

ПОДХОД ЗА ОПТИМИЗИРАНЕ НА АЛГОРИТЪМ ЗА БЕЗКОНФЛИКТНО РАЗПИСАНИЕ С ДИАГОНАЛНА АКТИВАЦИЯ НА МАТРИЦАТА НА ВРЪЗКИТЕ

AN APPROACH FOR ALGORITHM OPTIMIZATION OF NON-CONFLICT SCHEDULE BY DIAGONAL CONNECTIVITY MATRIX ACTIVATION

К.Колчаков, В.Моноу

Институт по информационни и комуникационни технологии Българска Академия на Науките, 1113, София, България, ул.Акад. Г. Бончев бл.2, тел.: (+359) 2 979 -32 -40 e-mail: kkolchakov@iit.bas.bg, vmonov@iit.bas.bg

Abstract: In this paper we propose an approach for optimization of an algorithm for non-conflict schedule in crossbar switch. The purpose is to achieve an improvement of the speed and common performance of the algorithm with diagonal connectivity matrix activation. The optimized algorithm is studied and analyzed by using specific connectivity matrices with different sizes.

Key words: network nodes, node traffic, crossbar switch, conflict elimination, packet messages.

УВОД

Целта на изследването е да се оптимизира алгоритъм с диагонална активация на матрицата на връзките за безконфликтно разписание на трафика в комутационен възел тип Кросбар. Предложеният подход цели увеличаване бързодействието и комплексната производителност на алгоритъма. Алгоритъмът е предназначен за процесора управляващ изпълнението на заявките за обслужване на трафика в Кросбар възела. Кросбар възелът работи в реално време и е от изключително значение бързодействието, комплексната производителност и необходимата памет. В статията се прави сравнение на оптимизирания и съществуващия алгоритми по отношение на бързодействие, комплексна производителност и необходима памет. За целта са използвани програмните им модели при различни характерни типове и размери на матрицата на връзките T .

Наличието на конфликти в работата на комутационните възли намалява драстично трафика.

Дисциплината при обработването на заявките чрез безконфликтно разписание, гарантира изцяло елиминирани на конфликтите.

Проблемът с конфликтите, съпътстващи работата на комутационните възли е следния: комутаторите в комутационните възли са с размери $N \times N$, като N на брой източници на пакетни съобщения се свързват през комутатора на комутационния възел с N на брой приемници на тези съобщения. Конфликтите се получават при два случая:

- когато източник на съобщение дава заявка за свързване към два или повече приемника на съобщения.
- когато един приемник на съобщения има заявка за свързване от два или повече източника на съобщения.

Състоянието на комутатора на един Кросбар възел се представя с т.н. матрица на връзките. За комутатор с размери $N \times N$ матрицата на връзките T също е с размери $N \times N$, като всеки член $T_{ij} = 1$, ако има заявка за връзка

между източник на съобщение i и приемник j . В противен случай $T_{ij} = 0$.

Конфликтна ситуация се създава, когато в който и да е ред от матрицата на връзките броя на единиците е по-голям от една, това съответства на случая, когато един източник заявява връзка с повече от един приемник. Наличието на повече от една единица, в която и да е колона от матрицата T е също указание за конфликтна ситуация и означава, че повече от един източник е заявил връзка към един и същ приемник [1,2,3].

Съществуват изследвания на проблемите с конфликтите в комутационни възли с моделиране чрез апарата на обобщените мрежи [5]. Ние използваме програмни модели написани на езика на Matlab.

ОПИСАНИЕ НА АЛГОРИТЪМА

В алгоритъма с диагонална активация на матрицата на връзките ADA (Algorithm by diagonal connectivity matrix activation) се използва знанието, че заявките разположени диагонално са безконфликтни [2]. Заявките, разположени в който и да е диагонал успореден на главния са безконфликтни помежду си. В матрица на връзките T с размери $N \times N$, броят на диагоналите със заявки е $2N - 1$. ADA активира главния диагонал от заявки за обслужване и след това последователно активира диагонали от заявки над и под него. При всяка една итерация се активира само един диагонал от заявки, като първата итерация е за главния диагонал. Броятът на итерациите е организиран така, че проверява дали броят им не е станал $2N - 1$, т.е. дали всички диагонали със заявки са били активирани.

Вярно е също и това, че заявките в диагонал, разположен успоредно на перпендикулярния диагонал на главния диагонал също са безконфликтни. Възможно е със същия успех при ADA да се избера за активирани диагонали със заявки, успоредни на диагонал перпендикулярен на главния. При този случай за размери $N \times N$ на T , броят на диагоналите със заявки отново е $2N - 1$.

Ние работиме с диагонали от заявки успоредни на главния диагонал.

Недостатък на ADA е това, че не се следи за изчерпването на заявките. Независимо дали на даден етап от работата на алгоритъма всички заявки са реализирани или не, $2N - 1$ на брой диагонала от заявки се активират последователно. Освен това алгоритъмът не проверява за нулева матрица на връзките T, събитие което макар и много рядко, но е възможно да се случи. В този случай, $2N - 1$ на брой нулеви диагонали ще започнат да се активират последователно. Пристигналите заявки във вече активирани бивши нулеви диагонали не могат да се обработват, докато и последния от диагоналите на нулевата входна матрица не бъде активиран.

ПОДХОД ЗА ОПТИМИЗИРАНЕ НА ADA

Оптимизирането на ADA (Algorithm by diagonal connectivity matrix activation) се реализира в посока отстраняване на посочените недостатъци.

1. Проверка за нулева матрица на връзките T. При $T = 0$, алгоритъмът преустановява работа.
2. Проверка за изчерпване на заявките при всяка итерация т.е. след активиране на текущ диагонал от заявки, броят на заявките в него се изваждат от текущата сума от заявки преди активацията му. Следи се дали текущата сума заявки не се е нулирала и ако това е така, алгоритъма преустановява работа преди всичките $2N - 1$ на брой диагонали да бъдат активирани.

Изследването на оптимизираният алгоритъм с диагонална активация на матрицата на връзките (ADAO) е направено чрез софтуерния му модел SMADAO. Изследването е по отношение на бързодействие, необходима памет и комплексна производителност. Използвани са характерни типове матрици на връзките при различни стойности на N. Сравнението между ADAO и ADA е чрез софтуерните им модели SMADAO и SMADA.

Програмните модели са написани на програмния език на Matlab и са тествани на компютърна конфигурация Dell OPTIPLEX 745 (Core 2 Duo E6400 2,13GHz, RAM 2048).

На Фигура 1. са показани пет типа характерни матрици на връзките.

<pre> 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 </pre> <p>2A</p>	<pre> 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 </pre> <p>2B</p>	<pre> 0 0 1 1 1 0 0 0 1 1 1 0 0 0 1 0 0 0 1 1 0 0 0 0 1 </pre> <p>2C</p>
<pre> 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 </pre> <p>2D</p>	<pre> 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 </pre> <p>2E</p>	

Фигура 1. Типове матрици на връзките.

Производителността P на алгоритмите за безконфликтно разписание се изчислява чрез формули 1 и 2. С $R(v)$ се означава броя на нулевите решения (решения при които не се реализират

заявки за обслужване), с $R(W)$ – ненулевите решения, а R е общия брой решения. P е отношение на ненулевите решения към общия брой решения. Когато броят на нулевите решения клони към нула, производителността P се доближава до 100%[1].

$$R = R(v) + R(w) \quad (1)$$

$$P = (R(w)/R).100[\%] \quad (2)$$

Комплексната производителност (CP) включва в себе си и бързодействието и се изчислява по Формула 3[1].

$$CP = P.t, \text{ при } N = \text{const.}, t = 1/S \quad (3)$$

Във Формула 3 с S е означено времето за реализиране на безконфликтното разписание от съответния софтуерен модел в секунди т.е. бързодействието. P е производителността. Колкото S е по-малко т.е. алгоритъма е по-бърз, толкова t е по-голямо и съответно CP има по-висока стойност.

ИЗСЛЕДВАНЕ НА ADAO ПРИ НУЛЕВА МАТРИЦА НА ВРЪЗКИТЕ И СРАВНЕНИЕ С ADA

При изследването се използват софтуерните модели на алгоритмите ADAO и ADA при нулеви входни матрици за различни стойности на N. В Таблица 1 и Таблица 2 са представени резултатите от изследването.

Таблица 1

	SMADAO	SMADAO	SMADA	SMADA
N	S[Sec.]	M[KB]	S[Sec.]	M[KB]
50	0,27	20,62	0,71	101,03
100	0,31	81,22	2,02	402,03
150	0,48	181,82	3,9	903,03
200	0,74	322,43	7,24	1604,03
250	0,96	503,03	14,11	2505,03
300	2,34	723,63	18,88	3606,03
350	3,03	984,23	22,30	4907,03

Таблица 2

	SMADAO	SMADAO	SMADA	SMADA
N	P[%]	CP	P[%]	CP
50	100	370,37	0	0
100	100	322,58	0	0
150	100	208,33	0	0
200	100	135,13	0	0
250	100	104,66	0	0
300	100	42,73	0	0
350	100	33,00	0	0

От Таблица 1 се вижда, че SMADAO е от 2,6 пъти по-бърз от SMADA при $N = 50$, до 14,6 пъти при $N = 250$.

Необходимата памет при SMADAO е средно 5 пъти по-малка от тази при SMADA.

Таблица 2 представя резултатите от изследването по отношение на производителността P и комплексната производителност CP, изчислени по Формули 2 и 3 съответно. При SMADAO P е 100% за всички размери на N, докато при SMADA е 0%. CP за SMADAO намалява с нарастване на N от 370,37 при $N = 50$, до 33 при $N = 350$, а CP при SMADA е нула за всички стойности на N.

СРАВНЕНИЕ НА ADAO И ADA ПРИ ХАРАКТЕРНИ МАТРИЦИ НА ВРЪЗКИТЕ

От представените на Фигура 1 характерни типове матрици на връзките, тип 2D отговаря на най-често срещания трафик, защото заявките са разпределени по случаен принцип.

Избрани са големи размери на матрицата на връзките с цел по-голяма достоверност при определяне на тенденциите в резултатите.

Таблица 3. Бърздействие и необходима памет при матрица на връзките 2A.

N	SMADAO 2A		SMADA 2A	
	S[Sec.]	M[MB]	S[Sec.]	M[MB]
100	0,23	0,240832	1,36	0,402456
500	8,41	6,004032	42,92	10,012056
1000	34,51	24,008032	105,63	40,024056
1500	80,10	54,012032	183,46	90,036056
2000	95,65	96,016032	199,15	160,040056

Таблица 4. Бърздействие и необходима памет при матрица на връзките 2B.

N	SMADAO 2B		SMADA 2B	
	S[Sec.]	M[MB]	S[Sec.]	M[MB]
100	1,74	0,409608	1,79	0,409632
500	33,52	10,048008	33,11	10,048032
1000	103,93	40,096008	101,05	40,096032
1500	253,57	90,144008	255,74	90,144032
2000	358,99	160,192008	269,60	160,184032

Таблица 5. Бърздействие и необходима памет при матрица на връзките 2C.

N	SMADAO 2C		SMADA 2C	
	S[Sec.]	M[MB]	S[Sec.]	M[MB]
100	2,14	0,199608	3,83	0,199632
500	30,18	4,798008	32,71	4,798032
1000	102,74	19,096008	109,54	19,096032
1500	258,31	42,894008	253,99	42,894032
2000	403,88	76,192008	263,14	76,184032

Таблица 6. Бърздействие и необходима памет при матрица на връзките 2D.

N	SMADAO 2D		SMADA 2D	
	S[Sec.]	M[MB]	S[Sec.]	M[MB]
100	1,65	0,402432	1,78	0,402456
500	30	10,012032	34,10	10,012056
1000	97,89	40,024032	97,96	40,024056
1500	236,86	90,036032	265,88	90,036056
2000	369,90	160,048032	284,18	160,040056

Таблица 7. Бърздействие и необходима памет при матрица на връзките 2E.

N	SMADAO 2E		SMADA 2E	
	S[Sec.]	M[MB]	S[Sec.]	M[MB]
100	1,69	0,482432	1,81	0,482456
500	34,52	12,012032	33,78	12,012056
1000	112,13	48,024032	112,26	48,024056
1500	253,85	108,036032	264,14	108,036056
2000	431,03	192,048032	302,66	192,040056

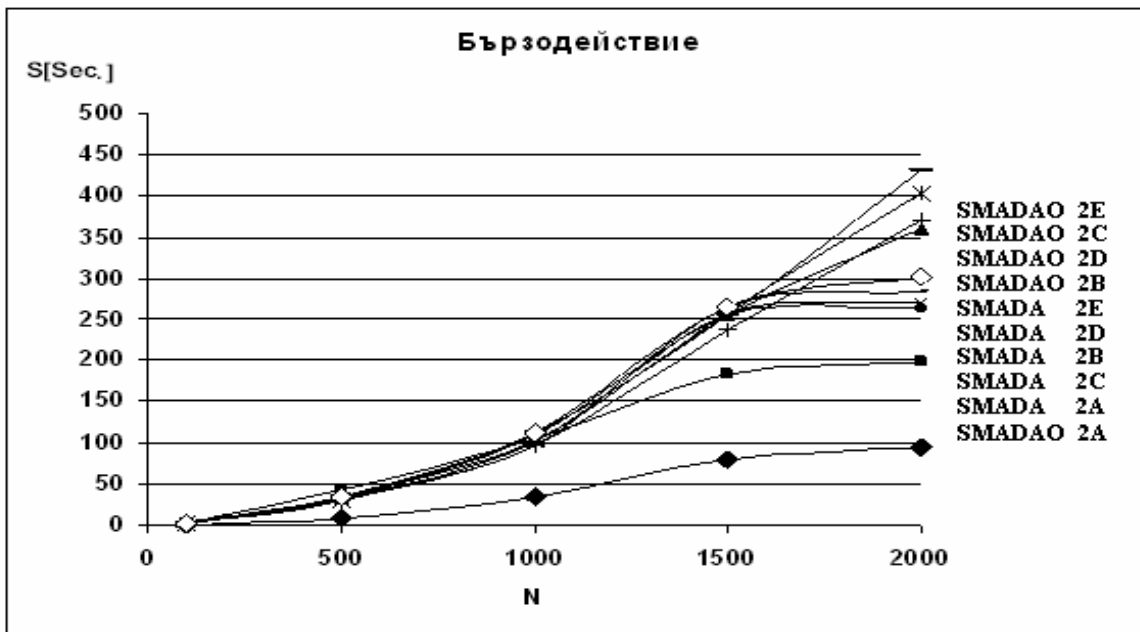
В таблици от 3 до 7 са представени резултатите от изследването по отношение на бърздействието и необходимата памет при различните типове матрици на връзките. На Фигура 2 са представени в графичен вид резултатите от изследването на бърздействието на SMADAO и SMADA.

От таблици 3 до 7 се вижда, че необходимата памет е почти еднаква за SMADAO и SMADA при матрици на връзките 2B, 2C, 2D и 2E. За тип матрица на връзките 2A необходимата памет при SMADAO е около 1,66 пъти по-малка от тази при SMADA за всички стойности на N. Важен резултат от изследването е повишението на производителността до 100% при SMADAO за матрица на връзките 2A. Резултатът се дължи на факта, че първо се обработват заявките в главния диагонал и при SMADAO след проверката за нереализирани заявки се оказва, че такива няма и работата спира, т.е. не се стига до нулеви решения.

При матрици на връзките 2B, 2C, 2D и 2E не се получава съществено избягване на нулеви решения, както при 2A, защото докато не се реализират всички заявки, нулевите решения са факт. Оказва се, че при N = 100 едно нулево решение се икономисва само, което е много малък процент от реализираните нулеви решения. В Таблица 8 са представени резултатите получени за комплексната производителност на SMADAO и SMADA при различни входни матрици на връзките за N = 500.

Таблица 8. Комплексна производителност.

Тип матрица	SMADAO CP	SMADA CP
2A	11,89	0,15
2B	0,59	0,60
2C	2,65	2,44
2D	2,66	2,34
2E	1,54	1,57



Фигура 2. Бързодействие.

ЗАКЛЮЧЕНИЕ

От направеното изследване се вижда, че при нулева входна матрица оптимизираният алгоритъм ADAO притежава безспорни предимства пред ADA по отношение на бързодействие, необходима памет и комплексна производителност за N от 50 до 350.

При входна матрица на връзките 2A, ADAO е от 5,9 пъти по-бърз за N = 100 до 2 пъти по-бърз при N = 2000. Комплексната производителност CP на ADAO за 2A е 11,89 спрямо 0,15 за ADA.

За входна матрица на връзките 2D (най-близка до обичайния трафик), ADAO е по-бърз от ADA за N = 100 до N = 1500 включително.

Тенденцията е при големите размери на N за 2B, 2C, 2D и 2E, бързодействието на алгоритмите да се изравнява, докато при N = 2000 ADAO е по-бавен от ADA при всички типове входни матрици с изключение на 2A.

Работата може да продължи в посока повишаване на бързодействието на ADAO при големи стойности на N.

ЛИТЕРАТУРА

1. Kolchakov K., V. Monov. Comparative Analysis of a Class of Algorithms for Traffic Management in a Crossbar Commutator with Respect to Complex Performance, Speed and Memory, Problems of Engineering Cybernetics and Robotics, Vol.66, pp.53- 62 ,Sofia, 2015, ISSN 0204-9848. Bulgarian Academy of Sciences.
2. Kolchakov K., "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.
3. 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.

4. Wanjari P., A. Chonhari, 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).

5. Tashev T. Computing 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.