



ZÁPADOČESKÁ  
UNIVERZITA  
V PLZNI

# Semestrální práce z KIV/VSP

Simulační model systému hromadné obsluhy

Ivan Habernal, A02226  
e-mail: habernal@students.zcu.cz

# Obsah

|            |   |           |
|------------|---|-----------|
| <b>I</b>   | <b>Zadání</b>                                     | <b>3</b>  |
| <b>1</b>   | <b>Obecné zadání</b>                              | <b>3</b>  |
| 1.1        | Konkrétní zadání . . . . .                        | 3         |
| <b>II</b>  | <b>Teoretický úvod</b>                            | <b>5</b>  |
| <b>2</b>   | <b>Systémy hromadné obsluhy</b>                   | <b>5</b>  |
| 2.1        | Elementární SHO . . . . .                         | 5         |
| 2.1.1      | Vstupní proud požadavků . . . . .                 | 5         |
| 2.1.2      | Fronta požadavků . . . . .                        | 6         |
| 2.1.3      | Kanály obsluhy . . . . .                          | 6         |
| 2.1.4      | Vztahy pro elementární SHO jako celek . . . . .   | 6         |
| 2.1.5      | M/M/1 . . . . .                                   | 7         |
| 2.2        | Otevřené sítě front . . . . .                     | 7         |
| 2.2.1      | Analýza středních frekvencí a zatížení . . . . .  | 7         |
| 2.2.2      | Určení délky front a doby odezvy . . . . .        | 8         |
| <b>3</b>   | <b>Generování náhodných čísel</b>                 | <b>9</b>  |
| 3.1        | Metoda inverzní transformace . . . . .            | 9         |
| 3.2        | Generování čísel s normálním rozdělením . . . . . | 9         |
| <b>III</b> | <b>Matematické řešení</b>                         | <b>10</b> |
| <b>4</b>   | <b>Charakteristiky uzlů</b>                       | <b>10</b> |
| 4.1        | Střední frekvence toku v uzlech . . . . .         | 10        |
| 4.2        | Ověření stacionárního stavu . . . . .             | 10        |
| 4.3        | Střední hodnoty uzlů . . . . .                    | 10        |
| <b>5</b>   | <b>Globální charakteristiky</b>                   | <b>11</b> |
| <b>IV</b>  | <b>Programové řešení</b>                          | <b>12</b> |
| <b>6</b>   | <b>Model sítě a popis balíčků</b>                 | <b>12</b> |
| <b>7</b>   | <b>Ovládání programu</b>                          | <b>13</b> |
| <b>V</b>   | <b>Zhodnocení výsledků</b>                        | <b>13</b> |

|          |  |           |
|----------|--|-----------|
| <b>A</b> | <b>Výpisy výstupů simulace</b>                   | <b>14</b> |
| A.1      | Exponenciální rozdělení . . . . .                | 14        |
| A.2      | Normální rozdělení s koeficientem 0.05 . . . . . | 15        |
| A.3      | Normální rozdělení s koeficientem 0.3 . . . . .  | 17        |
| A.4      | Normální rozdělení s koeficientem 0.8 . . . . .  | 19        |

## Část I

# Zadání

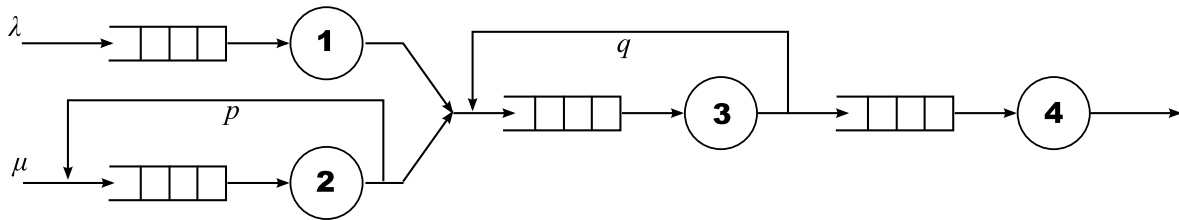
## 1 Obecné zadání

1. Vymyslete si otevřenou síť front (tj. propojení, vstupní proudy) obsahující alespoň 4 obslužné uzly (jeden kanál, fronta FIFO, neomez. délka), alespoň 2 vstupní proudy požadavků, alespoň 2 vnitřní zpětné vazby.
2. Parametry sítě (tj. střední frekvence vstupních proudů, střední doby obsluhy v jednotlivých kanálech a p-ti větvení) zvolte tak, aby síť pracovala ve stacionárním režimu. Doporučená hodnota zatížení pro všechny uzly:  $\rho > 0.5$ .
3. Určete výpočtem střední frekvence toků v uzlech. Dále určete veličiny  $L_{q_i}$  a  $T_{q_i}$  pro jednotlivé uzly a  $L_q$  a  $T_q$  pro celou síť pro případ, že všechny vstupní toky jsou Poissonovské a doby obsluhy ve všech uzlech mají exponenciální rozdělení.
4. Vypočtené hodnoty ověřte vlastnoručně vytvořeným simulačním programem. Použijte simulační knihovnu C-Sim nebo J-Sim.
5. Dále uvažujte případ, kdy všechny náhodné časové intervaly v modelu (příchody, obsluhy) mají Gaussovské pravděpodobnostní rozdělení  $N(a, \sigma)$  s (různou) střední hodnotou zvolenou v bodě 2. Vytvořte generátor tohoto rozdělení jako funkci v jazyce C nebo Java (s parametry např.  $a, \sigma$ ) a testováním ověřte správnou funkci generátoru - chce se tedy po Vás vytvoření a prokázání správné funkce generátoru, který napíšete VY - využít můžete pouze knihovní funkce pro generování rovnoměrného rozdělení (jako v průběžném příkladu č. 1). Použití knihovní funkce pro Gaussovo rozdělení z J-SIMu se nepočítá!
6. Simulací ověřte chování sítě (tj. určete stejné veličiny jako v bodech 3) a 4) pro případ, že všechna rozdělení (příchody, obsluhy) budou mít hustotu  $N(a, \sigma)$  se stejnou střední hodnotou jako pro exponenciální rozdělení alespoň pro 3 různé hodnoty koeficientu variace  $C = \sigma/a$ . Pokuste se o poměrně odlišné koeficienty, ať je vidět rozdíl v chování systému (např. 0.05 - tj. skoro konstantní generátor, 0.2 a 0.7 - ale to je pouze příklad). Poznámka: Simulační program je stejný jako v bodě 4, ale volá se jiný generátor podle bodu 5.
7. Simulační program upravte pro sledování další individuálně zadaných výkonnostních charakteristik sítě.
8. Řešení zpracujte formou písemného referátu (cca 10 stran, grafy, tabulky, barevné obrázky, hudební vložky, multimedia ap. - berte to jako přípravu na diplomku, navíc vlastní referát lze využít při zkoušce).

### 1.1 Konkrétní zadání

Na obrázku 1 je konkrétní schéma SHO. Jako další měření se mají provést statistiky čekacího času ve frontě č. 3. Hodnoty jsou následující:

$$\lambda = 0,9 \quad \mu = 0,4 \quad p = 0,5 \quad q = 0,5$$



Obrázek 1: Schéma zadaného systému hromadné obsluhy

a další hodnoty jsou pak v následující tabulce.

| Veličina | Server 1 | Server 2 | Server 3 | Server 4 |
|----------|----------|----------|----------|----------|
| $T$      | 1,0      | 0,3      | 1,0      | 0,7      |

## Část II

# Teoretický úvod

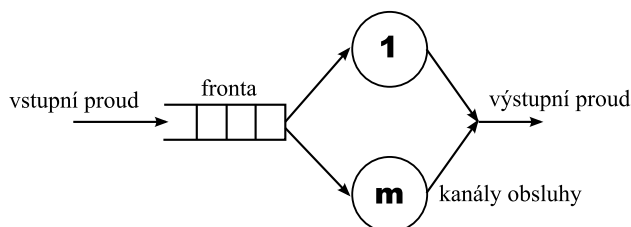
## 2 Systémy hromadné obsluhy

Jako *systémy hromadné obsluhy* se označují modely reálných systémů, jejichž funkce spočívá v poskytnutí nějaké *služby* (realizace *obsluhy*) pro velké počty souběžně přicházejících *požadavků* (někdy *transakcí*). Služby poskytují prvky zvané *kanály obsluhy*, před kanálem se obvykle vytvářejí *fronty* požadavků.

Významnou vlastností je abstrakce od semantiky služby, modeluje se pouze časová posloupnost přicházejících požadavků a jejich časová náročnost na služby. Vzhledem k velkým počtům požadavků je použit *pravděpodobnostní popis* časových charakteristik proudu požadavků.

### 2.1 Elementární SHO

Konceptuální model systému hromadné obsluhy je na obrázku 2. Jako *stav* systému figuruje celkový počet požadavků akumulovaných v systému (tj. ve frontě i obslužných kanálech).



Obrázek 2: Elementární systém hromadné obsluhy

Základním předpokladem pro další popis je *stacionární režim činnosti systému*, tj. časová stálost zdrojových statistických charakteristik.

#### 2.1.1 Vstupní proud požadavků

Předpokládáme, že požadavky vstupují do systému náhodně v časové posloupnosti  $\{t_0 < t_1 < t_2 \dots\}$ . Náhodná veličina  $\tau = t_k - t_{k-1}$  se nazývá *interval příchodů* a znamená velikost časového intervalu mezi dvěma vstupy.

Nejčastěji je vstupní proud zadáván *distribuční funkcí pravděpodobnostního rozdělení*  $F_a(t) = \mathcal{P}\{\tau \leq t\}$  nebo odpovídající *hustotou pravděpodobnosti*. Nejčastěji využívaným typem vstupního proudu je *poissonovský vstupní proud*, pro který má interval příchodů *exponenciální rozdělení*. Dále lze proud popsat těmito vztahy:

$T_a = 1/\lambda$  – střední perioda příchodů (střední hodnota intervalu mezi příchody).

$\lambda$  – střední frekvence příchodů

$F_a(t) = 1 - e^{-\lambda t}$  – distribuční funkce časových intervalů mezi příchody v poissonovském proudu

$C_a = \sigma\{\tau\}/T_a$  – koeficient variance, kde  $\sigma\{\tau\}$  je směrodatná odchylka intervalu mezi příchody; udává nahodilost příchodů, pro pravidelné má hodnotu 0, pro poissonovské hodnotu 1

### 2.1.2 Fronta požadavků

Fronta je charakterizována *maximální délkou* a *frontovou disciplínou*. Délka může být omezená nebo neomezená. Nejčastější frontovou disciplínou je FIFO. K popisu fronty se zavádí:

$w$  – okamžitý počet požadavků ve frontě

$L_w$  – střední délka fronty, tj. středí počet požadavků ve frontě

$t_w$  – doba čekání jednoho konkrétního požadavku ve frontě

$T_w$  – střední doba čekání požadavků ve frontě

### 2.1.3 Kanály obsluhy

Nejčastější je jednokanálová obsluha, počet kanálů označíme jako  $m$ . Doba  $t_s$  (doba obsluhy jednoho konkrétního požadavku) je *náhodná veličina*. Základní charakteristikou bude opět distribuční funkce uvažovaného rozdělení, tj.  $F_s(t) = \mathcal{P}\{t_s \leq t\}$ .

$F_s(t)$  – distribuční funkce pravděpodobnostního rozdělení doby obsluhy

$T_s = 1/\mu$  – střední doba obsluhy; veličina  $\mu$  představuje střední frekvenci obsluh, tj. střední počet požadavků obloužených za jednotku času (za předpokladu, že je kanál stále zatížený)

$F_s(t) = 1 - e^{-\mu t}$  – distribuční funkce exponenciálního rozdělení doby obsluhy

$C_s = \sigma\{t_s\}/T_s$  – koeficient variace doby obsluhy ( $\sigma\{t_s\}$  je směrodatná odchylka doby obsluhy); vyjadřuje nahodilost obsluh (pro konstatní má hodnotu 0, pro exponenciální má hodnotu 1)

### 2.1.4 Vztahy pro elementární SHO jako celek

Pro elementární SHO jako pro celek můžeme charakterizovat tyto vztahy a veličiny:

$q$  – okamžitý celkový počet požadavků v SHO (tj. ve frontě i v kanálech obsluhy); udává *stav SHO*

$L_q$  – střední celkový počet požadavků v SHO

$t_q$  – doba odezvy; doba průchodu celým systémem pro jeden konkrétní požadavek

$T_q$  – střední doba odezvy

**Zatížení kanálů** Nutnou podmínkou pro dosažení *stacionárního režimu* je hodnota  $\rho < 1$

$$\rho = \frac{1}{m} \frac{T_s}{T_a} = \frac{1}{m} \frac{\lambda}{\mu} \quad (1)$$

Veličina  $u = \lambda/\mu$  se nazývá *intenzita provozu* a k ní nejbližší vyšší celé číslo představuje maximální počet oblužných kanálů potřebných pro zajištění stacionárního režimu činnosti SHO.

Dále zřejmě platí:

$$\begin{aligned} L_q &= L_w + L_s = L_w + m \cdot \frac{\lambda}{\mu} \\ T_q &= T_w + T_s = T_w + \frac{1}{\mu} \end{aligned}$$

**Littleovy vzorce**

$$L_q = \lambda T_q, \quad L_w = \lambda T_w \quad (2)$$

### 2.1.5 M/M/1

Uvažujeme případ jednodokánalového SHO. Vstupní proud požadavků je poissonovský, charakterizovatelný parametrem  $\lambda$  – ten odpovídá střední frekvenci proudu a zároveň je parametrem exponenciální rozdělení  $F_a(t) = 1 - e^{-\lambda t}$ .

Doba obsluh má exponenciální rozdělení  $F_s(t) = 1 - e^{-\mu t}$ , kde parametr  $\mu$  je zároveň střední (dosažitelná) frekvence obsluh. Pro dosažení stacionárního režimu musí být zatížení SHO  $\rho = \lambda/\mu < 1$ .

Další vzorce pro veličiny  $L_q$ ,  $T_q$  a  $T_s$ :

$$L_q = \frac{\rho}{1 - \rho} \quad \text{pro } \rho < 1 \quad (3)$$

$$T_q = T_w + T_s, \quad T_s = \frac{1}{\mu} \quad (4)$$

## 2.2 Otevřené sítě front

Systém s více kanály obsluhy, kde každý kanál obsluhy má vlastní frontu, lze popsat jako síť, jejíž prvky jsou elementární SHO. Síť s explicitními vstupy z prostředí se nazývají *otevřené*.

Otevřenou síť front lze znázornit orientovaným grafem. Počet elementárních SHO v síti označíme jako  $n$ , číslování uzlů bude tedy v intervalu  $i \in \langle 1, n \rangle$ . Jako váhy hran se obvykle používají *pravděpodobnosti větvení*, tj. hrana vedoucí z uzlu  $i$  do uzlu  $j$  je vážena pravděpodobností  $p_{ij}$  přechodu do  $j$  po ukončení obsluhy v  $i$ . Součet vah hran vycházejících z uzlu  $i$  musí být roven jedné.

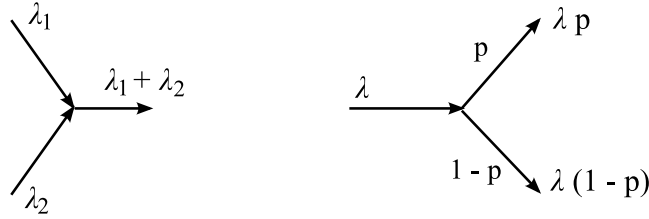
### 2.2.1 Analýza středních frekvencí a zatížení

Střední frekvenci vnitřního toku uzlem  $i$  označíme jako  $\Lambda_i$ . Frekvence  $\Lambda_0$  patří celkového toku požadavků vstupujících z okolí do systému. Rozdělují-li se požadavky po ukončení obsluhy v  $i$ -tém uzlu podle pravděpodobnosti větvení  $p_{ij}$ , bude střední frekvence toku požadavků z uzlu  $i$  do  $j$  dána jako  $\lambda_{ij} = \Lambda_i p_{ij}$ .

Na vstupu  $i$ -tého uzlu se sčítají frekvence toků hran vedoucích do uzlu  $i$ . Pro stacionární režim platí:

$$\sum_k \Lambda_k p_{ki} = \Lambda_i = \sum_j \Lambda_j p_{ij} \quad (5)$$





Obrázek 3: Pravidla pro spojování a rozdělování toků

kde index  $k$  nabývá hodnot odpovídající vstupním hranám a index  $j$  výstupním hranám uzlu  $i$ . Jedná se o *rovnici continuity* toku uzlem – součet frekvencí vstupních toků je roven frekvenci vnitřního toku a zároveň součtu frekvencí všech vystupujících hran.

Pro určení hodnot středních frekvencí všech toků v síti musí být dána hodnota frekvence  $\Lambda_0$  a matice  $\{p_{ij}\}$  pravděpodobností větvení. Hodnoty frekvencí  $\Lambda_i$  lze získat řešením soustavy  $n$  lineárních rovnic pro uzly 1 až  $n$  podle vzorce 5.

Po výpočtu středních frekvencí vnitřních toků lze provést kontrolu stacionarity systému. Pro tento výpočet ještě potřebujeme znát pro uzel  $i$  počet kanálů  $m_i$  a střední dobu obsluhy  $T_{si}$ . Zatížení je pak dáno jako

$$\rho_i = \frac{1}{m} \cdot \Lambda_i T_{si} < 1 \quad \text{pro } i = 1 \dots n \quad (6)$$

### 2.2.2 Určení délky front a doby odezvy

Dosud jsme neuvvažovali pravděpodobnostní rozdělení vstupu požadavků do systému a jejich obsluhy. Střední frekvence toků a zatížení uzlů na pravděpodobnostním rozložení nezávisí. Naopak střední délky front a doby odezvy narůstají s nepravidelností intervalů mezi příchody. Matematické řešení je pak možné jenom při splnění následujících podmínek (tzv. *Jacksonův teorém*):

- Všechny vstupní toky mají poissonovský charakter.
- Všechny obslužné uzly mají exponenciální rozdělení doby obsluhy se střední hodnotou  $T_{si}$ .
- Po ukončení obsluhy v uzlu  $i$  přechází požadavek s pravděpodobností  $p_{ij}$  do uzlu  $j$  bez zpoždění.

Na jednotlivé uzly lze pak nahlížet jako na elementární SHO a lze aplikovat vzorce ze sekce 2.1.4. Pro celou síť je pak možné určit průměrný počet požadavků  $L_q$  akumulovaných v celé síti jako

$$L_q = \sum_{i=1}^n L_{qi} \quad (7)$$

a střední dobu  $T_q$  průchodu požadavku sítí jako

$$T_q = \frac{1}{\Lambda_0} \cdot L_q \quad (8)$$

Pokud nejsou splněny stanovené předpoklady a některé příchody či obsluhy jsou pravidelnější než exponenciální (koeficienty variance jsou menší než jedna), lze výše uvedený postup použít k analýze nejhoršího případu (*worst case analysis*).

## 3 Generování náhodných čísel

### 3.1 Metoda inverzní transformace

Metoda umožňuje transformaci náhodných čísel  $Y$  s normalizovaným rovnoměrným rozdělením na čísla  $X$  zadaná distribuční funkcí  $F(x)$  jejich pravděpodobnostního rozdělení. Generátor normalizovaného rozdělení (tedy např. knihovni v `JDK Math.random()`) vygeneruje konkrétní hodnotu  $y$  a transformuje ji podle vzorce

$$x = F^{-1}(y) \tag{9}$$

Toho lze využít, pokud je distribuční funkce  $F(x)$  zadána analyticky a lze jednoduše určit její inverzní funkci  $F^{-1}(x)$ .

Pro *exponenciální rozdělení* dané distribuční funkcí  $F(x, \lambda) = 1 - e^{-\lambda x}$  lze po úpravách najít inverzní funkci ve tvaru

$$x = -\frac{1}{\lambda} \ln(1 - y),$$

ale vzhledem k tomu, že soubor prvků  $1 - y$  a  $y$  má stejné normované rovnoměrné rozložení, lze vzorec upravit na

$$x = -\frac{1}{\lambda} \ln(x).$$

### 3.2 Generování čísel s normálním rozdělením

Mějme dáno normální (někdy též *gaussovské*) rozdělení jako

$$\mathcal{N}(a, \sigma),$$

kde  $a$  je střední hodnota a  $\sigma$  je směrodatná odchylka.

Pro generování využijeme tzv. *centrální limitní větu*, která tvrdí, že součet náhodných čísel s libovolným rozdělením má asymptoticky normální rozdělení se střední hodnotou a rozptylem danými součtem středních hodnot a rozptylů jednotlivých rozdělení prvků součtu.

Uvedeme jenom výsledný vzorec pro realizaci generátoru náhodných čísel s normálním rozdělením (odvození viz např. [1]):

$$x = a + \sigma \left[ \left( \sum_{i=1}^{12} y_i \right) - 6 \right] \tag{10}$$

## Část III

# Matematické řešení

## 4 Charakteristiky uzlů

### 4.1 Střední frekvence toku v uzlech

Podle vzorce 5 sestavíme soustavu algebraických rovnic pro všechny uzly v síti.

$$\begin{aligned}\Lambda_1 &= \lambda \\ \Lambda_2 &= \mu + p \cdot \Lambda_2 \\ \Lambda_3 &= \Lambda_1 + (1 - p) \cdot \Lambda_2 + q \cdot \Lambda_3 \\ \Lambda_4 &= (1 - q) \cdot \Lambda_3\end{aligned}$$

Po dosazení zadaných hodnot dostaneme řešení

$$\Lambda_1 = 0,9 \quad \Lambda_2 = 0,8 \quad \Lambda_3 = 2,6 \quad \Lambda_4 = 1,3$$

### 4.2 Ověření stacionárního stavu

Podle 2.2.1 ověříme pro všechny uzly, zda je splněn požadavek stacionárního stavu. Podle vzorce 6 spočítáme, že

$$\begin{aligned}\varrho_1 &= 0,9 \cdot 1,0 = 0,9 & \varrho_2 &= 0,3 \cdot 2,6 = 0,78 \\ \varrho_3 &= 0,8 \cdot 1,0 = 0,8 & \varrho_4 &= 1,3 \cdot 0,7 = 0,91\end{aligned}$$

čili můžeme tvrdit, že je systém ve stacionárním stavu.

### 4.3 Střední hodnoty uzlů

Dále podle vzorců ze sekce 2.1.4 určíme pro jednotlivé uzly *střední celkový počet požadavků v uzlu*  $L_q$  a *střední dobu odezvy uzlu*  $T_q$ . Víme, že  $L_{qi} = \varrho_i / (1 - \varrho_i)$ .

$$\begin{aligned}L_{q1} &= \frac{0,9}{1 - 0,9} = 9 & L_{q2} &= \frac{0,78}{1 - 0,78} = 3,54 \\ L_{q3} &= \frac{0,8}{1 - 0,8} = 4 & L_{q4} &= \frac{0,91}{1 - 0,91} = 10,11\end{aligned}$$

Dále víme, že  $T_{qi} = T_{si} / (1 - \varrho_i)$ . Pak tedy

$$\begin{aligned}T_{q1} &= \frac{1}{1 - 0,9} = 10 & T_{q2} &= \frac{0,3}{1 - 0,78} = 1,37 \\ T_{q3} &= \frac{1}{1 - 0,8} = 5 & T_{q4} &= \frac{0,7}{1 - 0,91} = 7,78\end{aligned}$$

## 5 Globální charakteristiky

Podle vzorce 7 můžeme určit *střední celkový počet požadavků* v síti. Čili

$$L_q = 9 + 3,54 + 3 + 10,11 = 25,65$$

a podle vzorce 8 pak *střední dobu průchodu požadavku sítí* jako

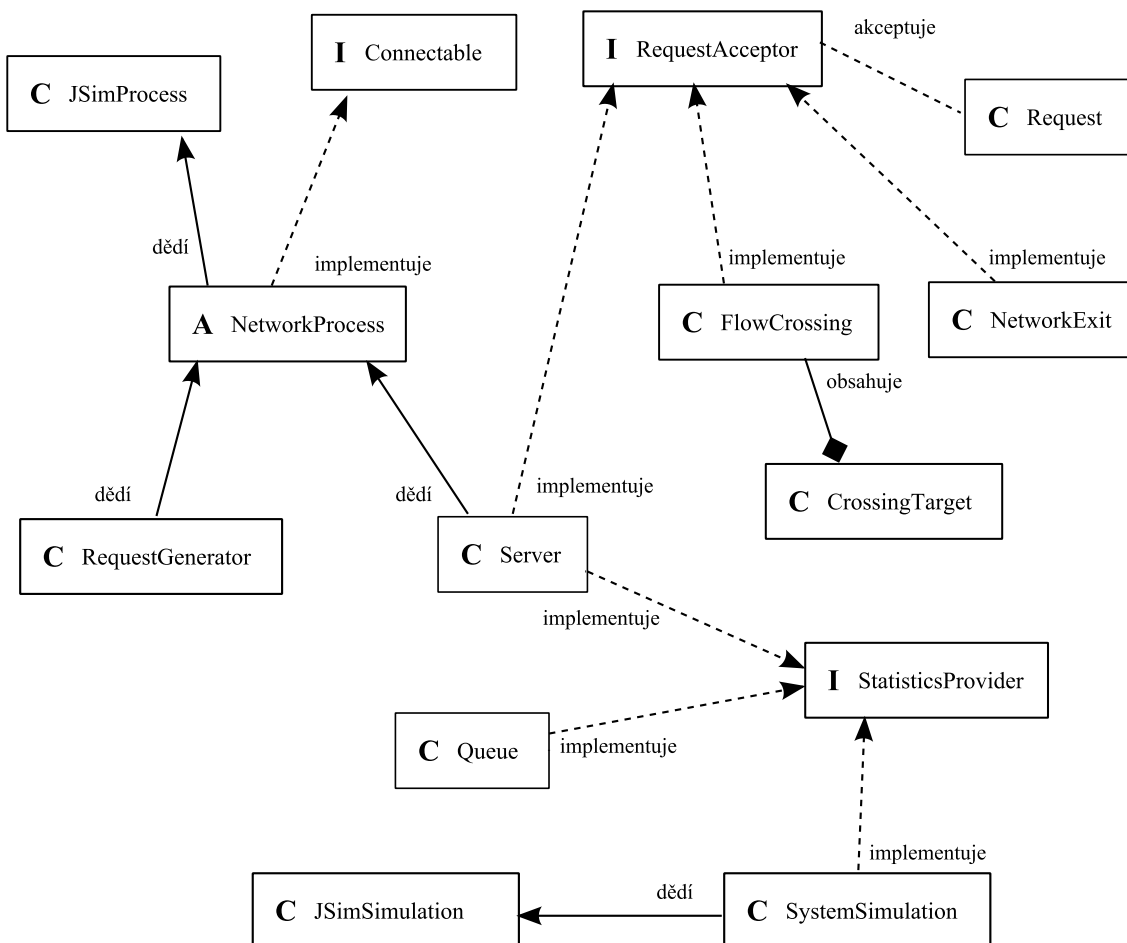
$$T_q = \frac{1}{1,3} \cdot 25,65 = 19,73 \quad \text{kde } 1,3 = \Lambda_0 = \lambda + \mu$$

## Část IV

# Programové řešení

## 6 Model sítě a popis balíčků

Problém jsem řešil v Javě s použitím knihovny *JSim*. Provedl jsem objektovou analýzu sítě front a snažil se co nejvíce uplatnit objektový návrh. UML digram je na obrázku 4.



Obrázek 4: UML diagram tříd – „I“ značí interface, „A“ abstraktní třídu a „C“ třídu.

Všechny tyto třídy jsou v balíčku `network`<sup>1</sup> a jsou velmi dobře komentované, proto je zde nebudu popisovat. Jejich API lze najít buď přímo ve zdrojových kódech, nebo taskem `ant javadoc` vygenerovat *JavaDoc*.

Pro generování náhodných čísel slouží třídy balíčku `distribution`. Společným rozhraním je interface `Distribution` s metodou `getRandomValue()`, která vrací náhodnou hodnotu daného rozdělení. Balíček obsahuje implementaci např. exponenciálního rozdělení, normálního rozdělení, Erlangova rozdělení druhého stupně, apod.

<sup>1</sup>Všechny názvy tříd a balíčků uvádím bez prefixu – ten je `cz.habi.vsp.sem`.

Pro otestování hodnot jednotlivých rozdělání jsem využil *unit testy*. Balíček `tests` obsahuje několik tříd pro *JUnit 4* a třída `DistributionsTest` pak přímo testy rozdělání. Pro každé rozdělání je vygenerováno milion náhodných hodnot, spočeny statistiky a ty pak porovnávány s předpokládanými analytickými výsledky. Konstantou `ALWAYS_SHOW_RESULTS` je možné nastavit zobrazení výsledků. Testování rozdělání lze spustit Ant taskem `ant tests-distribution`. Pro všechny testy z balíčku `tests` je pak Ant task `ant tests`.

Jelikož referát píšu v  $\text{\LaTeX}$ , obsahuje třída `Tools` i metodu pro výpis statistik v požadovaném formátu pro  $\text{\LaTeX}$ .

## 7 Ovládání programu

Pro překlad programu je zapotřebí mít nainstalovaný Ant. Tasky jsou následující:

- `ant clean` – vyčištění zkompileovaných tříd, distribučního jaru a JavaDocu
- `ant compile` – kompilace projektu
- `ant dist` – vytvoření *jar* souboru; nutné pro spuštění skriptem `run.bat`!
- `ant javadoc` – generování programátorské dokumentace (JavaDoc)
- `ant tests` – spouštění unit testů, viz výše

Program se spouští v příkazové řádce skriptem `run.bat`. Parametry mohou být následující:

bez parametrů – program simuluje exponenciální rozdělání

`N1` nebo `N2` nebo `N3` – program simuluje normální rozdělání pro koeficienty 0.05 nebo 0.3 nebo 0.8

*výše uvedené* a poslední parametr je `latex` – vytvoří výstup ve formátu pro sazbu v  $\text{\LaTeX}$ .

## Část V

# Zhodnocení výsledků

Program jsem nejdříve testoval na jednodušší síti `front` (ta z `QNEExample`), pro kterou dával stejné výsledky jako ukázkový příklad<sup>2</sup> i jako `QNAalyzer`. Z toho lze usoudit, že program funguje korektně, resp. framework pro simulaci sítí je naprogramován dobře.

Bohužel se u vlastního zadání rozcházím jak s teoretickými výpočty, tak i s programem `QNAalyzer`. Netuším, kde by mohla být chyba, možná je něco shnilého v samotném zadání systému.

---

<sup>2</sup>ten jsem u práce nechal, je to třída `cz.habi.vsp.sem.application.ExampleSimulation` a pro její spuštění stačí lehce pozměnit skript `run.bat`

## A Výpisy výstupů simulace

### A.1 Exponenciální rozdělení

(System): SystemSimulation simulation (*class: cz.habi.vsp.sem.network.SystemSimulation*  
)

| Veličina   | Hodnota           |
|------------|-------------------|
| Stopped at | 199952.2488762111 |
| Requests   | 100001            |
| Steps      | 1043364           |
| Tq         | 52624.95046718606 |
| Tq (DEV)   | 34374.05118736082 |
| Lq         | 80135.04659016465 |

(Server): S1 (*class: cz.habi.vsp.sem.network.Server* )

| Veličina | Hodnota            |
|----------|--------------------|
| Load     | 0.9017054271931144 |
| Tq       | 9.953433830130146  |
| Tq (DEV) | 10.24942497741308  |
| Requests | 180098             |
| Ts       | 1.001110661923774  |
| Flow     | 0.9007050483913152 |

(Queue) before (Server): S1 (*class: cz.habi.vsp.sem.network.Queue* )

| Veličina | Hodnota            |
|----------|--------------------|
| Tw       | 8.952294662028686  |
| Tw (DEV) | 10.200384582646379 |
| Lw       | 8.063484446971778  |

(Server): S2 (*class: cz.habi.vsp.sem.network.Server* )

| Veličina | Hodnota             |
|----------|---------------------|
| Load     | 0.23843266918969028 |
| Tq       | 0.39232255605734534 |
| Tq (DEV) | 0.3910054674159206  |
| Requests | 159406              |
| Ts       | 0.2990800121076764  |
| Flow     | 0.7972203408359114  |

(Queue) before (Server): S2 (*class: cz.habi.vsp.sem.network.Queue* )

| Veličina | Hodnota             |
|----------|---------------------|
| Tw       | 0.0932425439496785  |
| Tw (DEV) | 0.2538636475697551  |
| Lw       | 0.07433485266796933 |

(Server): S3 (*class: cz.habi.vsp.sem.network.Server* )

| Veličina | Hodnota            |
|----------|--------------------|
| Load     | 0.9999957971080904 |
| Tq       | 44515.90689346909  |
| Tq (DEV) | 25703.249924150834 |
| Requests | 200109             |
| Ts       | 0.9992124716955358 |
| Flow     | 1.0007839427896905 |

(Queue) before (Server): S3 (*class: cz.habi.vsp.sem.network.Queue* )

| Veličina | Hodnota            |
|----------|--------------------|
| Tw       | 44515.130360034345 |
| Tw (DEV) | 25703.376980583835 |
| Lw       | 80124.20975850131  |

(Server): S4 (*class: cz.habi.vsp.sem.network.Server* )

| Veličina | Hodnota            |
|----------|--------------------|
| Load     | 0.3489124020038047 |
| Tq       | 1.0728062052451812 |
| Tq (DEV) | 1.070901426673242  |
| Requests | 100001             |
| Ts       | 0.6976512179024347 |
| Flow     | 0.5001244075124649 |

(Queue) before (Server): S4 (*class: cz.habi.vsp.sem.network.Queue* )

| Veličina | Hodnota            |
|----------|--------------------|
| Tw       | 0.3751549873427326 |
| Tw (DEV) | 0.8143450206874557 |
| Lw       | 0.1876241657701306 |

## A.2 Normální rozdělení s koeficientem 0.05

(System): SystemSimulation simulation (*class: cz.habi.vsp.sem.network.SystemSimulation* )

| Veličina   | Hodnota            |
|------------|--------------------|
| Stopped at | 179836.4953622338  |
| Requests   | 100001             |
| Steps      | 1169850            |
| Tq         | 56959.09638443806  |
| Tq (DEV)   | 36213.87763244233  |
| Lq         | 149823.18814278953 |



(Server): S1 (*class: cz.habi.vsp.sem.network.Server* )

| Veličina | Hodnota            |
|----------|--------------------|
| Load     | 0.9999305821655169 |
| Tq       | 12.867672442571942 |
| Tq (DEV) | 10.001381975195596 |
| Requests | 199770             |
| Ts       | 0.9001552360322608 |
| Flow     | 1.1108423771138074 |

(Queue) before (Server): S1 (*class: cz.habi.vsp.sem.network.Queue* )

| Veličina | Hodnota            |
|----------|--------------------|
| Tw       | 11.967545376978316 |
| Tw (DEV) | 10.00140356830341  |
| Lw       | 13.29515484807234  |

(Server): S2 (*class: cz.habi.vsp.sem.network.Server* )

| Veličina | Hodnota            |
|----------|--------------------|
| Load     | 0.9999994423552333 |
| Tq       | 29977.346181401386 |
| Tq (DEV) | 17345.34681193317  |
| Requests | 199803             |
| Ts       | 0.9000685429015246 |
| Flow     | 1.111025877131051  |

(Queue) before (Server): S2 (*class: cz.habi.vsp.sem.network.Queue* )

| Veličina | Hodnota            |
|----------|--------------------|
| Tw       | 29976.596317269716 |
| Tw (DEV) | 17345.43335690394  |
| Lw       | 49954.027195296156 |

(Server): S3 (*class: cz.habi.vsp.sem.network.Server* )

| Veličina | Hodnota            |
|----------|--------------------|
| Load     | 0.9999894918194955 |
| Tq       | 44968.698066753306 |
| Tq (DEV) | 25914.451547522272 |
| Requests | 199818             |
| Ts       | 0.8999920207783044 |
| Flow     | 1.111109286229798  |

(Queue) before (Server): S3 (*class: cz.habi.vsp.sem.network.Queue* )

| Veličina        | Hodnota            |
|-----------------|--------------------|
| <b>Tw</b>       | 44968.022493695156 |
| <b>Tw (DEV)</b> | 25914.58097104979  |
| <b>Lw</b>       | 99851.36102369055  |

(Server): S4 (*class: cz.habi.vsp.sem.network.Server* )

| Veličina        | Hodnota             |
|-----------------|---------------------|
| <b>Load</b>     | 0.5004289831115964  |
| <b>Tq</b>       | 0.9203685335868323  |
| <b>Tq (DEV)</b> | 0.06338604077840915 |
| <b>Requests</b> | 100001              |
| <b>Ts</b>       | 0.8999449455553045  |
| <b>Flow</b>     | 0.5560662189205479  |

(Queue) before (Server): S4 (*class: cz.habi.vsp.sem.network.Queue* )

| Veličina        | Hodnota              |
|-----------------|----------------------|
| <b>Tw</b>       | 0.02042358803151147  |
| <b>Tw (DEV)</b> | 0.04477301592085418  |
| <b>Lw</b>       | 0.011356867373473425 |

### A.3 Normální rozdělení s koeficientem 0.3

(System): SystemSimulation simulation (*class: cz.habi.vsp.sem.network.SystemSimulation* )

| Veličina          | Hodnota            |
|-------------------|--------------------|
| <b>Stopped at</b> | 180126.17772590622 |
| <b>Requests</b>   | 100001             |
| <b>Steps</b>      | 1170163            |
| <b>Tq</b>         | 57007.29723256297  |
| <b>Tq (DEV)</b>   | 36253.23061838178  |
| <b>Lq</b>         | 150022.7248489626  |

(Server): S1 (*class: cz.habi.vsp.sem.network.Server* )

| Veličina        | Hodnota            |
|-----------------|--------------------|
| <b>Load</b>     | 0.99740259261434   |
| <b>Tq</b>       | 43.62245862366981  |
| <b>Tq (DEV)</b> | 36.927750500853    |
| <b>Requests</b> | 199976             |
| <b>Ts</b>       | 0.8983993912345993 |
| <b>Flow</b>     | 1.1101995419249877 |

(Queue) before (Server): S1 (*class: cz.habi.vsp.sem.network.Queue* )

| Veličina | Hodnota           |
|----------|-------------------|
| Tw       | 42.72440703835254 |
| Tw (DEV) | 36.92711041595655 |
| Lw       | 47.46984749575744 |

(Server): S2 (*class: cz.habi.vsp.sem.network.Server* )

| Veličina | Hodnota            |
|----------|--------------------|
| Load     | 0.9999943609858628 |
| Tq       | 30017.507078675964 |
| Tq (DEV) | 17368.233777692283 |
| Requests | 200025             |
| Ts       | 0.9005132458034923 |
| Flow     | 1.1104715734565431 |

(Queue) before (Server): S2 (*class: cz.habi.vsp.sem.network.Queue* )

| Veličina | Hodnota            |
|----------|--------------------|
| Tw       | 30016.75711871981  |
| Tw (DEV) | 17368.319908702928 |
| Lw       | 50024.983562892805 |

(Server): S3 (*class: cz.habi.vsp.sem.network.Server* )

| Veličina | Hodnota            |
|----------|--------------------|
| Load     | 0.9999913151234736 |
| Tq       | 44994.552600467374 |
| Tq (DEV) | 26000.06954218999  |
| Requests | 200099             |
| Ts       | 0.9001774789094074 |
| Flow     | 1.1108823965858308 |

(Queue) before (Server): S3 (*class: cz.habi.vsp.sem.network.Queue* )

| Veličina | Hodnota            |
|----------|--------------------|
| Tw       | 44993.877101058424 |
| Tw (DEV) | 26000.19837612273  |
| Lw       | 99946.10058087869  |

(Server): S4 (*class: cz.habi.vsp.sem.network.Server* )

| Veličina | Hodnota            |
|----------|--------------------|
| Load     | 0.499891436475516  |
| Tq       | 1.0366561678777082 |
| Tq (DEV) | 0.392079919131378  |
| Requests | 100001             |
| Ts       | 0.9004263330391431 |
| Flow     | 0.5551719425933146 |

(Queue) before (Server): S4 (*class: cz.habi.vsp.sem.network.Queue* )

| Veličina | Hodnota             |
|----------|---------------------|
| Tw       | 0.13622983483855394 |
| Tw (DEV) | 0.28524959935098076 |
| Lw       | 0.07563098204648554 |

#### A.4 Normální rozdělení s koeficientem 0.8

(System): SystemSimulation simulation (*class: cz.habi.vsp.sem.network.SystemSimulation* )

| Veličina   | Hodnota            |
|------------|--------------------|
| Stopped at | 187121.7617710086  |
| Requests   | 100001             |
| Steps      | 1156318            |
| Tq         | 59462.100979578085 |
| Tq (DEV)   | 37819.818583204324 |
| Lq         | 150523.4375382497  |

(Server): S1 (*class: cz.habi.vsp.sem.network.Server* )

| Veličina | Hodnota            |
|----------|--------------------|
| Load     | 0.9989875349737994 |
| Tq       | 497.73028936767287 |
| Tq (DEV) | 311.42573783666757 |
| Requests | 199454             |
| Ts       | 0.937220148663724  |
| Flow     | 1.0659048852056183 |

(Queue) before (Server): S1 (*class: cz.habi.vsp.sem.network.Queue* )

| Veličina | Hodnota            |
|----------|--------------------|
| Tw       | 496.79565417213604 |
| Tw (DEV) | 311.4253664281615  |
| Lw       | 532.3061257881681  |

(Server): S2 (*class: cz.habi.vsp.sem.network.Server* )

| Veličina | Hodnota            |
|----------|--------------------|
| Load     | 0.9999980489972488 |
| Tq       | 31301.65093109814  |
| Tq (DEV) | 18035.433510962714 |
| Requests | 199822             |
| Ts       | 0.9364404154494328 |
| Flow     | 1.067871519104942  |

(Queue) before (Server): S2 (*class: cz.habi.vsp.sem.network.Queue* )

| Veličina | Hodnota            |
|----------|--------------------|
| Tw       | 31300.871059355737 |
| Tw (DEV) | 18035.523950535393 |
| Lw       | 50236.02894039716  |

(Server): S3 (*class: cz.habi.vsp.sem.network.Server* )

| Veličina | Hodnota            |
|----------|--------------------|
| Load     | 0.9999874334730889 |
| Tq       | 46735.52060501931  |
| Tq (DEV) | 26991.196535582647 |
| Requests | 200023             |
| Ts       | 0.9354894702126938 |
| Flow     | 1.0689456859901703 |

(Queue) before (Server): S3 (*class: cz.habi.vsp.sem.network.Queue* )

| Veličina | Hodnota            |
|----------|--------------------|
| Tw       | 46734.81899710016  |
| Tw (DEV) | 26991.331656761995 |
| Lw       | 99751.29882345197  |

(Server): S4 (*class: cz.habi.vsp.sem.network.Server* )

| Veličina | Hodnota            |
|----------|--------------------|
| Load     | 0.5010624221060834 |
| Tq       | 1.4553888293236015 |
| Tq (DEV) | 1.1149759327775435 |
| Requests | 100001             |
| Ts       | 0.9375874559428314 |
| Flow     | 0.5344167297995881 |

(Queue) before (Server): S4 (*class: cz.habi.vsp.sem.network.Queue* )

| Veličina | Hodnota             |
|----------|---------------------|
| Tw       | 0.5178013733807523  |
| Tw (DEV) | 0.9021555110183733  |
| Lw       | 0.27672171664788175 |

## Reference

- [1] Racek S.: *Pravděpodobnostní modely počítačů*, (nevydáno).