

**Walenty ONISZCZUK**

Politechnika Białostocka, Wydział Informatyki

ul. Wiejska 45A, 15-351 Białystok

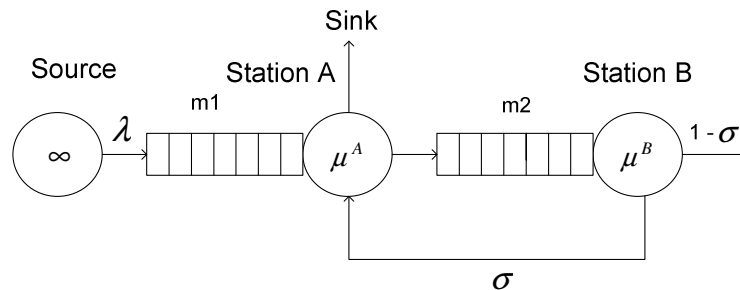
E-mail: w.oniszczyk@pb.edu.pl

## **Symulacja zjawisk zakleszczeń w systemach wielokomputerowych**

### 1 Wstęp

Wieloetapowa realizacja zadań na kolejnych serwerach jest zjawiskiem interesującym z punktu widzenia dostępności użytkownika do zasobów systemów i sieci komputerowych. W takich systemach wielokomputerowych częstym zjawiskiem jest potrzeba powtórnej realizacji zadań (ang. *feedback*) w tych samych serwerach. Biorąc pod uwagę ograniczoną pojemność buforów serwerów, tutaj mogą pojawiać się zjawiska wzajemnych ich blokad i zjawiska zakleszczeń (ang. *deadlock*), gdy serwery wzajemnie blokują swoje transmisje danych.

Uproszczony model układu stanowisk obsługi (serwerów) pokazany jest na poniższym rysunku.



Rys. 1. Trójwęzłowy model z zakleszczeniami i blokadami

Fig. 1. Three-station model with deadlocks and blocking

W tak zdefiniowanej konfiguracji węzłów obsługi (serwerów) zadania, po zakończeniu obsługi w węźle B, opuszczają sieć serwerów z prawdopodobieństwem  $1 - \sigma$  lub wracają do węzła A (*feedback*) dla kolejnej obsługi z prawdopodobieństwem  $\sigma$ . Zakłada się, że powracające zadania do serwera A, mają pierwszeństwo obsługi i obsługiwane są według nierugującego priorytetu (ang. *non-preemptive priority scheme*). Praktycznie oznacza to, że zadania powracające nie mogą być przyjęte do obsługi, jeżeli serwer A jest zajęty bieżącą obsługą kolejnych zadań i muszą czekać do zakończenia w nim obsługi, czasowo pozostając w serwerze B, jednocześnie blokując go. Po dodatkowej obsłudze w węźle A zadania wychodzą z systemu (ang. *sink*). Między serwerami A i B jest bufor o ograniczonej pojemności i jeżeli wystąpi jego przepełnienie, to blokowany jest serwer A. Podobnie, gdy bufor  $m1$  jest pełny, czasowo blokowany jest transfer ze źródła (blokada źródła).

Zadania przetrzymywane w czasie blokad znajdują się w swoich węzłach, lecz wznowienie transmisji i dalsza ich realizacja zależą od pracy innych węzłów, więc można je traktować, jako ulokowane w dodatkowych (wirtualnych) miejscach buforów serwerów  $A$  i  $B$  (jest to ważne przy tworzeniu pełnego grafu stanów takiego układu stanowisk obsługi).

W takiej konfiguracji sieci, gdzie występują zjawiska blokad i powtórnej obsługi zadań, mogą pojawiać się bardzo niekorzystne zjawiska wzajemnych zakleszczeń serwerów i przerwania transmisji zadań między nimi. Aby to zobrazować, załóżmy, że węzeł  $A$  jest blokowany przez węzeł  $B$  (pełny bufor  $m_2$ ) i zadanie, którego realizacja zakończyła się w serwerze  $A$ , musi w nim pozostać i czekać na miejsce w buforze  $m_2$  (transmisja czasowo wstrzymana aż do zakończenia realizacji zadania, które jest w przetwarzaniu w serwerze  $B$ ). Z kolei jeżeli zadanie kończące realizację w węźle  $B$  potrzebuje dodatkowej obsługi w serwerze  $A$ , który aktualnie przetrzymuje zablokowane własne zadanie, to zadanie z serwera  $B$  też zostanie zablokowane i musi pozostać w tym serwerze (serwer  $A$  jest w tym momencie zajęty). Jest to klasyczne zjawisko zakleszczenia (ang. *deadlock*) i całkowitego przerwania transmisji zadań między dwoma serwerami (węzłami obsługi). W takich przypadkach konieczne jest wyposażenie serwerów w programy monitorujące i wykrywające zjawiska zakleszczeń, które dodatkowo mogą realizować wzajemną wymianę zadań między tymi serwerami, likwidując kolejne zakleszczenia.

## 2 Model matematyczny

Do serii eksperymentów z takim układem serwerów (węzłów obsługi), w których możliwe są zjawiska zakleszczeń, przyjęto model matematyczny, szerzej opisany w [5], w którym przyjęto, że zewnętrzny strumień zadań do tandemu serwerów jest strumieniem Poissona z intensywnością  $\lambda$ , a czas obsługi w węzłach jest zmienną losową o rozkładzie wykładniczym, z wartością średnią odpowiednio  $1/\mu_1^A$ ,  $1/\mu_2^A$  (dla powtórnej priorytetowej obsługi) oraz  $1/\mu^B$ , gdzie  $\mu$  to intensywność obsługi. Dodatkowo przyjmuje się, że czas rozwiązania spraw zakleszczeń jest też zmienną losową o rozkładzie wykładniczym o wartości średniej równej  $1/\mu^d$ .

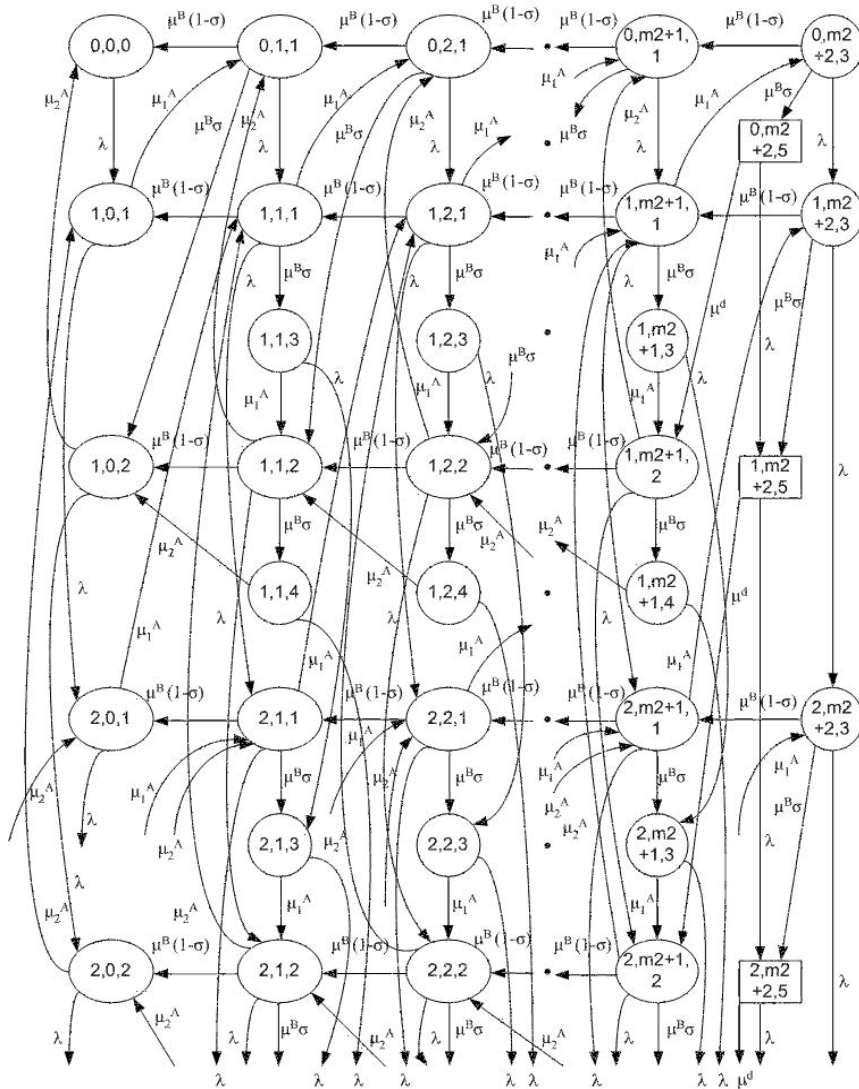
Wychodząc z przyjętych założeń, model węzłów obsługi można opisać standardowym łańcuchem Markowa, z ciągłym czasem i dyskretną przestrzenią stanów. Jeżeli założymy, że istnieją tu stany ustalone, to opisujący model łańcuch Markowa posiada stacjonarne prawdopodobieństwa dla wszystkich rozróżnialnych i możliwych stanów (graf stanów ma ograniczoną ich liczbę).

Procedura szczegółowej analizy działania serwerów połączonych szeregowo i stworzenia analitycznego ich modelu składa się z następujących etapów:

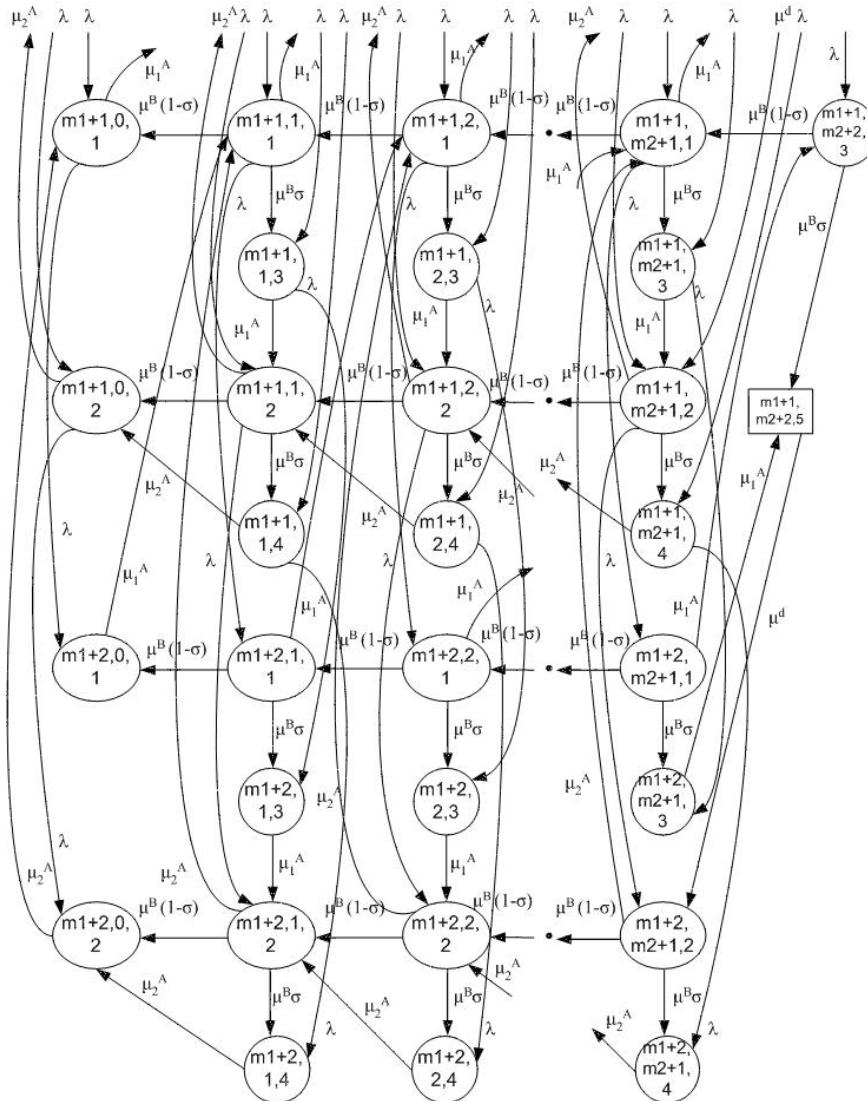
- a) zdefiniowanie wszystkich fizycznie możliwych stanów modelu i stworzenie ich pełnego grafu,
- b) określenie wszystkich możliwych przejść międzystanowych oraz podanie ich intensywności (gęstości prawdopodobieństw przejść),
- c) rozwiązanie układu równań liniowych dla obliczenia stacjonarnych prawdopodobieństw zdefiniowanych stanów modelu.

Podstawową czynnością analizy układu połączonych serwerów jest utworzenie pełnego grafu stanów systemu i określenie w nim intensywności przejść. Dla pełnego grafu stanów modelu przyjęto, że każdy stan można opisać przez trzy indeksy  $(i, j, k)$ , gdzie

przez  $i, j$  oznaczono liczbę zadań w pierwszym i drugim węźle, a indeks  $k$  to stan serwerów. I tak, gdy  $k = 0$ , to mamy przestój serwerów,  $k = 1$  – obsługę zwykłą,  $k = 2$  – obsługę priorytetową,  $k = 3$  – blokadę serwera,  $k = 4$  – blokadę i obsługę priorytetową w innym serwerze,  $k = 5$  – zakleszczenie. Pełny graf rozróżnialnych stanów modelu pokazany jest na rysunkach 2 i 3 [5].



Rys. 2. Graf stanów modelu (część pierwsza)  
 Fig. 2. Model state diagram (first part)



Rys. 3. Graf stanów modelu (część druga)  
 Fig. 3. Model state diagram (second part)

Wychodząc z grafu stanów, można wypisać układy równań dla stanów ustalonych, pokazujące ewolucję systemu.

I tak dla stanów z zakleszczeniami mamy:

$$(\lambda + \mu^d) \cdot p_{0,m2+2,5} = \mu^B \cdot \sigma \cdot p_{0,m2+2,3} \quad \text{dla } i = 0, j = m2+2, k = 5;$$

$$\begin{aligned}
 (\lambda + \mu^d) \cdot p_{i,m2+2,5} &= \lambda \cdot p_{i-1,m2+2,5} + \mu^B \sigma \cdot p_{i,m2+2,3}; \\
 &\text{dla } i = 1, \dots, m1-1, j = m2+2, k = 5; \\
 \mu^d \cdot p_{m1,m2+2,5} &= \lambda \cdot p_{m1-1,m2+2,5} + \mu^B \sigma \cdot p_{m1,m2+2,3} \\
 &\text{dla } i = m1, j = m2+2, k = 5; \\
 \mu^d \cdot p_{m1+1,m2+2,5} &= \mu^B \sigma \cdot p_{m1+1,m2+2,3} + \mu_1^A \cdot p_{m1+2,m2+1,3} \\
 &\text{dla } i = m1 + 1, j = m2+2, k = 5.
 \end{aligned}$$

Układy równań dla stanów z blokadami i stanów bez blokad, dokładnie pokazane są w [5]. Kompletny układ równań modelu można rozwiązać metodami numerycznymi, otrzymując stacjonarne prawdopodobieństwa stanów. Otrzymane prawdopodobieństwa podlegają procedurze normalizacyjnej na sumę prawdopodobieństw wszystkich fizycznie możliwych i rozróżnialnych stanów.

Po wyliczeniu stacjonarnych prawdopodobieństw stanów, w dość standardowych procedurach, można obliczyć podstawowe miary wydajności i jakości obsługi [5]. Ważne to jest, gdy końcowym efektem analizy matematycznej są analizatory komputerowe pozwalające prześledzić ewolucje sieci stanowisk obsługi dla wielu wariantów konfiguracji systemu i różnorodnych parametrów wejściowych.

### 3 Przykłady numeryczne

Dla pierwszego modelu i przykładowych obliczeń numerycznych oraz analizy działania sieci stanowisk obsługi wybrano następującą konfigurację:

- a) wymiary buforów:  $m1 = 12, m2 = 6$ ;
- b) intensywność obsługi:  $\mu_1^A = 4.0, \mu_2^A = 3.0, \mu^B = 2.0$ , zakleszczenia:  $\mu^d = 0.4$ ;
- c) intensywność napływu zadań ze źródła  $\lambda$  zmieniana od 0.5 do 5.0 (dla pokazania reakcji modelu na zmieniające się obciążenia serwerów);
- d) prawdopodobieństwo powtórnej obsługi  $\sigma = 0.9$ .

Dla powyższej konfiguracji wyróżniano 456 stanów: 216 bez blokad, 226 z blokadami i 14 z zakleszczeniami, a wybrane parametry jakości obsługi pokazane są na rysunku 4.

Dla drugiego eksperymentu wybrano następującą konfigurację:

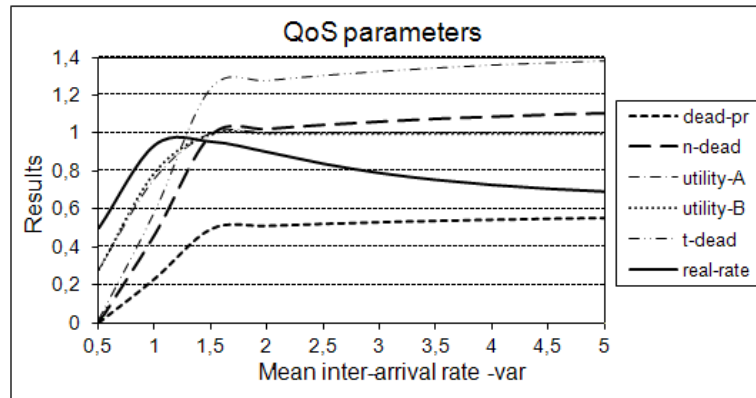
- a) wymiary buforów:  $m1 = 10, m2$  zmieniane od 1 do 10;
- b) intensywność obsługi:  $\mu_1^A = 1.8, \mu_2^A = 1.4, \mu^B = 1.1$ , zakleszczenia:  $\mu^d = 0.3$ ;
- c) intensywność napływu zadań ze źródła  $\lambda = 0.6$ ;
- d) prawdopodobieństwo powtórnej obsługi  $\sigma = 0.8$ .

Dla tej konfiguracji wybrane parametry jakości obsługi pokazane są na rysunku 5.

Dla trzeciego eksperymentu wybrano konfigurację:

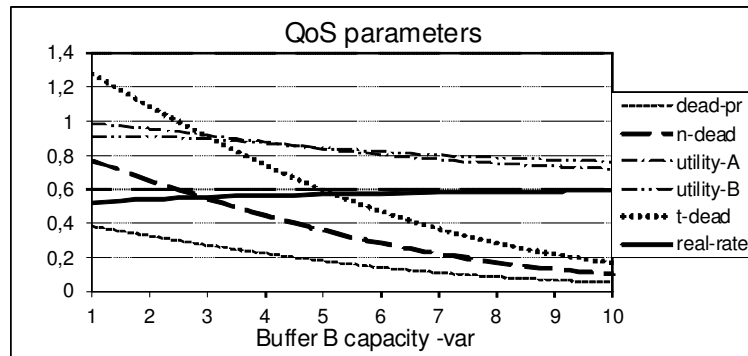
- a) wymiary buforów:  $m1 = 10, m2 = 4$ ;
- b) intensywność obsługi:  $\mu_1^A = 0.5, \mu_2^A = 0.4, \mu^B = 0.25$ , intensywność obsługi zakleszczeń zmieniana od 0.050 do 0.275;
- c) intensywność napływu zadań ze źródła  $\lambda = 0.8$ ;
- d) prawdopodobieństwo powtórnej obsługi  $\sigma = 0.7$ .

Wyniki eksperymentu na rysunku 6.



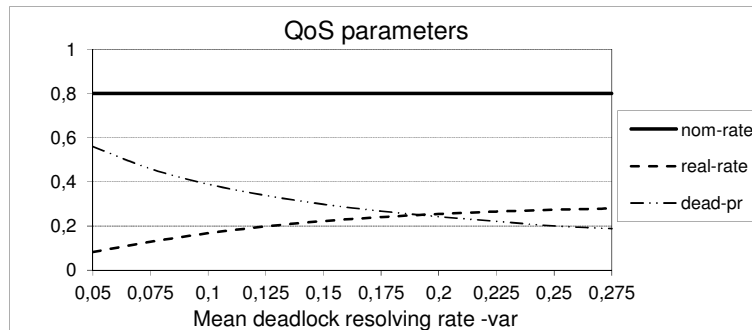
Rys. 4. Parametry jakości obsługi 1, gdzie *dead-pr* - prawdopodobieństwo zakleszczeń, *n-dead* - liczba zakleszczonych zadań, *utility-A* - obciążenie węzła A, *utility-B* - obciążenie węzła B, *t-dead* - czas zakleszczeń, *real-rate* - realna intensywność napływu zadań

Fig. 4. Graph QoS parameters 1, where *dead-pr* - deadlock probability, *n-dead* - number of deadlocked tasks, *utility-A* - station A utilization factor, *utility-B* - station B utilization factor, *t-dead* - deadlock time, *real-rate* - real task input rate



Rys. 5. Parametry jakości obsługi 2, gdzie *dead-pr* - prawdopodobieństwo zakleszczeń, *n-dead* - liczba zakleszczonych zadań, *utility-A* - obciążenie węzła A, *utility-B* - obciążenie węzła B, *t-dead* - czas zakleszczeń, *real-rate* - realna intensywność napływu zadań

Fig. 5. Graph QoS parameters 2, where *dead-pr* - deadlock probability, *n-dead* - number of deadlocked tasks, *utility-A* - station A utilization factor, *utility-B* - station B utilization factor, *t-dead* - deadlock time, *real-rate* - real task input rate



Rys. 6. Parametry jakości obsługi 3, gdzie dead-pr - prawdopodobieństwo zakleszczeń, nom-rate - deklarowana intensywność napływu zadań, real-rate - realna intensywność napływu zadań

Fig. 6. Graph QoS parameters 3, where dead-pr - deadlock probability, nom-rate - declared task input rate, real-rate - real task input rate

#### 4 Podsumowanie

Wyniki przeprowadzonych eksperymentów pokazują, że w dynamicznie zmieniających się warunkach działania sprzężnych serwerów (stanowisk obsługi) sprawy określania wpływu niekorzystnych zjawisk zakleszczeń ich monitorowania i rozwiązywania w sposób znaczący mogą wpływać na warunki pracy i komfort użytkowników takich systemów. Prawdopodobieństwa pojawiania się zjawisk zakleszczeń są minimalne przy niewysokich obciążeniach serwerów i gwałtownie rosną, gdy użytkownicy pracują w okresach spiętrzeń napływu zadań do obsługi. Zwykle takie zjawisko występuje w nowoczesnych systemach i sieciach komputerowych.

#### Literatura

1. Balsamo S., De Nito Persone V., Onvural R.: *Analysis of Queueing Networks with Blocking*, Kluwer Academic Publishers, Boston, 2001
2. Filipowicz B.: *Modelowanie i analiza sieci kolejkowych*, Wydawnictwa AGH, Kraków, 1997
3. Kim C.S., Klimenok V., Tsarenkov G., Breuer L., Dudin A.: The BMAP/G/1->·/PH/1/M tandem queue with feedback and losses, *Performance Evaluation*, vol. 64, pp. 802-818, 2007
4. Martin J.B.: Large Tandem Queueing Networks with Blocking, *Queueing Systems*, vol. 41(1/2), pp. 45-72, 2002
5. Oniszczyk W.: Blocking and Deadlock Factors in Series Linked Servers with HOL Priority Feedback Service, *Polish Journal of Environmental Studies*, vol. 16(5B), pp. 145-151, 2007
6. Oniszczyk W.: An Intelligent Service Strategy in Linked Networks with Blocking and Feedback, *Studies in Computational Intelligence* N. 134 "New Challenges in Applied Intelligence Technologies", N.T. Nguyen, R. Katarzyniak (Eds.), Springer-Verlag, Berlin, Heidelberg, pp. 351-361, 2008

7. Oniszczyk W.: Semi-Markov-based approach for analysis of open tandem networks with blocking and truncation, *International Journal of Applied Mathematics and Computer Science*, vol. 19(1), pp. 151-163, 2009
8. Perros H.G.: *Queuing Networks with Blocking. Exact and Approximate Solution*, Oxford University Press, New York, 1994

### Streszczenie

W niniejszej pracy pokazano możliwości matematycznego modelowania połączonych serwerów, w których możliwe są ich blokady (czasowe przerwania transmisji danych) ze względu na ograniczone wymiary buforów oraz występują zjawiska zakleszczeń (całkowite przerwanie transmisji i pracy serwerów). W prezentowanym modelu występuje tzw. recykling zadań, to jest powtórna ich obsługa. To właśnie powtórna obsługa powoduje powstawanie zakleszczeń. W eksperymentach symulacyjnych zobrazowano skalę tego zjawiska i jego powiązania ze stopniem obciążenia serwerów.

## **Deadlock simulation in multi-computer systems**

### Summary

In this publication, the mathematical modelling of specialized multi-computer systems with blocking and deadlocks are presented. Generally, deadlock means that some process of accumulation of new tasks between two stations is temporarily suspended, when the buffer is full. In this special type of multistage network, at the end of a service in station *B* the task may returns to the station *A* (feedback) for additional service independently of all other events. Of course, in the network with feedback a deadlock may occur. We assume that a deadlock is detected instantaneously and resolved with some delay time by exchanging both the blocked tasks simultaneously. The results of experiments show that depending on the model parameters the mathematical modelling allows for finding the proper deadlock probabilities and adequate rate range for an input stream that guarantees congestion avoidance in the multi-computer systems.