ZÁPADOČESKÁ
UNIVERZITA
V PLZNI

Semestrální práce z KIV/VSP

Diskrétní simulační modely

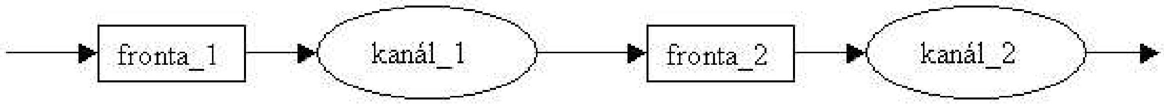
Martin Sloup, A08N0111P

msloup@students.zcu.cz

20.12.1984, zadání č. 3.2

Zadání

Pro zadanou otevřenou síť front vypočítejte odhad hodnot L_q a T_q. Časové intervaly mezi vstupy požadavků mají exponenciální rozdělení s parametrem **$\boldsymbol{\lambda}$** = 0.5, oba kanály obsluhy mají gaussovsky rozdělenou dobu obsluhy se středními hodnotami $T_{s_1} = 1$, $T_{s_2} = 1{,}33$.



Dále určete střední frekvenci výstupního toku požadavků.

Zadanou otevřenou síť se pokuste nasimulovat s využitím programových nástrojů C-Sim, J-Sim nebo Cpp-Sim. Hodnoty získané ze simulace porovnejte s teoretickými hodnotami získaných z odhadu.

Teoretický výpočet

Ze zadání zajisté vyplívá, že se jedná o případ SHO označený v Kendallově klasifikaci jako M/G/1. Pokud se zamyslíme nad zadání, narazíme na dva problémy. První z nich je, že neznáme koeficient variace doby obsluhy C_s pro výpočet střední délky fronty SHO M/G/1 :

$$L_w = \frac{\varrho^2}{2(1-\varrho)}(1 + C_s^2)$$

Pro gaussovskou dobu obsluhy se variance doby obsluhy pohybuje v rozmezí $0 < C_s < 1$. $C_s = 0$ odpovídá shodné (nenáhodné) době obsluhy (tj. M/D/1) a střední délka fronty L_w vyjde přibližně dvakrát menší než pro SHO M/M/1. Pokud budeme uvažovat nejhorší případ, tj. $C_s = 1$, jedná se o variantu SHO M/M/1.

Ze střední délky fronty se již dají jednoduchým způsobem spočítat hodnoty L_q (střední celkový počet požadavků v SHO):

$$L_q = L_w + L_s = L_w + m \cdot \lambda \cdot T_s \text{ [1]}$$

A pomocí Littleový vztahů i T_q (střední doba průchodu požadavků elementárním SHO – střední doba odezvy):

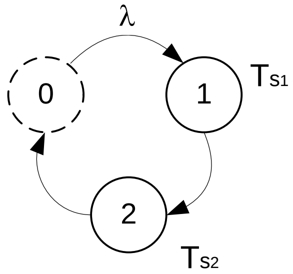
$$T_q = \frac{L_q}{\lambda}$$

Druhým problémem je, že pro učení celkového T_q obou SHO je zapotřebí znát celkové L_q. To je možné spočítat použitím Jacksonového teorému:

- Všechny toky požadavků z okolí do sítě mají poissonovský charakter.
- Všechny obslužné uzly mají exponenciální rozdělení doby obsluhy se střední hodnotou T_{s_i}.
- Po ukončení obsluhy v uzlu i přechází požadavek zcela náhodně do dalšího uzlu j (tj. s pravděpodobností p_{ij}), přitom přechod se uskuteční bez zpoždění.

[1] V našem případě je počet kanálů 1, tedy $m = 1$.

Z toho vyplívá, že celkové L_q lze spočítat jen pro SHO M/M/1. Proto L_q a T_q spočítáme pro nejhorší případ (tj. M/M/1) s vědomím, že výsledné hodnoty pro SHO M/G/1 budou nižší. Nyní již tedy výpočet pro SHO M/M/1.



Obrázek 1 - Markovská reprezentace našeho SHO

Nejprve vypočítáme tok uzly:

$$\Lambda_1 = \Lambda_0 p_{01} = \lambda = 0{,}5$$
$$\Lambda_2 = \Lambda_1 p_{12} = \lambda = 0{,}5$$

Následně i zatížení:

$$\varrho_1 = \Lambda_1 T_{s_1} = \lambda T_{s_1} = 0{,}5 \cdot 1 = 0{,}5$$
$$\varrho_2 = \Lambda_2 T_{s_2} = \lambda T_{s_2} = 0{,}5 \cdot 1{,}33 = 0{,}665$$

Platí podmínka stacionarity, tj. $\varrho < 1$. Spočteme L_q (střední celkový počet požadavků v SHO) pro každý uzel. Nejdříve však odvodíme tento vzorec použitím střední délky fronty L_w, variance doby obsluhy $C_s = 1$ pro SHO M/M/1 a vzorce $\varrho = \lambda T_s$:

$$L_q = \frac{\varrho^2}{2(1-\varrho)}(1 + C_s^2) + \lambda T_s = \frac{\varrho^2}{1-\varrho} + \varrho = \frac{\varrho}{1-\varrho}$$
$$L_{q_1} = \frac{\varrho_1}{1-\varrho_1} = \frac{0{,}5}{1-0{,}5} = 1$$
$$L_{q_2} = \frac{\varrho_2}{1-\varrho_2} = \frac{0{,}665}{1-0{,}665} = 1{,}985$$

Použitím Jacksonova teorému vypočítáme L_q celého našeho SHO:

$$L_q = L_{q_1} + L_{q_2} = 1 + 1{,}985 = 2{,}985$$

Pomocí jednoho z Littleových vzorců získáme T_q (střední dobu odezvy):

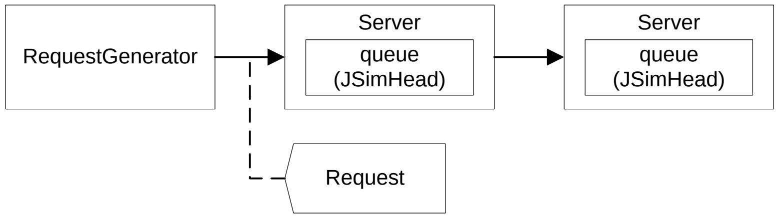
$$T_q = \frac{L_q}{\lambda} = \frac{2{,}985}{0{,}5} \doteq 5{,}970$$

Nyní již zbývá vypočítat střední frekvenci výstupního toku. Tu vypočítáme z převrácené hodnoty střední doby obsluhy (v sekundách) druhého uzlu:

$$\mu = \frac{1}{T_{s_2}} = \frac{1}{1{,}33} = 0{,}752 Hz$$

Simulace sítě

Simulovaná síť byla rozdělena na několik prvků reprezentující třídu v J-Sim:



Obrázek 2 - Diagram komunikace mezi třídami

Třída `RequestGenerator` provádí generování požadavků. Doba mezi vygenerováním dalšího požadavku je prováděna pomocí generátoru náhodných čísel založeném na exponenciálním rozdělení. Parametr λ generátoru je v našem případě 0.5. Vygenerovaný požadavek je předán severu serveru pomocí metody `addToQueue`.

Třída Server zpracovává příchozí požadavky zařazené do fronty pomocí metody `addToQueue`. Pokud ve frontě není požadavek, provede své uspání. Server svou dobu zpracování požadavku a předání dál simuluje normálním (Gaussovo) rozdělením. Vstupními parametry normálového rozdělení byly použity následující hodnoty (střední hodnota a rozptyl):

$$E(X) = \mu = \frac{1}{T_s}$$
$$D(X) = \sigma = \frac{C_s}{T_s}$$

Pro první sever server jsou tyto hodnoty $\mu = 1$ *a* $\sigma = 0{,}5825$*; pro druhý* $\mu = 1{,}33$ *a* $\sigma = \frac{0{,}5825}{1{,}33} \doteq$ $\doteq 0{,}4379$*. Parametr* C_s *byl zvoleno tak, aby rozptyl* σ *vycházel v obou případech okolo hodnoty 0,5; tedy* $C_s = 0.5825$*.*

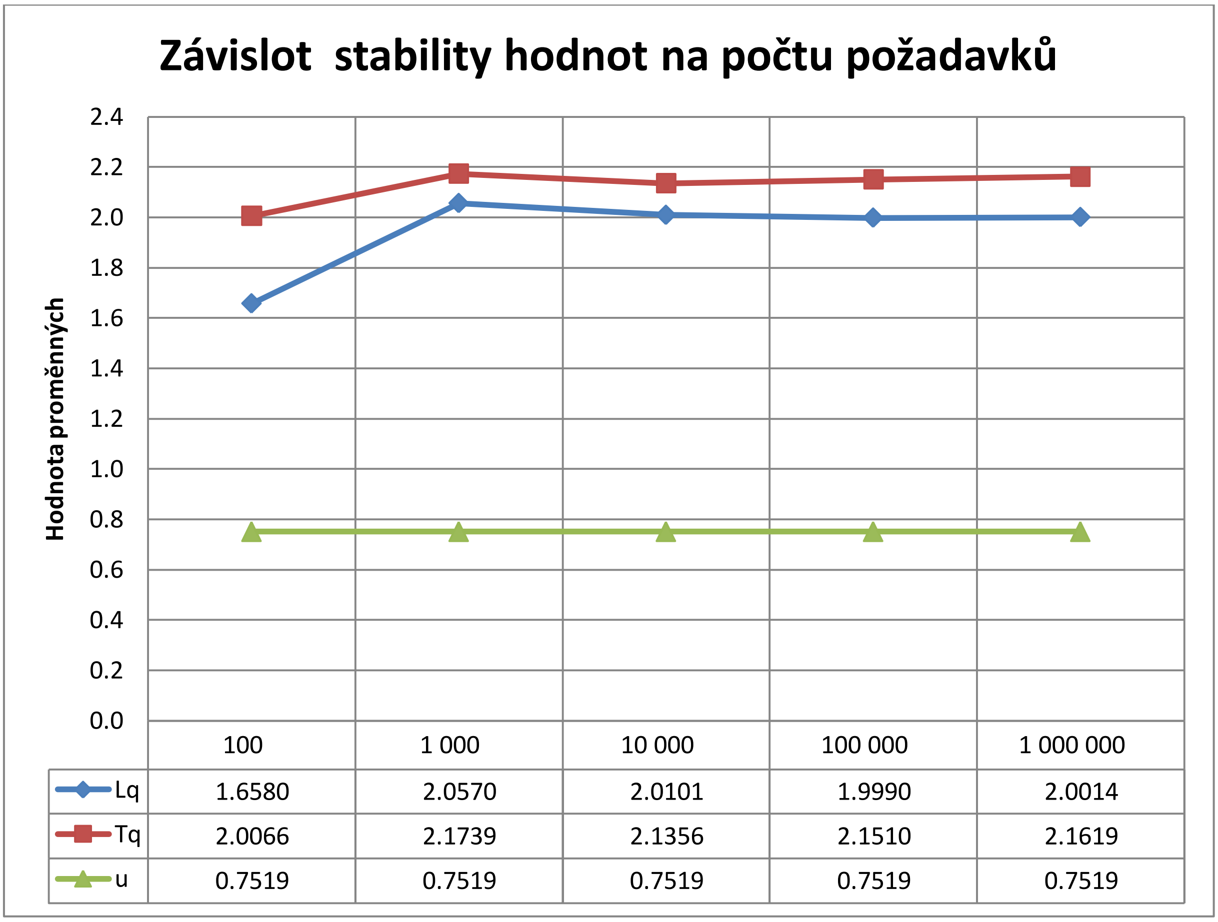
Po zpracování požadavku je požadavek předán druhému serveru stejně jako v případě třídy `RequestGenerator` pomocí metody `addToQueue`.

Třída `Request` je zde jakýmsi obalem nad časem vytvořením požadavku. Prakticky jinou funkci neposkytuje, než tento čas.

Výsledné hodnoty simulace vypisuje na obrazovku třída `Main`, která zároveň vytváří objekty simulace a nastavuje propojení mezi nimi.

Výsledky simulace

Simulace byla provedena pro 100, 1 000, 10 000 a 100 000 generovaných požadavků, aby byla vidět stabilita hodnot. Z grafu na obrázku 3 jasně vyplívá, že čím více vygenerovaných požadavků, tím vyší přesnost výsledků pro náš testovací případ.



Obrázek 3 - Graf stability hodnot na počtu požadavků

Závěr

Cílem bylo vypočítat hodnoty L_q, T_q a μ pro SHO M/G/1. Pro neznalost hodnoty variance doby obsluhy (C_s) jsme provedli výpočet hodnot pro nejhorší případ, tj. pro SHO typu M/M/1. Hodnoty pro M/G/1 budou tedy nižší než námi vypočítané pro nejhorší případ.

Tedy přesněji, teoretický výpočet nám dává odhady následujících výsledků:

$$L_q(pro\ \ M/G/1) < 2{,}985$$
$$T_q(pro\ \ M/G/1) < 5{,}970$$
$$\mu = 0{,}752Hz$$

Porovnáme-li výsledky hodnot ze simulace pro 1 000 000 požadavků, vychází nám, že naše odhady jsou správné a tedy pro reálný případ dostaneme menší hodnoty:

	Teoretický výpočet	Porovnání	Výsledek ze simulace
L_q	2,985	>	2,001
T_q	5,970	>	2,162
μ	0,752 Hz	=	0,752 Hz

Přílohy

Výstup simulace pro 100 požadavků

```
Simulace prerusena v case 165.79774744686367
Odeslanych pozadavku: 100
Statistika front:
  Server 1 - Queue: Lw = 0.9086443441886476
  Server 2 - Queue: Lw = 0.603265744704561
Statistika serveru:
  Server 1: Tq = 1.5065118549681045
  Server 2: Tq = 2.5067128708068145
Celkova statistika:
  Lq = 1.6579774744686366
  Tq = 2.0066123628874597
  u  = 0.7518796992481203
```

Výstup simulace pro 1 000 požadavků

```
Simulace prerusena v case 2057.0390450978366
Odeslanych pozadavku: 1000
Statistika front:
  Server 1 - Queue: Lw = 0.8380143714766647
  Server 2 - Queue: Lw = 0.43757400217339576
Statistika serveru:
  Server 1: Tq = 1.7238282824806221
  Server 2: Tq = 2.6239350900710243
Celkova statistika:
  Lq = 2.0570390450978366
  Tq = 2.173881686275823
  u  = 0.7518796992481203
```

Výstup simulace pro 10 000 požadavků

```
Simulace prerusena v case 20100.942532198696
Odeslanych pozadavku: 10000
Statistika front:
  Server 1 - Queue: Lw = 0.8352002535448966
  Server 2 - Queue: Lw = 0.4544262968514945
Statistika serveru:
  Server 1: Tq = 1.6788312299383745
  Server 2: Tq = 2.5922709177515504
Celkova statistika:
  Lq = 2.0100942532198696
  Tq = 2.135551073844962
  u  = 0.7518796992481203
```

Výstup simulace pro 100 000 požadavků

```
Simulace prerusena v case 199900.54796833548
Odeslanych pozadavku: 100000
Statistika front:
  Server 1 - Queue: Lw = 0.843055938152488
  Server 2 - Queue: Lw = 0.4659466855046309
Statistika serveru:
  Server 1: Tq = 1.6852734400464147
  Server 2: Tq = 2.616703417610474
Celkova statistika:
  Lq = 1.9990054796833547
  Tq = 2.1509884288284447
  u  = 0.7518796992481203
```

Výstup simulace pro 1 000 000 požadavků

```
Simulace prerusena v case 2001414.5425024151
Odeslanych pozadavku: 1000000
Statistika front:
  Server 1 - Queue: Lw = 0.8485088850824097
  Server 2 - Queue: Lw = 0.4633911562233045
Statistika serveru:
  Server 1: Tq = 1.6982180220464451
  Server 2: Tq = 2.625655820978827
Celkova statistika:
  Lq = 2.001414542502415
  Tq = 2.1619369215126363
  u  = 0.7518796992481203
```