БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ • BULGARIAN ACADEMY OF SCIENCES

ПРОБЛЕМИ НА ТЕХНИЧЕСКАТА КИБЕРНЕТИКА И РОБОТИКАТА, **70**
PROBLEMS OF ENGINEERING CYBERNETICS AND ROBOTICS, **70**

София • 2018 • Sofia

An approach to weights distribution of requests in two algorithms for non-conflict scheduling

Kiril Kolchakov, Vladimir Monov

Institute of Information and Communication Technologies 1113 Sofia
Email: ***kkolchakov@iit.bas.bg***

Abstract: *The paper presents an approach for determining the weights of requests in the connection matrix of a crossbar switch node. We consider two algorithms for non-conflict scheduling: Adaptive algorithm for management by weight coefficient of the traffic in crossbar commutator (**AAM**) and Optimum adaptive algorithm for management by weight coefficient of the traffic in crossbar commutator (**AAMO**). In both algorithms the weights are positioned from top to bottom and right to left. In this way each request has a constant weight and hence a constant priority in the execution. Here we present an alternative determination of weights improving the execution of requests.*

Key words: *crossbar commutator, algorithms for non-conflict scheduling*

1. Introduction

In the ccrossbar commutator, N number of sources of packet massages are associated with N number receivers of packet messages through the so-called T connection matrix with dimensions N x N. In matrix T, an element value is equal to

1 when a request for transmission of packet message is available (Tij = 1, when source i wants to transmit a packet message to receiver j). Two types of conflict situations are available:

1. when two or more sources of packet messages want to send messages to one and the same receiver (the unities in any column of T are more than one).
2. when one source of packet messages wants to transmit to two or more packet message receivers (the unities in any raw of T are more than one) [1,2,4,6].

There is a great number of algorithms for a conflict-free schedule by which these conflicts are avoided. Two of these are Adaptive algorithm for management by weight coefficient of the traffic in Crossbar commutator (**AAM**) and Optimum adaptive algorithm for management by weight coefficient of the traffic in Crossbar commutator (**AAMO**) [3].

2. Weights distribution and results

Figure 1 shows how the weight factors are determined, namely from top to bottom and from left to right in the case of a connection matrix T with size N=4.

1	**0**	**1**	**1**
1	**0**	**1**	**1**
0	**0**	**1**	**0**
1	**1**	**0**	**1**

T

N=4

1	**0**	**5**	**8**
2	**0**	**6**	**9**
0	**0**	**7**	**0**
3	**4**	**0**	**10**

W1

Figure 1. Weight factors assigned from top to bottom and left to right

Each request is executed in time, according to the weight coefficient. We assume conditionally that orders with a lower weight coefficient are executed earlier. The new approach for weight definition is bottom-up and right-to-left. Figure 2 presents this approach of determining the weights for the same connection matrix T of Figure 1.

1	**0**	**1**	**1**
1	**0**	**1**	**1**
0	**0**	**1**	**0**
1	**1**	**0**	**1**

T

N=4

10	**0**	**6**	**3**
9	**0**	**5**	**2**
0	**0**	**4**	**0**
8	**7**	**0**	**1**

W2

Figure 2. Weight factors assigned from bottom to top and right to left

From Figure 3 it is seen that the sum of the respective weight factors for each request is constant and equal to 11 for this specific example.

1	**0**	**5**	**8**
2	**0**	**6**	**9**
0	**0**	**7**	**0**
3	**4**	**0**	**10**

10	**0**	**6**	**3**
9	**0**	**5**	**2**
0	**0**	**4**	**0**
8	**7**	**0**	**1**

11	**0**	**11**	**11**
11	**0**	**11**	**11**
0	**0**	**11**	**0**
11	**11**	**0**	**11**

W1 *W2* *W = W1+W2*

Figure 3. Sum of the weight factors

Applied to both **AAM** and **AAMO** algorithms, the new approach represents an alternative weighting (top to bottom and left to right or bottom to top and right to left) for each new connection matrix.

Determining the weights from top to bottom and from left to right is performed by means of the software model **SMRIGHT**, Figure 4. Determining the weights from bottom to top and from right to left is performed by the software model **SMBACK**, e.g., see Figure 5.

The software models **SMRIGHT** and **SMBACK** are written in MATLAB language and our experiments are performed on a computer configuration Dell OPTIPLEX 745 (Core 2 Duo E6400 2,13GHz, RAM 2048).

```
tic;
   T = randsrc(N,N,[0,1])
B = sparse(T)
C = ones(N)
p = 0
for i = 1: N
   for j = 1: N
if B(i,j) = = 1
C(i,j) = C(i,j) + p % request - weight coeffitsient.
 p = p + 1
else C(i,j)= 0
end
end
end
T,C
toc
```

Figure 4. Software model SMRIGHT

```
tic;
   T = randsrc(N,N,[0,1])
B = sparse(T)
C = ones( N )
L = sum(sum(T))%total number of requests.
p = L - 1
for i = 1: N
   for j = 1: N
if B(i,j) = = 1
C(i,j) = C(i,j) + p % request - weight coeffitsient.
 p = p - 1
else C(i,j)= 0
end
end
end
T,C
toc
```

Figure 5. Software model SMBACK.

Table 1 shows the results of the study of **SMRIGHT** and **SMBACK** software models with respect to the performance and memory required for different sizes N of the T connection matrix. It is seen that for values of N from N = 4 to N = 32 the results are almost identical. For N = 64, there is a difference of 24% for performance and 0.5% for memory, while for N = 128 the difference is 4% for performance and there is almost no memory difference.

Table 1 Performance and memory required of models SMRIGHT and SMBACK

N	SMRIGHT, S[Sec.]	SMBACK, S[Sec.]	SMRIGHT, M[B]	SMBACK, M[B]
4	0,0438	0,0408	524	508
8	0,1438	0,1218	1956	1988
16	1,5126	1,5126	7856	7900
32	19,0970	19,2438	30768	30944
64	392,0600	486,4200	123572	122980
128	4955,8000	4769,1000	493728	493736

From these results we can conclude that the alternative application of software models **SMRIGHT** and **SMBACK** in the determination of the weight coefficients is equivalent in terms of performance and memory required. Also the alternative application of these models enables us to avoid the attachment of permanent weights to the requests.

The weight factor determines when to run the corresponding request for the **AAM** and **AAMO** algorithms. The alternative application of models **SMRIGHT** and **SMBACK** in determining the weigh factors is a prerequisite for evenly over time executing of the requests as long as they are not assigned to one and the same weight factors.

In Table 2, we have denoted by **w** the number of weights for software models **SMAAM** and **SMAAMO** corresponding to algorithms **AAM** and **AAMO** for different values of N. It is important to note that for **AAMO** the number of weights is smaller for one and the same T-connections matrix (Table 2) because the requests in one and the same diagonal are of equal weight, i.e. they are non-conflict to one another [3].

Table 2 Weights for software models SMAAM and SMAAMO

N	**w SMAAM**	w **SMAAMO**
4	9	6
8	26	12
16	133	28
32	502	63

3. Conclusion

From this study, it can be concluded that the alternative application of **SMRIGHT** and **SMBACK** in weight factor determination is equivalent in terms of performance and memory required and from this perspective it will not slow the performance of **AAM** and **AAMO**. However, as a result of this alternative weight determination we can avoid the attachment of constant weights to requests and it is possible to achieve comparative equalization with respect to their execution.

References

1. 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, 69, Prof. Marin Drinov Academic Publishing House, 2018, ISSN:Print 0204-9848 , Online 1314-409X, 83-90.

2. Kolchakov K.**,** V. Monov. An approach for algorithm optimization of non-conflict schedule by diagonal connectivity matrix activation. Proceedings of the International Conference Automatics and Informatics`2017, Bulgaria, Sofia, October 4-6, 2017, Federation of the scientific engineering unions, John Atanasoff Society of Automatics and Informatics, 2017, ISSN:Proceedings ISSN 1313-1850, CD ISSN 1313-1869, 161-164
3. Kolchakov K., Monov V. Adaptive Algorithm for Management by Weight Coefficients of the Traffic in Crossbar Commutator, International Journal " Information Models and Analyses" Vol.4, № 1, pp. 53-60. **http://www.foibg.com/ijima/vol04/ijima04-01-p05.pdf**
4. Tashev, T. , Atanasova, T. Computer Simulation of MIMA Algorithm for Input Buffered Crossbar Switch. International Journal "Information Technologies & Knowledge", Volume 5, Number 2, 2011. ITHEA® , Sofia, Bulgaria. Pages 183-189.
5. Kim D., K. Lee and H. Yoo, A Reconfigurable Crossbar Switch with Adaptive Bandwidth Control for Networks-on-Chip, IEEE International Symposium on Circuits and Systems, 2005.
6. 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.

Подход к весовому распределению запросов в двух алгоритмах для бесконфликтного расписания

Кирил Колчаков, Владимир Монов

Институт информационных и коммуникационных технологии

Резюме

В статье представлен подход для определения весов запросов в матрице соединений перекрестного коммутатора. Мы рассматриваем два алгоритма для вычисления бесконфликтного расписания коммутации : Адаптивный алгоритм управления по весовому коэффициенту трафика в перекрестном коммутаторе (AAM) и Оптимальный адаптивный алгоритм управления по весовому коэффициенту трафика в перекрестном коммутаторе (AAMO). В обоих алгоритмах веса располагаются сверху вниз и справа налево. Таким образом, каждый запрос имеет постоянный вес и, следовательно, постоянный приоритет при выполнении. Здесь мы приводим альтернативное определение весов, улучшающее выполнение запросов.