EXAMINATION OF AN ALGORITHM FOR NON CONFLICT SCHEDULE WITH DIAGONAL ACTIVITION OF JOINT SUB MATRICES IN A LARGE SCALE SWTCHING MATRIX

Kiril Kolchakov, Vladimir Monov

Institute of Information and Communication Technologies, Bulgarian Academy of Sciences,
Acad. G. Bonchev Str., Sofia, Bulgaria,
kkolchakov@abv.bg , vmonov@iit.bas.bg.

Keywords: Network nodes, Message switching Node traffic, Crossbar switch, Conflict elimination, Packet messages.

Abstract: The aim of this study is to determine the influence of the size of submatrices in a lagre scale connections matrix on the performance of an algorithm for non conflict scheduling with diagonal activation of joint submatrices (ADAJS). We have used a software model (SMADAJS) of the algorithm developed in MATLAB programming environment. A comparison is made with similar algorithms with diagonal activation of the connections matrix with respect to performance time and memory required.

1 INTRODUCTION

The traffic via Crossbar switching nodes is casual and depends on the users. The formulation of a conflict issue during operation of the switching nodes is as follows. The dimensions of switches in the switching nodes are N x N, where N sources of packet messages are connected to N receivers via the switch of the switching node. The traffic is random by nature and conflicts are available in the following two cases:

- When one source of message requests communication to two or more message receivers
- When one message receiver receives communication requests from two or more message sources.

The evasion of conflicts is directly related to the switching node performance.

The status of the switch of the switching node is represented with the so called connection matrix. For N x N dimensional switch the dimension of the connection matrix T is N x N also, where every element $T_{ij} = 1$ if the connection request from i- source to j- receiver exists. In the opposite case $T_{ij} = 0$.

A conflict situation arises if any row of the connection matrix has more than a single 1, which corresponds to the case when one source requests a connection with more than one receiver. The presence of more than a single 1 in any column of the matrix T also indicates a conflict situation, it means that two or more sources have requested a connection with the same receiver (Kolchakov K, Monov V, 2013).

The problem of conflict situations is solved by compiling a non-conflict schedule. Algorithms using different approaches to obtain non-conflict schedules are described in (Tashev T. , 2010). Algorithms with diagonal activation of connections matrix to obtain non-conflict schedule are described in (Kolchakov K., Monov V., 2013, Kolchakov K., 2013, Kolchakov K., 2012). Software models to obtain a non-conflict schedule through the sparse matrix-masks are described in (Kolchakov K., 2009). In tis paper, we study the performance of an algorithm for non conflict scheduling with diagonal activation of joint submatrices (ADAJS) by using a software model (SMADAJS) of the algorithm developed in MATLAB programming environment.

2 DESCRIPTION OF THE ALGORITHM

The connections matrix T with N x N size, where N is being the degree of two, is divided into sub matrices (S) with dimension n x n, (n also is a degree of two), i.e:

$$T = [\ S\,i\,j\ ],\ \ i = 1\text{-}\ n, j = 1\text{-}\ n$$

The sets of sub matrices located along the main diagonal are processed simultaneously in each of the diagonals. For submatritces in diagonals parallel to the main one, the principle of reconciliation is used (Kolchakov K, Monov V, 2013).

The idea of synthesis of the algorithm ADAJS (Algorithm with diagonal activations of joint sub-switching matrices) is based on the knowledge that the diagonal sub matrices with requests for service in the matrix T are non-conflict in the diagonal where they are located. There are diagonals with sub matrices of requests that are non-conflict to one another. Figure 1 shows joint couples of non-conflict diagonals with sub matrices of requests for service and the main diagonal of sub matrices that can not be jointed with anyone else (Kolchakov K, Monov V, 2013).

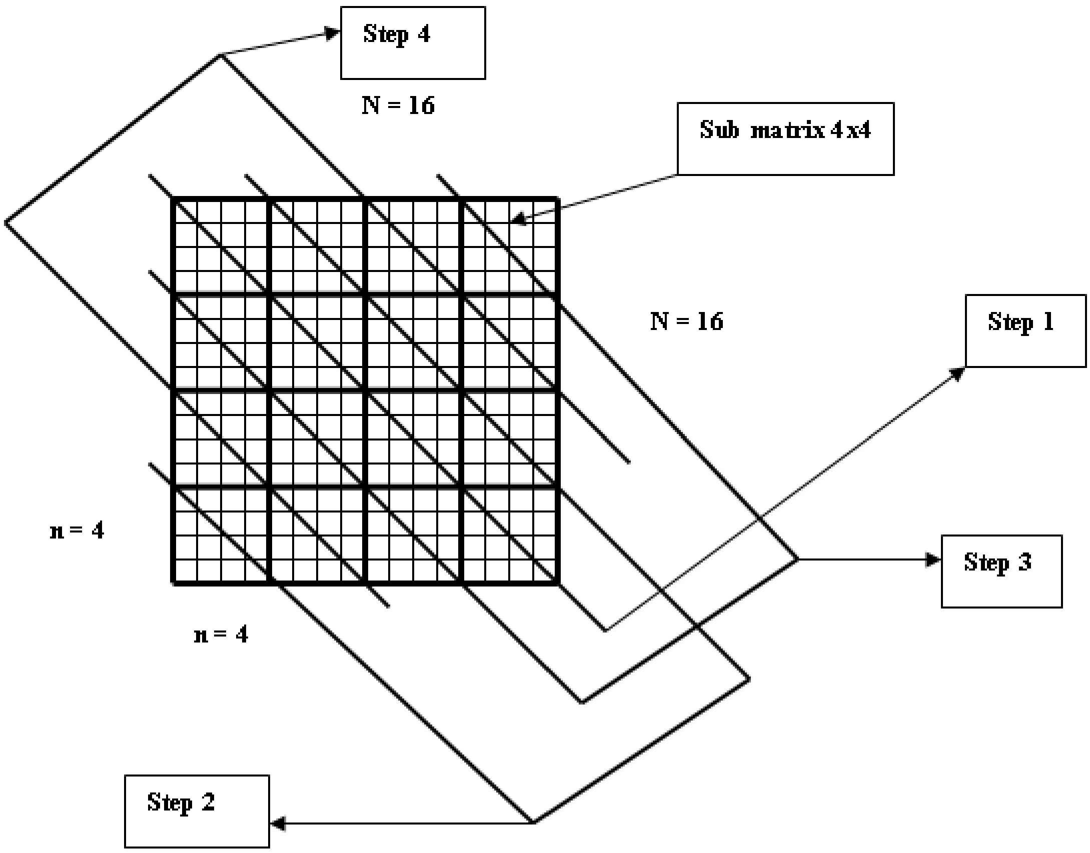

Figure 1: Diagonal activation of joint sub matrices.

The whole process of the implementation of ADAJS algorithm for obtaining a non-conflict schedule is divided into steps. The first step refers to the main diagonal sub matrices processed simultaneously and without conflict. The next steps are related to the reconciliation of the diagonals parallel to the main diagonal by pairs (Figure 1).

The analytical description of the steps shown in Figure 1 is as follows:

Step1 : S_{11}, S_{22}, S_{33}, S_{44} Step3 : S_{21}, S_{32}, S_{43}, S_{14}

Step2 : S_{41}, S_{12}, S_{23}, S_{34} Step4 : S_{31}, S_{42}, S_{13}, S_{24}

$T = [\, S_{ij} \,]$, $i = 1 - 4$, $j = 1 – 4$

The size (n) of the sub matrix determines the number of steps (I) as follows

$$I=N/n \qquad (1)$$

For **N** = const., I = f (n), where $1 < n \le N / 2$.

Table 1: Optimal values of the size n.

N	1024	2048	4096	16384
$n_{opt.}$	8	8	4	8
N	32768	65536	131072	262144
$n_{opt.}$	8	8	4	4
N	524288	1048576	-	-
$n_{opt.}$	4	8	-	-

3 TEST RESULTS WITH ADAJS FOR LARGE VALUES OF N

The software model SMADAJS, describing the algorithm ADAJS is written in MATLAB programming language. Our study of SMADAJS software model is performed on Dell OPTIPLEX 745 (Core 2 Duo E6400 2,13GHz, RAM 2048).

Figure 2 shows in a grphic form the performance time of the algorithm for values of N from N = 1024 to N=32768. Figure 3 presents data on the speed, when N is ranging from 65536 to 1048576.

The results from figures 2 and 3 enable us to determine the optimal values of the size size n of submatrices with respect to the performance time of the algorithm for different sizes N of the connections matrix. These values are shown in Table 1.

Our study with SMADAJS also shows that the required memory M [KB] of the algorithm depends only on the size of the submatrix n, at N = const. This is illustrated in figure 4.

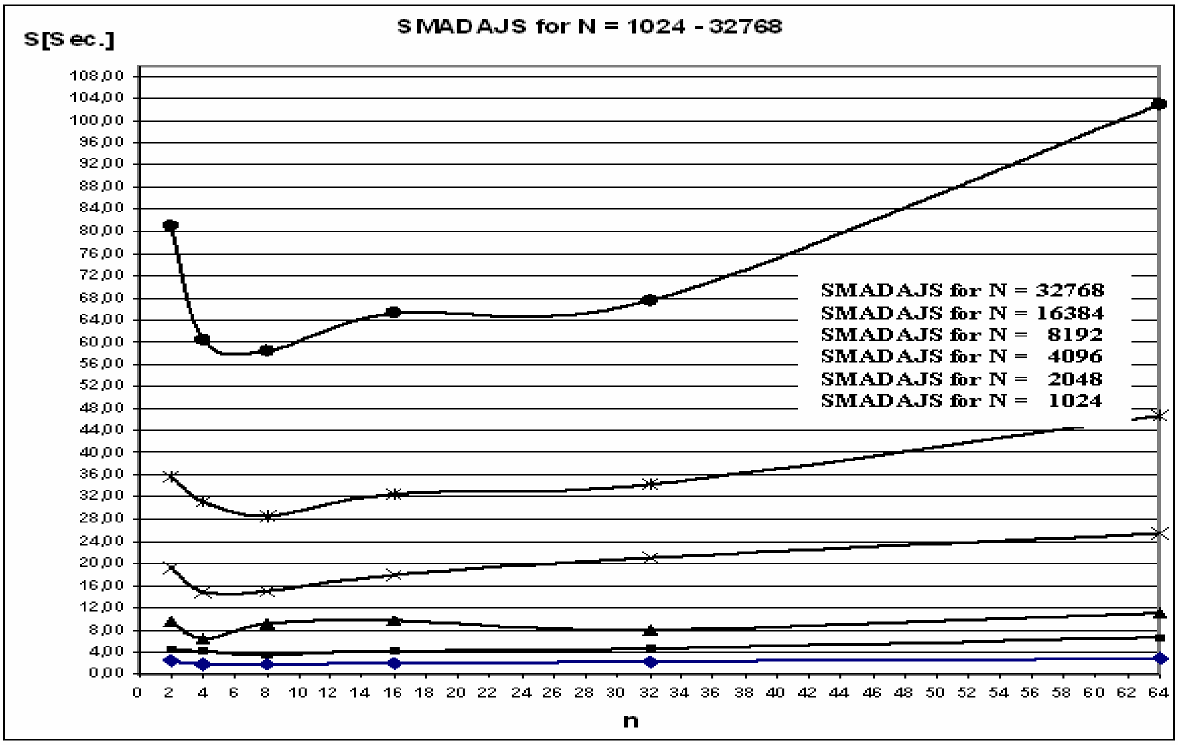

Figure 2: The performance time of the algorithm for values of N from N = 1024 to N=32768.

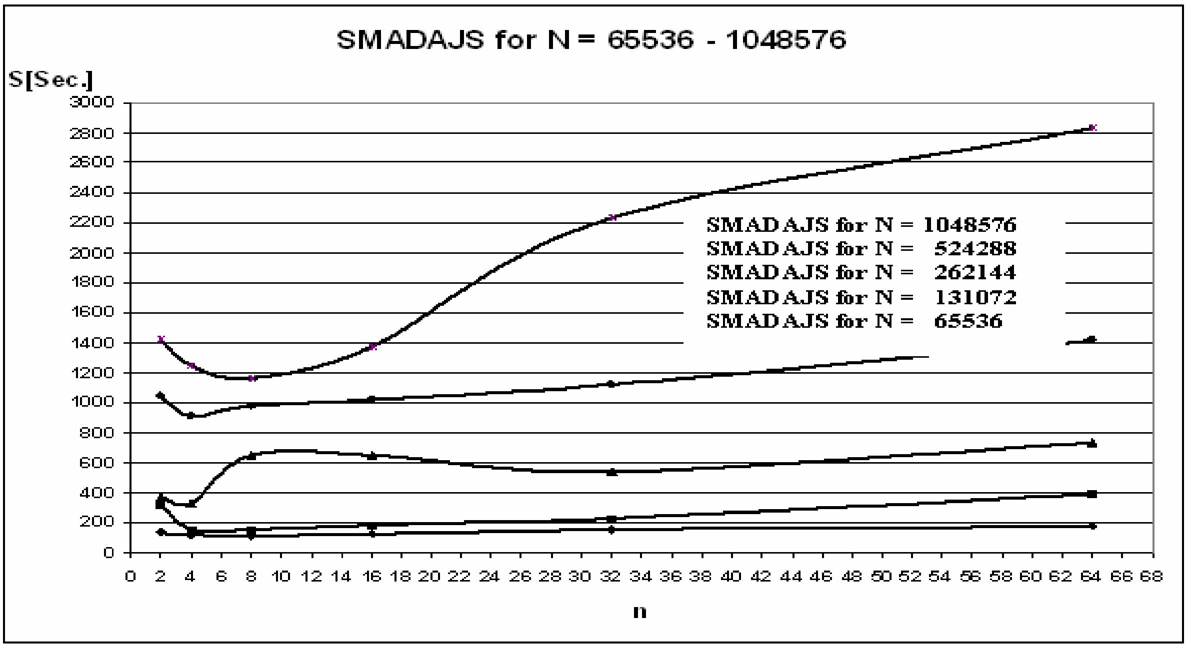

Figure 3: The speed, when N is ranging from 65536 to 1048576.

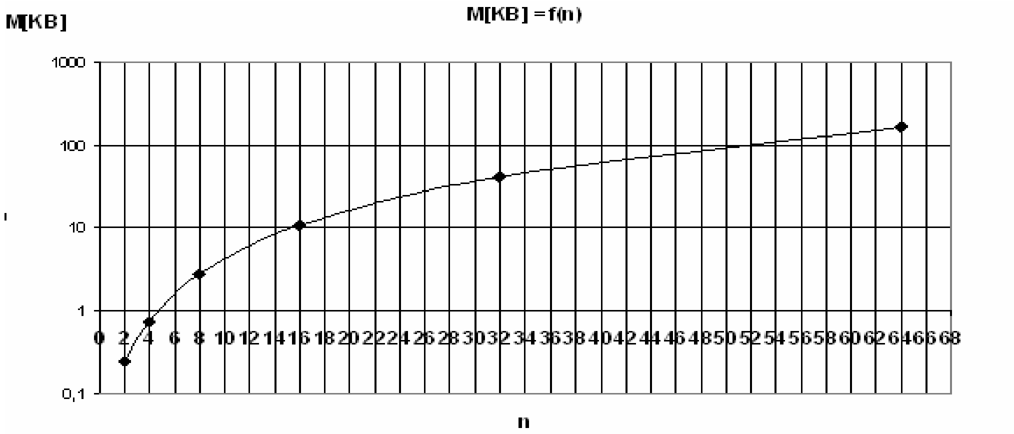

Figure 4: Required memory.

4 A COMPARISON OF THREE SOFTWARE MODELS PERFORMANCE

A software model's performance (P) is defined as a ratio of the non- nil resolutions to the total number of the solutions. R(v) is the set of the nil solutions, R(w) is the set of the non-nil solutions, and R is a set of the all solutions (Kolchakov K., 2013).

$$R=R(v)+R(w) \qquad (2)$$

$$P=(R(w)/R).100[\%] \qquad (3)$$

From formula (3) it is seen that when the number of nil solutions R(v) vanishes to nil, then the performance P tends to 100% (Kolchakov K., 2013).

To facilitate the performance examination, 5 kinds of matrices for simulation of the input connectivity matrix T are chosen. The special input matrices 2A, 2B, 2C, 2D and 2E (Kolchakov K., 2013) are represented on Figure 5.

$$\begin{matrix} 1&0&0&0&0 \\ 0&1&0&0&0 \\ 0&0&1&0&0 \\ 0&0&0&1&0 \\ 0&0&0&0&1 \end{matrix} \qquad \begin{matrix} 1&1&0&0&0 \\ 1&1&1&0&0 \\ 0&1&1&1&0 \\ 0&0&1&1&1 \\ 0&0&0&1&1 \end{matrix} \qquad \begin{matrix} 0&0&1&1&1 \\ 0&0&0&1&1 \\ 1&0&0&0&1 \\ 1&1&0&0&0 \\ 1&1&1&0&0 \end{matrix}$$

2A 2B 2C

$$\begin{matrix} 0&0&0&1&1 \\ 1&0&1&0&1 \\ 0&1&1&0&0 \\ 1&1&1&1&0 \\ 1&0&1&1&1 \end{matrix} \qquad \begin{matrix} 1&1&1&1&1 \\ 0&1&1&1&1 \\ 0&0&1&1&1 \\ 0&0&0&1&1 \\ 0&0&0&0&1 \end{matrix}$$

2D 2E

Figure 5: The special input matrices.

In Table 2, we have represented the investigation results related to the performance P of the software models SMADA, SMAJDA and SMADAJS for each of the above input matrices.

Table 2: Comparison in terms of performance.

P[%]	2A	2B	2C	2D	2E
SMADA	6,66	20	80	80	53,3
SMAJDA	6,66	20	80	80	100
SMADAJS	12,5	37,5	87,5	100	100

A comparison of the results in Table 2 shows that the performance of SMADAJS is superior than the performance of SMADA and SMAJDA.

5 CONCLUSIONS

The main conclusions of the paper are as follows.The optimal size of the submatrices n_opt. with respect to the speed is characterized by two values n_opt. = 4 and n_opt. = 8 for the different sizes of the connections matrix. The memory required is determined by the size of submatrices n and it does not exceed 200 MB. The performance of SMADAJS is faster than that of SMADA and SMAJDA.

ACKNOWLEDGEMENTS

The research work reported in the paper is supported by the project AComIn "Advanced Computing for Innovation", grant 316087, funded by the FP7 Capacity Programme (Research Potential of Convergence Regions).

REFERENCES

Kolchakov K., Monov V., 2013. An Algorithm for Non – Conflict Schedule with Diagonal Activation of Joint Sub Matrices, *17-th International Conference on "DISTRIBUTED COMPUTER AND COMMUNICATION NETWORKS (DCCN-2013):"CONTROL, COMPUTATION, COMMUNICATIONS, Moscow, October 07-10, 2013, pp. 180-187, ISBN 978-5-94836-366-0.*

Kolchakov K., 2013. Comparative Analysis of Class Algorithms for Non-Conflict Schedule in Switching Nodes, *17-th International Conference on "DISTRIBUTED COMPUTER AND COMMUNICATION NETWORKS (DCCN-2013): "CONTROL, COMPUTATION, COMMUNICATIONS, Moscow, October 07-10, 2013, pp.172-179, ISBN 978-5-94836-366-0.*

Kolchakov K., 2012. Examination on Algorithms for Non Conflict Schedule with Diagonal Activation in Case of a Large Size Switching Matrix. *International Conference Automatics and Informatics `12 03-05.10.2012 Sofia, Bulgaria. Proceedings CD: ISSN 1313 – 1869. pp.341-344.*

Kolchakov K., 2011. “An Algorithm Synthesis of Non-Conflict Schedule by Diagonal Connectivity Matrix Activation” *Proceedings of the International Conference AUTOMATICS AND INFORMATICS`11, John Atanasoff Society of Automatics and Informatics, Bulgaria, Sofia 03.10-07.10.2011., pp. B-247 – B251, Proceedings ISSN 1313-1850, CD ISSN 1313-1869.*

Tashev T. , 2010. Computering simulation of schedule algorithm for high performance packet switch node modelled by the apparatus of generalized nets. *Proceedings of the 11th International Conference CompSysTech’2010, 17-18 June 2010, Sofia, Bulgaria. ACM ICPS, Vol.471, pp.240-245.*

Kolchakov K., 2009. Software Models with Sparse Mask Matrixes for Non-Conflict Schedule obtaining in a Switching Node, In *“Distributed Computer and communication networks DCCN 2009”, Moscow: R&D Company “Information and Networking technologies” ISBN 978-5-9901871-1-5.*

Wanjari P., Chonhari A., 2011. “Implementation of 4x4 crossbar switch for Network Processor”, *International Journal of Emerging Technology and Advanced Engineering,* Website: www.ijetae.com (ISSN 2250 – 2459, Volume 1, Issue 2, December 2011).

Kim D., Lee K. and Yoo H., 2005. A Reconfigurable Crossbar Switch with Adaptive Bandwidth Control for Networks-on-Chip, *IEEE International Symposium on Circuits and Systems, 2005.*