ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2014

Computing the Determinants in the Multiprocessor Computer

Salman H. Abbas

Associate Professor, Department of Mathematics, College of Science, University of Bahrain, Kingdom of Bahrain

ABSTRACT: This paper presents a new parallel algorithm for computing the determinants of block matrices of order $w \times w$, with Computational Complexity $O\left(\frac{w^3 q^2}{2}\right)$, using the Gauss elimination method and Abbas's block method[7] as references. We present our algorithm using Ada-multitasking. The algorithm is numerically stable and has been tested in the Sequent Balance machine with efficient utilization of all processors.

KEYWORDS: Determinants, Gauss elimination method, Multitasking, Theoretical and experimental costs.

I.INTRODUCTION

Current research into parallel algorithm and parallel Computing driven the need to increase computing power.Two aims are achieved by increasing power: more problems can be computed in a given interval of time, and problems whose solutions have been too expensive to calculate can be solved.
In this paper, we present an efficient algorithm for the parallel computing of the determinant using block Gauss elimination method. The algorithm is developed with partitioning the $n \times n$ matrix into small blocks of size$w \times w$and the number of block rows$\left(q = \frac{n}{w}\right)$, where $1 \leq i \leq q$. We extend the idea to describe the effect of multiprocessors communication in the execution time.

The time spent on inter-processor communication will be dominant constraint on the performance of optimal algorithm when the matrix and multiprocessor sizes are Large. Some authors have discussed the effect of the communication network of multiprocessor on the communication costs incurred when using a multiprocessor[1 , 2 , 3],and the effect of the Communication costs on the execution time of parallel algorithm [4 , 5] . The outline of this paper is as follows: we describe our algorithm and problem assumptions in Section 2. In Section 3, we describe the parallel algorithm. In Section 4,we give the total costs of the sequential and parallel algorithms. We briefly summarize the results and make Concluding remarks in Section 5.

II. RELATED WORK

A parallel algorithm for calculating large determinants with arbitrary precision on parallel machine was implemented[11] using Gaussian elimination. The size of the determinants are *NxN* and all the calculation was done with complexity $N^2 + O(N)$.In[12],proposed efficient procedure for computing determinant of structured sparse matrices was implemented using *LU* factorization .As the matrix $A=P^{T}LU$ of size *nxn* , where P is the matrix permutation , *L* and *U* are respectively a lower triangular matrix with unit main diagonal and an upper triangular matrix. Therefore *det(A)=det(P).det(U), det (P)=± 1* and *det (U)* is the product of the diagonal entries of *U*.This approach comparing to our work is slightly similar specially when $det(A)= \prod_{i=1}^{k} det(P_i) . det(U_i)$ in case of repeating the process *(k-1)* times.The speedup for the full factorization is of complexity $O(k^3 n^3)$ and for parallel implementation is $O(Kn^3)$.In [13],MAPLE and MATLAB are used for computing determinants .Therefor comparing our code against those implementation is meaningful. From our computational results in section(VII) the

DOI: 10.15680/IJIRSET.2014.0308062

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2014

implementationCan be used to improve the efficiency in terms of the speedup and the number of processors. It is mentioned in [13]The total cost for computing the determinant is bounded by *O(m²)* arithmetic operations ,where m is the size of the determinant which is exactly similar to our algorithm. Finally in [14] , a new method was developed to calculate every determinant of size *nxn (when n>2)*,by calculating four determinants of size *(n-1)x(n-1)* and one determinant of size *(n-2)x(n-2)* on condition that *(n-2)x(n-2)* to be different than zero.It is reported that the time complexity of this method is *O(n)*.

ALGORITHM

Our focus in this paper is on MIMD (Multiple Instruction Multiple Data) Multiprocessor [6] and modeling parallelism at the level of concurrent of floating point operations. Consider the matrix A of size $n \times n$,

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & & a_{2n} \\ \vdots & \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & a_{n3} & & a_{nn} \end{pmatrix},$$

We partition the matrix A into q^2 blocks of size $w \times w$ each, where $q = \frac{n}{w}$, then the matrix A becomes,

$$A = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1q} \\ A_{21} & A_{22} & & A_{2q} \\ \vdots & \vdots & \cdots & \vdots \\ A_{q1} & A_{q2} & \cdots & A_{qq} \end{pmatrix}.$$

The forward elimination algorithm can be written as:

For $i = 1$ to q_1 loop
Compute A_{ii}^{-1} ;
For $j = i + 1$ to qLoop
$M_{ji} = A_{ji} * A_{ii}^{-1}$;
For $k = i + 1$ to q loop

$$A_{jk} = A_{jk} - M_{ji} * A_{ik}$$

End Loop;
End Loop;
End Loop;

This involves getting Zeros below the diagonal. The upper triangular block matrix can be written as

$$A = \begin{pmatrix} A_{11} & A_{12} & A_{13} & \cdot\cdot & A_{1q} \\ 0 & A_{22} & & & \\ 0 & 0 & \ddots & \neq & 0 \\ \vdots & \vdots & & \ddots & \\ 0 & 0 & \dots.. & 0 & A_{qq} \end{pmatrix}$$

then

DOI: 10.15680/IJIRSET.2014.0308062

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2014

$$det\ A = (det\ A_{11})\ (det\ A_{22}) \ldots \quad (det\ A_{qq})$$
$$= \Pi_{i=1}^{q} \det A_i$$

III. PARALLEL ALGORITHM

In this Section we consider the parallel implementation of block Gauss elimination using multitasking. Assuming that p processors are available and $p < q$[8,9,10], thenthe updated blocks in each block row can be carried in parallel. Therefore, the following procedures are needed:

1) Procedure to compute the inverse block,
2) Procedure to multiply two blocks,
3) Procedure to subtract two blocks,
4) Procedure to evaluate the determinant of a block,
5) Procedure to multiply the determinant of the blocks in the diagonal.

The first task is to reduce the blocks below the diagonal to Zeros:

Accept block row (i), block row $(j), i$;
Call procedure to compute A_{ii}^{-1} ;
E valuate the multipliers
$M_{ji} = A_{ji} * A_{ii}^{-1}$;
Call a task to update the blocks;
Compute $A_{jk} = A_{jk} - M_{ji} * A_{ik}$;

The second task is to compute the determinant of the blocks in the diagonal Accept block A_{ii}
Call procedure to computedet A_{ii}

$$i = 1 \quad to \quad q$$

Collect the results.

The third task is to evaluate the multiplication of the determinant of the blocks
$\det = \Pi_{i=1}^{q} \det A_{ii}$

PREDICTED COST

In this Section we give the sequential and parallel costs of the block Gauss elimination algorithm using Abbas's block methods[7].

A) SEQUENTIAL ALGORITHM COST

In the sequential mode the cost depends only on the number of arithmetic operations required for the algorithm. So, reducing the blocks below the diagonal to zeros requires:

1

$$t_1 = \sum_{i=1}^{q-1} \{\frac{w^3}{2} + \sum_{j=i+1}^{q} [\, w^3 + \sum_{k=i+1}^{q} (\, w^3 + w^2)]\}$$

DOI: 10.15680/IJIRSET.2014.0308062

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2014

$$= \sum_{i=1}^{q-1} \{\frac{w^3}{2} + \sum_{j=i+1}^{q} [w^3 + (w^3 + w^2)(q-i)]\}$$

$$= \sum_{i=1}^{q-1} [\frac{w^3}{2} + [w^3(q-i+1)(q-i+1) + w^2 (q-i)^2]$$

$$= \frac{w^3}{2}(q-1) + w^3 q^2 (q-1) - 2w^3 q^2 \frac{(q-1)}{2} + w^3 q(q-1)$$

$$= w^3 \frac{q(q-1)}{2} + w^3 q \frac{(q-1)(2q-1)}{6} + w^2 q^2 (q-1)$$

$$= 2w^2 \frac{q^2(q-1)}{2} + w^2 \frac{q(q-1)(2q-1)}{-6}$$

$$= \frac{w^3}{2}(q-1) + \frac{w^3}{6}(2q^3 - 3q^2 + q + 3q^2 - 3q) + \frac{w^2}{6}(2q^3 - 3q^2 + q)$$

$$= \frac{w^3}{2}(q-1) + \frac{w^3}{6}(2q^3 - 2q) + \frac{w^2}{6}(2q^3 - 3q^2 + q)$$

$$= \frac{w^3}{2}(3q - 3 + 2q^3 - 2q) + \frac{w^2}{6}(2q^3 - 3q^2 + q)$$

$$= \frac{w^3 q^3}{3} + \frac{w^2 q^2}{3} = O\left(\frac{w^3 q^3}{3}\right).$$

To findt_2 , it is known that one can compute a determinant of size w in the time of $O(w^2)$. Since the blocks below the diagonal are no longer required, so the time required to compute the, block determinant $\prod_{i=1}^{q} \det A_{ii}$ is $t_2 = w^3 q/_3$.

Hence the total cost required to compute the determinants using the sequential block Gauss elimination is :

$$T = t_1 + t_2 = \frac{w^3}{3} q^3 + \frac{w^3}{3} q$$
$$= O\left(\frac{w^3 q^3}{3}\right). \quad (2)$$

DOI: 10.15680/IJIRSET.2014.0308062

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2014

IV. PARALLEL ALGORITHM COST

Here,we give the predicted cost of the algorithm mentioned in the previous section[3] in terms of arithmetic operation count, communication cost and data sending. A test program was written to measure these quantities. The estimate of the cost of one arithmetic operation, t_f, is0.26 millisec, the time to set up one rendezvous, t_r, is 2 millisec and the time to send one data item,t_c ,is 0.02 millisec.

The following table describes the number of multiplication and additions and the number of steps required for each operation. It is assumed that there are at least q processors.

Table 4.1

Operation	Number of multiplications and additions	Number of steps
Inverse A_{ji}	$\frac{w^3}{2}$	$q-1$
Compute the multipliers	w^3	$q-1$
Reduce the blocks below the diagonal to zero	w^3	$q(q-1)/2$
Compute the block …	$w^3/3$	q

If we assume that the number of processors $p < q$, then the total number of arithmetic operation should be multiplied byq/p .Therefore, the total cost of arithmetic operations is approximately $\left(\frac{11}{6}w^3\,q^2/p\right)t_f$.

At stage i , we have $2(q-i)$ rendezvous and $3q^2(q-1)\,w^2/2$ data sent. Therefore,

$$\sum_{i=1}^{q-1}(q-i)^2$$

Which is equal to $q(q-1)\,(2q-1)/3$ rendezvous and $2q(q-1)\,w^2/3$ data are sent.

Hence, the predicted time for this algorithm is:

$$[2q(q-1)(q+1)]\,t_\tau/3 + [q(q-1)(17q-4)w^2]\,{}^{t_c}/_6 + \left(\frac{11}{6}w^3\,q^2/p\right)t_f. \quad (3)$$

COMPUTATIONAL RESULTS

In this section, we give the experimental timings, t_c, where the program was implemented on the sequent balance machine with sevenprocessors. Also we give the predicted timings, t_p. All Speedups and efficiencies are given.

DOI: 10.15680/IJIRSET.2014.0308062

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2014

Table5.1: Experimental and Predicted timings

Determinant Size	Processor	Experimental timing t_c	Predicted timing t_p
$n = 100$, $q = 10$, $w = 10$	1	48.987	53.961
	2	27.501	30.132
	3	21.457	22.194
	4	17.519	18.220
	5	14.440C	15.832
	6	12.352	14.250
	7	11.929	13.109
$n = 200$, $q = 20$, $w = 10$	1	235.163	243.867
	2	129.252	148.533
	3	110.401	116.756
	4	97.325	100.867
	5	89.678	91.333
	6	82.910	84.977
	7	79.813	80.438
$n = 256$, $q = 32$, $w = 8$	1	397.852	407.841
	2	279.481	282.882
	3	235.120	241.232
	4	216.281	220.404
	5	201.120	207.902
	6	198.541	199.578
	7	192.113	193.628
$n = 300$, $q = 20$, $w = 15$	1	732.120	749.900
	2	414.235	428.150
	3	311.612	320.900
	4	252.282	267.275
	5	222.357	235.100
	6	210.320	213.650
	7	191.246	198.328

DOI: 10.15680/IJIRSET.2014.0308062

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2014

Table5.2: Speed ups and Efficiencies

Determinant Size	Processor	Actual Speed up	Actual Efficiency	Predicted Speed up	Predicted Efficiency
100	1	1	1	1	1
	2	1.78	0.89	1.79	0.89
	3	2.28	0.76	2.43	0.81
	4	2.79	0.70	2.96	0.74
	5	3.39	0.68	3.41	0.68
	6	3.97	0.66	3.79	0.63
	7	4.11	0.59	4.12	0.59
200	1	1	1	1	1
	2	1.82	0.91	1.64	0.82
	3	2.130	0.71	2.09	0.69
	4	2.416	0.60	2.42	0.60
	5	2.62	0.52	2.67	0.53
	6	2.84	0.47	2.87	0.48
	7	2.95	0.42	3.03	0.43
256	1	1	1	1	1
	2	1.42	0.71	1.44	0.72
	3	1.69	0.56	1.69	0.56
	4	1.84	0.46	1.85	0.46
	5	1.97	0.39	1.96	0.39
	6	2.00	0.33	2.04	0.34
	7	2.07	0.29	2.11	0.30
300	1	1	1	1	1
	2	1.76	0.88	1.75	0.86
	3	2.35	0.78	2.34	0.78
	4	2.90	0.73	2.81	0.70
	5	3.29	0.66	3.18	0.64
	6	3.49	0.58	3.51	0.58
	7	3.83	0.55	3.78	0.54

From the above tables we have seen that several runs for different sized determinant is made. Table 5.1 is presented the predicted timings which was calculated from the formula (3) and the corresponding the experimental timings. It is noteworthy to point out that performance improves with increasing the number of processors. In table 5. 2, we give the predicted speed up and efficiency and corresponding for the experimental.

We observe that our theoretical estimate of the speed up and efficiency predicts accurately the performance of the algorithm. This occurs because the sizes of the determinants are very large. In practice if the size of the determinants have the same number of processors we might have a poor speed up and efficiency [10].

V. CONCLUSION

In this paper we have developed a parallel algorithm for computing the determinants in the multiprocessor computer. The advantages of the algorithm are ease of implementation and good suitability for parallel computing. It is designed to handle large determinants whose storage requirements cannot fit on a single computer. Also we can conclude that our proposed algorithm reaches good efficiency even with increasing of the number of processors.

DOI: 10.15680/IJIRSET.2014.0308062

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2014

VI. ACKNOWLEDGMENTS

The author is grateful for financial support from the University of Bahrain, the project No. 2012/55.

REFERENCES

[1] Cvetanovic,Z,"The Effect of Problem Partitioning Allocation and Granularity on the Performance of Multiprocessor System", IEEE Trans. Computers, vol. C-36, pp. 42-432, 1987.

[2] Gannon,D. B. and Van Rosendale,J, "On the Impact of the Communication Complexity on Design of Parallel Numerical Algorithm, " IEEE Trans. Computers, vol. C-33, pp. 1180-1194, 1984.

[3] Lundsrom,S. L, "Application Consideration in the System Design of Highly Concurrent Multiprocessors", IEEE Trans. Computers, vol. C-36, pp. 1292-1309, 1987.

[4] Greiner,J. and Blelloch,G. E, "Connected Components Algorithms". In Sabot,G.W, editor, High Performance Computing: Problem Solving with Parallel and Vector Architectures. Addison Wesley, Reading , MA, 1995.

[5]Ladd, S., "Guide to Parallel Programming", Springer – verlag, 2004.

[6] Kopta,D. and Davis,A, "Efficient MIMD Architectures for High – Performance Ray Tracing", Int. Rep., University of Utah, School of Computing, 2009.

[7] Abbas,Salman H.,"Complexity of Parallel Block Gauss Jordan Algorithm", Journal of Association of Arab Universities for Basic and Applied Science, vol. 7, pp.57-65, 2009.

[8] Abbas,Salman H.,"On the Cost of Sequential and Parallel Algorithm for Solving Linear System of equations", Int. J. of Computer Mathematics, vol. 74, Issue 3, pp.391-403,2000.

[9] Abbas,Salman H.,"Parallel Solution of Linear System of Equations using a Column Oriented Approach", Chaos, Soliton& Fractals, vol. 30, Issue 4, pp. 864-874, 2006.

[10] Abbas,Salman H.,"Parallel solution of Linear System of Equations on Transputer Array", Applied Mathematics & Information Sciences vol. 2, No. 2, pp. 225-235, 2008.

[11] Beliakov,G. and Matiyasevich,Y.,"A Parallel Algorithm for Calculating of Large Determinants with Accuracy for GPUS and MPI Clusters" ariXiv.1308.1536v2,Accessed Augest 2013.

[12] Kamgin,E.,Nguenange,L. and Phillippe,B.,"Some Efficient Methods For Computing the Determinants of Large Sparse Matrices",ARIMA, Volume xx,pp.1-10,2012.

[13]Haque,S and Maze,M.,"Determinant Computation on the GPU using condensation Method", Technical Report,pp.1-10 ,University of western Ontario,London N6A 1MA ,Canda,2012.

[14] Saliha,A.,"A New Method to Calculate the Determinants of *nxn* Matrix by reducing determinants to second order", International Conference on Applied Analysis and Algebra,YildizTechnocal University ,20-24 June 2012,Istanbul,Turkey.

DOI: 10.15680/IJIRSET.2014.0308062