VOL. 4, NO. 12, December 2014 ISSN 2225-7217
ARPN Journal of Science and Technology

http://www.ejournalofscience.org

Matrix Multiplication by (Map Reduce) Segmentation

[1] Egbenimi Beredugo Eskca, [2] Eltaeib Tarik
[1, 2] Department of Computer Science, University of Bridgeport, Bridgeport Connecticut

[1] eberedug@my.bridgeport.edu

ABSTRACT

This paper was produced as a result of my interest in metric multiplication of high ordered matrix. Following several attempts to simplify the task of matrix multiplication, there abound a plethora of information pertaining to various methods of achieving, varying degree of simplicity and accuracy in the multiplication of matrices of high order. My research primarily focused on the need to split large matrices into smaller segment, multiply the smaller segments in a systematic manner and reassembling the results to form the product of what would be the result of multiplying the large matrices. I have envisioned my approach as a new approach to the common map Reduce approach for metric multiplication.

The result of my research and analysis, and the consequent methodology, is a fast, and practical matrix multiplication system that is applicable to parallel and distributed computing environment. It is my hope that the methods included in this article will greatly improve the speed of multiplying large matrices, as well as reduce the accompanying computational complexity required to implement the matrix multiplication system. Initial experimentation with a simple C++ simulation produced very impressive results.

This article is mainly meant to expose the logic, algorithm and the mathematically deduction that serve as the foundation of the proposed system without diminishing the broader applicability of the techniques outlined in the paper.

Keywords: *Segmentation*, *Matrix*, *Map reduce, Second square, Computational-mapping*, *Hama*

1. AIM

To achieve fast parallel matrix multiplication

1.1 Procedure

Given two square matrix A and B of order mxn, reduce each matrix into four square matrices by segmenting the source matrices into four equal component matrices.

1.2 Expectation

Since this is a matrice multiplication of two square matrices, the product of the multiplication will be a square matrice of the same order. For a square matrix m = n; thus multiplying two matrices of the order mxn will produce a matrix of the same order as the original matrices.

It follows that the product matrix can be segmented into four equal components matrices just like the source matrices.

1.3 Requirement

Find the minimum number of source component matrix multiplication that would result into a component of the product matrix.

See figure 1 below.

$$A \begin{array}{c} A_0 \qquad A_1 \\ \left(\begin{array}{cc|cc} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ \hline a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{array}\right) \\ A_2 \qquad A_3 \end{array} \quad \times \quad B \begin{array}{c} B_0 \qquad B_1 \\ \left(\begin{array}{cc|cc} b_{00} & b_{01} & b_{02} & b_{03} \\ b_{10} & b_{11} & b_{12} & b_{13} \\ \hline b_{20} & b_{21} & b_{22} & b_{23} \\ b_{30} & b_{31} & b_{32} & b_{33} \end{array}\right) \\ B_2 \qquad B_3 \end{array} \quad = \quad C \begin{array}{c} C_0 \qquad C_1 \\ \left(\begin{array}{cc|cc} c_{00} & c_{01} & c_{02} & c_{03} \\ c_{10} & c_{11} & c_{12} & c_{13} \\ \hline c_{20} & c_{21} & c_{22} & c_{23} \\ c_{30} & c_{31} & c_{32} & c_{33} \end{array}\right) \\ C_2 \qquad C_3 \end{array}$$

Fig 1:

Component matrices: A0, A1, A2, A3; B0, B1, B2, B3; C0, C1, C2, C3

VOL. 4, NO. 12, December 2014 ISSN 2225-7217
ARPN Journal of Science and Technology

http://www.ejournalofscience.org

2. COMPUTATIONAL MAPPING

$$\begin{array}{c} C_0 \\ \begin{pmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \end{pmatrix} \\ c_{10}\ \ c_{11} \end{array} = \left\{ \begin{array}{c}\begin{pmatrix} a_{00} & a_{01} \\ a_{10} & a_{11}\end{pmatrix} \\ c_{10}\ \ c_{11}\end{array} \text{ X } \begin{array}{c}\begin{pmatrix} b_{00} \\ b_{01}\end{pmatrix} \\ b_{10}\ \ b_{11}\end{array} \right\} + \left\{ \begin{array}{c}\begin{pmatrix} a_{02} & a_{03} \\ a_{12} & a_{13}\end{pmatrix} \\ c_{10}\ \ c_{11}\end{array} \text{ X } \begin{array}{c}\begin{pmatrix} b_{20} \\ b_{21}\end{pmatrix} \\ b_{30}\ \ b_{31}\end{array} \right\}$$

It follows from the above that

$$C_{00} = ((a_{00} \text{ x } b_{00}) + (a_{01} \text{ x } b_{10})) + ((a_{02} \text{ x } b_{20}) + (a_{03} \text{ x } b_{30}))$$

$$C_{01} = ((a_{00} \text{ x } b_{01}) + (a_{01} \text{ x } b_{11})) + ((a_{02} \text{ x } b_{21}) + (a_{03} \text{ x } b_{31}))$$

$$C_{10} = ((a_{10} \text{ x } b_{00}) + (a_{11} \text{ x } b_{10})) + ((a_{12} \text{ x } b_{20}) + (a_{13} \text{ x } b_{30}))$$

$$C_{11} = ((a_{10} \text{ x } b_{01}) + (a_{11} \text{ x } b_{11})) + ((a_{12} \text{ x } b_{21}) + (a_{13} \text{ x } b_{31}))$$

Similarly,

$$C_1 = \left(A_0 \text{ x } B_1\right) + \left(A_1 \text{ x } B_3\right)$$

$$C2 = \left(A_2 \text{ x } B_0\right) + \left(A_3 \text{ x } B_2\right)$$

$$C3 = \left(A_2 \text{ x } B_1\right) + \left(A_3 \text{ x } B_3\right)$$

I refer to the source segments like (A_0 , B_1 , A_1 , B_3) that multiplies to produce a product segment such as C_1 as the **segment copies**. Thus (A_0 , B_1 , A_1 , B_3) are segment copies of C_1 .

It follows that, to multiply two matrices of order 4 (i.e. m= n = 4), eight source component matrices need to be multiplied in order to correctly map their product into the product matrices.

This conclusion is true for any square even number ordered matrix whose order is greater than two. Thus, for any even number ordered matrix greater than two (m x n; m=n and n > 2), it is possible to segment the multiplier and the multiplicand into 4 segment.

On successful segmentation of the source matrices, a possible base level 2n multiplications are required to map the reduced segmentations into the required n segments of the product matrix(n = number of segments).

3. NON EVEN–SQUARE MATRIX

With a minimal modification, I was able to apply this approach to the multiplication of odd number square matrices.

For odd number, I padded the last column and the last row with zeros to convert the matrix into an even number square matrix, thus allowing me to apply the same rule to the odd pair of square matrices to produce the same result as the even number ordered square matrices. See figure 2 below

Multiplying matrices A and B of order = mxn ; n = m =3

$$A \begin{array}{c} \begin{array}{cc} A_0 & A_1 \end{array} \\ \left(\begin{array}{cc|cc} a_{00} & a_{01} & a_{02} & 0 \\ a_{10} & a_{11} & a_{12} & 0 \\ \hline a_{20} & a_{21} & a_{22} & 0 \\ 0 & 0 & 0 & 0 \\ A_2 & & & A_3 \end{array}\right) \end{array} \text{ X } B \begin{array}{c} \begin{array}{cc} B_0 & B_1 \end{array} \\ \left(\begin{array}{cc|cc} b_{00} & b_{01} & b_{02} & 0 \\ b_{10} & b_{11} & b_{12} & 0 \\ \hline b_{20} & b_{21} & b_{22} & 0 \\ 0 & 0 & 0 & 0 \\ B_2 & & & B_3 \end{array}\right) \end{array} = C \begin{array}{c} \begin{array}{cc} C_0 & C_1 \end{array} \\ \left(\begin{array}{cc|cc} c_{00} & c_{01} & c_{02} & 0 \\ c_{10} & c_{11} & c_{12} & 0 \\ \hline c_{20} & c_{21} & c_{22} & 0 \\ 0 & 0 & 0 & 0 \\ C_2 & & & C_3 \end{array}\right) \end{array}$$

Fig 2:

VOL. 4, NO. 12, December 2014 ISSN 2225-7217
ARPN Journal of Science and Technology

http://www.ejournalofscience.org

On mapping to the product matrix, the excess, zeros row and column should be discarded to produce the final matrix product of A x B.

4. SECOND SQUARE REDUCTION

In a process I call the second square reduction (SSR); attempt is made to reduce any given matrix into optimum segments made up of 2x2 matrices. Corresponding segment copies of these minimum segments are then transmitted to various parallel processing unites to be multiplied. The various products are reassembled in the mapping phase to form the final product.

Recall that if a matrix is reduced to n SSR segments, then a total of 2n multiplications are required to map the products of the reduced components to the final product. For matrices that would not reduce to 2x2 matrices on their final segmentation, a zero padding of final matrix is required to reduce to the SSR format.

Pseudo code for metric segmentation

```
If((m=n) and  n > 2)//prerequisite for segmentation apply
to matrices of order higher than 2x2

Segment_count = 0;
For(int I = 0; I < row(A) ; i++)
For (int j = 0; j < col(A); j++){
A0[I,J] = A[I,J)
If(J > n/2){
J =  col //exit inside loop; re-enter for entry of second row
of segment.
If(I > n)//signal segment full and reset for next segment
Segment_count++}}
```

5. OBSERVATION

This approach to matrix multiplication requires increasingly large total number of multiplication as the order of the multiplier and the multiplicand matrices increases. However, it is worth noting that the objective of this method is to reduce the source matrices into simple 2x2 matrix which can be multiplied with little computing resource and overhead. Second, multiplying 2x2 matrices and assembling the final products would be faster than most of the methods of multiplying matrices that are available in the computing industry as well as the academia. Very large matrices, reduction can be limited to available computing resource.

A comparison of the above method is made with a version based on the modification of the hadoop approach.

In their attempt to modify the map Reduce approach to matrix multiplication outlined in the Hadoop HAMA project, **Botong Huang and You Wu (Will)** of the computer science department of the Duke University outlined a process of matrix multiplication as shown below.

As shown in Figure 3 below, they partition each of the input matrices into n x n small square blocks of equal size. The size of each block according to them would be M/n x M/n assuming that the matrices to be multiplied are square matrices of order M.

They reckoned that the output matrix would consists of nxn blocks, each resulting from the addition of n block matrix multiplications. They let each map task handle one block matrix multiplication. So there would be totally n^3 map tasks.

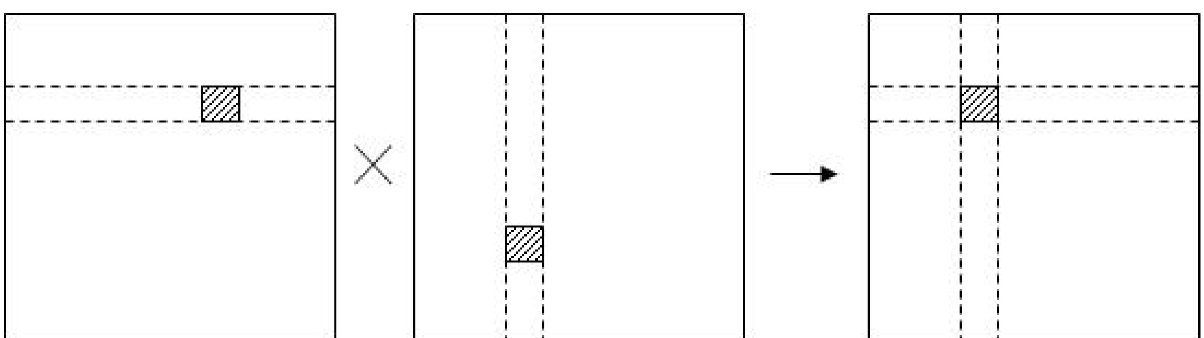

Fig 3:

For clarification, they posit that for a 4x4 matrix like the one I used to demonstrate my case, it would follow from their assumption that a partition of four 2x2 matrices are possible where n = 2. Their approach attempt to partition the sources matrices into small square matrices of equal size without recourse to the final size of the partitioned matrices. This approach compromise efficiency, as the size of the source matrices get larger; although their approach would result into fewer number of multiplication.

6. CONCLUSION

With the availability of computing resources, our approach to matrix map Reduce multiplication ensures fast and efficient way of multiplying large matrices. Although some other algorithm is not shown here for purpose of clarity, are required to standardize none square matrices and large matrices to the SSR format, We believe that our second square reduction process presents a new and efficient way to multiply large matrices in a parallel computing environment.

ACKNOWLEDGEMENT

Without the consistent encouragement and support of Professor Tarik El Taeib, this article would never have seen in the light of day. We thus use this medium to express our gratitude to his generous intellectual prowess.

REFERENCES

[1] Project Report: Matrix Multiplication in Map Reduce: Botong Huang and You Wu (Will) - Department of Computer Science

[2] J. Norstad, A Map Reduce Algorithm for Matrix Multiplication", 2009

[3] S. Seo, E.J. Yoon, J. Kim, S. Jin, J.S. Kim, and S. Maeng, \HAMA: An E_cient Matrix Computation with the Map Reduce Framework", 2010.