Západočeská univerzita v Plzni

Fakulta aplikovaných věd

Katedra informatiky a výpočetní techniky

Semestrální práce

Výkonnost a spolehlivost číslicových sytémů
Otevřená síť front

Jan Bařtpán (A03043)

bartipan@students.zcu.cz

leden 2006

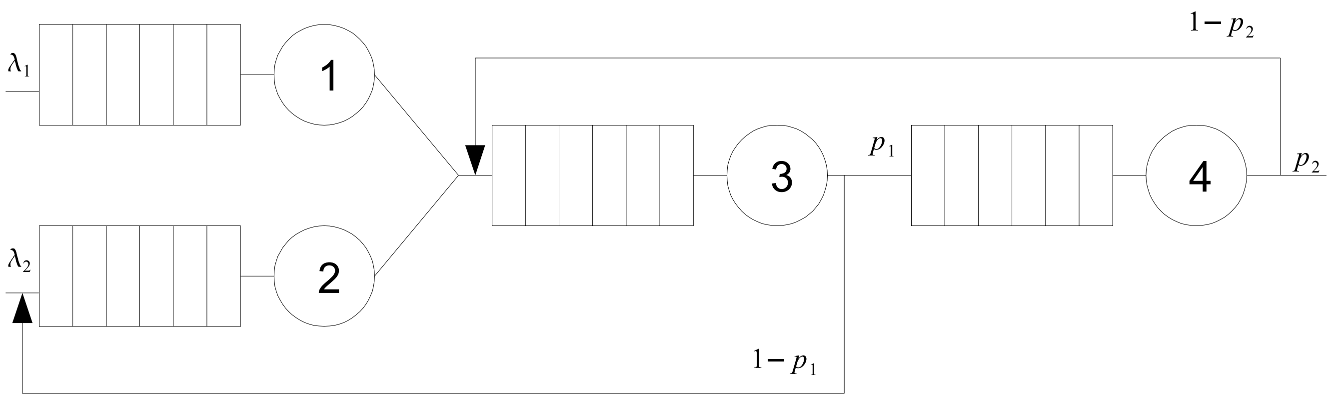
Obsah

1. Úvod

1.1 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 Lq_i a Tq_i pro jednotlivé uzly a Lq a Tq 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.** Příklad na simulační model otevřené sítě front je součástí standardní součástí distribuce u obou knihoven..
5. Dále uvažujte případ, kdy všechny náhodné časové intervaly v modelu (příchody, obsluhy) mají Gaussovské pravděpodobnostní rozdělení $N(a, \sigma)$ s (různou) sřední hodnotou zvolenou v bodě 2. Vytvořte generátor tohoto rozdělení jako funkci v jazyce **C** nebo **Java** (s parametry např.a, σ) a testováním ověřte správnou správnou funkci generátoru.
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 $C=\sigma/a$. 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, multimédia ap. - berte to jako přípravu na diplomku, navíc vlastní referát lze využít při zkoušce).

Schéma sítě



Další sledované charakteristiky: Statistika počtu požadavků ve třetím uzlu (E, D, …).

1.2 Legenda

Abych měl představu, co jednotlivé veličiny, uzly a požadavky v této síti znamenají, vymyslel jsem si tuto legendu:

Tato síť představuje obsluhu zákazníků blíže neurčeného úřadu. Jednotlivé uzly pak představují kanceláře, obsluhující jednotlivé klienty úřadu.

1.3 Parametry sítě

Střední frekvence vstupních proudů:

$$\lambda_1 = 8\,hod^{-1}$$
$$\lambda_2 = 6\,hod^{-1}$$

Střední doba obsluhy v jednotlivých uzlech:

$$T_{s_1} = 6\,min = 0{,}1\,hod$$
$$T_{s_2} = 2{,}4\,min = 0{,}04\,hod$$
$$T_{s_3} = 1{,}5\,min = 0{,}025\,hod$$
$$T_{s_4} = 3\,min = 0{,}05\,hod$$

Pravděpodobnosti větvení:

$$p_1 = 0{,}7$$
$$p_2 = 0{,}8$$

2. Výpočet

2.1 Výpočet střední frekvence obsluhy

Ze střední doby obsluhy určíme střední frekvenci obsluhy jako $\mu_i = \dfrac{1}{T_{s_i}}$:

$$\mu_1 = 10$$
$$\mu_2 = 25$$
$$\mu_3 = 40$$
$$\mu_4 = 20$$

2.2 Určení frekvence toků v uzlech

Protože doba obsluhy požadavků v jednotlivých uzlech a doba příchodů požadavků do sítě mají exponenciální rozdělení, určíme střední frekvenci vstupů do jednotlivých uzlů z této soustavy rovnic:

$$\Lambda_1 = \lambda_1$$
$$\Lambda_2 = \lambda_2 + (1 - p_1) \cdot \Lambda_3$$
$$\Lambda_3 = \Lambda_1 + \Lambda_2 + (1 - p_2) \cdot \Lambda_4$$
$$\Lambda_4 = p_1 \cdot \Lambda_3$$

Tato soustava vyjadřuje vztahy frekvencí jednotlivých vnitřních toků sítě.

Řešení této soustavy je:

$$\Lambda_1 = 8$$
$$\Lambda_2 = 13{,}5$$
$$\Lambda_3 = 25$$
$$\Lambda_4 = 17{,}5$$

2.3 Výpočet zatížení jednotlivých uzlů sítě

Nyní spočteme zatížení uzlů jako $\rho_i = T_{s_i} \cdot \Lambda_i$ tedy:

$$\rho_1 = 0{,}8$$
$$\rho_2 = 0{,}54$$
$$\rho_3 = 0{,}625$$
$$\rho_4 = 0{,}875$$

Platí, že $\rho_i < 1$, a proto je síť ve stacionárním stavu (nikde se nehromadí požadavky).

2.4 Výpočet středního celkového počtu požadavků na úřadě

Nyní již můžeme určit střední počet požadavků a střední dobu průchodu požadavků v jednotlivých uzlech a v celém systému jako:

$$L_{q_i} = \frac{\rho_i}{1 - \rho_i}$$
$$T_{q_i} = \frac{T_{s_i}}{1 - \rho_i}$$
$$L_q = \sum_{i=1}^{4} L_{q_i}$$
$$T_q = \frac{L_q}{\sum_{i=1}^{2} \lambda_i}$$

Po dosazení do těchto vzorečků nám vyjde:

$$L_{q_1} = 4 \qquad T_{q_1} = 0{,}5$$
$$L_{q_2} = 1{,}174 \qquad T_{q_2} = 0{,}087$$
$$L_{q_3} = 1{,}\overline{6} \qquad T_{q_3} = 0{,}0\overline{6}$$
$$L_{q_4} = 7 \qquad T_{q_4} = 0{,}9$$
$$L_q = 13{,}841 \qquad T_q = 0{,}989$$

3. Simulace

3.1 Analýza

V této simulaci jistě budeme potřebovat třídu, která bude reprezentovat jednotlivé kanceláře (třída Office). Tato třída vystupuje aktivně v simulaci, proto bude odvozena od třídy JSimProcess. Dalším aktivním prvkem v simulaci budou vstupní dveře do úřadu („generátor klientů“ - třída Door). I tato třída bude odvozena od třídy JSimProcess. Dále musíme reprezentovat fronty jednotlivých klientů úřadu – třída Queue. Protože jde o frontu, odvodíme ji od třídy JSimHead. Nakonec nám zbývá třída, která bude představovat jednotlivé klienty úřadu (třída Person).

3.1.1 Třída Office

Atributy:

Aby byl zajištěn chod simulace, musí mít tato třída tyto fronty:

- jednu vstupní frontu (atribut inQueue) – tato fronta je vytvořena v konstruktoru třídy (třída Office je jejím vlastníkem)
- dvě výstupní fronty (atributy outQueue1, outQueue2)– tato fronta není vlastněna třídou Office (slouží jako odkaz na vstupní frontu jiné instance třídy Office). Pokud třída má jen jednu výstupní frontu (uzly 1, 2), je druhá nastavena na null. Podobně, pokud je fronta výstupem ze sítě (uzel 4), je také nastavena na null.

Dále potřebujeme atribut, který bude vyjadřovat pravděpodobnost, že zpracovaný požadavek půjde do výstupní první fronty číslo 1 (outQueue1) – atribut p. Pokud daná instance třídy Office má pouze jednu výstupní frontu, musí být vždy p = 1.

Nakonec ještě budeme potřebovat další parametry pro určení:

- doby obsluhy – parametr mu
- sledovaných statistických veličin

Aktivita:

```
while (true){
    if (inQueue.empty())
        passivate();
    else{
        // aktualizování parametrů pro výpočet statistických veličin
        inQueueTestCount++;
        long cnt = inQueue.cardinal();
        inQueueCountSum += cnt;
        inQueueCountSquareSum += cnt * cnt;
        hist.add(cnt);

        // další zákazník ve frontě
        JSimLink personLink = inQueue.first();
        // vyřazení z fronty
        personLink.out();

        // simulace obsluhy
        double delta = rand();
        hold(delta);

        // výběr výstupní fronty
        Queue out = (JSimSystem.uniform(0,1) <= p) ? outQueue1 : outQueue2;
        // nejde-li o výstupní frontu (zákazník odchází z úřadu)
        if (out != null){
            // klient přejde do jiné fronty
            personLink.into(out);
            // případné vzbuzení (lelkujícího) úředníka
```

```
            out.activateServer();
        }
        else{
            // zákazník odchází z úřadu
            // aktualizování parametrů pro výpočet statistických veličin
            personCount++;
            serviceTimeSum += currentTime - person.getTimeOfArrival();
            personLink = null;
        }
    }
}
```

3.1.2 Třída Door

Atributy:
Třída Door slouží jako „generátor klientů“ úřadu. Noví klienti se zařazují do výstupní fronty (atribut outQueue). Tato fronta je vstupní frontu jednotlivých kanceláří. Jde tedy o odkaz (tj. třída Door se nestará o vytváření této fronty).
Dále tato třída potřebuje parametr pro určení frekvence s jakou přicházejí zákazníci úřadu (resp. doby mezi jednotlivými příchody).

Aktivita:

```
while (true){
    // vytvoření nového zákazníka
    JSimLink link = new JSimLink(new Person(myParent.getCurrentTime()));

    // zařazení do výstupní fronty
    link.into(outQueue);
    Person p = (Person)link.getData();
    // aktivování kanceláře
    outQueue.activateServer();
    // uspání do dalšího příchodu klienta (lelkování)
    double delta = rand();
    hold(delta);

}
```

3.1.3 Třída Queue

Atributy:
Protože po této třídě budeme požadovat, aby případně vzbudila lelkující obsluhu, musíme si udržet odkaz na kancelář, ke které tato obsluha patří – atribut server.

Vzbuzení obsluhy:

```
// pokud obsluha lelkuje
if (server.isIdle())
    // probudíme jí
    server.activate(myParent.getCurrentTime());
```

3.2 Generování náhodných čísel s Gaussovským rozdělením pravděpodobnosti

3.2.1 Návrh

Ke generování náhodných čísel y s Gaussovským pravděpodobnostním rozdělením budeme používat generátor náhodných čísel s uniformním pravděpodobnostním rozdělením. Takto vygenerované náhodné číslo x_i bude v intervalu <0; 1). S využitím centrální limitní věty lze pak náhodné číslo y vypočítat takto:

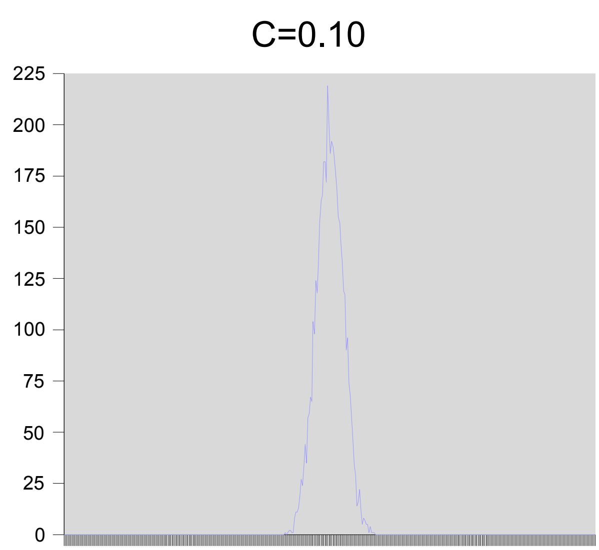
$y = \mu + \sigma \cdot (\sum_{i=1}^{12} x_i)$, kde σ je směrodatná odchylka a μ je střední hodnota.

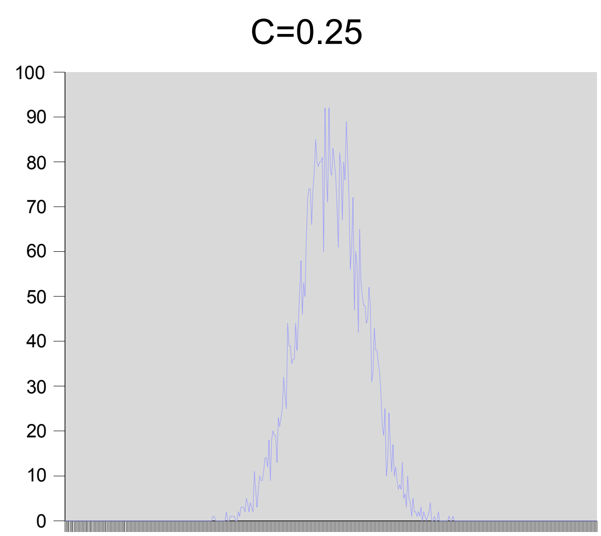
Protože bude třeba generovat v závislosti na střední hodnotě a variačním koeficientu, určíme směrodatnou odchylku jako:

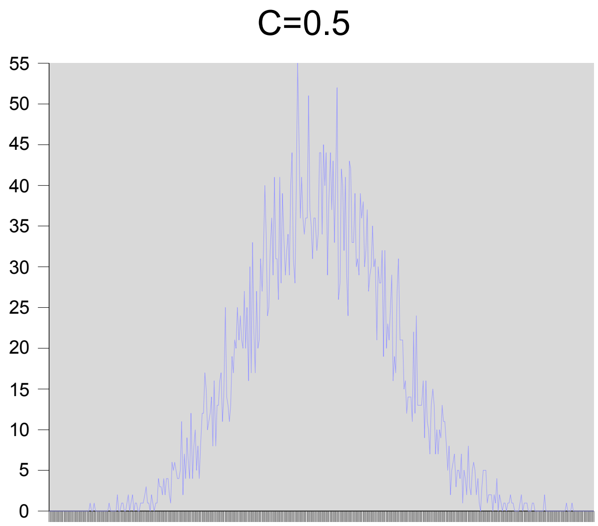
$$C = \frac{\sigma}{\mu}$$
$$\sigma = C \cdot \mu$$

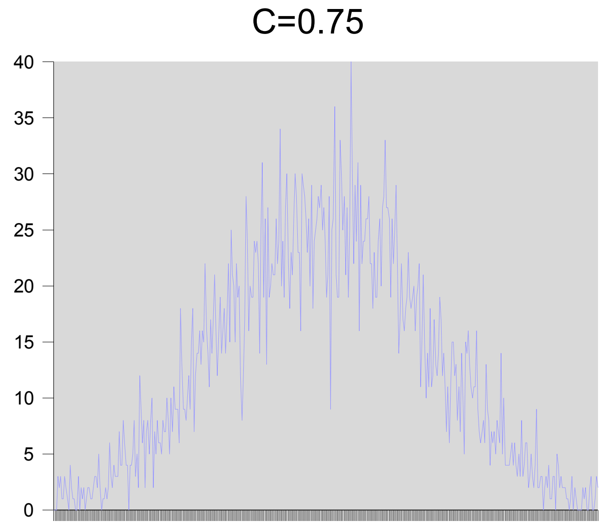
3.2.2 Ověření

Abych ověřil správnost takto generovaných čísel, vytvořil jsem si histogramy pro 5000 vygenerovaných čísel pro různé variační koeficienty:









Jak je vidět, histogramy odpovídají Gaussovskému rozložení hodnot.

3.3 Výsledky simulace

Pro určení přesných hodnot je třeba co možná nejdelší běh simulace. S větším počtem kroků simulace se tedy bude zpřesňovat výsledek. Bohužel se mi nepovedlo nalézt vhodný počet kroků, pro než by simulace trvala snesitelně dlouhou dobu a zároveň výsledky simulace (tj. hodnoty Tq a Lq) by odpovídaly přesně (tj. na tři desetinná místa) vypočteným výsledkům.

3.3.1 Pro exponenciální rozdělení

Následující tabulka ukazuje výsledky simulace v závislosti na maximálním simulačním čase. Sloupec E a D jsou střední hodnoty a rozptyl pro počet požadavků v uzlu tři.

Max time	Lq	Tq	Lw1	Tw1	Lw2	Tw2	Lw3	Tw3	Lw4	Tw4	E	D
500	11.525	0.840	2.71	0.35	0.62	0.05	0.91	0.04	4.52	0.27	1.91	4.23
1000	13.100	0.935	2.87	0.36	0.63	0.05	1.07	0.04	5.71	0.33	2.06	5.44
3000	13.488	0.968	2.85	0.36	0.6	0.04	1.02	0.04	6.21	0.36	2.01	5.23
5000	13.659	0.973	3.30	0.41	0.67	0.05	1.07	0.04	5.78	0.33	2.07	5.72

Jak je vidět hodnoty L_q a T_q se blíži vypočteným výsledkům ($L_q = 13{,}841, T_q = 0{,}989$).

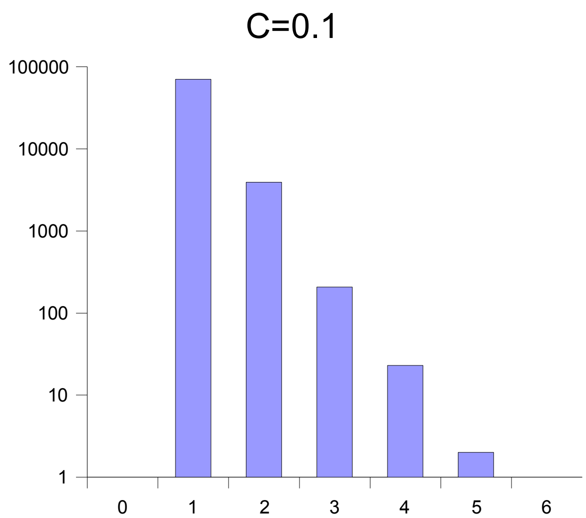
3.3.2 Pro normální rozdělení

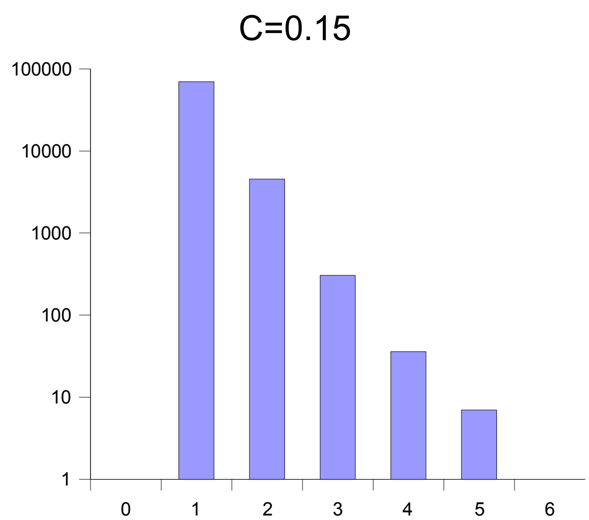
Obdobně následující tabulka ukazuje výsledky simulace v závislosti na maximálním simulačním čase. A stejně sloupec E a D jsou střední hodnoty a rozptyl pro počet požadavků v uzlu tři.

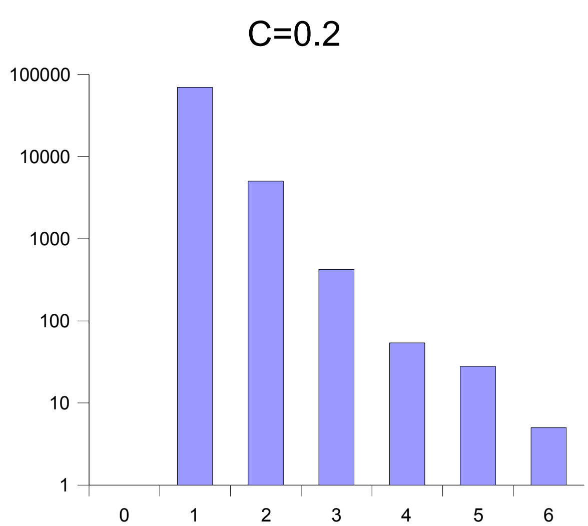
C	Lq	Tq	Lw1	Tw1	Lw2	Tw2	Lw3	Tw3	Lw4	Tw4	E	D
0.1	4.03	0.29	0.004	0.000	0.14	0.01	0.17	0.0068	0.88	0.050	1.06	0.13
0.15	4.11	0.29	0.021	0.003	0.15	0.01	0.18	0.0073	0.92	0.053	1.07	0.15
0.2	4.29	0.31	0.059	0.007	0.16	0.01	0.19	0.0077	1.04	0.059	1.08	0.19
0.25	1.56	0.31	0.096	0.014	0.06	0.01	8269.77	0.0084	0.37	0.060	1.1	0.22

Povšimněte si hodnoty Lw3 pro C=0,25. Nepoměr vůči ostatním Lw3 hodnotám pro jiná C je zapříčiněn tím, že se systém dostal do nestacionárního stavu a požadavky (zákazníci) se hromadí ve vstupní frontě uzlu tři.

Dále uvádím histogramy počtu požadavků v uzlu číslo 3 pro různé variační koeficienty:







4. Závěr

Při simulaci, kdy měly všechny náhodné časové intervaly v modelu (příchody, odchody) exponenciální rozdělení, vycházely hodnoty aplikačně zajímavých veličin velmi blízko analyticky zjištěných hodnot.
V případě Gaussovského rozdělení můžeme vysledovat rostoucí hodnoty Lq a Tq s rostoucím koeficientem variace, přičemž hodnoty pro C=0,25 vzhledem k nestacionárnosti systému je třeba brát s rezervou (jako nevalidní).
Dále je vidět z histogramu počtu požadavků v třetím uzlu (kanceláři), že ačkoliv se jedná o nejzatíženější uzel, nejčastěji je požadavek vyřízen ihned (nečeká ve frontě).