A MATRIX DECOMPOSITION APPROACH TO THE CONTROL OF CLOS REARRANGEABLE SWITCHING NETWORKS

I. Sakho[a] Y. Langue[b]

[a]Ecole Nationale Supérieure des Mines - Centre SIMADE
158 Cours Fauriel, 42023 St-Etienne, France
Tel : (33) 77 42 01 66 - Fax : (33) 77 42 00 00 - Email : sakho@emse.fr

[b]RIS Technologies, 84 Rue du 1er Mars 43, 69625 Villeurbanne, France

Abstract : Communication issues remain the key for the development of Distributed Memory MIMD computers. Two main approaches prevail in the search for adequate communication paradigms : the use of static or reconfigurable interconnection networks. In this paper, we are interested in the second approach. Reconfigurable Distributed Memory MIMD computers with a large number of processors need multistage switching networks to interconnect the processors. The Clos rearrangeable switching network belongs to the most used in the industry. For this family of switching networks the literature proposes several kind of control algorithms. The decomposition of the interconnection matrix of the switches, induced by a given configuration, into permutation matrices constitutes an interesting approach.

For each permutation matrix the algorithms of this class proceed in two phases whose the second, the most costly expensive, needs for a mxm interconnection matrix at most m/2 iteration steps.

This paper discusses two modifications of these algorithms. The first results in an algorithm whose second phase needs less than m/3 iteration steps instead of m/2. When after all second phase is needed, the second modification shows how to carry out it according to the divide and conquer strategy.

Keywords : Parallel computer, interconnection network, message conveying, permutation matrix, Clos rearrangeable switching network.

I . Introduction

The experience gained with the development of parallel processing reveals that communication issues are at least as important as computation issues to obtain better performances. In order for interconnection network to be as general as possible, target applications are modeled as graphs of processes which must be mapped onto the processor's graph. The performance of the application can then be measured by the distance between the two graphs.

Distributed Memory MIMD computers with fixed interconnection network are not always well suitable to bring closer these two graphs; according to the application, in general, costly expensive communication between distant processes is the price to pay. However note that completely connected networks constitute an exception; unfortunately, they do not support a large number of processors.

Although they offer more flexibility than the previous, the bus based interconnection networks also limit the number of the processors because of the high communication contention they can induce.

Dynamic or reconfigurable interconnection networks constitute an intermediate class. Comparatively to the precedents, their interest comes from their ability to bring neighbours two distant processes by connecting directly the processors on which they are running. To interconnect a very large number of processors, these networks consist in several crossbars organised in stages and are called multistage switching networks.

In the large variety of multistage switching networks, the Clos three stages rearrangeable network [1] is one of the most attractive. Indeed it needs a low cost of crosspoints, induces a low delay and because of its rearrangeability is able to perform any permutation that is to say any processors interconnection network. Many industrial realisations of parallel computers use this kind of switching networks [2], [3], [4], etc. The price to pay for this flexibility is the cost of the control: the computation of the commands required to perform a given configuration.

Many control algorithms have been proposed in the literature; except for some particular classes of networks [5], their principles remain the same and they differ only in the formalism they use. We distinguish bipartite graph edges coloring [6], set theory [7] and matrix decomposition approaches [9]. Recently another method called scheduling was proposed [8].

The algorithms based upon matrix decomposition consist in two phases whose the second, costly expensive, is necessary only if the first does not succeed. In [10] Tsao-Wu reports a strategy for the first phase to lower the probability of the second phase. Although interesting for large switching networks, this modification does not induce a noticeable improvement of the cost of the second .

More recently, Jajszczyk [11] proposed a method for matrix decomposition which should not require a second phase. In [8] it is proved that this algorithm does not always provide information enough to solve the problem without a second phase.

This paper proposes to reduce the cost of the second phase too. In the next sections, we first introduce some definitions and generalities about the matrix decomposition approach to the calculus of the settings of Clos three stages rearrangeable switching networks. Then we show how to reduce the cost of the second phase to m/3 iteration steps instead of m/2 for a mxm interconnection matrix. We also show how the second phase could be carry out more efficiently with a divide and conquer strategy.

II . The matrix decomposition process

A Clos three stages rearrangeable switching network consists in m dxd input modules, d mxm intermediate modules and m dxd output modules. The modules are interconnected as follows: the k-th output (input) of the i-th (j-th) input (output) module is connected to the i-th (j-th) input (output) of the k-th intermediate module (see figure 1 below).

Let π be a permutation of N=md elements representing a processors interconnection. To realise π, the switching network must connect any of its inputs $i \in [0, N-1]$ to its outputs $\pi(i) \in [0, N-1]$. More precisely π induces the following interconnections between the outermost modules of the switching network: an input module I is said to be connected to an output module J if there is an input $i \in I$ such as $\pi(i) \in J$. This set of interconnections can be defined by the matrix H_π whose the element H_π (I, J) indicates the number of the connections that π induces between the modules I and J. In the following H_π will be noted simply H.

Let P be a permutation matrix extracted from H. It performs an one to one correspondence between the outermost modules hence a sub-permutation π_k of m elements of π. To realise the connections that π_k represents it suffices to transit them by one of the intermediate modules. Thus d distinct permutation matrices must be extracted from H to achieve π. Figure 2 illustrates this process for the permutation of the figure 1.

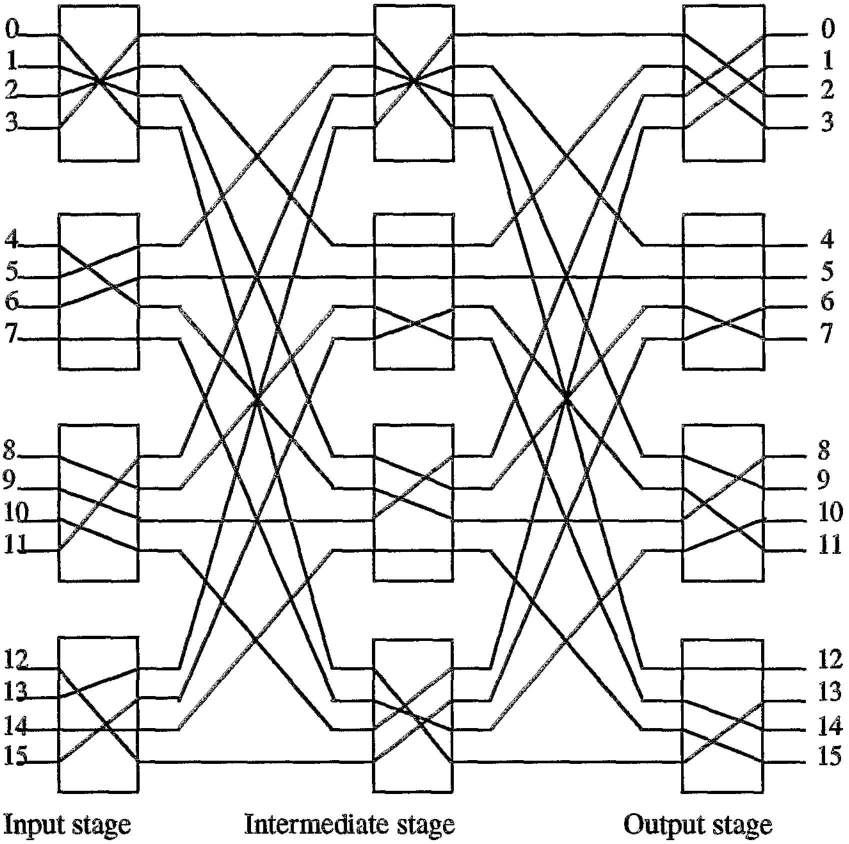


Figure 1 : A Clos three stages rearrangeable switching network configured to realise the interconnection corresponding to the permutation
$\pi = (13,7,3,12,8,9,5,10,14,1,0,4,6,2,15,11)$

$$\begin{bmatrix} 1 & 1 & 0 & 2 \\ 0 & 1 & 3 & 0 \\ 2 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Figure 2a : The interconnection matrix associated to the permutation of the figure 1.

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Figure 2b : Permutation matrices corresponding respectively to the switching commands for the first, second, third and fourth switches of the intermediate stage of the figure 1. The commands for the outermost stages are inferred.

Figure 2 : A configuring process

As described by Neiman, the extraction of each permutation matrix P_k, consist in "marking" the non-null elements of $H\text{-}\Sigma_{k<i}P_k$ such as there is exactly one by row and by column. In the following, we will use the * sign to mark a non null element. It is proved [9] that with a blind marking, for a mxm matrix, the number of elements one can mark is at least m/2.

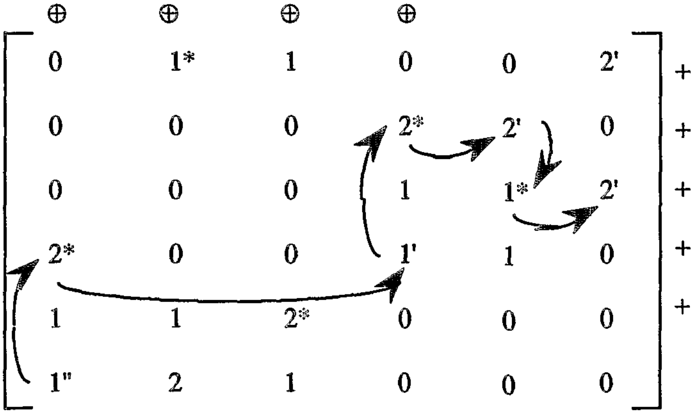
It is uncommon that this process results in a complete permutation matrix. In such a case a completion phase is necessary. Any step of this one is similar to the search, in a bipartite graph, for an alternated path whose extremities are unsatured to construct a maximum matching [12]. A systematic way to carry out the completion process, was reported by Neiman [9]. It consists in two steps. For the first step :

1 - Distinguish the columns which contain a marked element with for instance + sign.
2 - Find a non null element α in an undistinguished column.
3 - If the row of α contains a marked element β, then mark α with for instance the ' sign, undistinguish, by circling the + sign, the column of β and distinguish the row of α with + sign.
4 - Else mark α with another sign, for instance " sign.

At the second step :

1 - From an element marked with " sign and whose the row does not contain an element marked with * sign construct a sequence of non null elements marked alternatively with * sign and ' sign by moving along a column then along a row. The last element of such a sequence is marked with ' sign.
2 - Alternate on this sequence * sign with the other signs.

The figure 3 illustrates this process.



$$\begin{array}{cccccc|c} \oplus & \oplus & \oplus & \oplus & & & \\ 0 & 1^* & 1 & 0 & 0 & 2' & + \\ 0 & 0 & 0 & 2^* & 2' & 0 & + \\ 0 & 0 & 0 & 1 & 1^* & 2' & + \\ 2^* & 0 & 0 & 1' & 1 & 0 & + \\ 1 & 1 & 2^* & 0 & 0 & 0 & + \\ 1'' & 2 & 1 & 0 & 0 & 0 & \end{array}$$

Figure 3 : The process for building a stretcher.

This process could be very expensive because it requires an exhaustive search and furthermore, the sequences which allow to increase the number of the marked elements called *stretcher* in the following are longer than necessary.

To lower the probability of this process, Tsao-Wu [10] proposes to modify the first phase. Instead of blind marking, he proposes the following strategy for the first phase :

1- In each column of H mark the largest unmarked element.
2- After an element H(i, j) has been marked, choose the next column k such as H(i, k) be the largest unmarked element in the i-th row.

He then shows that for a switching network with dxd outermost modules, the number of the marked elements x holds $x \geq dm/(2d-2)$. Asymptotically, this does not induce no noticeable improvement of the lower bound of the number of marked elements.

A more interesting marking strategy was reported by Jajszczyk [11]. This one consists in marking the non null element whose the row contains the largest number of zeros. In [13], this strategy has been modified. The new strategy consists in marking the non null elements H(i*, j*) such as firstly the row (column) i* (j*) contains the largest number of zeros and then the column (row) j* (i*) also contains the largest number of zeros among all the column (row) j (i) that verify H(i*, j) (H(i, j*)) ≠ 0.

This modification results in an algorithm which in some cases does not need a completion phase.

$$\left[\begin{array}{cccccc} 0 & 1 & 1 & 0 & 0 & 2^* \\ 0 & 0 & 0 & 2^* & 2 & 0 \\ 0 & 0 & 0 & 1 & 1^* & 2 \\ 2^* & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 2^* & 0 & 0 & 0 \\ 1 & 2^* & 1 & 0 & 0 & 0 \end{array}\right] \begin{array}{c} 4 \\ 1 \\ 2 \\ 3 \\ 5 \\ 6 \end{array}$$

Figure 4 : A modified Jajszczyk's strategy.
The numbers on the right correspond to the marking order.

III . Some improvements

In this section, we will deal with the improvements of the completion process described earlier. To that purpose suppose that the modified Jajszczyk's strategy has been used unsuccessfully. Let x be the number of the marked elements. There are two permutation matrices L and C such as

$$LHC = \begin{bmatrix} H_1 & H_2 \\ H_3 & H_4 \end{bmatrix}$$

where the diagonal of H_1 consists in all the marked elements, H_2 and H_3 are both non null and $H_4 = 0$.

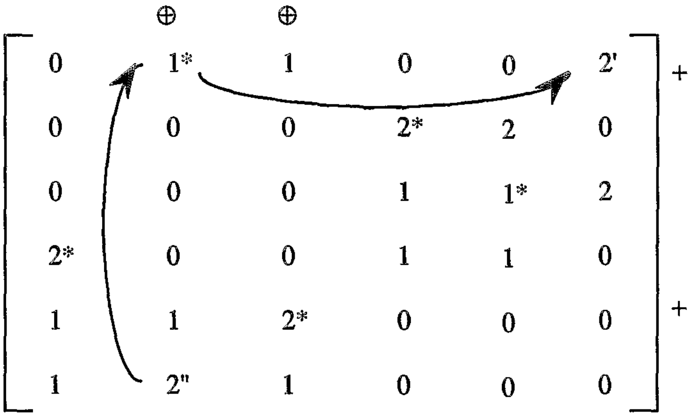
Let $i \in [1, x]$ such as $H_2(i, .)$, the i-th row of H_2 and $H_3\ (., i)$, the i-th column of H_3 are both non null. The second step of the completion process can take place because there are k_1 and $k_2 \in [x+1, m]$ such as the sequence $(H_2(i, k_1)\ H_1(i, i), H_3(k_2, i)\ )$ be a stretcher. By iterating this process while there is such a sequence, it follows that for any $i \in [1, x]$, $H_2(i, .) = 0$ or $H_3\ (., i) = 0$. From where :

Proposition : There is $t \in [1/2, 1]$ such as $x \geq m/(2\text{-}t)$.

Proof : Let $S_x(H_1) = \Sigma_{1 \leq i \leq m} \Sigma_{1 \leq j \leq m} H_1(i, j)$. One verifies easily that $S_x(H_1) = (2x - m)\ d$ as $\Sigma_{1 \leq i \leq m} H_1(i, j) = \Sigma_{1 \leq j \leq m} H_1(i, j) = d$.
Let t be the maximum of the proportion of the non null rows of H_2 and the proportion of the non null columns of H_3. As $H_2(i, .)$ and $H_3(., i)$ for any $i \in [1, x]$ are such as at least one of them is null then $t \geq 1/2$ and we have

$S_x(H_1) \geq (t(d-1) + 1)x$.
As furthermore $S_x(H_1) = (2x - m)d$ it follows that $x \geq md/((2-t)d + t - 1)$ which asymptotically converges towards $m/(2-t)$.

The figure below illustrates this proposition. We can see that a stretcher with length 3 would suffice to complete the number of the marked elements.



$$\begin{array}{c|cccccc|c} & & \oplus & \oplus & & & & \\ \hline & 0 & 1^* & 1 & 0 & 0 & 2' & + \\ & 0 & 0 & 0 & 2^* & 2 & 0 & \\ & 0 & 0 & 0 & 1 & 1^* & 2 & \\ & 2^* & 0 & 0 & 1 & 1 & 0 & \\ & 1 & 1 & 2^* & 0 & 0 & 0 & + \\ & 1 & 2'' & 1 & 0 & 0 & 0 & \end{array}$$

Figure 5

Now let us generalise this process. Consider that the number of the null columns of H_3 is larger than the number of null rows of H_2. Let $(r(i)\ ;\ 1 \leq i \leq p)$ be these column numbers.

Denote :

$H_1{}^{(i)}$ the diagonal block $(H_1(a,b)\ ;\ a, b \in [r(i-1), r(i)])$ where $r(0) = 1$,
$H_2{}^{(i)}$ be the block $(H_2\ (a,b)\ ;\ a \in [r(i-1), r(i)]\ , b \in [x+1, m])$,
$H_3{}^{(i)}$ be the block $(H_3\ (a,b)\ ;\ a \in [x+1, m], b \in [r(i-1), r(i)])$,
(see figure 6).

If $H_2{}^{(i)}$ is non null then there are two probable stretchers whose extremities belong to $H_2{}^{(i)}$ and $H_3{}^{(i)}$.

More precisely, according to the process for building a stretcher, the non null elements with the ' sign of each probable stretcher belong either to the lower or to the upper triangular part of $H_1{}^{(i)}$.

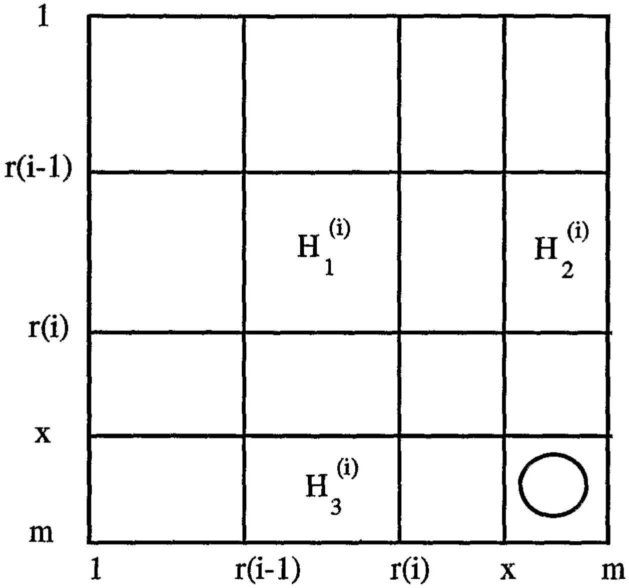


Figure 6

We can now apply the completion process on each ($H_2^{(i)}$, $H_1^{(i)}$, $H_3^{(i)}$). Unlike the case where $H_1^{(i)}$ wass reduced to one marked element, two situations can arise.

- There is $i \in [1, p]$ such as the completion process succeeds. Then permute the rows of the related stretchers such as the new marked elements be on the diagonal of H_1 and repeat the process.

- All the completion processes fail. Then extend $H_3^{(i)}$ to its next non null column ; $H_1^{(i)}$ and $H_2^{(i)}$ are so extended too; apply the completion process on each of the new sequences ($H_2^{(i)}$, $H_1^{(i)}$, $H_3^{(i)}$).

In the worst cases ($H_2^{(i)}$, $H_1^{(i)}$, $H_3^{(i)}$) converges towards the sequence (H_2, H_1, H_3); then the process is similar to the one described earlier which is known to be successful.

Otherwise at each step we can expect to mark more than one new non null element with shorter stretchers.

This divide and conquer approach to the completion process should be analysed carefully as it can result in a parallel algorithm. Indeed we can imagine that for each ($H_2^{(i)}$, $H_1^{(i)}$, $H_3^{(i)}$) the search for the probable stretchers be allocated to a distinct processor.

IV . Conclusion

This paper reports some modifications to the algorithms for decomposing in permutation matrices the interconnection matrix associated to a configuration of a Clos three stages rearrangeable switching network. It improves the results of similar studies. Indeed we have proved that if the completion phase remains necessary then, for a mxm interconnection matrix, it needs strictly less than m/3 iteration steps instead of m/2.

When after all the completion phase is needed we show that the cost of this latter can be reduced by a divide and conquer strategy. This approach when analysed carefully should induce a parallel algorithm. This will be the next step of our study.

To carry out these improvements, mainly the one based upon the divide and conquer strategy, the marked elements must constitute the diagonal of H_1. This seems to be expensive. In fact, we just need a circular permutation of the rows of H which contain the elements of the stretchers .

We focused on the use of reconfigurable networks for producing interconnection schemes. While other techniques are available, this one represents the better adequation between an application processes graph and a processors graph. However, the associated control is generally complex, especially if a dynamic behaviour is required.

As much effort was dedicated to build computation processors, a great deal of efforts must be devoted to producing communication processors which could handle such a complex control and execute the distributed algorithm required to build communication schemes.

V. References

[1] C. Clos : "A study of non blocking switching networks", The Bell system technical journal, March 1953.

[2] J. Beetam, M. Denneau, D. Weigarten : The GF11 supercomputer", Proc. 12th annual International Symposium on computer architectures, IEEE 1985.

[3] V. P. Bhatkar : "Parallel Computing. An indian perspective", Proc. of CONPAR 90 - VAPP IV, LNCS 457, Springer-Verlag, H. Burkhart (Ed).

[4] P. Waille, T. Muntean : "Introduction à l'architecture des machines supernodes", La lettre du Transputer, n°7 1990.

[5] S. Andresen : "The looping algorithm extended to 2^t rearrangeable switching networks", IEEE Trans. on Comp., Vol c-25, 1977, pp 1057-1063.

[6] H. N. Gabow : "Using Euler Partitions to edge color bipartite multigraph", International Journal of Computer and Information Sciences, Vol 5 n°4 1976, pp 345-355.

[7] H. R. Ramanujam : "Decomposition of permutation networks", IEEE Trans. on Comp., Vol c-22, July 1973, pp 639-643.

[8] J. Gordon, S. Sritkanthan : "Novel algorithm for Clos-type networks", Electronics lettres, Vol 26 n°21 October 1990, pp 1772-1774.

[9] V. I. Neiman : "Structures et commandes optimales de réseaux sans blocage", Annales des Telecom, Juillet-Aout 1969.

[10] Tsao-Wu : "On the Neiman's algorithm for the control of rearrangeable switching networks", IEEE Trans. on Comp., Vol c-22, June 1974, pp 737-742.

[11] A. Jajszczyk : "A simple algorithm for the control of rearrangeable switching networks", IEEE Trans. COM-33, pp 169-171.

[12] C. Berge : "Graphes", Galuthiers Villars 1983.

[13] I. Sakho : "Control of clos rearrangeable switching networks : on the Neiman's algorithm", to appear in the proceedings of SMS TPE 94, Moscow, 21-23 September 1994.