Proxy Cache Szerverek hatékonyságának vizsgálata

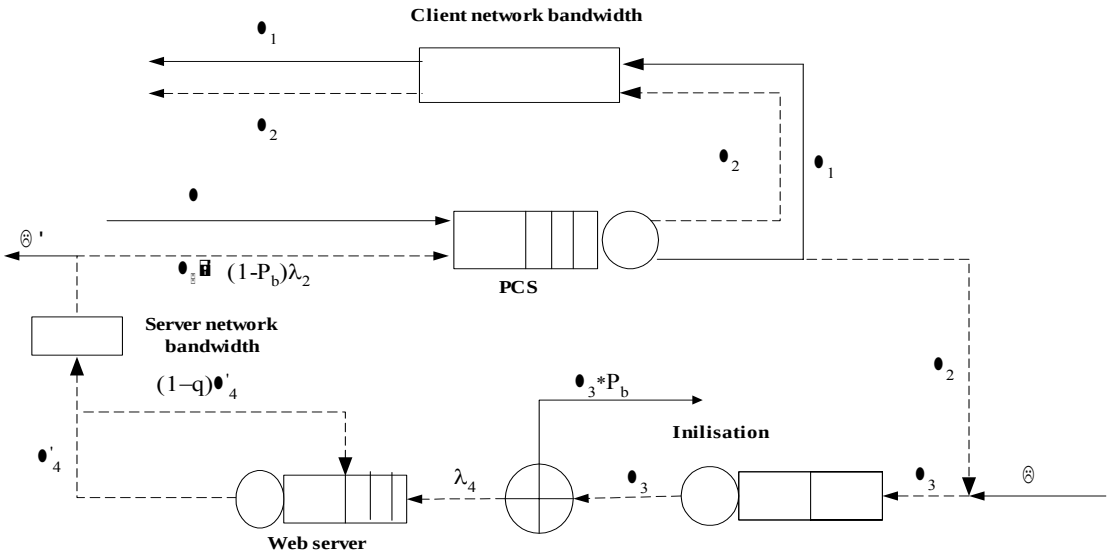
The Performance of the Proxy Cache Server

Bérczes Tamás, berczest@inf.unideb.hu
IFSZ KFT, Debrecen Péterfia u. 4

Sztrik János, sztrik.janos@inf.unideb.hu
Debreceni Egyetem, Informatikai Kar

1. Bevezetés

Napjainkban az egyik leginkább közkedvelt információszerzési lehetőség az internet használata. Az internet gyors és egyszerű lehetőséget biztosít több ezer webszerver adatainak a megismerésére, letöltésére.
Az internet használata az elmúlt években rohamosan növekedett. A felhasználók száma a 2001-es 474 millióról 2002-re 590 millióra növekedett. Becslések szerint 2006-ra az internetet használók száma eléri a 948 milliót. Figyelembe véve, hogy 1996-ban mindösszesen 40 millió - an használták az internetet, a növekedés üteme igen jelentékeny.
A felhasználók számának növekedésével párhuzamosan növekedett az internet forgalma is. Ennek hatására egyre nagyobb igény mutatkozik a színvonalas és gyors internet elérésre és kiszolgálásra.
Az információ keresés és letöltés közben a válasz a távoli web szervertől a kliens gépéig gyakran igen sok időt vesz igénybe. A probléma egyik oka, hogy ugyanabban az időben ugyanazt a fájlt más felhasználó is le akarja tölteni. Ebből adódóan ugyanazon fájlok másolatai mennek keresztül a hálózaton. Ez tulajdonképpen a kiszolgálási idő növekedését eredményezi. Természetes megoldásnak mutatkozik az információk tárolása. Ennek egyik megoldási lehetősége a böngésző szoftverbe való implementálás. Ebben az esetben a tárolt adatokhoz azonban csak egy személy férhet hozzá. Egy másik lehetőség Proxy Cache szerver használata. Jelen előadás célja ez utóbbi megoldás hatékonyságának vizsgálata.
A felhasználó szemszögéből nézve lényegtelen, hogy az általa keresett fájl fizikailag hol található: egy Proxy Cache Serveren (PCS) valahol a munkahelyének belső hálózatán vagy a világ túlsó felén egy távoli Web szerveren. A keresett dokumentum érkezhet a távoli Web szervertől vagy a Proxy Cache Szervertől (PCS). Kliens oldalról nézve a PCS funkciója ugyanaz mintegy Web szervernek valamint a Web szerver felől nézve ugyanolyan, mint egy kliens.
Feltételezzük, hogy az igények érkezési intenzitása λ paraméterű Poisson folyamat valamint a külső igények érkezési intenzitása szintén Poisson folyamat Λ paraméterrel.



1. ábra

2. A modell

Jelen előadásban a Bose és Cheng által készített analitikus modellt egészítjük ki. Feltételezzük, hogy a távoli Web szerverhez más felhasználóktól is érkeznek kérések, így jelen modellben figyelembe vesszük ezen a külső igényeket is, valamint a még realisztikusabb vizsgálat érdekében feltételezzük, hogy a Web szerver véges kapacitású.

Proxy Cash szervert használva, ha egy fájlt le akarunk tölteni egy távoli Web szerverről először meg kell vizsgálni, hogy a keresett fájl egy példánya megtalálható-e a PCS-en (Ennek valószínűségét jelöljük p-vel). Amennyiben a keresett dokumentum megtalálható a PCS-en, egy másolat továbbítódik a felhasználónak. Amennyiben a PCS-en nem található meg, az igény továbbítódik a távoli Web szerverhez. Miután az igényelt fájl megérkezett a PCS-re egy másolat azonnal a felhasználóhoz kerül.
A Proxy Cache Server hatékonysága a következő tényezőktől függ:

- a találati arány (a kért dokumentum milyen valószínűséggel található meg a PCS-en)
- a PCS sebessége
- a kliens oldali sávszélesség
- a szerver oldali sávszélesség
- a külső igények intenzitása
- a Web szerver karakterisztikája

Az 1.-es ábra mutatja egy igény lehetséges útját a felhasználótól kiindulva egészen a visszaérkezésig. Legyen F a keresett dokumentumok átlagos mérete. Az alábbiakban definiáljuk az ábrán szereplő változókat.

$$\lambda_1 = p * \lambda; \qquad (1)$$

$$\lambda_2 = (1-p) * \lambda; \qquad (2)$$

$$\lambda_3 = \lambda_2 + \Lambda; \tag{3}$$

Az egyenes vonal (λ_1) reprezentálja azt az esetet, mikor a keresett dokumentum egy példánya megtalálható a PCS-en. λ_2 jelöli azon igények útját (szaggatott vonallal rajzolva), melyek nem találhatóak a Proxy szerveren, így ezen igények továbbítódnak a távoli Web szervernek. λ_3 jelöli a Web szerverhez érkező összes igény érkezési intenzitását. A Web szerverhez érkező igényeknek először fel kell állítaniuk egy TCP kapcsolatot. Legyen I_s ezen egyszeri inicializáláshoz szükséges idő. A várakozó igények tárolására szolgáló puffer kapacitása legyen K. Legyen P_b annak a valószínűsége, hogy a beérkező igényt a szerver elutasítja.
A Web szerver hatékonyságát a következő három jellemzővel írhatjuk le [1]: A szerver kimenő pufferének kapacitása B_s, a statikus szerveridő Y_s valamint R_s a dinamikus szerver arány. Az M/M/1/K sorbanállási modell alapján meghatározható a P_b valószínűség:

$$P_b = \mathrm{P}\left(\mathrm{N{=}K}\right) = \frac{(1-\rho)*\rho^K}{1-\rho^{K+1}} \tag{4}$$

ahol

$$P_b = \frac{\lambda_3 F\left(Y_s R_s + B_s\right)}{R_s B_s} \tag{5}$$

így látható, hogy a Web szerver pufferéhez érkező igények intenzitása Poisson folyamat

$$\lambda_4 = \left(1-P_b\right)*\lambda_3 \tag{6}$$

intenzitással. Az előzőekhez hasonlóan a Proxy Cache Szerver karakterisztikáját a B_{xc}, Y_{xc}, R_{xc} paraméterhármassal határozhatjuk meg.
Ha a felhasználó által kért fájl mérete nagyobb, mint a szerver kimenő puffere, akkor egy visszacsatolási ciklus kezdődik, mely addig tart, míg az igény kiszolgálása be nem fejeződik. Legyen

$$q = \min\left(1, \frac{B_s}{F}\right) \tag{6}$$

annak a valószínűsége, hogy a szerver az igényt elsőre ki tudja szolgálni és nem következik be visszacsatolási ciklus. Jelöljük λ_4' -vel a Web szerver kiszolgáló egységéhez érkező igények intenzitását figyelembe véve az igények esetleges visszacsatolását. Felhasználva az egyensúlyi egyenleteket kapjuk:

$$\lambda_4 = q * \lambda_4' \tag{7}$$

A fenti eredményeket felhasználva kapjuk egy igény válaszidejét. Jelölje T_{xc} valamint T a válaszidőt PCS használata esetén, illetve PCS hiányában:

$$T_{xc} = \frac{1}{\frac{1}{I_{xc}} - \lambda} + p * \left(\frac{1}{\frac{B_{xc}}{F * \left(Y_{xc} + \frac{B_{xc}}{R_{xc}} \right)} - \lambda_1} + \frac{F}{N_c} \right) +$$

$$+ (1-p) * \left(\frac{1}{\frac{1}{I_s} - \lambda_3} + \frac{1}{\frac{B_s}{F * \left(Y_s + \frac{B_s}{R_s} \right)} - \frac{\lambda_4}{q}} + \frac{F}{N_s} + \frac{1}{\frac{B_{xc}}{F * \left(Y_{xc} + \frac{B_{xc}}{R_{xc}} \right)} - \lambda_5} + \frac{F}{N_s} \right) \tag{8}$$

valamint,

$$T = \frac{1}{\frac{1}{I_s} - (\lambda + \Lambda)} + \frac{1}{\frac{B_s}{F * \left(Y_s + \frac{B_s}{R_s} \right)} - \frac{(1-P_b) * (\lambda + \Lambda)}{q}} + \frac{F}{N_s} + \frac{F}{N_c} \tag{9}$$

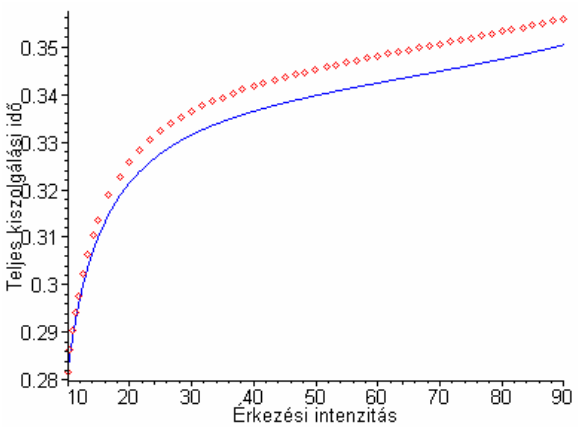
A T_{xc} válaszidő három részből tevődik össze: Az első annak az időtartama, míg eldől, hogy a Proxy szerver tartalmazza-e az igényelt fájlt. Ez a sorbanállás elméletből jól ismert M/M/1 folyamat várakozási idejéből adódik, ahol λ az érkezési intenzitás valamint $\frac{1}{I_{xc}}$ a kiszolgálási idő. A képlet második tagja annak a válaszideje, amikor az igény megtalálható a PCS-en, ahol a Proxy szerver kiszolgálási ideje $\frac{B_{xc}}{F * \left(Y_{xc} + \frac{B_{xc}}{R_{xc}} \right)}$, valamint $\frac{F}{N_c}$ az „utazási” idő míg az igényelt fájl keresztüljut a kliens hálózatán (N_c a kliens sávszélessége).

A képlet harmadik tagja reprezentálja annak az igénynek a válaszidejét, mely nem található meg a PCS-en. Ez további három részre bontható. Az első az egyszeri TCP inicializáláshoz szükséges idő, a második a kiszolgálóegységnél töltött idő, ahol a Web szerver kiszolgáló egységéhez érkező igények érkezési intenzitása $\lambda_4' = \frac{\lambda_4}{q}$, mely már tartalmazza a visszacsatolási ciklus intenzitását. A harmadik tag harmadik része a fentiekhez hasonlóan, a PCS-hez visszaérkező igények kliens felé való továbbításának az időtartamát tartalmazza.

Proxy szerver nélkül a modellünk a fentebb tárgyalt modellnek a leegyszerűsített változata.

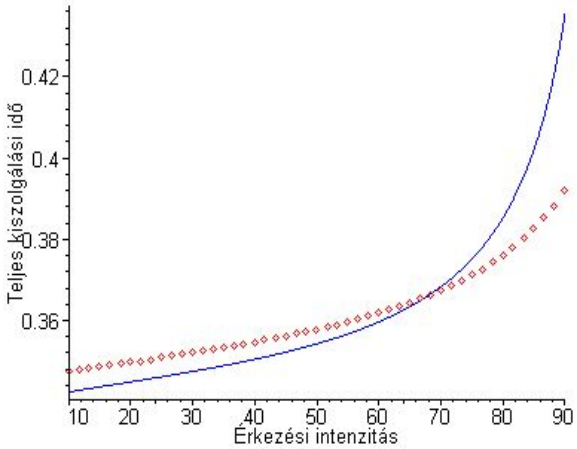
3. Numerikus eredmények

A numerikus számításokhoz a Bose és Cheng [1], [3] által használt paraméterértékeket használjuk: $I_s = I_{xc} = 0.004$ másodperc, $B_s = B_{xc} = 2000$ byte, $Y_s = Y_{xc} = 0.000016$ másodperc, $R_s = R_{xc} = 1250$ Mbyte/s, $N_s = 1544$ Kbit/s és $N_c = 128$ Kbit/s.
Az összes tárgyalt grafikonon szaggatott vonallal ábrázoltuk a teljes válaszidőt PCS létezésekor, míg a sima vonal a PCS nélküli válaszidőt mutatja.



$p = 0.1, F = 5000, \Lambda = 100, K = 100$

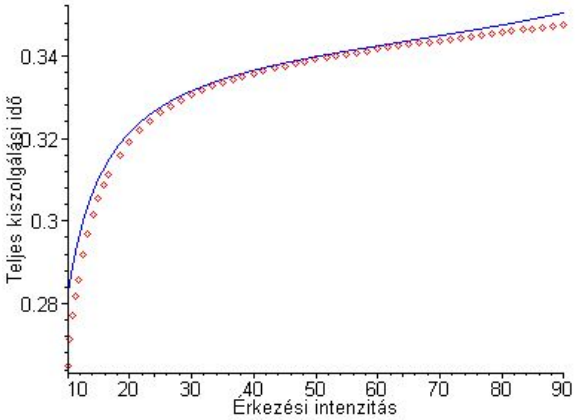
Kép 2.



$p = 0.1, F = 5000, \Lambda = 150, K = 100$

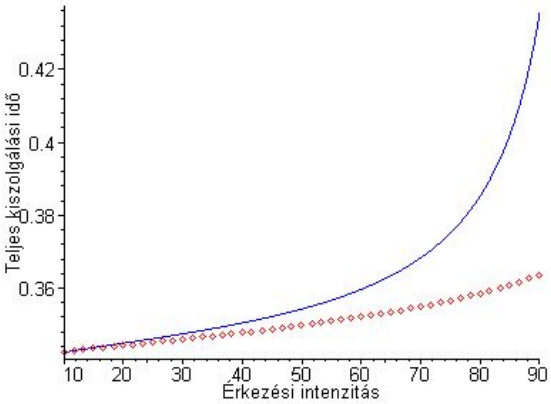
Kép3

Mind a két fenti grafikon esetében a találati valószínűség 0.1, az igényelt fájlok mérete 5000 byte, a Web szerver kapacitása K=100. Mint látható, amennyiben a külső érkezési intenzitás 100 igény/másodperc úgy a PCS beiktatása nagyobb válaszidőket eredményez. Viszont ha megnöveljük a külső érkezési intenzitást 150-re úgy $\lambda =$ 70 igény/s fölött a PCS haszna egyértelművé válik.



$p = 0.25, F = 5000, \Lambda = 100, K = 100$

Kép 4.



$p = 0.25, F = 5000, \Lambda = 150, K = 100$

Kép5.

A következő két grafikonon (Kép 4., Kép 5.) minden paramétert változatlanul hagytunk, kivéve a találati valószínűséget, melyet mindkét esetben 0.25-re emeltünk. Mint ahogyan látható, most már a kisebb külső érkezési intenzitás esetén is egy minimális haszna van, ha PCS-t használunk, nagyobb külső érkezési intenzitás esetén pedig a PCS előnye nyilvánvaló.
Mint ahogyan a numerikus eredményekből látszik annak eldöntése, hogy megéri-e egy Proxy Cache Szervert üzemeltetni nagyban függ az Internetet használók szokásaitól. Amennyiben a Proxyt használók nagyobb valószínűséggel akarnak ugyanazon dokumentumokat letölteni, vagy olyan oldalak iránt érdeklődnek melyek igen leterheltek a PCS használata számottevő javulást eredményezhet a válaszidők tekintetében.

λ	A klienstől érkező igények intenzitása
Λ	A külső igények érkezési intenzitása
F	Az igényelt fájl mérete
P	A PCS találati valószínűsége
B_{xc}	A PCS kimenő puffere
B_s	A Web szerver kimenő puffere
I_{xc}	A PCS –en való keresési idő
Y_{xc}	A szerver statikus ideje a PCS - esetén
R_{xc}	A PCS dinamikus szerver ideje
I_s	Egyszeri kapcsolat inicializálási idő
Y_s	A Web szerver statikus ideje
R_s	A Web szerver dinamikus szerver ideje
N_c	A kliens sávszélessége
N_s	A szerver sávszélessége

Táblázat 1.

Felhasznált irodalom:

[1] **Bose, I. , Cheng, H.K.,** Performance models of a firms proxy cache server. *Decision Support Systems and Electronic Commerce.,* **29** (2000), 45-57.
[2] **CacheFlow Inc.,** 1999. CacheFlow White Papers. Avaiable from htp://cacheflow.com/technology/
[3] **Menasce, D.A. , Almeida, V.A.F.,** *Capacity Planning for Web Performance: Metric, Models, and Methods.* Prentice Hall., (1998)
[4] **L.P. Slothouber**, A model of Web server performance. *5th International World Wide Web Conference, Paris, France.,* (1996)
[5] **C. Aggarwal, J.L. Wolf, P.S. Yu,** Caching on the World Wide Web, *IEEE Transactions on Knowledge and Data Engineering 11 (1999)*