P573

Assignment 4

1) A lower triangular solve with one right-hand side is a level-2 BLAS operation (why?). So how does it compare to the other two such creatures you've seen so far, matrix-vector multiply and rank-1 update? "Compare" means in terms of which should run fastest/slowest.

Solution →

We have been given the following loop:

```
for k = 1:n
        u(k) = v(k)
        for i = 1:k-1
            u(k) = u(k) - L(k,i)*u(i)
        end for
        u(k) = u(k)/L(k,k)
    end for
```

A Matrix – Vector multiplication is a Level 2 BLAS operation because in the equation is L*u = v L is a Lower triangular matrix u and v are vectors hence this becomes a case of Matrix – vector multiplication which is a level 2 BLAS operation.
Load/Store Analysis for the above loop :

Step 1) Calculate Memory references :

- Read Matrix L
- Read v
- Write u

After adding all the operations we get the following equation :
n*(n+1) / 2 + n + n = (n^2 + 5*n) / 2

Step 2) Calculate Flops

- Number of addition and multiplication operations
- Number of division operation

The equation looks like
n(n-1) + n = n^2
As per the Load/store analysis formula we have the following result
Mem ref/flops = ((n^2+5*n) / 2) / n^2 = 1/ 2 = 0.5

Comparison between single right-hand side case L , Matrix vector multiply and rank – 1 update

Matrix vector multiply : -
$y = y + Ax$

Number of memory references : -

- Read A
- Read x
- Read y
- Write y

The equation becomes : n^2 + n + n + n = n^2 + 3*n
Number of flops : 2*n^2
Memory ref/flops = (n^2 + 3*n)/(2*n^2) = 1/2 = 0.5

Rank -1 update
Number of memory references :-
- Read A
- Read u
- Read v
- Write A

The equation becomes : n^2 + n + n + n^2 = 2*n^2 + 2*n
Number of Flops : 2*n^2
Memory ref/flops = (2*n^2 + 2*n)/2*n^2 = 1

Hence it is clear that lower triangle solve and matrix-vector are similar and faster that rank – 1 update.

2) The algorithm above can actually be done using one array for both *u* and **v**, which is the way it usually is done. How does using overwriting this way change the load/store analysis?

Solution→
By using both u and v there wont be any difference in the load / store analysis because the number of reading / writing n elements. But we can say that by removing one array we gain on the performance.

3) What level-1 BLAS operation is implemented by the innermost loop in the code above? Does the load/store analysis imply that implementing it as a sequence of those level-1 BLAS operations is near-optimal, or that there is a better way?

Solution→
The dot product, level-1 BLAS operation is implemented by the innermost loop in the code.
Load Store Analysis with the inner loop implemented as level-1 BLAS dot product operation.
Number of memory references for level-1 BLAS = 2(k-1) + 1

Total memory references = (summation from k=1 to n.. 2(k-1) + 1) + (Read v) + (Write u) = n(n+1) + n = n^2 + 2*n
Number of flops for level-1 BLAS = 2(k-1) - 1
Total flops = (summation from k=1 to n……. 2(k-1) - 1) + (n divisions) = n^2 - n
Memory ref/flops = (n^2 + 2*n)/(n^2 - n) -> 1

As per load store analysis, implementing it as a sequence of those level-1 BLAS operations is not optimal. The lower triangular solve with one right side has better performance as per load store analysis.

4) The inner and outer loops can be interchanged in the algorithm give above, provided the division by diagonal elements *L(k,k)* is handled properly. How does that change the load/store analysis for solving *L*u = v* ?

Solution→

If the inner and outer loops are inter changed then the Load/store analysis is given as follows

Number of memory references in the new inner loop :

- Read u
- Read v
- Write u

The equation becomes = (k-1) + (k-1) + (k-1) = 3*(k-1)

Total number of memory references = (summation from k=1 to n.. 3(k-1) + 1) = (3*n^2 - n)/2
flops = n^2
Memory ref/flops = ((3*n^2 - n)/2)/n^2 -> 3/2 = 1.5

Thus the implementation is slower.

<u>Multiple right – hand side case</u>

Solution 1→

This is BLAS level- 3 operation because it involves Matrix-Matrix multiplication. The above described implementation has an outer loop with the inner loop implemented by BLAS level -2 operation.

Solution 2→

The Load / Store Analysis for the operation is as follows
Number of memory references = (Memory references for Single right-hand side case)*n = ((n^2 + 5*n)/2)*n
flops = (flops for single right hand side case) * n = (n^2)*n = n^3
Memory ref/flop = (((n^2 + 5*n)/2)*n)/n^3 -> 1/2 = 0.5

As level-3 BLAS operation-
The number of memory references will change because faster subcomponents may be written to obtain optimized accesses.

Number of memory references =

- read L
- read V
- write U

$n(n+1)/2 + n^2 + n^2 = (5*n^2 + n)/2$
flops = n^3
Memory ref/flops = $((5*n^2 + n)/2)/n^3 \rightarrow 1/n$

Solution 3→
We see that the performance of multiple right-hand side version is same as the performance of the single right-hand side version. This makes sense as multiple right-hand side case essentially executes the single right-hand side version n times.

<u>More!</u>

Solution 1→
The load store analysis does not change for solving an upper triangular system $U*x = y$. Nor does it change for the multiple right-hand side upper triangular system

Solution 2→
If the diagonal elements are not included then number of elements in the lower triangular matrix is $n*(n-1)/2$. Also the division operation can be excluded.
Number of memory references =

- Reads L
- Read v
- Write u

$n*(n-1)/2 + n + n = (n^2 + 3*n)/2$
Flops = (Number of addition and multiplication operations) = $n(n-1) = n^2 - n$
Memory references/flops = $((n^2 + 3*n)/2)/(n^2 - n)$

theoretically the number of memory references should reduce by $(n/(n*(n-1)/2)$ i.e $2/(n-1)$

Sources :

1) www.cs.indiana.edu/classes/p573-bram/notes.html
2) Discussed the assignment with Rohan Pillai