Modele kolejkowe w IT

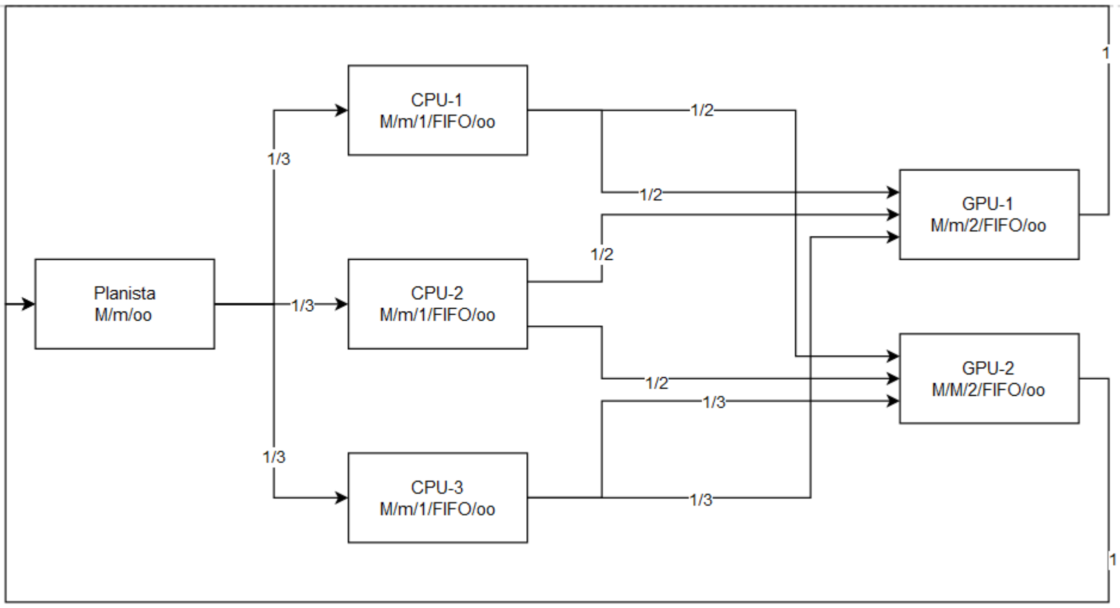
1. Wstęp

Celem jest zamodelowanie zamkniętej sieci kolejkowej BCMP2 dla systemu złożonego z planisty, modułów CPU oraz GPU w którym krąży N klas zgłoszeń. Projekt zakłada utworzenie programu pozwalającego na zbadanie parametrów sieci określenie przez użytkownika.

Założeniami projektu jest wykorzystanie metody SUM oraz przyjęty brak możliwości zmiany przynależności do klasy przez zgłoszenie. Użytkownik musi znać parametry sieci, którą chce zamodelować takie jak: liczba zgłoszeń każdej klasy w systemie, ilość oraz typy systemów w sieci oraz prawdopodobieństwa przejść dla każdej klasy.

2. Opis zagadnienia

Badana w projekcie sieć kolejkowa zakładała się z Planisty będącego systemem typu $M/m/\infty$, trzech CPU będących systemami $M/M/1$ oraz dwóch GPU będących typu $M/M/m/FIFO/\infty$ gdzie m zostało wybrane jako $= 2$. W systemie krążyły 4 klasy zgłoszeń z różnym przetwarzaniem przez systemy.



Rysunek 1. Schemat sieci dla klasy zgłoszeń 1.

W celu znalezienia parametru lambda opisującego każdy system wykorzystana iteracyjną metodę SUM opisaną w instrukcji do ćwiczeń. Następnie parametry dla każdego systemu były wyznaczane zgodnie z wzorami im odpowiadającymi opisanymi w [1], [2].

Opisy klas zgłoszeń:

- ➔ Klasa 1: zgłoszenie musi zostać przetworzone najpierw przez CPU a następnie przez GPU.
- ➔ Klasa 2: zgłoszenie musi zostać przetworzone w kolejności GPU -> CPU.
- ➔ Klasa 3: zgłoszenie musi zostać przetworzone tylko przez CPU.

➔ Klasa 4: zgłoszenie musi zostać przetworzone tylko przez GPU.

Program udostępnia możliwość zdefiniowania różnej ilości klas.

Parametry dla systemów $M/M/\infty$ obliczane są z wykorzystaniem wzorów [1]:

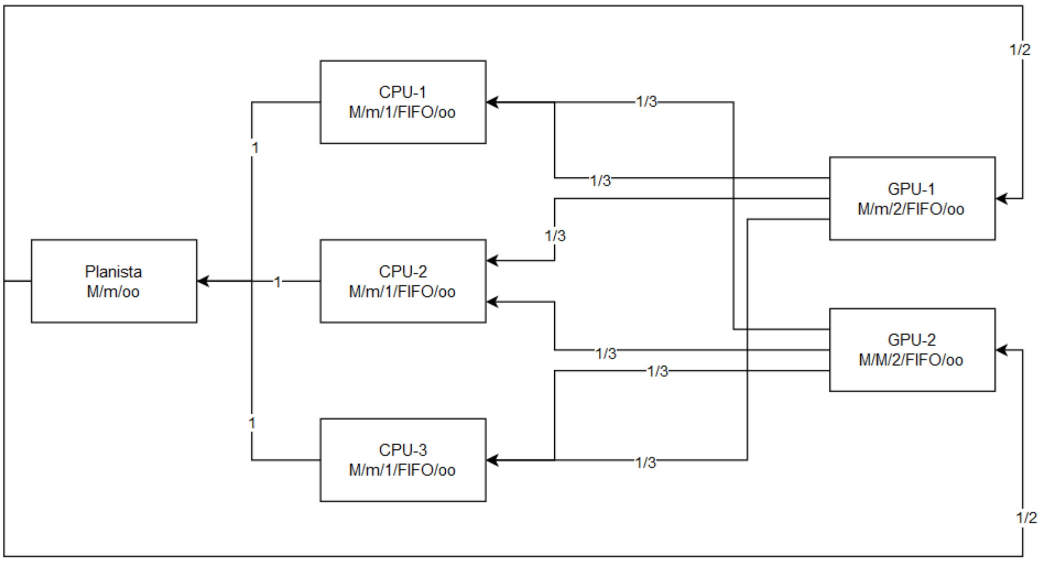
Prawdopodobieństwa k zgłoszeń w systemie:

$$\pi_0 = e^{-\rho}$$

$$\pi_k = \frac{\rho^k}{k!}\pi_0$$

Oraz średni czas w systemie:

$$\bar{T} = \frac{1}{\mu}$$



Rysunek 2. Schemat systemu dla zgłoszeń klasy 2

Parametry dla systemów $M/M/m$ obliczane są z wykorzystaniem wzorów [2]:

Prawdopodobieństwo k zgłoszeń w systemie:

$$\pi_0 = \left[\sum_{k=0}^{m-1} \frac{(m\rho)^k}{k!} + \frac{(m\rho)^m}{m!}\frac{1}{1-\rho}\right]^{-1}$$

$$\pi_k = \begin{cases} \pi_0 \dfrac{(m\rho)^k}{k!} \; dla\; 0 \le k \le m \\ \pi_0 \dfrac{\rho^k m^m}{m!} \; dla\; k \ge m \end{cases}$$

Średnia długość kolejki:

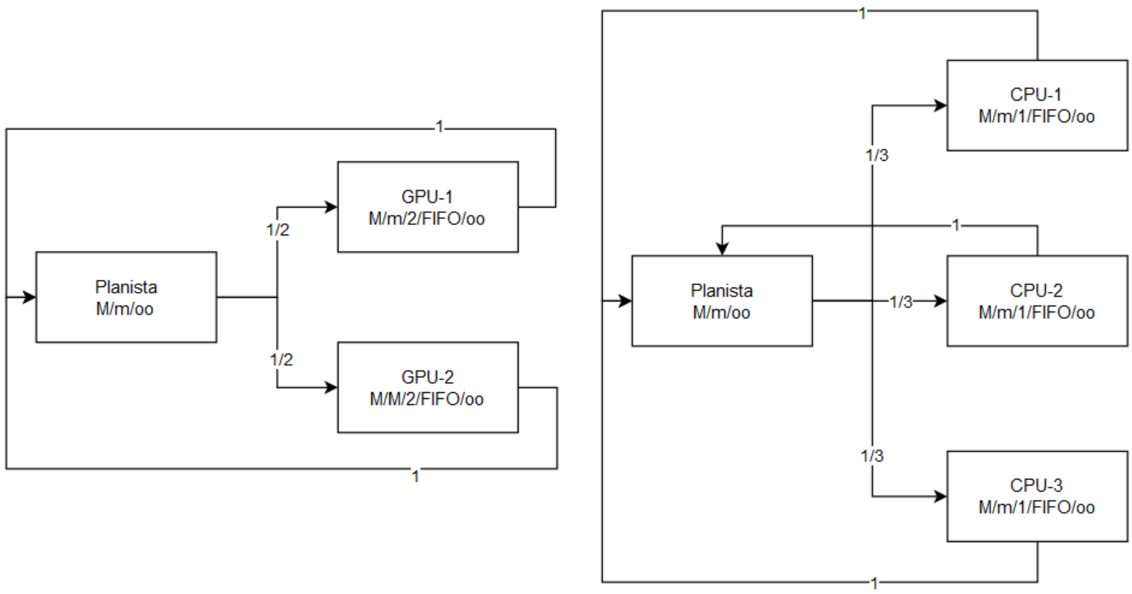
$$\bar{Q} = \frac{\rho}{1-\rho}\frac{(m\rho)^m}{m!\,(1-\rho)}\pi_0$$

Średnia czas w kolejce:

$$\bar{W} = \frac{\bar{Q}}{\lambda}$$

Oraz średni czas odpowiedzi:

$$\bar{T} = \frac{\bar{K}}{\lambda}$$



Rysunek 3. Schemat sieci dla klasy 3(prawo) oraz 4(lewo)

Gdzie K_{ir} wyznaczane jest z wykorzystaniem funkcji:

$$f_{ir}(\lambda_{ir}) = \bar{K}_{ir} = \begin{cases} \dfrac{\rho_{ir}}{1 - \dfrac{K-1}{K}\rho_i}, & Typ\,1,2,4\,(m_i = 1) \\ m_i\rho_{ir} + \dfrac{\rho_{ir}}{1 - \dfrac{K-m_i-1}{K-m_i}\rho_i} \cdot P_{m_i}, & Typ\,1\,(m_i > 1) \\ \dfrac{\lambda_{ir}}{\mu_{ir}}, & Typ\ 3 \end{cases}$$

Macierze prawdopodobieństw przejść zostały stworzone na podstawie Rysunków 1,2 oraz 3.

K1:

0	0	0	0	1	1
1/3	0	0	0	0	0
1/3	0	0	0	0	0

1/3	0	0	0	0	0
0	1/2	1/2	1/2	0	0
0	1/2	1/2	1/2	0	0

K2:

0	1	1	1	0	0
0	0	0	0	1/3	1/3
0	0	0	0	1/3	1/3
0	0	0	0	1/3	1/3
1/2	0	0	0	0	0
1/2	0	0	0	0	0

K3:

0	1	1	1	0	0
1/3	0	0	0	0	0
1/3	0	0	0	0	0
1/3	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

K4:

0	0	0	0	1	1
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
1/2	0	0	0	0	0
1/2	0	0	0	0	0

Macierz stosunku wizyt e jest obliczana z wykorzystaniem funkcji opartej na wzorze:

$$e_i = \sum_{j=1}^{N} e_j p_{ji}.$$

Przy założeniu, że $e_1 = 1$.

1	1/3	1/3	1/3	1/2	1/2
1	1/3	1/3	1/3	1/2	1/2
1	1/3	1/3	1/3	0	0
1	0	0	0	1/2	1/2

3. Dokumentacja

Wymagane środowisko: Python 3.6,
Wymagane biblioteki: numpy, math, matplotlib, sympy

Aplikacja zawiera w sobie 2 programy, jeden pozwalający na obliczenia dla systemu zamkniętego w przypadku, kiedy posiadamy tylko jedną klasę zgłoszeń oraz drugi oferujący możliwość zbadania sieci z wieloma zgłoszeniami. Oba programy są odpowiednio podpisane.

Aby uruchomić program użytkownik musi wypełnić macierze prawdopodobieństwa przejść, których może być dowolna ilość. Następnie należy uruchomić funkcję get_e() z parametrami:

$get\_e(ilość\ klas\ zgłoszeń, ilość\ systemów, macierz\ przejść\ klasy\ 1, macierz\ przejść\ klasy\ 2, \ldots, macierz\ przejść\ klasy\ N)$

Program następnie wyświetli otrzymaną macierz stosunku wizyt.

Następnie użytkownik musi wypełnić parametry inicjalizacyjne programu:

K – wektor zgłoszeń k-tej klasy w sieci
class_num – ilość klas zgłoszeń w sieci
sys_num – ilość systemów w sieci
mu – macierz μ
mi - m_i, ilość serwerów w i-tym systemie
typ – typ systemu, jeśli i-ty system należy to typu 3 należy wpisać wartość 2, w przeciwnym razie należy wpisać wartość 1.

Następnie należy uruchamiać po kolei fragmenty kodu. Pozwoli nam to uzyskać parametry takie jak:

- ➔ Stabilność sieci – jeśli któryś z systemów wykazuje niestabilność musimy zmienić parametry, ponieważ pozostałe parametry będą niemiarodajne.
- ➔ Średnia liczba zgłoszeń w poszczególnych systemach zostanie wyświetlona na wykresie z podziałem na system oraz klasę zgłoszeń.
- ➔ Macierz λ_{ir}
- ➔ Średnia długość kolejki dla każdego systemu $\bar{Q}$
- ➔ Wykresy prawdopodobieństw π_k dla każdego systemu
- ➔ Macierz średnich czasów odpowiedzi $\bar{T}$

4. Eksperymenty

Dla opisanego w zagadnieniach modelu sieci przyjęto dodatkowo założenia:

- ➔ Liczba zgłoszeń w systemie: K1=2, K2= 3, K3=2, K4=5.
- ➔ Macierz μ:

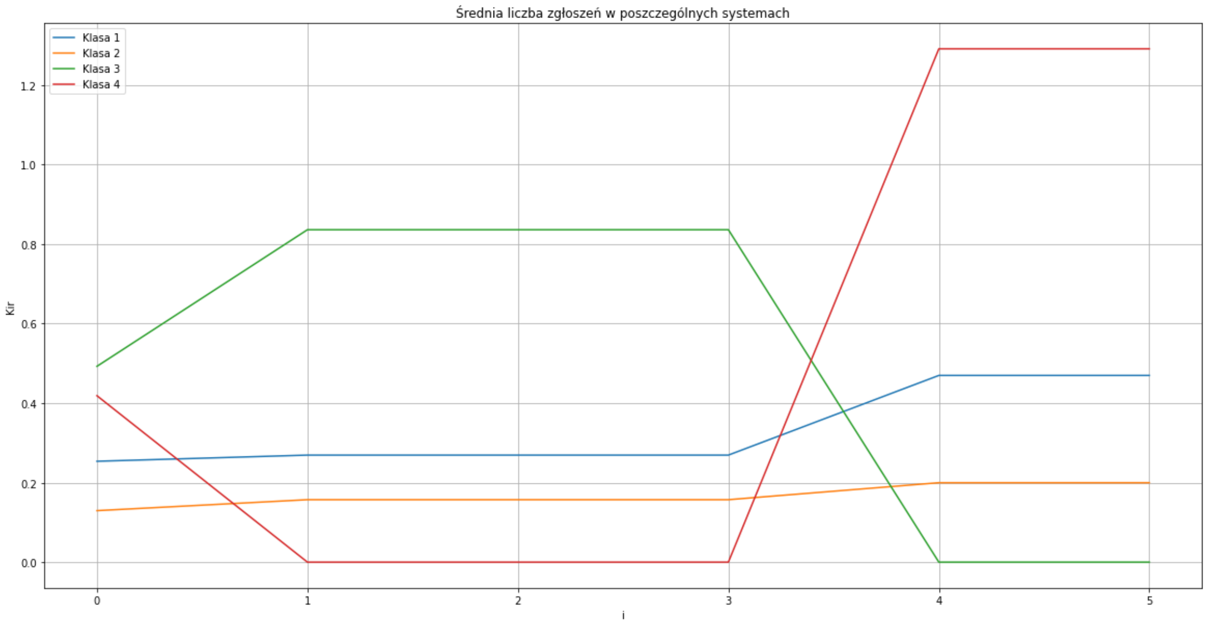
	System 1	System 2	System 3	System 4	System 5	System 6
Klasa z. 1	120	80	80	80	50	50
Klasa z. 2	120	70	70	70	60	60
Klasa z. 3	120	50	50	50	-	-
Klasa z. 4	120	-	-	-	30	30

Następnie po uruchomieniu programu, otrzymaliśmy wyniki:

Stabilności systemów:

```
System 0  is stable
System 1  is stable
System 2  is stable
System 3  is stable
System 4  is stable
System 5  is stable
```

Średnią liczbę zgłoszeń każdej klasy w poszczególnych systemach:



Wartości λ_{ir}:

```
Lambda ir:
[[30.442 10.147 10.147 10.147 15.221 15.221]
 [15.541  5.18   5.18   5.18   7.77   7.77 ]
 [59.092 19.697 19.697 19.697  0.     0.   ]
 [50.217  0.     0.     0.    25.109 25.109]]
```

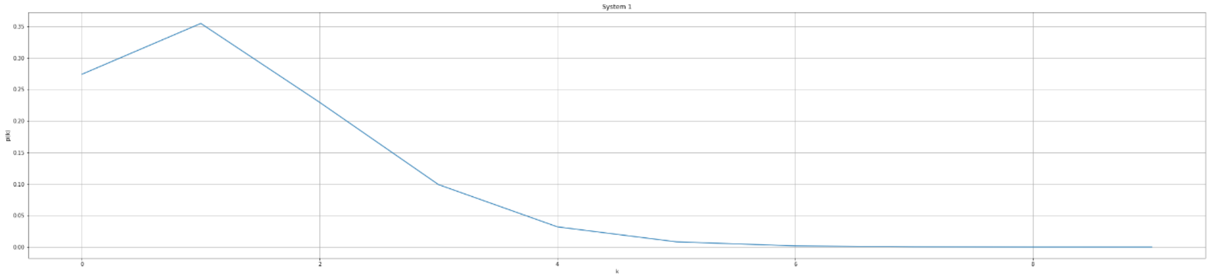
Średnie długości kolejki:

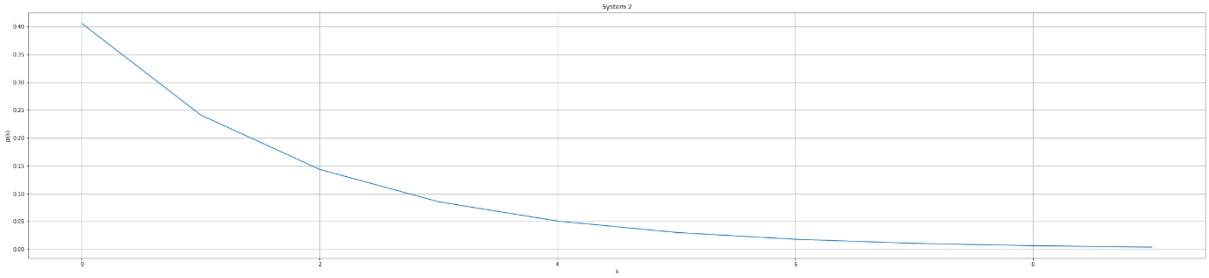
```
Średnia długość kolejki:
    System  0 :  0.0
    System  1 :  0.873053271834679
    System  2 :  0.873053271834679
    System  3 :  0.873053271834679
    System  4 :  0.01789268868488875
    System  5 :  0.01789268868488875
```

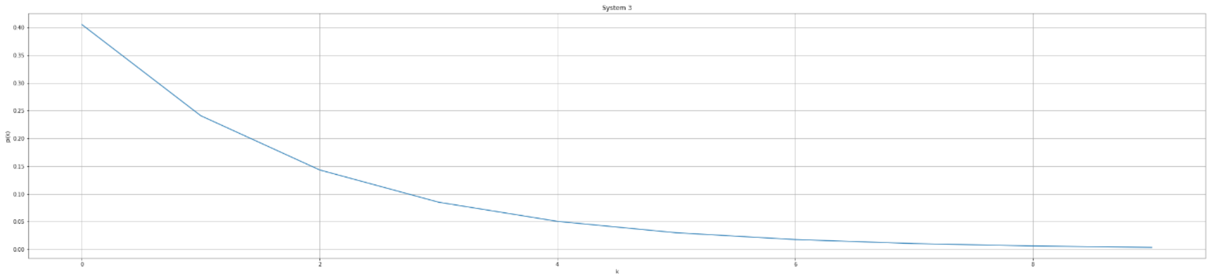
Średnie czasy odpowiedzi:

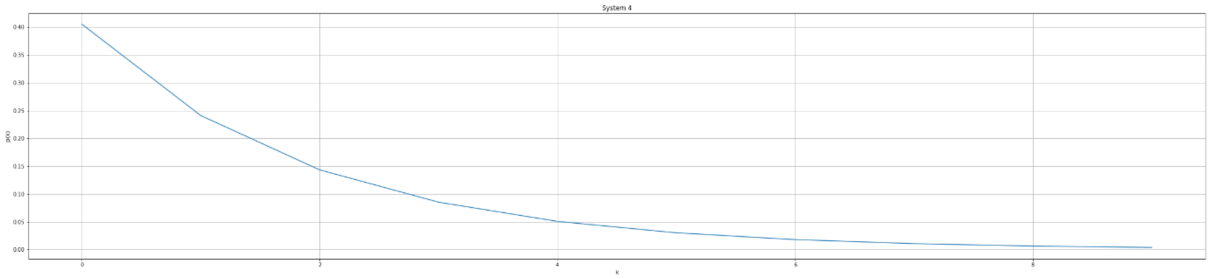
```
Średni czas odpowiedzi:
[[0.008 0.008 0.008 0.008]
 [0.027 0.03  0.042   nan]
 [0.027 0.03  0.042   nan]
 [0.027 0.03  0.042   nan]
 [0.031 0.026   nan 0.051]
 [0.031 0.026   nan 0.051]]
```

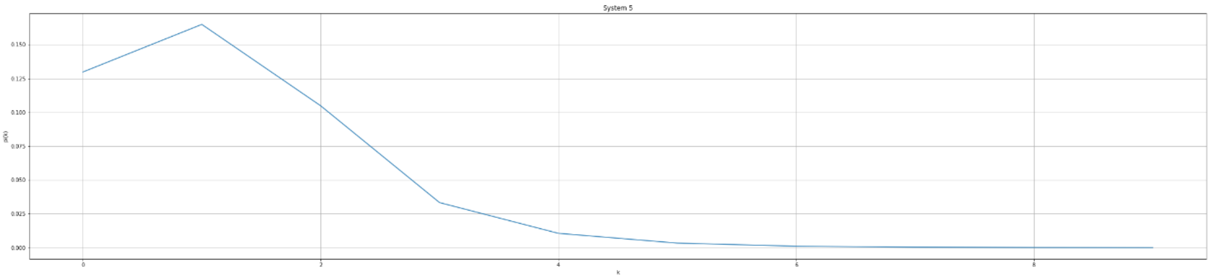
Prawdopodobieństwa π_k:

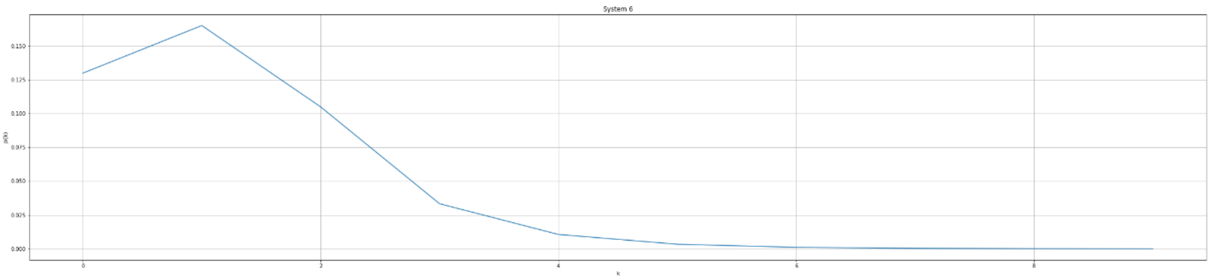












5. Podsumowanie

Najcięższym punktem projektu jest dobranie parametrów tak aby system był stabilny. Podczas obliczeń mogą wystąpić 2 problemy, pierwszy w którym brak zbieżności wyników może uniemożliwić obliczanie parametrów systemów w sieci oraz drugi – system nawet po obliczeniu lambdy jest niestabilny. Jeśli obliczenia przeciągają się, oznacz to, że najprawdopodobniej rozwiązanie dla badanego systemu nie istnieje, co czyni go niestabilnym.

6. Literatura:

[1]. B. FILIPOWICZ and J. KWIECIEŃ, Queueing systems and networks. Models and applications, BULLETIN OF THE POLISH ACADEMY OF SCIENCES TECHNICAL SCIENCES Vol. 56, No. 4, 2008

[2]. Bolch, S. Greiner, H. de Meer, and K.S. Trivedi, Queueing Networks and Markov Chains. Modelling and Performance Evaluation with Computer Science Applications, John Wiley&Sons, Inc., London, 1998