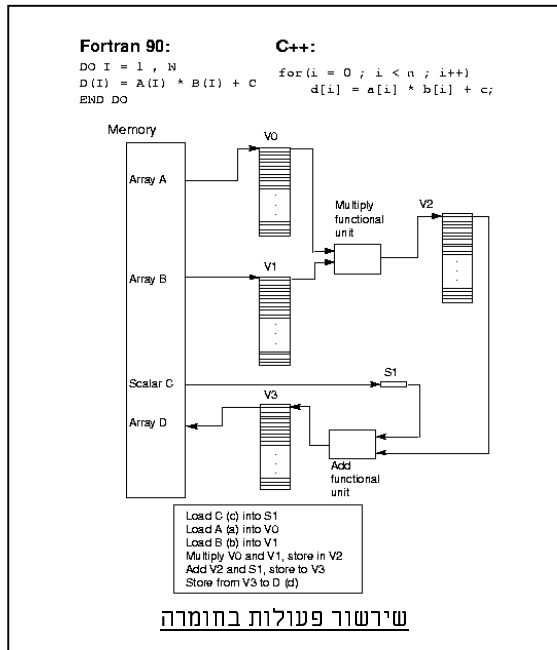


מבוא

מחשבים וקטוריים מיועדים לביצוע חישובים בצורה יעילה על וקטורים ארוכים יחסית, כאשר ישנן פעולות החוזרות על עצמן עבור כל או חלק מאברי הוקטורים. מחשבים אלו אינם מיועדים לשימוש כללי כתחנת עבודה או מחשב אישי, אלא לצורך הרצת יישומים הנדסיים, מדעיים ואחרים שהותאמו במיוחד לניצול הארכיטקטורה הוקטורית. על ידי ניצול הארכיטקטורה בצורה יעילה, ניתן להגיע לביצועי מעבדים הגדולים בסדרי גודל יחסית למעבדים המקובלים בשוק. בנוסף לפעולות הוקטוריות, ניתן לבצע פעולות בלתי תלויות בו זמנית על מעבדים שונים על מנת להאיץ את זמן ביצוע המשימה החישובית.

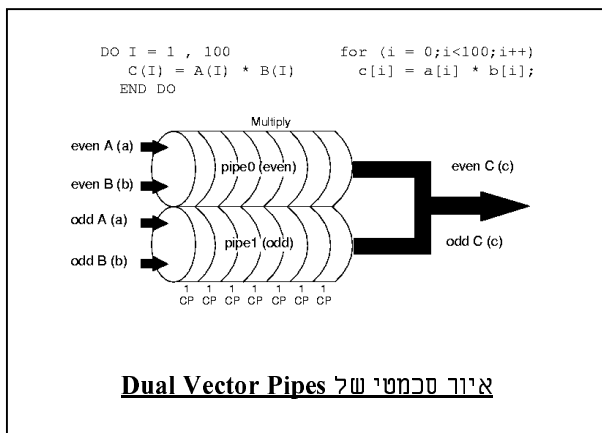
במאמר זה אציג את עקרונות החישוב בעזרת מחשבים וקטוריים, ואסביר היכן יתרונותיהם ובמה הם שונים ממחשבים מבוססי מיקרו מעבדים סטנדרטיים. לצורך דוגמאות והשוואות אתייחס למחשבים הנמצאים ברשות מרכז החישובים הבין אוניברסיטאי (מחב"א) וכן למחשבים אחרים מתוצרת חברת Cray, המתמחה בפיתוח וייצור מחשבי על. בשנת 1976 סיפקה

הוקטורי) והופך לזניח ככל שאורך וקטור הנתונים גדל. מסיבה זו, יעילות החישוב במחשבים וקטוריים גדולה יותר בבעיות חישוביות גדולות.



יחידות וקטוריות שונות קיימות עבור פעולות בשלמים (Integers), פעולות על ביטים (Bits Manipulation), פעולות לוגיות ופעולות בנקודה צפה (Floating Point), כאשר גם לפעולות הכבדות ביותר, כגון חילוק, ישנו מימוש וקטורי, וזאת בניגוד למיקרו-מעבדים סטנדרטיים.

במעבדים הוקטוריים המודרניים פועלים בד"כ מספר Vector Pipes בו זמנית על אברים שונים של וקטורי האופרנדים ועל ידי כך מגדילים את קצב העיבוד של כל הוראה וקטורית.



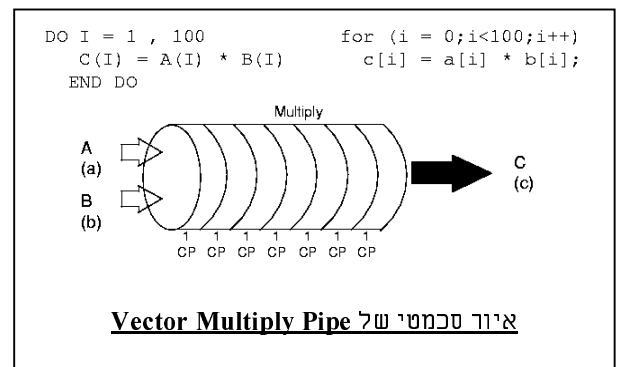
על ידי שירשור פעולות, ריבוי Pipes ואי תלות בין פעולות המתבצעות ביחידות חישוב שונות, משיג המעבד הוקטורי קצב חישוב גבוה עם תדר

Cray את המחשב הוקטורי הראשון למעבדת המחקר האמריקאית ב Los Alamos, ומאז ועד היום פיתחה מספר דורות של מחשבים וקטוריים וארכיטקטורות אחרות. מחשבים וקטוריים נפוצים אחרים, אשר אינם משווקים בארץ, הינם מתוצרת חברות יפניות.

ארכיטקטורת המעבדים

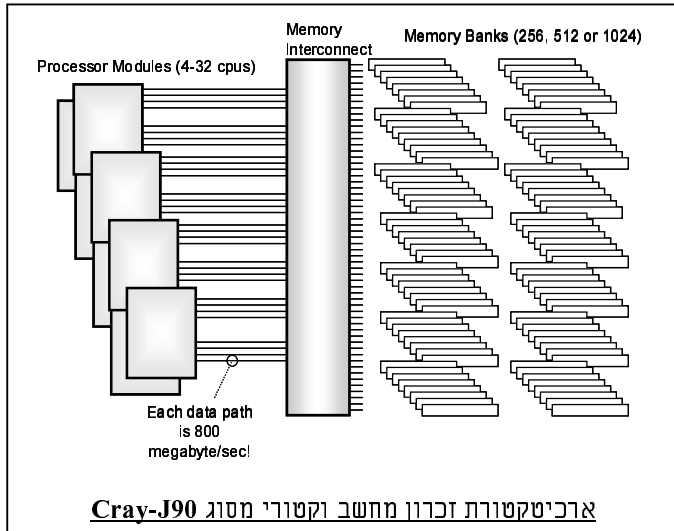
לצורך הטיפול בפעולות וקטוריות כוללים מעבדי המחשב הוקטורי אוגרים שהם עצמם וקטוריים (Vector Registers), כלומר אוסף של אלמנטים (בד"כ 64 או 128) המיועדים לאכסון האופרנדים והתוצאות של הפעולות השונות. לצורך דוגמא: מחשב וקטורי עם 8 אוגרים וקטוריים בעלי 64 אברים כל אחד יכול להחזיק סך הכל 512 אברים באוגרים, לעומת מיקרו מעבדים מודרניים בעלי 32 אוגרים למספרי נקודה צפה ו 32 אוגרים למספרים שלמים. הוראה (פקודה) וקטורית אחת גוררת חישוב עוקב על כל אברי וקטורי האופרנדים ויצירת כל אברי וקטורי התוצאה. מעבדים בהם כל הוראה יוצרת תוצאה אחת ולא וקטור של תוצאות, נקראים מעבדים סקלריים. הוראות סקלאריות במעבדים וקטוריים מיועדות לפעולות על מילים בודדות או על וקטורים קצרים מאד.

יחידות החישוב של המעבדים הוקטוריים (Vector Functional Units) הינן בעלי זמן אתחול האופייני לכל יחידה, בהתאם לסוג הפעולה שהיא מבצעת. בסיום זמן האתחול ניתן לקבל מכל יחידה תוצאה לאחר כל מחזור שעון (Clock). אופן פעולה זה, דומה להשייה ביציאת נוזל מצינור באורך מסויים ולכן נקראות היחידות הוקטוריות גם בשם Vector Pipes.

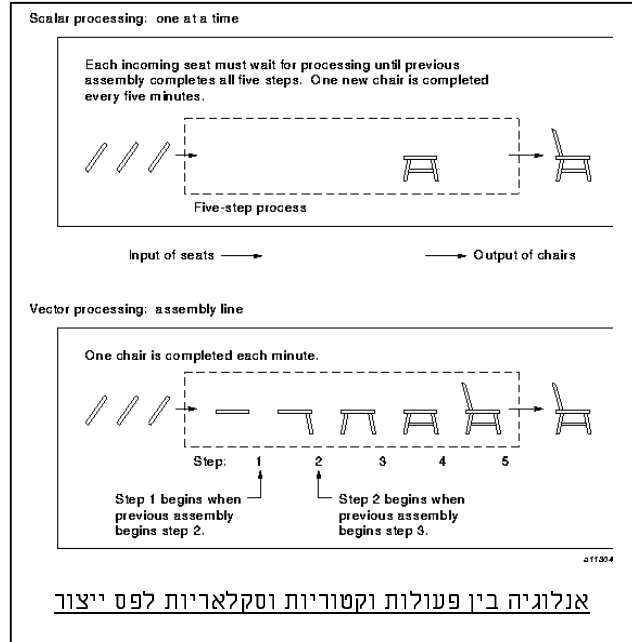


תוצאות חישוב מיחידה אחת יכולות להיות מוזנות ליחידות אחרות וליצור שרשרת של פעולות (Chaining) ועל ידי כך לאפשר ביצוע של מספר פעולות בכל מחזור שעון. הזמן עד לקבלת התוצאה הראשונה במקרה של שירשור פעולות יהיה ארוך יותר (כי "הצינור" ארוך יותר). באופן כללי, זמן האתחול הינו משמעותי עבור פעולות על וקטורים קצרים (יחסית לאורך האוגר

ארכיטקטורת המחשבים הוקטוריים על חלוקה למספר גדול של בנקים (Memory Banks), כאשר ניתן לפנות לבנקים שונים בגישות עוקבות מבלי לרדת מביצועי הגישה לזכרון, גם אם הגישה הנה בקפיצות גדולות או אקראיות (דוגמא לגישה אקראית לזכרון ניתן למצוא בפעולות על מטריצות דלילות).



שעון נמוך יחסית. באופן עקרוני שקול אופן פעולת המעבד הוקטורי לפס ייצור במפעל. מספר הפועלים בכל פס ייצור שקול לשירשור פעולות, ומספר פסי ייצור הפועלים במקביל שקול למספר ה- Vector Pipes עבור אותה הפעולה.



לאופי הגישה לזכרון השפעה ניכרת על ביצועי תוכניות מסוימות, בעיקר על כאלו המבצעות מעט פעולות חישוביות עבור כל גישה לזכרון (Memory bounded programs). מבחן ביצועים (Benchmark) ידוע, המודד קצב גישה סדרתי לזכרון נקרא Stream Benchmark ומפורסם באתר האינטרנט: <http://www.cs.virginia.edu/stream/>. יש לציין שגישות אקראיות לזכרון או גישות בקפיצה גדולה, גורמות לנפילת הביצועים בכל המחשבים מבוססי מיקרו-מעבדים עם Cache, אולם לא פוגעות בביצועי מחשבים וקטוריים, כל זמן שהפניות העוקבות אינן לאותו בנק זכרון (מספר הבנק בו מאוכסנת מילה כלשהי הינו כתובת המילה מודולו מספר הבנקים). לצורך המחשה מובאות בעמוד הבא דוגמאות לקצבי גישה לזכרון של מספר מחשבים וקטוריים לעומת מחשבים סקלריים.

וקטוריזציה

התהליך של יצירת קוד המשתמש בחומרה הוקטורית נקרא וקטוריזציה (Vectorization). תהליך זה נעשה אוטומטית על ידי הקומפיילר עבור לולאות חישוב מסוימות (ב C/C++/Fortran) בתנאי שאין גורמים המונעים זאת. בין הגורמים אשר מונעים וקטוריזציה אוטומטית של לולאות: פקודות קלט פלט בלולאת חישוב, קריאה לרוטינות מתוך לולאת חישוב, קוד לא מובנה (Spaggetti Code). תלות בתוצאות שחושבו באיטרציות קודמות של הלולאה (Data Dependencies). רוב תהליך האופטימיזציה

ארכיטקטורת הזכרון

בדרך כלל, קצב החישוב התאורטי המקסימאלי אינו מאפיין תוכניות רבות. אחת הסיבות לכך הינה שהנחנו כי הנתונים נמצאים כבר באוגרים. אולם כאשר מטפלים בוקטורים ארוכים, ישנו הכרח להביא את הנתונים מן הזכרון. לעיתים אופן הגישה לזכרון הינו סדרתי ולפעמים בקפיצות קבועות או בסדר אקראי. חשיבות רבה נודעת לקצב הבאת הנתונים מהזכרון. קצב זה נמדד ב Mbytes/sec ונקרא רוחב סרט (Memory Bandwidth). במקרים רבים, רוחב הסרט הינו זה שמגביל את ביצועי התוכנית ולא קצב החישוב התאורטי.

ארכיטקטורת הזכרון של המחשבים הוקטוריים מאפשרת את קבלת הקצבים הגבוהים ביותר בגישת כל מעבד לזכרון, בסדרי גודל יותר מאשר ארכיטקטורות אחרות. הסיבה לכך הינה שמחשבים בייצור המוני כגון שרתים, תחנות עבודה ו PCs משתמשים בד"כ בזכרון Cache מהיר, שגודלו מוגבל למספר MBytes, הנמצא מבחינת הירארכית בין המעבד לזכרון הראשי. אל ה Cache ניתן לפנות במהירות בתנאי שהנתונים הדרושים כבר נמצאים בו, אחרת ישנו Cache miss, שעולה במספר גדול יחסית של מחזורי שעון. מסיבות טכנולוגיות וכלכליות לא ניתן ליצר את כל זכרון המחשב כך שיהיה בעל ביצועים זהים ל Cache. לעומת זאת, מבוססת

דוגמא לסוג לולאות Fortran שניתנות לוויקטור
אוטומטי (מתוך קוד לסימולציה של דינמיקת
חלקיקים):

```
DO I=1, N
  U(I)=0.
  R=X(ID(I))*2+Y(ID(I))*2+Z(ID(I))*2
  R=SQRT(R)
  IF(R.LE.e) U(I)=-1./R
ENDDO
```

דוגמא לקוד שלא ניתן לוויקטור אוטומטי בגלל
קיום יחס רקורסיבי (אולם ניתן לוויקטור ע"י
פתרון הרקורסיה):

```
A(1)=1
DO I=2,64
  A(I)=A(I-1)*2+1
ENDDO
```

דוגמא לאיטרציה גאוס-זיידל אשר איננה מנצלת
וקטורזציה בגלל תלות בתוצאות קודמות של
U(I)

```
DO I=2,N-1
  U(I)=(U(I-1)+U(I+1))*0.5
ENDDO
```

לעומתה, איטרציה יעקובי מנצלת וקטורזציה
כי אין תלות של Uold ב Unew

```
DO I=2,N-1
  Unew(I)=(Uold(I-1)+Uold(I+1))*0.5
ENDDO
```

יכולת המיקבול לאחר מיצוי היכולת הוקטורית
על מנת להאיץ את ביצועי התוכנית מעבר
ליכולת של מעבד בודד. אוגרים מיוחדים
משמשים לסינכרון והעברת אינפורמציה בין
המעבדים השונים. ארכיטקטורת המחשבים
הוקטוריים הינה של זכרון משותף, בה לכל מעבד
אותה קלות גישה לכל מרחב הזכרון (Symmetric
SMP=Multi Processing). תהליך יצירת קוד
שיכול להשתמש במספר מעבדים נקרא מיקבול
(Parallelization). המיקבול יכול להעשות ברמת
הלולאות, קריאה לרוטינות בלתי תלויות, ביצוע
קטעי קוד בלתי תלויים או ביצוע אותם קטעי
קוד על נתונים שונים. בדרך כלל, יש לבצע
שינויים ידניים לקוד התוכנית, על מנת
שהקומפילר יוכל לייצר קוד מקבילי. שיטות
מיקבול בעזרת OpenMP, autotasking, MPI, PVM
SHMEM משתמשות בעובדה שהזכרון הוא
משותף ומנצלות את הפשטות בשיתוף הנתונים
בעזרת הזכרון משותף. שיטות המיקבול השונות
הוצגו בכנס איש"מ האחרון בהוצאת הלומדה
של ד"ר משה גולדברג.

דוגמאות מתוך מבחן ביצועי STREAM עבור מספר מחשבים וקטוריים
(Cray-J90,SV1,T90) וסקלריים (Cray-T3E, SGI-O2K, PC).
התוצאות בטבלה הינם קצבי גישה לזכרון (למעבד בודד) ביחידות
Mbytes/sec עבור סוגי הפעולות המוסברים למטה.

Machine	COPY	SCALE	ADD	TRIAD
Cray_T94	11341	10717	14783	13920
Cray_SV1	2429	2434	2512	2511
Cray_J932	1433	1408	1260	1270
Cray_T3E-1200	474	480	566	558
Origin2000 400MHz	380	363	422	422
PC Cluster 700MHz	350	350	414	415

Name	Kernel	Bytes/iter	FLOPS/iter
COPY:	a(i) = b(i)	16	0
SCALE:	a(i) = q*b(i)	16	1
SUM:	a(i) = b(i) + c(i)	24	1
TRIAD:	a(i) = b(i) + q*c(i)	24	2

ביצועי Transpose למטריצה בגודל 2560x2560 ביחידות
Mbytes/sec על מספר מחשבים וקטוריים (Cray: J90,SV1,T90) ועל
מחשב סקלרי (Cray-T3E).

Machine (1 cpu):	J90	SV1	T90	T3E
B=A ^T BW (MB/s)	1110	2255	10145	351

דוגמא לגישה אקראית לזכרון באמצעות וקטור אינדקסים עם פיזור
אחיד. הלולאה הבאה ב C הורצה 10 פעמים תוך מדידת הזמן הכולל.
הזמנים (בשניות) שהתקבלו, נתונים בטבלה עבור מספר מחשבים
וקטוריים (Cray: J90,SV1,T90) וסקלאריים (Cray-T3E, SGI-O2000).

```
for(int i=0;i<1000000;i++) d[i]=a[c[i]]+b[i];
```

a,b,c,d הינם וקטוריים של אלמנטים מטיפוס long int (64 ביט).

J90	SV1	T90	T3E	O2000
0.33s	0.15s	0.023s	2.15s	2.5s

בתוכניות למחשבים וקטוריים קשור ביצירת קוד
שרוב ההוראות בו הן הוראות וקטוריות. קוד כזה
נקרא קוד וקטורי והוא מנצל ביעילות את
הארכיטקטורה. הרווח בביצועים של קוד וקטורי
לעומת קוד סקלארי יכול להגיע לפי 10 ויותר.
לפעמים, על מנת שהקומפילר יוכל לייצר קוד
וקטורי, יש לבצע אופטימיזציה ידנית לתוכנית,
כלומר שינוי קוד (כמובן מבלי לשנות את
התוצאות). במקרים מסוימים, כדאי להשתמש
באלגוריתמים שונים על מנת לאפשר
וקטורזציה.

חישוב מקבילי

אחת התכונות החזקות של מחשבים וקטוריים
מקביליים הינה היכולת לשיתוף פעולה של
מספר מעבדים (Multi-Tasking), לצורך השלמת
המשימה החישובית יותר מהר. בדרך כלל תמומש