VICTORIA UNIVERSITY OF TECHNOLOGY

DEPARTMENT OF COMPUTER AND MATHEMATICAL SCIENCES

A Parallel Algorithm for the
Calculation of Kronecker Products /
Sums on Hypercube Processor Topologies

Roderick White
Charles Osborne (Monash University)

(33 COMP 7)
November 1993

TECHNICAL REPORT

VICTORIA UNIVERSITY OF TECHNOLOGY
(P O BOX 14428) MELBOURNE MAIL CENTRE
MELBOURNE, VICTORIA, 3000
AUSTRALIA

TELEPHONE (03) 688 4249 / 4492
FACSIMILE (03) 688 4050

Footscray Campus

A PARALLEL ALGORITHM FOR THE CALCULATION OF KRONECKER PRODUCTS/SUMS ON HYPERCUBE PROCESSOR TOPOLOGIES

Roderick White
Charles Osborne

ABSTRACT

Kronecker algebra is widely used in the field of reliability. Calculation of Kronecker products and sums is a cumbersome exercise requiring calculations of order $O(N^4)$. This paper shows that Kronecker products/sums can be viewed as simple algebraic operations performed on the elements residing on (n,b,k)-hypercubes.This technique provides a simple algorithm for the parallel calculation of Kronecker products and sums using hypercube processor topologies. This method minimises the algorithm complexity usually associated with parallel architectures as well as providing calculation speedup.

1. INTRODUCTION

Kronecker algebra is widely used in the field of reliability when modelling the behaviour of systems. Both Kronecker product and Kronecker sum operations are extensively employed when building transition rate matrices (Amoia[1],Cafaro[2]).

A characteristic of Kronecker algebra is that the size of calculations grows rapidly, given a matrix **A** of size (m x m) and a matrix **B** of size (p x p) the Kronecker product or sum of the two matrices will be of size (mp x mp). Modelling a realistic system requires a large number of Kronecker operations, which results in a computationally intensive task handling large matrices. Applications such as this can benefit from a parallel processing algorithm which decreases processing time without adding extra complexity to the program implementation.

2. KRONECKER PRODUCTS

Given a matrix **A** of size $m \times n$ and a matrix **B** of size $p \times q$ the Kronecker product is a matrix of size $mp \times nq$ determined as shown by (1)

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & . & . & . & a_{1n}\mathbf{B} \\ . & & & & . \\ . & & & & . \\ . & & & & . \\ a_{m1}\mathbf{B} & . & . & . & a_{mn}\mathbf{B} \end{bmatrix}. \tag{1}$$

Rao[5] shows that given (1) the following results

$$\mathbf{A}_1\mathbf{A}_2 \otimes \mathbf{B}_1\mathbf{B}_2 = (\mathbf{A}_1 \otimes \mathbf{B}_1)(\mathbf{A}_2 \otimes \mathbf{B}_2). \tag{2}$$

where $\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}_1, \mathbf{B}_2$ are matrices of the type defined above. Choosing $\mathbf{A}_1 = \mathbf{I}_\mathbf{A}, \mathbf{B}_2 = \mathbf{I}_\mathbf{B}$ gives

$$\mathbf{I}_\mathbf{A}\mathbf{A} \otimes \mathbf{B}\mathbf{I}_\mathbf{B} = \mathbf{A} \otimes \mathbf{B} = (\mathbf{I}_\mathbf{A} \otimes \mathbf{B})(\mathbf{A} \otimes \mathbf{I}_\mathbf{B}). \tag{3}$$

Kronecker products are thus expressable as the product of Kronecker products of the original matrix and the appropriate identity matrix. Thus repeated Kronecker products can be calculated as matrix products by implementing (3) recursively.

Equation (3) can be viewed as the product of two matrices which have undergone a restructuring operation. An alternative method is available for effecting the matrix restructuring performed by the identity-matrix Kronecker products shown in (3). The Kronecker product can be expressed as

$$\mathbf{A} \otimes \mathbf{B} = \begin{cases} \mathbf{B}^{*}\mathbf{A}^{*} & p \geq m \\ \mathbf{A}^{\#}\mathbf{B}^{\#} & p < m \end{cases} \tag{4}$$

where $\mathbf{B}^{*} = \mathbf{B}$ for $m = 1$, $\mathbf{A}^{*} = \mathbf{A}$ for $q = 1$, $\mathbf{B}^{\#} = \mathbf{B}$ for $n = 1$, $\mathbf{A}^{\#} = \mathbf{A}$ for $p = 1$

given that $\mathbf{A}^{*}, \mathbf{B}^{*}, \mathbf{A}^{\#}, \mathbf{B}^{\#}$ are restructured forms of the original matrices **A** and **B**.

Equation (4) shows that the Kronecker product can be viewed as two distinct operations, a matrix mapping or restructuring operation and an algebraic operation. If the matrices can be restructured appropriately the Kronecker product can be reduced to a matrix multiplication.

Where restructuring is required the structure of the matrices is determined by altering the binary representation of the matrix row and column indices by adding a new "dummy" variable with the required number of bits to the row and column indices to create a matrix of the same size as the Kronecker product resultant matrix. This is illustrated with the following example.

Example 1:

Given $\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ $\quad \mathbf{B} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$

the row and column indices of matrix **B** can be represented in binary form as $R_0 = 0 \rightarrow 1$, $C_0 = 0 \rightarrow 1$ as shown in table 1. Addition of the dummy bit x_0 to the row and column indices $x_0R_0 = 00 \rightarrow 11$, $x_0C_0 = 00 \rightarrow 11$ creates the matrix $\mathbf{B}^{*}$ as shown in table 2.

Expressing these tables in matrix form gives

$$\mathbf{B} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \qquad \mathbf{B}^{*} = \begin{bmatrix} b_{11} & b_{12} & 0 & 0 \\ b_{21} & b_{22} & 0 & 0 \\ 0 & 0 & b_{11} & b_{12} \\ 0 & 0 & b_{21} & b_{22} \end{bmatrix}.$$

The matrix $\mathbf{A}^*$ is formed by moving the dummy bit one place to the right in the row and column indices to give R_0x_0, C_0x_0. Expressing this in matrix form gives

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \qquad \mathbf{A}^* = \begin{bmatrix} a_{11} & 0 & a_{12} & 0 \\ 0 & a_{11} & 0 & a_{12} \\ a_{21} & 0 & a_{22} & 0 \\ 0 & a_{21} & 0 & a_{22} \end{bmatrix}.$$

The product of these restructured matrices is $\mathbf{B}^*\mathbf{A}^* = \mathbf{A} \otimes \mathbf{B}$.

When p<m the restructuring uses the same technique as for the case $p \geq m$ with the exception that the position of the dummy bits added to the column index are reversed in position. For example given

Example 2:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} b_{11} & b_{12} \end{bmatrix}$$

The matrix $\mathbf{B}^\#$ has the row index R_0x_0, and the column index x_0C_0, this gives the matrix

$$\mathbf{B}^\# = \begin{bmatrix} b_{11} & b_{12} & 0 & 0 \\ 0 & 0 & b_{11} & b_{12} \end{bmatrix}$$

The Kronecker product can then be given as $A \otimes B = AB^\#$. For the case of square matrices of size 2^n the above method can be expressed as

$$\mathbf{C}_1 \otimes \mathbf{C}_2 \otimes \ldots \otimes \mathbf{C}_n = \prod_{i=n}^{1} \mathbf{C}_i^* \qquad (5)$$

where the binary row-column indices restructuring is given by

Matrix	Row index	Column index
C_n	$x_j \ldots x_0 R_i \ldots R_0$	$x_j \ldots x_0 C_i \ldots C_0$
C_2	$x_j R_i \ldots R_0 \ldots x_0$	$x_j C_i \ldots C_0 \ldots x_0$
C_1	$R_i \ldots R_0 x_j \ldots x_0$	$C_i \ldots C_0 x_j \ldots x_0$

Kronecker products can be reduced to a matrix multiplication of matrices which have been restructured by the addition of an independant dummy variable to the row and column index values of the matrix elements. This can be taken a step further. The introduction of two independant dummy variables in the matrix restructuring results in the algebraic operations which have to be performed on the matrices being reduced to a simple element-element multiplication.

Example 3.

$$\mathbf{A}=\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \qquad \mathbf{B}=\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

$$\mathbf{A}\otimes\mathbf{B}=\mathbf{B}^{*}.\mathbf{A}^{*}$$

given $\mathbf{B}^{*}_{\text{row index}}=x_1R_0$, $\mathbf{B}^{*}_{\text{column index}}=x_0C_0$, $\mathbf{A}^{*}_{\text{row index}}=R_0x_1$, $\mathbf{A}^{*}_{\text{column index}}=C_0x_0$, and where the dot product operator represents the multiplication of corresponding matrix elements. This element mapping results in the matrices given in (6)

$$\mathbf{A}\otimes\mathbf{B}=\begin{bmatrix} b_{11} & b_{12} & b_{11} & b_{12} \\ b_{21} & b_{22} & b_{21} & b_{22} \\ b_{11} & b_{12} & b_{11} & b_{12} \\ b_{21} & b_{22} & b_{21} & b_{22} \end{bmatrix} . \begin{bmatrix} a_{11} & a_{11} & a_{12} & a_{12} \\ a_{11} & a_{11} & a_{12} & a_{12} \\ a_{21} & a_{21} & a_{22} & a_{22} \\ a_{21} & a_{21} & a_{22} & a_{22} \end{bmatrix}. \qquad (6)$$

3. KRONECKER SUMS

The Kronecker sum of $\mathbf{A}$ $(m\times m)$ and $\mathbf{B}$ $(p\times p)$ is defined as

$$\mathbf{A}\oplus\mathbf{B}=(\mathbf{A}\otimes\mathbf{I}_p)+(\mathbf{I}_m\otimes\mathbf{B}). \qquad (7)$$

The similarity between equations (7) and (3) indicates that Kronecker sums can be treated in a similar manner to Kronecker products, thus Kronecker sums can be expressed as

$$\mathbf{C}_1\oplus\mathbf{C}_2\oplus\ldots\oplus\mathbf{C}_n=\sum_{i=1}^{n}\mathbf{C}_i^{*} \;. \qquad (8)$$

4. HYPERCUBE IMPLEMENTATION

The matrix restructuring operations can be given a geometric interpretation by combining the matrix row-column indices to give the address of the corresponding matrix element in a geometric structure. For example the binary representations of the row and column indices of matrix **B** in the first example can be considered as the addresses of the elements on a two-dimensional cube.

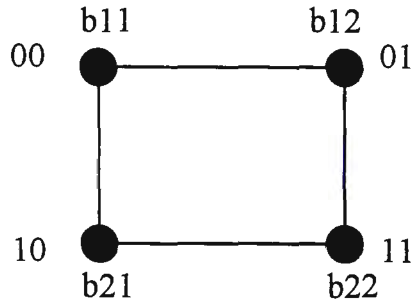


Figure 1. Hypercube representation of matrix **B**.

From this viewpoint the matrix restructuring in this example represents a mapping of the matrices **A** and **B** onto a four dimensional base 2 hypercube, and the matrix multiplications required to determine the Kronecker product correspond to data transfers and multiplication of matrix elements or vertex values.

The introduction of the dummy variable provides a partial mapping to the hypercube, requiring a matrix multiplication to perform the Kronecker product. If the matrices **A** and **B** are mapped onto the hypercube using two independant dummy variables the Kronecker product reduces to a simple multiplication of each of the elements at each hypercube vertex. This can be illustrated using example 3. Combining the row-column indices of the matrix elements and mapping the elements onto a four dimensional hypercube results in the structure given in figure 2.

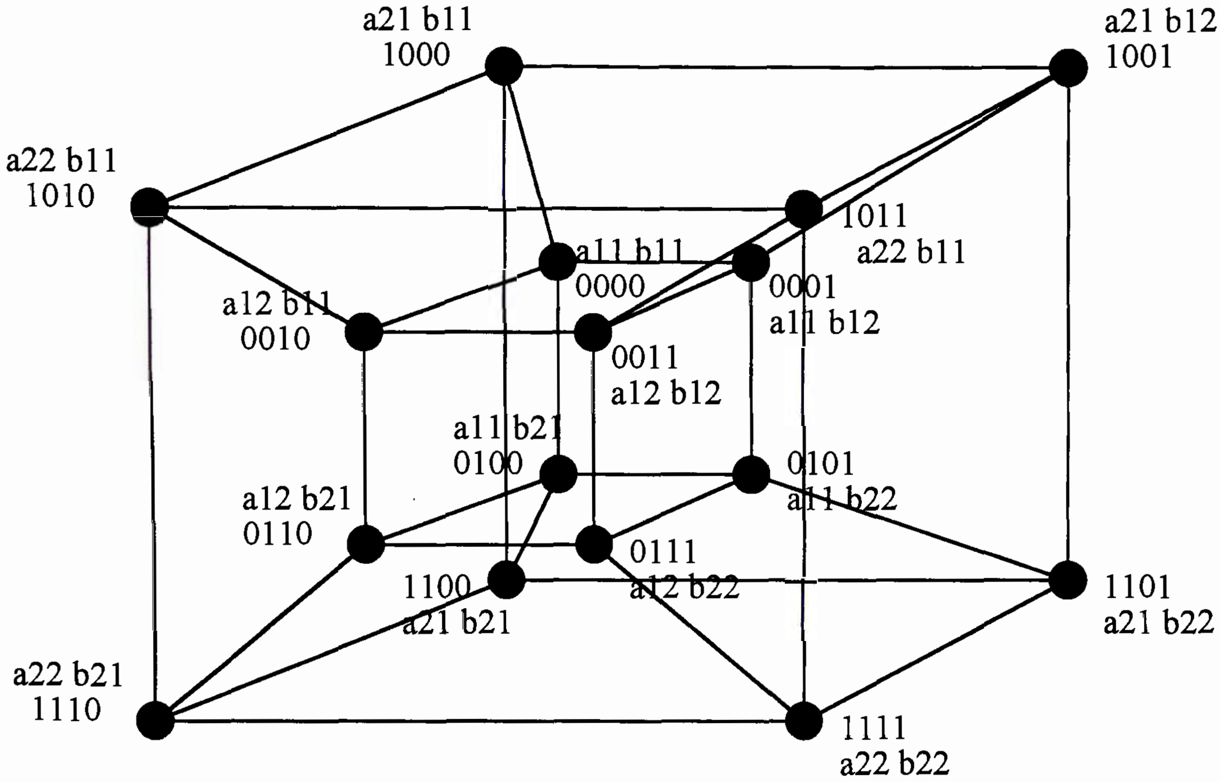


Figure 2. Mapping of matrix elements to a four dimensional hypercube.

The Kronecker product can now be found by multipliying the elements residing on each node of the cube.

Kronecker products of matrices of size other than $2^a \times 2^b$ can be represented as elementwise multiplications on (n,b,k) cubes (Laksmivarahan[4]) of which the base 2 complete hypercube is a member.

Similarly Kronecker sums can be mapped to hypercube structures, the only variation in the operation being that the elements mapped to cube nodes are added rather than multiplied. These algorithms could be easily ported to either MIMD or SIMD architectures

If the number of processors does not match the size of the resultant matrix each node may be loaded with more element operations by mapping onto the hypercube the restructured matrices determined for the matrix multiplication technique.

Example 4

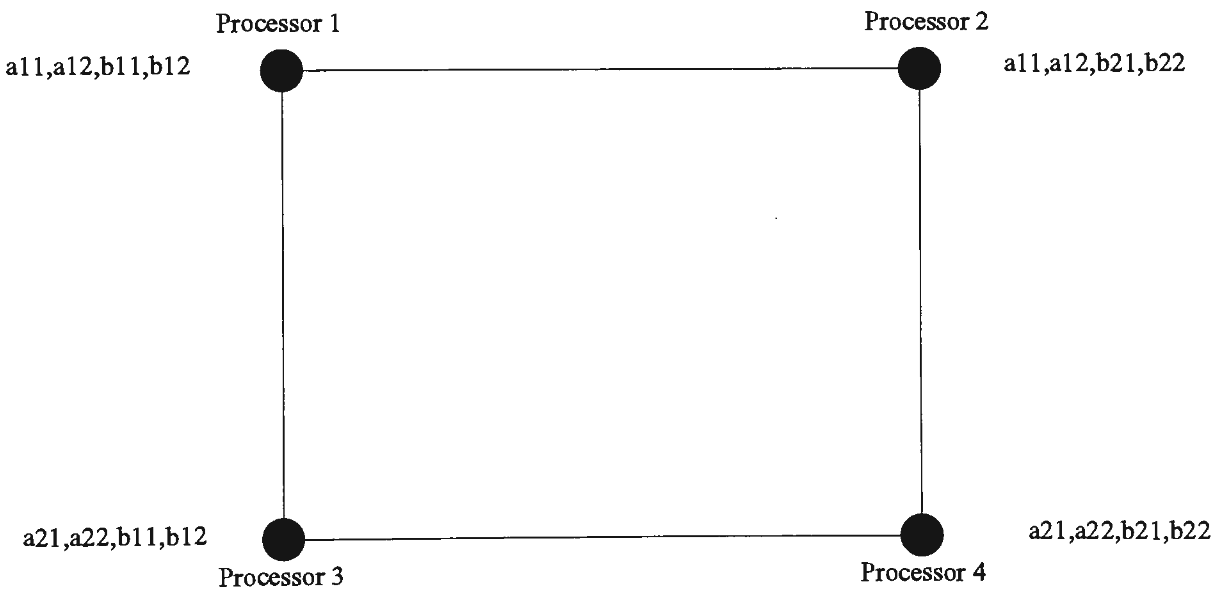
Calculating a Kronecker product on a four node hypercube

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

$$\mathbf{A} \otimes \mathbf{B} = \mathbf{B}^{*}\mathbf{A}^{*}$$

The matrix rows are divided between the available hypercube processors

$$\begin{bmatrix} b_{11} & b_{12} & 0 & 0 \\ \hdashline b_{21} & b_{22} & 0 & 0 \\ \hdashline 0 & 0 & b_{11} & b_{12} \\ \hdashline 0 & 0 & b_{21} & b_{22} \end{bmatrix} \begin{bmatrix} a_{11} & 0 & a_{12} & 0 \\ \hdashline 0 & a_{11} & 0 & a_{12} \\ \hdashline a_{21} & 0 & a_{22} & 0 \\ \hdashline 0 & a_{21} & 0 & a_{22} \end{bmatrix} \begin{matrix} \text{Processor 1} \\ \text{Processor 2} \\ \text{Processor 3} \\ \text{Processor 4} \end{matrix}$$



Multiplication of elements on each processor results in the following resultant matrix distributed over the hypercube.

$$\begin{bmatrix} a_{11}b_{11} & a_{11}b_{12} & a_{12}b_{11} & a_{12}b_{12} \\ \hdashline a_{11}b_{21} & a_{11}b_{22} & a_{12}b_{21} & a_{12}b_{22} \\ \hdashline a_{21}b_{11} & a_{21}b_{12} & a_{22}b_{11} & a_{22}b_{12} \\ \hdashline a_{21}b_{21} & a_{21}b_{22} & a_{22}b_{21} & a_{22}b_{22} \end{bmatrix} \begin{matrix} \text{Processor 1} \\ \text{Processor 2} \\ \text{Processor 3} \\ \text{Processor 4} \end{matrix}$$

Kronecker sums can be calculated in a similar manner, the matching matrix elements of each matrix resident on a processor are summed to give the row elements of the corresponding Kronecker sum matrix.

Conclusion

The Kronecker product can be determined by means of either a matrix multiplication, or a set of simple scalar multiplications dependant upon the structure of the constituent matices. Kronecker sums can be determined in a similar manner. This ability makes them amenable to parallel calculation on hypercubes utilising a simple mapping algorithm. The advantages of these techniques are that Kronecker product/sum applications can be calculated in parallel without employing application specific processor topologies and avoiding programming complexity.

5. REFERENCES

[1] V.Amoia, G. De Micheli, M. Santomauro, Computer-Oriented Formulation of Transition-Rate Matrices via Kronecker Algebra, IEEE Transactions on Reliability, Vol. R-30, No. 2, June 1981.

[2] G. Cafaro, F. Corsi, F. Vacca, Multistate Markov Models and Structural Properties of the Transition-Rate Matrix, IEEE Transactions on Reliability, Vol. R-35, No. 2, June 1986.

[3] M. Davio, Kronecker Products and Shuffle Algebra, IEEE Transactions on Computers, Vol. C-30, No. 2, February 1981.

[4] S. Laksmivarahan, S.K. Dhall, Analysis and Design of Parallel Algorithms, McGraw-Hill, 1990.

[5] C. Radhakrishna Rao, Linear Statistical Inference and its Applications, 2nd ed., John Wiley & sons, 1973.

Appendix

Table 1:

Matrix B

Row R_0	Column C_0	Matrix Element
0	0	b_{11}
0	1	b_{12}
1	0	b_{21}
1	1	b_{22}

Table 2:

Matrix B*

Row x_0R_0	Column x_0C_0	Matrix Element
00	00	b_{11}
10	10	b_{11}
00	01	b_{12}
10	11	b_{12}
01	00	b_{21}
11	10	b_{21}
01	01	b_{22}
11	11	b_{22}