БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ • BULGARIAN ACADEMY OF SCIENCES

ПРОБЛЕМИ НА ТЕХНИЧЕСКАТА КИБЕРНЕТИКА И РОБОТИКА, 71
PROBLEMS OF ENGINEERING CYBERNETICS AND ROBOTICS, 71

София • 2019 • Sofia

Comparative analysis of algorithms for non-conflict scheduling in crossbar switch with large scale connections matrix

K. Kolchakov

Institute of Information and Communication Technologies 1113 Sofia, Bulgarian Academy of Sciences,
e-mail:kkolchakov@iit.bas.bg

Abstract: *A comparison is made between two conflict-free scheduling algorithms for crossbar switch in terms of performance and memory requirements in the case of very large sizes of the connections matrix. The optimal size of the submatrices has been found, where the maximum speed of execution is achieved at minimum required memory.*

Key words: *network nodes, crossbar switch, conflict elimination, packet messages*

1. Introduction

In modern communication systems, the packet switch routes the packet traffic from the source to the corresponding destination under the control of a conflict-free switching algorithm. Effective management of information traffic is a major problem in communications, systems for processing and integrating heterogeneous data [1], wireless sensor networks and systems [2]. In the crossbar switch, N in number of packet message sources are connected to N in number of packet message receivers by a so-called matrix of connections T with dimensions N x N. An element in matrix T with value one indicates the presence of a packet message request (Tij = 1 when source i wants to transmit a packet message to receiver j).

There are two types of conflict situations:

• when one packet message source wants to transmit to two or more packet message receivers (the units in any row of T are more than one).
• when two or more packet message sources want to send messages to the same receiver (the units in any column of T are more than one).

Conflict-free scheduling algorithms are available to avoid such conflicts [3,4,8]. In this paper, we compare two conflict-free scheduling algorithms in the case of very large sizes of the connections matrix.

2. Description of algorithms ADAJS and ADAJSFA

In the conflict-free scheduling algorithm ADAJS (Algorithm with Diagonal Activation of Joint Submatrices), the size N of the connections matrix T is a degree of the number two and the same applies to the size n of the submatrices. The number of iterations performed by the algorithm is I = N / n.

The first iteration implements the requests which are arranged in the main diagonal of the connections matrix without conflict (Figure 1). Each successive iteration implements requests in joint pairs of diagonal submatrices that are conflict-free. At N = constant, the number of iterations is determined by the size n of the submatrices [5].

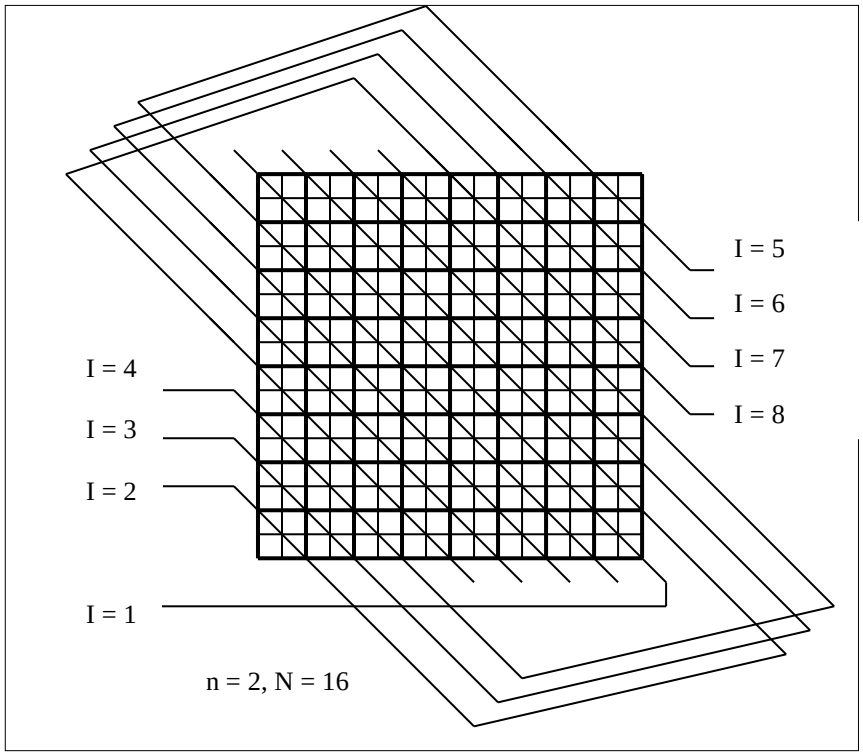

Figure 1. Connections matrix with diagonal activation of submatrices.

The essence of the ADAJSFA algorithm (Algorithm with Diagonal Activation of Joint Submatrices with Finite Automata) is based on the knowledge that diagonally located submatrices with requests for service in the connections matrix T are conflictless in the diagonal where they are located. Also, there are pairs of diagonals with submatrices of requests that are conflict-free with each other. ADAJSFA synthesizes completely conflict-free schedules in Crossbar switches, as well as the ADAJS algorithm.

The process of implementing a conflict-free schedule using the ADAJSFA algorithm is divided into several steps. The first step concerns submatrices in the main diagonal, which are processed simultaneously and without conflict. The next steps are related to the combination of diagonals with submatrices parallel to the main diagonal in pairs. The whole process of conflict-free execution of requests under the ADAJSFA algorithm is carried out by means of finite state automata [6].

In the algorithm of diagonal activation of joint submatrices with finite automata the control is accomplished at two levels - finite state automata are activating diagonals in joint submatrices (subordinates) and a finite state automation is activating joint diagonals of submatrices (main).

Under the activation of signal Z1, the automation at the first level passes through the states from A0 to AK consecutively and to each state from A1 to AK corresponds an output signal St.1 (Step1) to St.K (StepK). Thus the automation activates one diagonal with requests of combined two diagonal submatrices at the first control level of ADAJSFA.

The second-level finite state automation is a master and it activates pairs of diagonal joint submatrices in a certain order. Under the signal Z2, the automation goes into state A0 with an output signal R, which is an indication to the switch node processor that the automation is free and can be started at any time (Figure 2) [7].

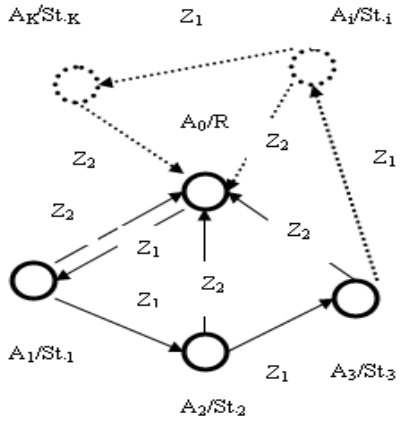

Figure 2. State graph of Moore's finite state automation.

3. Numerical simulations

Algorithms ADAJS and ADAJSFA are programed as software modules in MATLAB programming environment. Numerical simulations are performed in order to compare the two algorithms with respect to performance (speed of execution) and required memory of each algorithm in the cases of very large sizes of connections matrixes of the switch. Table 1 contains the results concerning speed of execution of both algorithms in seconds as a function of the size of connections matrices. Graphically, the results are illustrated in figures 3 an 4.

Table 1. Performance as a function of matrix size for ADAJS and ADAJSFA algorithms.

n	**S[Sec.] N = 1048576**		**S[Sec.] N = 2097152**		**S[Sec.] N = 4194304**		**S[Sec.] N = 8388608**	
	ADAJS	**ADAJSFA**	**ADAJS**	**ADAJSFA**	**ADAJS**	**ADAJSFA**	**ADAJS**	**ADAJSFA**
2	1228,8	2125,3	2457,6	4615,7	4915,2	10016,0	9830,4	21734,7
4	1292,8	3420,5	2585,6	7422,4	5171,2	16106,7	10342,4	34951,7
8	2013,4	9586,1	4026,8	20801,8	8053,7	45139,9	16107,5	97953,7
16	3649,2	28758,3	7298,5	62405,5	14597,1	135419,9	29194,2	293861,2

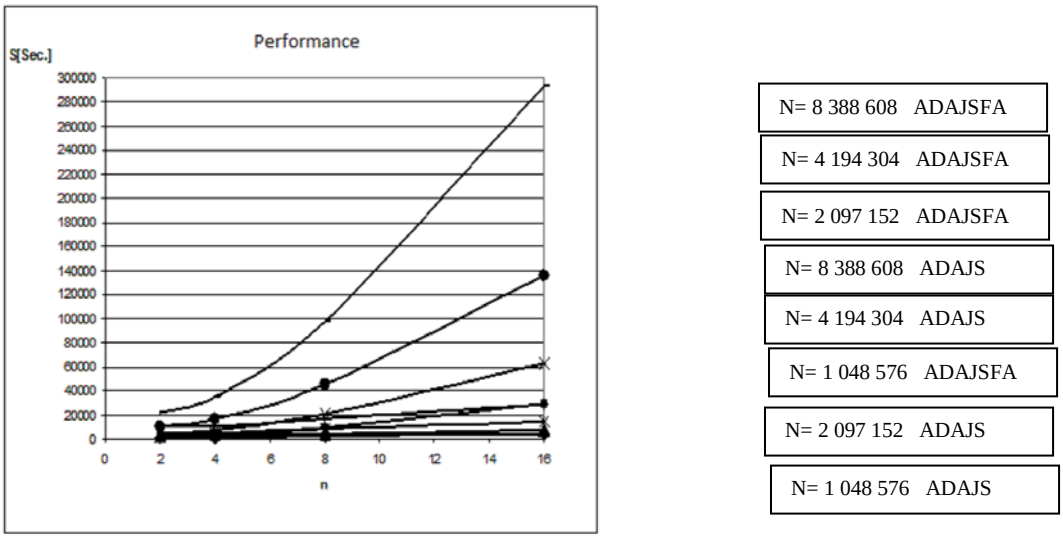

Figure 3. Performance of ADAJS and ADAJSFA at very large sizes of N and different values of n.

Figure 5 shows the ratio between ADAJSFA and ADAJS performance at N = const. and different values of n.

Table 2 presents the results for the required memory at different values of n. Figure 6 is a graphical representation of the memory requirements for the two algorithms.

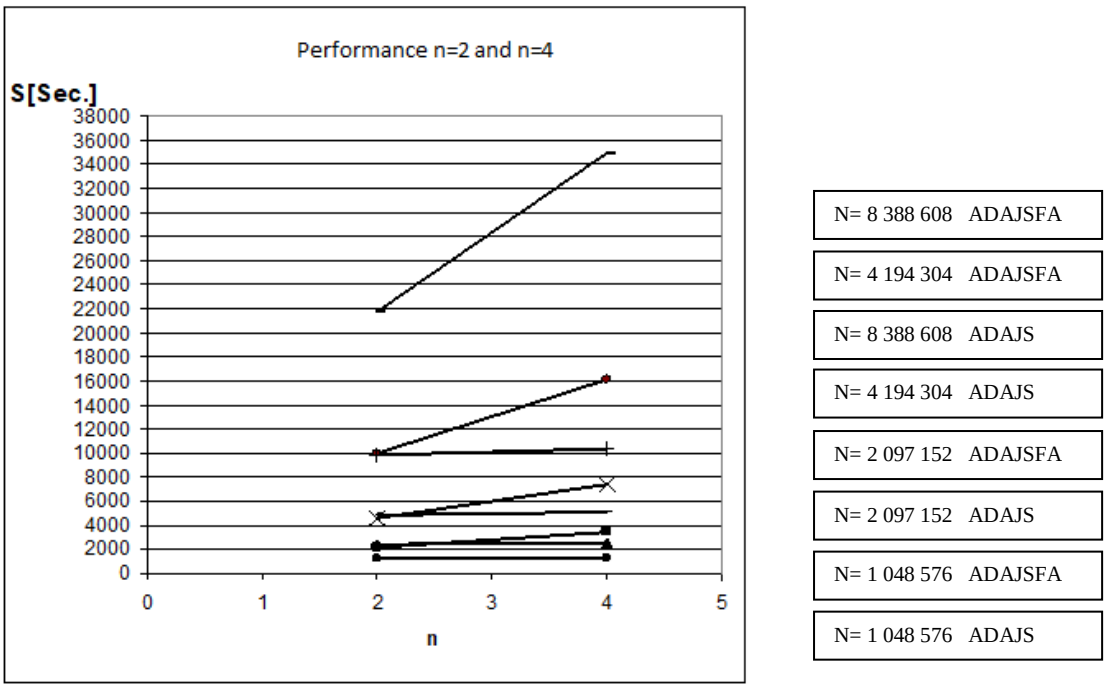

Figure 4. ADAJS and ADAJSFA performance at very large N sizes and values at n = 2 and n = 4.

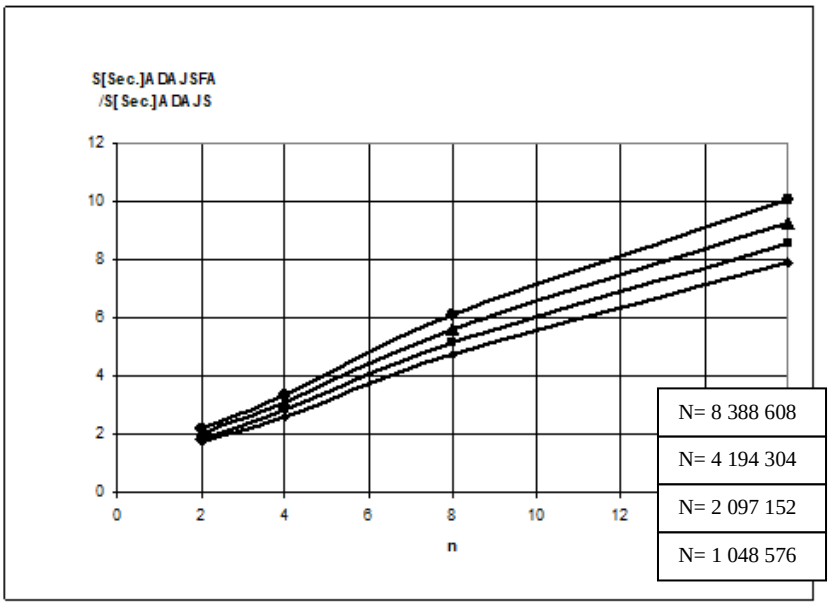

Figure 5. Ratio between ADAJSFA and ADAJS performance at N = const. and different values of n.

Table 2. Memory required.

n	M[B] ADAJSFA	M[B] ADAJS
2	280	224
4	1048	736
8	4120	2720
16	16480	10528

For values of n = 2 and n = 4, the required memory is almost the same for both algorithms, and then, as n increases, the difference increases significantly, with ADAJSFA requiring more memory.

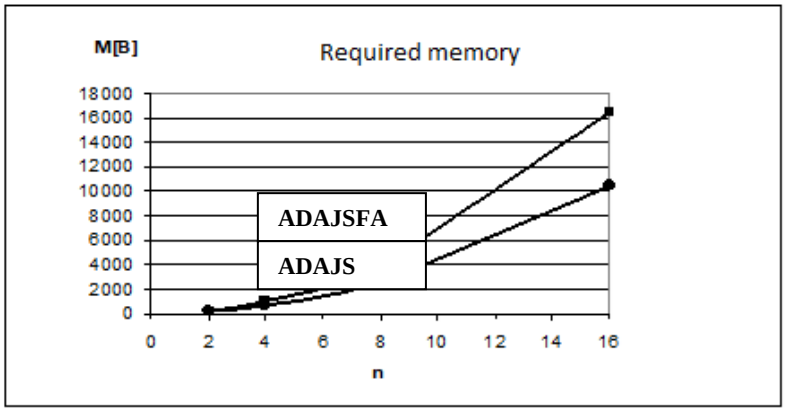

Figure 6. Required memory at different values of n.

4. Conclusion

The comparative analysis shows that for very large sizes N of the connections matrix for both conflict-free scheduling algorithms, there is an optimal sub-matrix size (n = 2), with maximum speed and minimal memory required. This result is valid for all N. The numerical simulations indicate that ADAJS is a better algorithm with respect to speed of execution and memory requirements.

References

1. Alexandrov, A., V. Monov, Implementation of a service oriented architecture in smart sensor systems integration platform, Proc. of the Third International Conference on Telecommunications and Remote Sensing – ICTRS'14, 26-27 june 2014, Luxembourg, Grand Duchy of Luxembourg, , pp. 114-118, 2014, ISBN 978-989-758-033-8.
2. Alexandrov, A., V. Monov, ZigBee smart sensor system with distributed data processing, Proc. of the 7-th IEEE Conference Intelligent Systems, Warsaw Poland,

Vol. 2, pp. 259-268, September 24-28, 2014., In: Advances in Intelligent Systems and Computing, Springer Vol. 323, ISBN 978-3-319-11309-8.
3. Kolchakov K., V. Monov. Query Conflicts in an Algorithm for Non-Conflict Scheduling for Crossbar Commutator Proceedings of the International Conference Automatics and Informatics`2018, Bulgaria, Sofia, October 4-6, 2018, Federation of the scientific engineering unions, John Atanasoff Society of Automatics and Informatics, 2018, ISSN: Proceedings ISSN 1313-1850, CD ISSN 1313-1869, pp.153-156.
4. K. Kolchakov, V. Monov. Hardware acceleration of a scheduling algorithm for crossbar switch node via decomposition of the connection matrix, Problems of Engineering Cybernetics and Robotics, Vol.69, pp.83-90, 2018, Sofia. Bulgarian Academy of Sciences.
5. К. Колчаков, В. Монов. Изследване на алгоритъм за безконфликтно разписание с голяма размерност на матрицата на връзките. BULGARIAN ROBOTIC SOCIETY, Proceedings International Conference "Robotics, Automation and Mechatronics` 18 RAM 2018" July 24-26,2018, Sofia, Bulgaria ISSN 1314-4634 , pp.42-46
6. Kolchakov K., V. Monov. "An Approach for syntesis of Non-conflict Schedule with Optimal Performance of Sub Matrices in a Crossbar Switching Node". " Proceedings of the International Conference AUTOMATICS AND INFORMATICS`2016, John Atanasoff Society of Automatics and Informatics, Bulgaria, Sofia, 04.10-05.10.2016, pp. 33-35., Proceedings ISSN 1313-1850, CD ISSN 1313-1869.
7. Kolchakov K. Non- conflict schedule through finite automats in switching nodes. International Conference Automatics and Informatics `13 03-07.10.2013 Sofia, Bulgaria. Proceedings: ISSN 1313 – 1850, Proceedings CD: ISSN 1313 – 1869. pp. I-191 - I-194.
8. Tashev T. 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.

Сравнительный анализ алгоритмов для бесконфликтного планирования в коммутационной панели с крупномасштабной матрицей соединений

К. Колчаков

Институт Информационных и Коммуникационных Технологий
e-mail: kkolchakov@iit.bas.bg

Аннотация: *Проведено сравнение между двумя бесконфликтными алгоритмами планирования для коммутационной перемычки с точки зрения производительности и требований к памяти в случае очень больших размеров матрицы соединений. Был найден оптимальный размер подматриц, где максимальная скорость выполнения достигается при минимально необходимой памяти.*

Ключевые слова: *сетевые узлы, межблочный коммутатор, устранение конфликтов, пакетные сообщения.*