Module-2

CSEN 3104

Lecture 19

22/08/2019

Dr. Debranjan Sarkar

SIMD Algorithms

Matrix multiplication

Matrix Multiplication basics

- Let $A = [a_{ik}]$ and $B = [b_{kj}]$ be n x n matrices
- Product matrix $C = A X B = [c_{ij}]$ of dimension n x n
- The elements of the product matrix C is related to elements of A and B by:

$$c_{ij} = \sum_{k=1}^{n} a_{ik} \times b_{kj} \text{ for } 1 \leq i \leq n \text{ and } 1 \leq j \leq n$$

- There are n^3 cumulative multiplications to be performed.
- Cumulative multiplication refers to the linked multiply-add operation

  $c \leftarrow c + a X b$.
- Addition is merged into the multiplication because the multiply is equivalent to multi-operand addition
- Unit time is considered as the time required to perform one cumulative multiplication

Matrix multiplication in SISD computer

For i = 1 to n Do

For j = 1 to n Do

$C_{ij} = 0$ (Initialization)

For k = 1 to n Do

$C_{ij} = C_{ij} + A_{ik} . B_{kj}$ (Scalar additive multiply)

End of k loop

End of j loop

End of i loop

- In a conventional SISD uniprocessor system, the n^3 cumulative multiplications are carried out by a serially coded program with 3 levels of DO loops corresponding to three indices to be used
- The time complexity of this sequential program is proportional to n^3

Matrix multiplication in SIMD computer

For i = 1 to n Do

Par for k = 1 to n Do

$C_{ik} = 0$ (Vector load)

For j = 1 to n Do

Par for k = 1 to n Do

$C_{ik} = C_{ik} + A_{ij} . B_{jk}$ (Vector multiply)

End of j loop

End of i loop

Matrix multiplication in SIMD computer

- There are n PEs
- The algorithm construct depends heavily on the memory allocations of the A, B, and C matrices in the PEMs
- Each row vector of the matrix is stored across the PEMs (Show figure)
- Column vectors are then stored within the same PEM
- This allows parallel access of all the elements in each row vector of the matrices
- First parallel do operation corresponds to vector load for initialization
- Other parallel do operation corresponds to vector multiply for the inner loop of additive multiplications
- The time complexity has been reduced to $O(n^2)$
- SIMD algorithm is n times faster than the SISD algorithm for matrix multiplication

Matrix multiplication in SIMD computer

- Vector load operation is performed to initialize the row vectors of matrix C, one row at a time
- For vector multiply operation, the same multiplier a_{ij} is broadcast from the CU to all the PEs to multiply all n elements of the i^{th} row vector of B
- In total, n^2 vector multiply operations are needed in the double loops
- Show table illustrating the successive contents of the C Array in memory
- Each vector multiply instruction implies n parallel scalar multiplications in each of the n^2 iterations

Sorting on a mesh-connected parallel computer

Parallel Sorting on mesh

- Sorting of $N = n^2$ elements on an $n \times n$ mesh-type processor array
- Architecture (show figure) is similar to Illiac IV with exceptions
  - No wraparound connections, i.e.,
  - PEs at the perimeter have 2 or 3 rather than 4 neighbours
  - This simplifies the array sorting algorithm
- Two time measures are required to estimate the time complexity of the algorithm:
  - Routing time (t_R) to move one data item from a PE to one of its neighbours
  - Comparison time (t_C) for one comparison step (conditional interchange on the contents of two registers in each PE

Parallel Sorting on mesh

- Concurrent data routing is allowed
- Upto N numbers of concurrent comparisons may be performed
- This means that a comparison-interchange step between two items in adjacent processors can be done in time $2t_R + t_c$ *(route left, compare, and route* right)
- A number of these comparison-interchange steps may be performed concurrently in time $(2t_R + t_c)$ *if* they are all between distinct, vertically adjacent processors
- A mixture of horizontal and vertical comparison-interchanges will require at least $(4t_R + t_c)$ time unit

Thank you