

PRIMJENA RAČUNA REDOVA ČEKANJA POMOĆU ERLANG DISTRIBUCIJA ZA WEB APLIKACIJE

APPLICATION OF QUEUING THEORY AND ERLANG DISTRIBUTION FOR WEB APPLICATIONS

Dubravko Miljković

HEP-SIT, Vukovarska 37, Zagreb

Mob: 098 9825602

E-mail: dubravko.miljkovic@hep.hr

SAŽETAK

Opisana je primjena računa redova čekanja (queuing theory, teorije masovnog posluživanja) na dimenzioniranje broja elemenata sustava. Teorija redova čekanja moćan je alat za analizu složenih sustava. Cilj Erlang teorije je određivanje koliko točno elemenata za osiguranje servisa je potrebno da se zadovolje potrebe korisnika bez uskih grla i rastrošnog predimenzioniranja sustava. Uz uobičajena pojednostavljena i pretpostavke broj elemenata i vrijeme opsluživanja korisnika može se prilično precizno odrediti. Sustav se može pustiti u produkciju bez potrebe za obično nedostupnim i skupim stres testerom, a naknadno su obično potrebna samo manja podešenja na sustavu. Karakteristični primjeri riješeni uz pomoć Erlang-B i Erlang-C kalkulatora.

ABSTRACT

Application of waiting lines (queuing theory, theory of mass service) for sizing of system elements (and avoiding bottlenecks) is described. The theory of queues is a powerful tool for analyzing complex systems. The aim of Erlang traffic theory is to determine exactly how many elements are required to satisfy the needs of users without wasteful oversizing the system. With the usual simplifying assumptions the number of elements and user wait time can be quite accurately determined. The system can be put into production without the need for expensive and often inaccessible stress tester. Subsequently only minor adjustments to the system are usually required. Typical examples are solved with the help of Erlang-B and Erlang-C calculators.

1. UVOD

Realni računarski sustavi nemaju neograničeni kapaciteta. Ponekad se određena aktivnost na jednom serveru može odvijati samo za jednog ili nekoliko korisnika. Potrebno je naći odgovore na slijedeća pitanja: kako procijeniti dopušteno opterećenje servera, kako odrediti potreban broj uslužnih kanala, kako odrediti čekanje korisnika na uslugu i kolika je potrebno maksimalno trajanje usluge za određeno vrijeme čekanja i određen broj uslužnih kanala. Na ova i slična pitanja odgovori se mogu naći primjenom teorije čekanja. Rad je usredotočen na praktičnu primjenu i gdje god je to moguće izbjegava se komplicirana matematska notacija. Čitatelj kojeg interesira teoretski pristup tematici može naći puno informacija tutorijalnog karaktera na internetu.

2. AGNER KRARUP ERLANG

Agner Krarup Erlang (slika 1) je danski matematičar, statističar i inženjer koji je zasnovao područja modeliranja prometa i teorije čekanja, [1]. Modeliranje prometa je postupak izrastao u telefonskoj industriji, a teorije koje je razvio u uporabi danas razvijene su između 1909 i 1917. Erlang je računao udio korisnika koji moraju čekati na uspostavu poziva van sela budući da su sve telefonske linije zauzete. Erlangovi rezultati na polju Teorije telefonskog saobraćaja najbolje se reprezentiraju putem Erlangovih formula od kojih su u ovo radu za nas bitne Formula gubitaka, B formula, EBF) i Formula čekanja, C formula, ECF (postoji i treća interkonekciona ili D formula).



Slika 1 Agner Krarup Erlang (izvor Wikipedia)

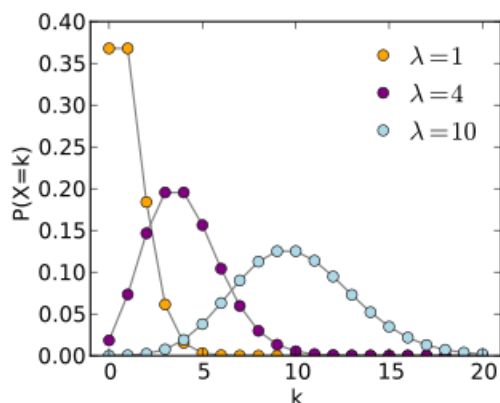
3. POLAZIŠTE U PROBLEMIMA IZ TELEKOMUNIKACIJA

Erlangov teorijski pristup prometnim problemima preko statističke ravnoteže, kao i eksplicitna rešenja, imali su presudan utjecaj na razvoj teorije telekomunikacijskog prometa posljednjih 100 godina. Teorija je našla veliku primjenu, posebice u telefonskim centrala i call centrima. Dolazak telefonskih poziva modeliran je kao Poissonov proces, [2], a isti se može primijeniti i na korisnike web aplikacija. Poissonova razdioba, [3], opisana je sa funkcijom gustoće razdiobe (1).

$$f(k, x, \lambda) = \frac{(\lambda x)^k e^{-\lambda x}}{k!} \quad x, \lambda \geq 0 \quad (1)$$

Poissonova distribucija daje vjerojatnost boja dolazaka tijekom datog vremena x , gdje je λ intenzitet dolazaka zahtjeva.

Sama razdioba prikazana je na slici 2. Horizontalna os je indeks k , odnosno broj pojava. Funkcija je definirana samo na cjelobrojnim vrijednostima k . Vrijednosti funkcije za cjelobrojne iznose k povezan je linijom zbog lakšeg uočavanja oblika funkcije.



Slika 2 Poissonova razdioba (izvor Wikipedia)

Vjerojatnost da diskretna varijabla poprima točno određenu vrijednost dana je sa (2):

$$f(k, \lambda) = \Pr(X = k) = \frac{\lambda^k e^{-\lambda}}{k!} \quad (2)$$

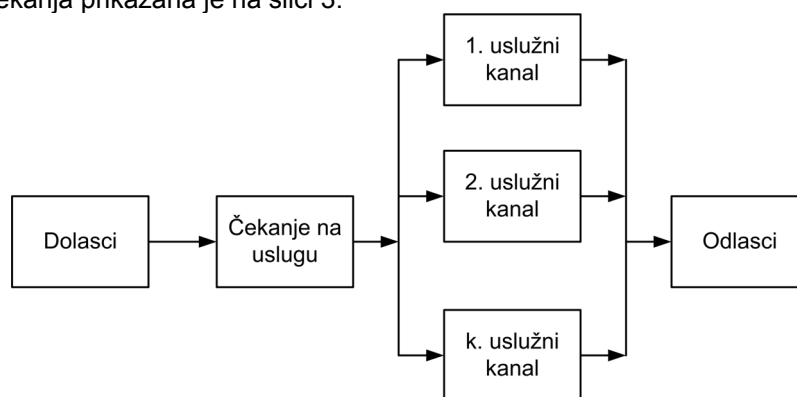
4. REDOVI ČEKANJA

Red čekanja je problem koji se pojavljuje u praksi kada određeni broj jedinica (ljudi, predmeti...) koji traže odgovarajuću uslugu ili obradu moraju čekati. Nastanak reda čekanja je posljedica nesklada između kapaciteta uslužnih mjesta i zahtjevima korisnika usluge. Sustav poslužuje dolazne zahtjeve, ali je istovremeno i konačnih mogućnosti.

Osnovni pojmovi vezani uz redove čekanja su, [4]:

- Ulazne jedinice (korisnici usluga, klijenti, potrošači, stranke, eng. customers)
- Uslužni kanali (uslužna mjesta, eng. servers)
- Red čekanja (eng. queue)

Struktura reda čekanja prikazana je na slici 3.



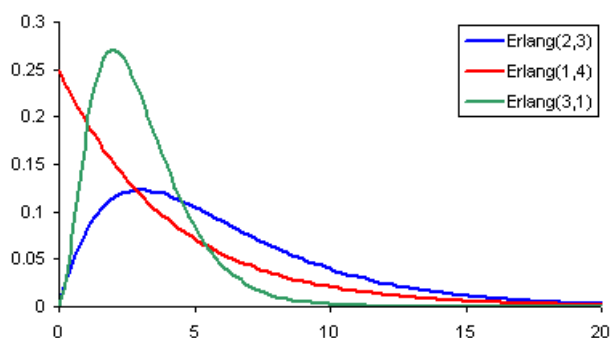
Slika 3 Struktura redova čekanja

5. ERLANG DISTRIBUCIJA

Erlang distribucija je kontinuirana razdioba vjerojatnosti sa širokim primjenom, [5]. Funkcija gustoće razdiobe Erlang distribucije (slika 4), dana je sa

$$f(x; k, \lambda) = \frac{\lambda^k x^{k-1} e^{-\lambda x}}{(k-1)!} \quad x, \lambda \geq 0 \quad (3)$$

Parametar k zove se shape parametar (parametar oblika), a λ intenzitet dolazaka zahtjeva.



Slika 4 Erlang distribucije za različite parametre (izvor Vose Software)

Događaji koji se javljaju neovisno s nekom prosječnom frekvencijom modelirani su kao Poissonov proces. Vremena čekanja između k događaja su distribuirani u skladu s Erlang distribucijom. Erlang distribucija daje vjerojatnost vremenskog intervala za dati broj dolazaka k . Za razliku od Poissonove distribucije, kod Erlang distribucije pretpostavlja se da imamo dolazak u trenutku $t=0$ (koji se ne broji u broj dolazaka k) i trenutku $t=x$ (koji se broji u broj dolazaka k).

Postoje još dvije druge Erlang distribucije: Erlang B distribucija i Erlang C distribucija s pripadnim formulama.

6. WEB APLIKACIJE

Web aplikacije koriste ograničene resurse prisutne na aplikacijskom serveru kao i na drugim serverima (file server, print server) potrebnim za korištenje aplikacije. Pri tome se susrećemo s ograničenim (konačnim) brojem uslužnih kanala i redovima čekanja. Primjena Erlang B i Erlang C formule biti će prikazana za slijedeće probleme:

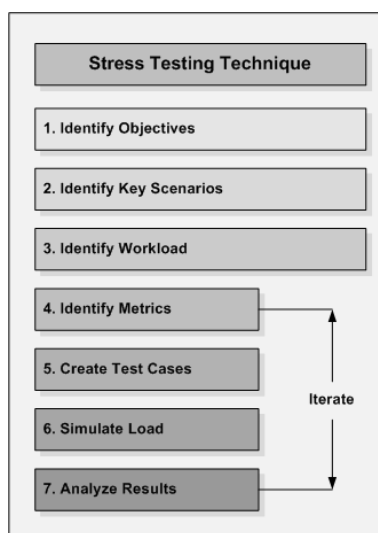
- Forms sesije
- Reports serveri
- Host komande
- Komunikacija s file serverom
- Komunikacija s print serverom
- Printanje - Print job-ovi i red čekanja
- Korištenje aplikacija koje dopuštaju pokretanje samo jedne instance (na terminal serverima)

7. PERFORMANCE/LOAD/STRES TEST SUSTAVA

Prije uporabe računarskog sustava od strane krajnjih korisnika bilo bi poželjno obaviti load test da bi utvrdili performanse sustava pod realnim opterećenjem. Postoji i sistemski stres test koji predstavlja testiranje performansi sustava pod povećanim opterećenjem (to ga razlikuje od load test-a). Takvo testiranje je fokusirano na robusnost aplikacije, raspoloživost i pouzdanost pod ekstremnim uvjetima. Stres test pronalazi odgovore na slijedeća pitanja:

- Koje vrijeme odziva će postizati korisnici?
- Kako dugo sustav može izdržati povećanu aktivnost korisnika?
- Kako često će se korisnici susretati s pogreškama na sustavu i trebati se rekonektirati?
- Može li se sustav vratiti sam u normalno stanje nakon stres testa s abnormalnim opterećenjem?

Load/stres tester je uređaj (software ili kombinacija hardware i software) koje simulira opterećenje sustava. Koristi se za određivanje stabilnosti sustava pod opterećenjem. Simulira simultani dolazak korisnika sa odvojenih PC strojeva (različite IP adrese). Uporabom load/stres testera postižu se vrlo dobre procjene, ali uređaj je vrlo skup (alternativa je vrlo skupi najam) i zahtijeva iskusnog operatera. Sam postupak za stre test prikazan je na slici 5.



Slika 5 Postupak stress test-a (izvor Microsoft)

Budući da su ovakvi uređaji većinom nedostupni, ideja je da se mnoge performanse sustava mogu predvidjeti analitičkim pristupom primjenom Erlang distribucije uz uporabu svima dostupnih i besplatnih web Erlang B i C kalkulatora.

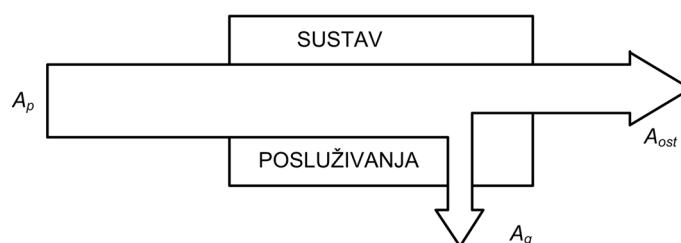
8. ERLANG

Erlang je bezdimenziona jedinica koja se koristi u telefoniji kao statistički pokazatelj telekomunikacijskog prometa [6, 7]. Definira se kao iskorištenost kanala u vremenu. "Erlang" je korišten u skandinavskim zemljama i označava jedinicu telefonskog prometa. Striktno govoreći Erlang predstavlja kontinuiranu uporabu jednog govornog kanala. Jedna telefonska linija koja je neprekidno jedan sat zauzeta pozivom nosi jedan Erlang prometa. Međunarodno priznanje jedinice slijedilo je nakon kraja drugog svjetskog rata. Na slijedećem primjeru pokazano je kako se određuje promet u Erlanzima. Imamo 30 poziva po satu pri čemu svaki poziv ima prosječno trajanje od 5 minuta.

Minuta prometa u satu = Broj poziva x Trajanje
 Minuta prometa u satu = 30 x 5
 Minuta prometa u satu = 150
 Sati prometa u satu = 150 / 60
 Sati prometa u satu = 2.5 Erlanga

Dakle, za zadani broj poziva sa zadanim prosječnim trajanjem promet je 2.5 Erlanga.

9. ERLANG B



Slika 6 Poopćeni model sustava posluživanja s gubicima

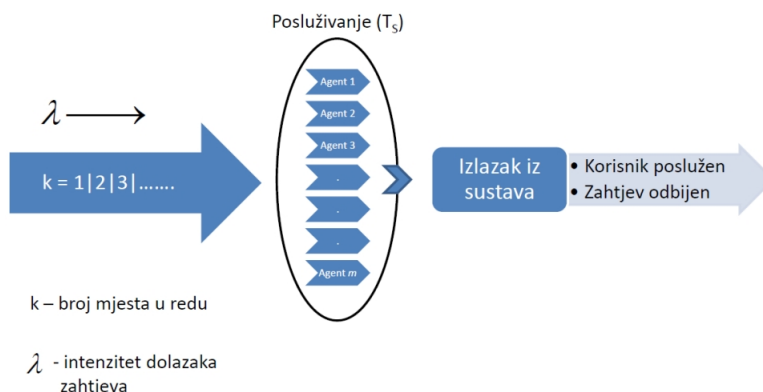
Ukoliko se odustaje od neopsluženih zahtjeva za uslugom, "target service" (slika 6), za rješavanje koristimo Erlang B formulu, [6, 7, 8]:

$$p_B(A_p, m) = \frac{\frac{A_p^m}{m!}}{\sum_{i=0}^m \frac{A_p^i}{i!}} \quad (4)$$

pri čemu je

p_B - vjerojatnost da će zahtjev biti odbijen (vjerojatnost da su svi poslužitelji zauzeti)
 m - broj poslužitelja
 A - promet (Erlang-a)

10. ERLANG C



Slika 7 Erlang C formula pogodna je za primjenu kod redova čekanja s jednim ili više uslužnih kanala

Ukoliko neopsluženi zahtjevi za uslugom ulaze u red čekanja sve dok se ne opsluže (slika 7), [6], za rješavanje koristimo Erlang C formulu [7, 8]:

$$p_C(A_p, m) = \frac{A_p^m}{A_p^m + m! \left(1 - \frac{A_p}{m}\right) \sum_{i=0}^{m-1} \frac{A_p^i}{i!}} \quad (5)$$

pri čemu je

- p_C - vjerojatnost da će zahtjev biti odbijen
(vjerojatnost da su svi poslužitelji zauzeti)
- m – broj poslužitelja
- A - promet (Erlang-a)

11. DEFINIRANJE KVALITETE USLUGE

Cilj Erlang u teoriji prometa je točno odrediti koliko elemenata za pružanje usluge treba osigurati kako bi zadovoljio korisnike, bez rasipanja resursa (overprovisioning). Kako bi se to postiglo potrebno je definirati ciljnu vrijednost za Grade of Service (GoS) ili Quality of Service (QoS), [7]. Glavni parametri bili bi vjerojatnost blokiranja usluge (npr. 0.01) i vrijeme čekanja na uslugu (npr. na 80% zahtjeva za uslugom čeka se ispod 10 s). Navedene vrijednosti potrebno je definirati za Busy hour ili glavni prometni sat. Glavni prometni sat je jedan najprometniji sat u razdoblju dana, kada je najvjerojatnije da će dolazni zahtjevi za uslugom biti odgođeni ili blokirani. To je opterećenje za koji se kapacitete resursa treba izračunati.

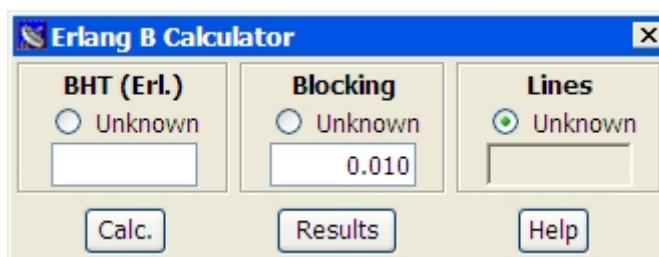
12. ERLANG WEB KALKULATORI

Na Internetu postoji veliki broj interaktivnih programa koji omogućavaju pronalaženje parametara prometa po Erlangovim formulama. Ovi kalkulatori su jednostavni za korištenje, zadovoljavaju većinu potreba i ne zahtijevaju detaljno poznavanje teorije već samo operativno, [9, 10, 11, 12, 13]. Kod jednostavnijih kalkulatora potrebno je specificirati (ili će se dobiti kao rezultat) promet u Erlang jedinicama dok bolji kalkulatori to izračunaju na osnovu broja zahtjeva za uslugom po satu i prosječnom trajanju usluge.

12.1. Erlang B kalkulator

Tri varijable koje koristi kalkulator su Busy Hour Traffic (BHT), Blocking i Lines (slika 8), [9]:

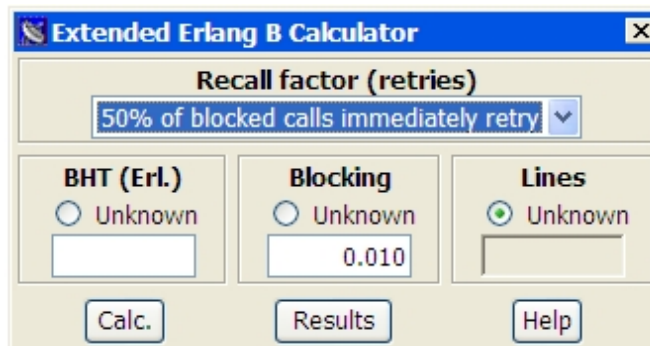
- Busy Hour Traffic (Promet glavnog sata) u Erlanzima je broj sati usluge tijekom glavnog prometnog sata sustava
- Blocking je vjerojatnost blokiranje prometa ovisno o opterećenju (npr. 0.03 znači da na 100 zahtjeva za uslugom 3 su blokirana)
- Broj uslužnih agenata za prihvatljivu vjerojatnosti blokiranja



Slika 8 Erlang B web kalkulator

12.2. Prošireni (extended) Erlang B kalkulator

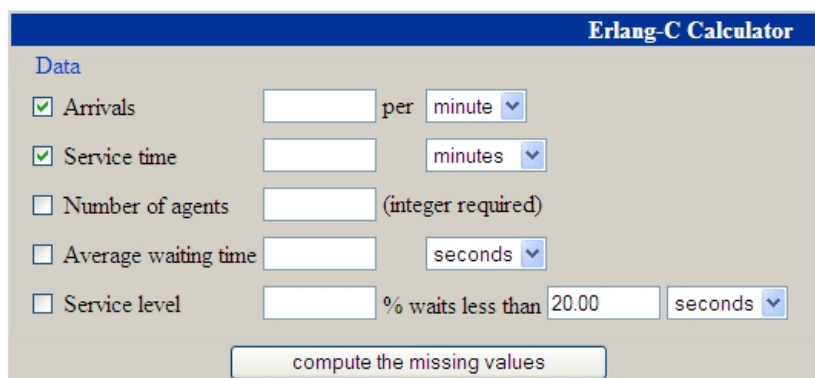
Uključena je dodatna mogućnost specificiranja Recall Factor-a (postotak korisnika koji nakon neuspjeha odmah pokušavaju ponovo dobiti uslugu (slika 9), [10]. Prikladno kada u aplikaciju ugradimo provjeru u petlji za retry.



Slika 9 Prošireni (Extended) Erlang B web kalkulator

12.3. Erlang C kalkulator

Za razliku od prethodnih Erlang B web kalkulatora, ovaj Erlang C web kalkulator (slika 10) ne zahtijeva specificiranje prometa u Erlanzima već sam izračunava promet na temelju specificiranih parametara, [11]. Potrebno je specificirati barem tri parametra (pri tome odabrati barem dva od prva tri parametra)



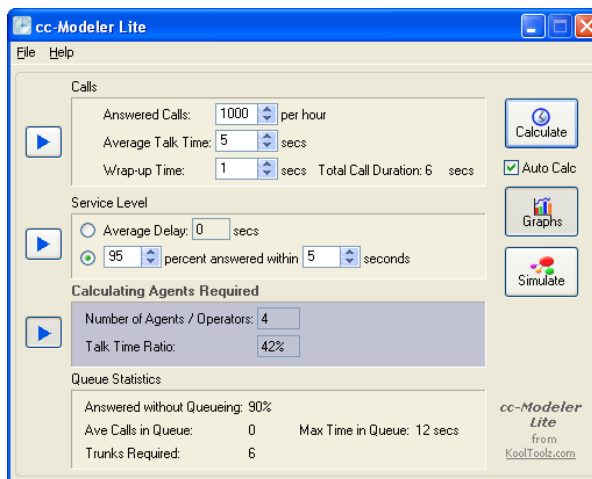
Slika 10 Erlang C web kalkulator

13. STANDALONE SOFTWARE

Osim Erlang WEB kalkulatora postoje i specijalizirani programi koji obavljaju iste proračune s dodatnim brojnim opcijama (npr. besplatan Erlang CC Modeler Lite). Postoji i Erlang CC Modeler Pro koji omogućava analizu prometa koji se mijenja tijekom dana (zanimljivo premda smo vezani za Busy Hour), [14].

Omogućava grafički prikaz očekivanog boravka u redu čekanja i prosječnog čekanje ovisno o broju uslužnih agenata.

Izgled dialog boxa za Erlang CC Modeler Lite prikazan je na slici 11, a izgled izvještaja koje generira program na slici 12.



Slika 11 Erlang CC Modeler Lite Dialog Box

- 1 Calculate**
The number of agents you need.
What the average speed to answer will be.
- 2 Model**

 - How many extra calls could be handled with a certain number of agents.
 - The effect of reducing talk times or wrapup time.
 - The productivity of your staff.
- 3 Show**

 - How many calls will be queued.
 - The longest time that a caller will wait.
 - The number of incoming lines you need.

- 4 Auto Calculate**
vary any of the parameters and instantly see the effect on others.
- 5 Forecast**
See the impact of variations or growth in call volumes on:

 - The number of agents needed.
 - The average speed to answer.

Slika 12 Erlang CC Modeler Lite Graph Reports

14. KADA KORISTITI ERLANG B, A KADA ERLANG C

Odluka koju formulu ćemo koristiti ovisi o tome da li korisnici u slučaju blokiranja usluge ulaze ili ne ulaze u red čekanja. Ukoliko je usluga blokirana, ali korisnik ne ulazi u red čekanja primijeniti ćemo Erlang B. Ako je pak usluga blokirana, ali korisnik ulazi u red čekanja dok se ne oslobodi uslužni kanal (agent) primijeniti ćemo Erlang C. Ako pak nema reda čekanja, a vjerojatnost blokiranja usluge je dovoljno mala (npr. $p < 0.1$), možemo u aplikaciju ugraditi kod koji provjerava da li je usluga ostvarena i ako nije ponavlja zahtjev za uslugom (Extended Erlang B).

15. PRIMJERI

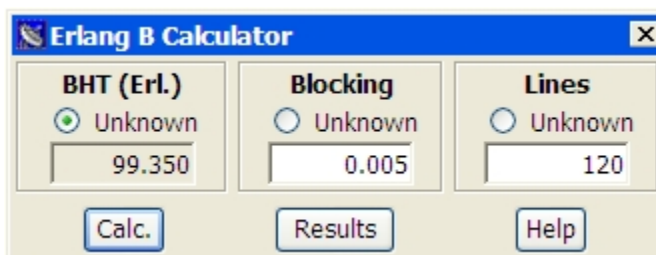
15.1. Dimenzioniranje iAS servera – Forms sesije

Ukoliko se korisnici povremeno spajaju na web aplikacije, Erlang B kalkulator možemo koristiti za dimenzioniranje iAS servera (slike 13 i 14). Najveći istovremeni broj korisnika Forms sesija određen je s Windows heap size, a postoje i memorijska ograničenja OS-a. Ukoliko znamo koliko je prosječno trajanje Forms sesije i dopuštenu vjerojatnost blokiranja kod uspostave sesije možemo odrediti ukupan broj uspostava Forms sesija (Busy Hour). Rezultat ćemo dobiti kao BHT u Erlanzima, a ukupan broj (ne istovremenih korisnika možemo dobiti dijeleći ga s prosječnim trajanjem Forms sesije).

Parametri s kojima radimo su:

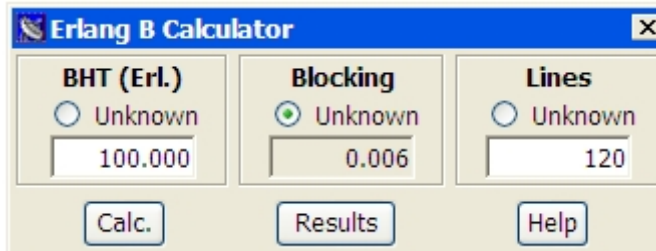
- Najveći istovremeni broj korisnika Forms sesija
- Dopusštena vjerojatnost blokiranja kod uspostave sesije
- Ukupan broj uspostava sesija tijekom busy hour-a
- Prosječno trajanje Forms sesije

Produkt posljednja dva parametra specificiraju BHT (odnosno poznavajući BHT i jedan od parametara možemo odrediti drugi parametar).



BHT (Erl.)	Blocking	Lines
<input checked="" type="radio"/> Unknown 99.350	<input type="radio"/> Unknown 0.005	<input type="radio"/> Unknown 120
Calc.	Results	Help

Slika 13 Dimenzioniranje iAS-a – određivanje ukupnog broja uspostava sesija tijekom Busy hour-a



BHT (Erl.)	Blocking	Lines
<input type="radio"/> Unknown 100.000	<input checked="" type="radio"/> Unknown 0.006	<input type="radio"/> Unknown 120
Calc.	Results	Help

Slika 14 Dimenzioniranje iAS-a – određivanje vjerojatnosti blokiranja kod uspostave sesije za zadani promet

15.2. Reports serveri

Erlang C kalkulator može se koristiti za određivanje potrebnog broja prometa reports servera i procjenu vremena koju će korisnici provesti čekajući u redu za uslugu. Reports serveri mogu biti in-process i stand-alone. U slučaju korištenja starog iAS-a (iAS 1.0.2. i Forms/reports 6) in-process reports server posjeduje samo jedan reports engine, koji može opsluživat istovremeno samo jedan job i kod velikog broja zahtjeva za uslugom stvaraju se dugački redovi. Kod novih aplikacijskih servera (iAS 10g Rel. 1 i Rel.2) in-process reports server ima svoju konfiguracijsku datoteku gdje se može definirati više reports engine-a, može obrađivati više job-ova istovremeno pa korisnici usluge ne ovise samo o jednom engine-u, a redovi čekanja su bitno kraći. Stand alone reports serveri (već i na starom iAS 1.0.2 pa nadalje) redovito imaju mogućnost konfiguriranja i korištenja više reports engine-a pa us redovi i vremena čekanja kraći. Može se pomisliti zašto ne postaviti broj reports engine-a na neku veliku vrijednost, pa ne moramo razmišljati puno o redu čekanja. Međutim, postavljanje broja engine-a iznad neke vrijednosti (obično nekoliko engine-a) osim velike potrošnje memorije za svaki engine (a time i manjeg mogućeg broja korisnika Forms sesija) neće imati skoro nikakav utjecaj na doživljaj

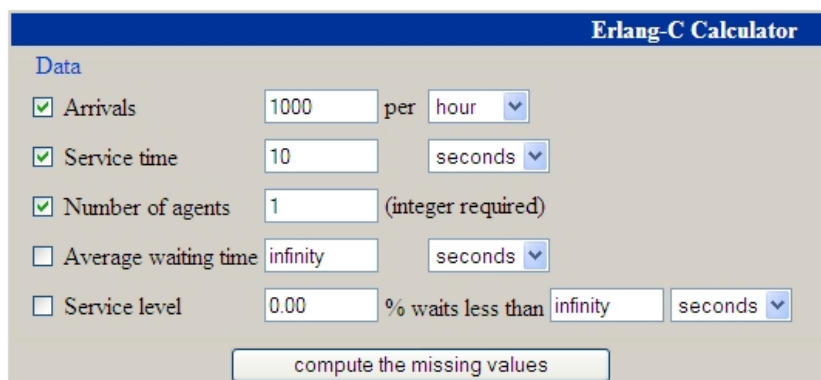
krajnjeg korisnika. S druge strane konfiguriranje premalo engine-s će uzrokovat formiranje repova i duga vremena čekanja na report.

Prilikom uporabe Erlang C kalkulatora posebno je unijeti prosječno vrijeme izvođenja reporta. njega možemo odrediti pokretanje reporta iz aplikacije ili praćenjem procesa job-ova uz pomoć Task manager-a (posebno dobar je Process Explorer od Sysinternals). Nakon izvođenja više desetaka reporta u vrijeme kad je sustav slabije opterećen (off Peak Hours) imat ćemo dobru procjenu prosječnog trajanja izvođenja reporta za pojedini reports server. Potom dobiveni podatak zajedno s brojem zahtjeva na uslugu u jedinici vremena i brojem uslužnih kanala (agenata, u konkretnom slučaju reports engine-a) unesemo u web kalkulatori. Zatim inkrementalno (ili uz male skokove) mijenjamo broj engine-a dok ne dođemo do zadovoljavajućeg vremena čekanja na report. (primjeri 1-3). Alternativno možemo specificirati Service level (Primjer 4).

Idućih nekoliko primjera (slike 15-18) ilustrira vremena čekanja na report za različiti broj reports engine-a.

□ Primjer 1

- 1000 reporta po satu
- vrijeme izvođenja reporta 10s
- 1 reports engine

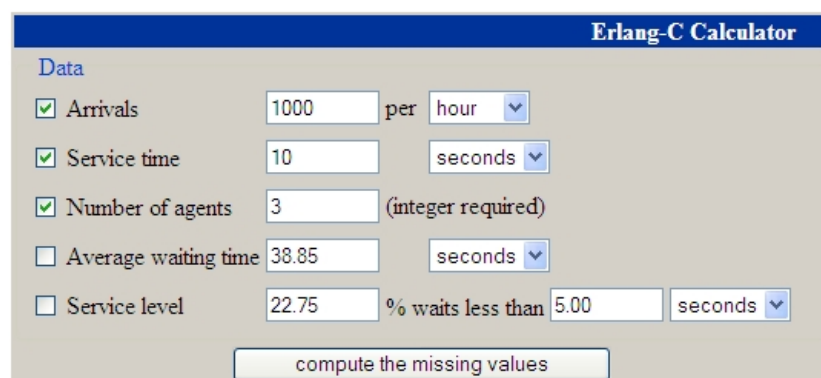


Erlang-C Calculator			
Data			
<input checked="" type="checkbox"/> Arrivals	1000	per	hour
<input checked="" type="checkbox"/> Service time	10		seconds
<input checked="" type="checkbox"/> Number of agents	1	(integer required)	
<input type="checkbox"/> Average waiting time	infinity		seconds
<input type="checkbox"/> Service level	0.00	% waits less than	infinity seconds
compute the missing values			

Slika 15 Primjena Erlang C kalkulatora za izračun čekanja na uslugu i service level-a

□ Primjer 2

- 1000 reporta po satu
- vrijeme izvođenja reporta 10s
- 3 reports engine

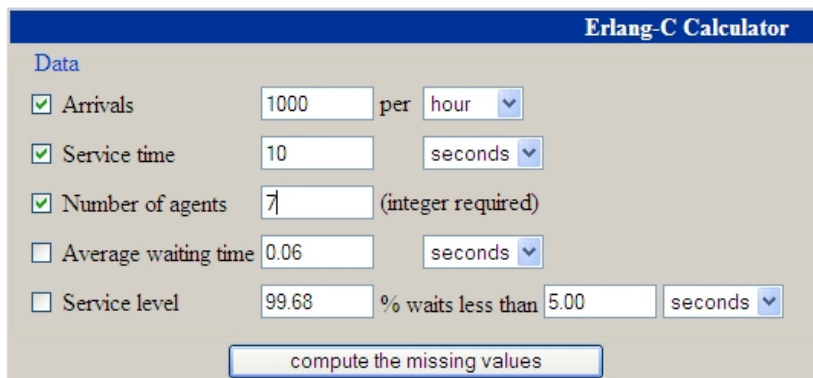


Erlang-C Calculator			
Data			
<input checked="" type="checkbox"/> Arrivals	1000	per	hour
<input checked="" type="checkbox"/> Service time	10		seconds
<input checked="" type="checkbox"/> Number of agents	3	(integer required)	
<input type="checkbox"/> Average waiting time	38.85		seconds
<input type="checkbox"/> Service level	22.75	% waits less than	5.00 seconds
compute the missing values			

Slika 16 Primjena Erlang C kalkulatora za izračun čekanja na uslugu i service level-a II

□ Primjer 3

- 1000 reporta po satu
- vrijeme izvođenja reporta 10s
- 7 reports engine



Erlang-C Calculator

Data

Arrivals 1000 per hour

Service time 10 seconds

Number of agents 7 (integer required)

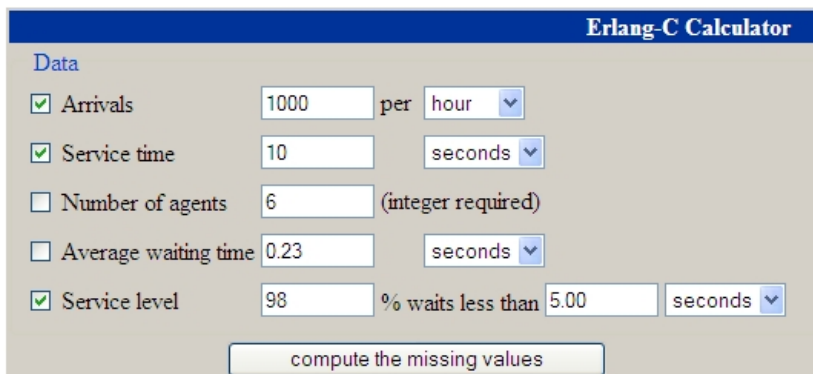
Average waiting time 0.06 seconds

Service level 99.68 % waits less than 5.00 seconds

compute the missing values

Slika 17 Primjena Erlang C kalkulatora za izračun čekanja na uslugu i service level-a III

- Primjer 4 – specificira potrebni Service level (slika 18)
 - 1000 reporta po satu
 - vrijeme izvođenja reporta 10s
 - Service level: 98% čeka manje od 5s
 - Potrebno 6 agenata (reports engines)



Erlang-C Calculator

Data

Arrivals 1000 per hour

Service time 10 seconds

Number of agents 6 (integer required)

Average waiting time 0.23 seconds

Service level 98 % waits less than 5.00 seconds

compute the missing values

Slika 18 Primjena Erlang C kalkulatora za izračun čekanja na uslugu i potrebnog broja reports engine-a uz specificirani service level

Iz gornjih primjera vidi se da nakon 6 reports engine-a nema značajnog poboljšanja usluge (7 engine-a postaje pomalo već luksuz).

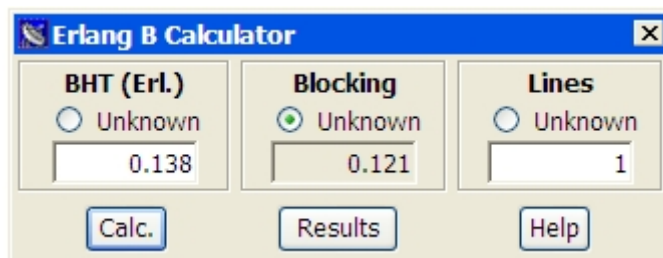
15.3. HOST command

Ponekad nije moguće istovremeno pokrenuti više istih HOST komandi na istom serveru. Npr. prvi korisnik može pokrenuti HOST komandu, a drugi je ne može pokrenuti dok prethodna ne završi. Ponekad se može pokušati s Run-as (Sysinternals utilities) s kredencijama drugog user-a, ali obično ni to ne pomaže. Isto tako pojedine instance komande mogu se pokretati pod različitim imenima:

- Ponekad pomaže RENAME trik
 - 1st instance -> run original App.exe
 - 2nd instance -> run copied App_1stCopy.exe
 - 3rd instance -> run 2nd copied App_2ndCopy.exe

No, kad sve to ne pomaže potrebno je procijeniti najveći broj pokretanja aplikacije u jedinici vremena uz prihvatljivu vjerojatnost blokiranja zahtjeva.

- Primjer (slika 19)
 - 100 zahtjeva po satu, trajanje zahtjeva 5 sekundi (BHT = 0.138)
 - Vjerojatnost blokiranja uz 1 uslužni kanal čak 12% (u praksi previše!)



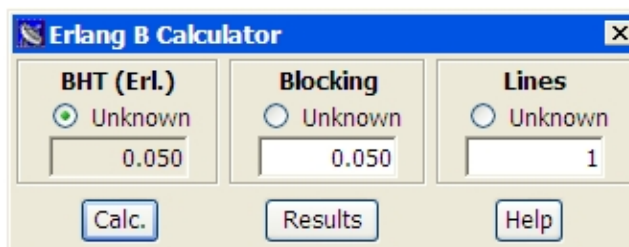
Slika 19 Procjena vremena blokiranja usluge koja se ostvaruje pokretanjem uz pomoć HOST komande

Ukoliko je dobivena vjerojatnost blokiranja usluge prevelika preostaje nam da pokušamo nekakvom optimizacijom skratiti vrijeme koje zauzima izvođenje dotične HOST komande. Ako smo pak zadovoljni s dobivenom vjerojatnosti blokiranja i ona je dovoljno mala (npr. $p < 0.1$) možemo u kod same aplikacije ugraditi provjeru da li je HOST komanda uspjela i ako nije ponavljati poziv HOST komande.

15.4. File server

Sličnu situaciju imamo i kod pristupa iAS-a na File Server. Jedan korisnik s iAS-a može pristupiti na file server. Može i više njih, ali sa različitih iAS-a (npr. ako su u NLB clusteru), no ne mogu dva korisnika istovremeno s istog iAS-a. Osim toga ako na File Serveru čuvamo npr. dokumente u pdf formatu možda na File Serveru imamo instaliran programi koji omogućuje pretraživanje pdf dokumenata (ponavlja se i problematika slična kao kod pokretanja uz pomoć HOST komande, ali ovaj put s remote exec. – PsExec - Sysinternals). Budući da nema reda čekanja već samo blokiranje usluge koristi se kao i u prethodnom primjeru Erlang B formula.

- Primjer (slika 20)
 - Dopuštena vjerojatnost blokiranja 0.05
 - BHT 0.05
 - Ekvivalentno 200 zahtjeva u trajanju 0.9 s ili 100 zahtjeva od 1.8 s



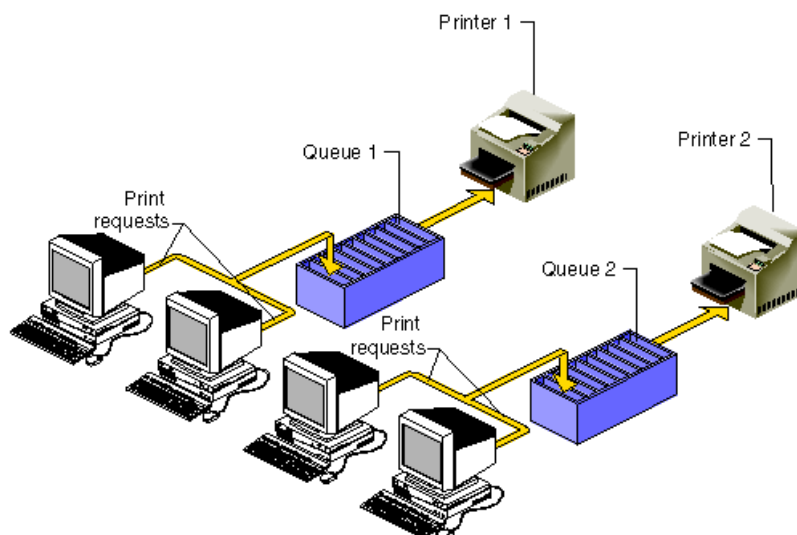
Slika 20 Izračun BHT uz specificiranje vjerojatnosti blokiranja usluge na jednom uslužnom kanalu

15.5. Print server

Identičnu situaciju imamo i kod komunikacije sa print serverom. Jedan korisnik s iAS-a može pristupiti na print server. Može i više njih, ali sa različitih iAS-a (npr. ako su u NLB clusteru), no ne mogu dva korisnika istovremeno s istog iAS-a. Potrebno je odrediti koliko prosječno dugo smije trajati komunikacija s Print Serverom za dati priljev zahtjeva i uz prihvatljivu vjerojatnost blokiranja zahtjeva za komunikacijom. Budući da nema reda čekanja već se usluga blokira koristi se Erlang B formula.

15.6. Printanje

Rezultati obrade web aplikacije u vidu reporta mogu se ispisati na grupi štampača. Budući da svaki ispis traje neko vrijeme, a na raspolaganju su jedan ili više (linijskih?) štampača doći će do formiranja reda za čekanje (print queue), slika 21. Budući da sada imamo red čekanja, primjenom

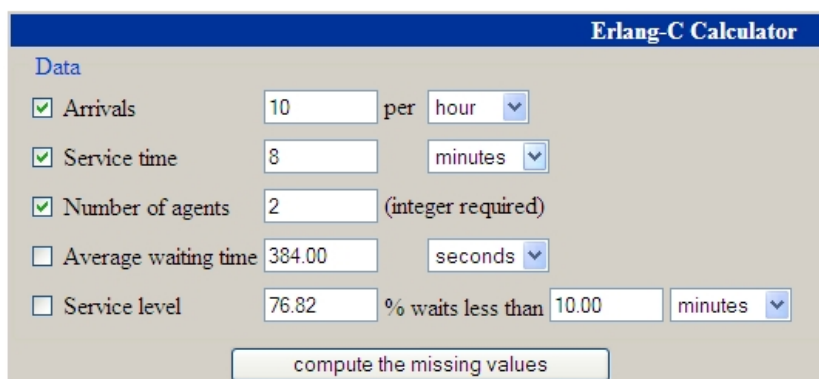


Slika 21 Redovi čekanja za printanje (izvor SGI)

Erlang C formule možemo odrediti koliko štampača trebamo i koliko ćemo vremena provesti u redu za ispis.

Primjer

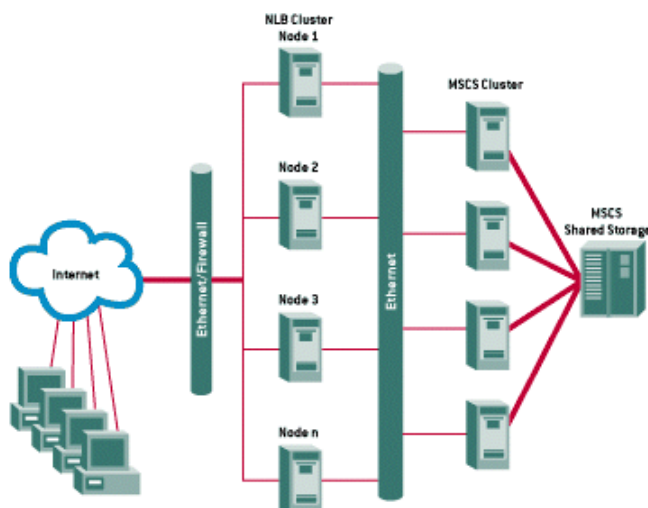
- Broj print job-ova po satu (busy hour) je 10
- Prosječno trajanje print job-a je 8 minuta
- Broj raspoloživih printera je 2
- Prosječno čekanje print job-a je 384 sekunde
- 76.82% korisnika čeka na printanje job-a manje od 10 minuta



Slika 22 Izračun vremena čekanja za print job-ove

15.7. NLB cluster

Kad iscrpimo sve mogućnosti smanjivanja vremena izvođenja usluge na nodu, a ne možemo povećati broj uslužnih kanala na nodu, ako ne ostvarujemo potreban ulazni promet možemo dodavati nodove u NLB cluster i time povećati broj uslužnih kanala, slika 23. Nažalost, time se ne dobiva i red čekanja, pa se promet ponovo analizira s Erlang B formulom.



Slika 23 NLB cluster - broj uslužnih kanala možemo povećati i dodavanjem novog noda (izvor Orion eSolutions)

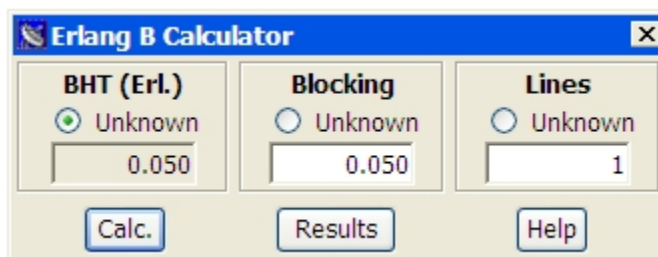
15.8. Terminal server

Erlang B formula može se osim za web aplikacije primijeniti za proračun vjerojatnosti blokiranja programa instaliranih na terminal serveru, a za koje pokušaj istovremenog pokretanje više instanci nije predviđen. Na temelju prosječnog trajanja korištenja programa i priljeva zahtjeva za usluge u glavnom satu prometa može se upotrebom Erlang B kalkulatora procijeniti vjerojatnost blokiranja usluge.

Primjer

- Dopuštena vjerojatnost blokiranja usluge je 0.05
- Broj uslužnih kanala je 1
- Izračunati BHT je 0.05

Iz dobivenog BHT koji je produkt broja ulaznih zahtjeva i prosječnog vremena korištenja instance programa na terminal serveru, uz specifikaciju jednog od parametara jednostavno izračunamo drugi.



Slika 24 Izračun BHT uz specificiranje vjerojatnosti blokiranja usluge na jednom uslužnom kanalu (jedna instanca programa koja se može pokrenuti na cjelokupnom terminal serveru)



Slika 25 Terminal server (izvor Winderam)

Erlang B formula može se iskoristiti i za određivanje broja licenci za uporabu terminal servera, slika 25. Pri tome treba znati prosječno trajanje sesije i dopuštenu vjerojatnost blokiranja usluge (login). Nakon unosa izračunatog BHT i vjerojatnosti blokiranja dobivamo potreban broj licenci.

16. ZAKLJUČAK

Teorija redova čekanja i Erlang distribucije omogućuje inicijalno dimenzioniranje sustava koje se kasnije fino tjunira. Analitičkim pristupom primjenom Erlang B i Erlang C kalkulatora (ovisno da li postoji red čekanja) možemo procijeniti vjerojatnosti blokiranja usluge, broj korisnika za prihvatljivo vrijeme čekanja, vremena čekanja korisnika pod opterećenjem, potreban broj uslužnih kanala. Na taj način mogu se dobiti okvirne procjene budućeg iskustva korisnika bez uporabe skupih performance/load/stress testera.

17. LITERATURA

1. Agner Krarup Erlang http://en.wikipedia.org/wiki/Agner_Krarup_Erlang
2. Poisson process, http://en.wikipedia.org/wiki/Poisson_process
3. Poisson distribution, http://en.wikipedia.org/wiki/Poisson_distribution
4. Redovi čekanja, <http://www.pfri.uniri.hr/~bdrascic/OI/redovi-cekanja.pdf>
5. Erlang distribution, http://en.wikipedia.org/wiki/Erlang_distribution
6. Osnove tehnologije prometa, http://e-student.fpz.hr/Predmeti/O/Osnove_tehnologije_prometa/Materijali/Vjezbe_TK_promet.pdf
7. I. Angus, An Introduction to Erlang B and Erlang C, <http://www.tarrani.net/linda/ErlangBandC.pdf>
8. Erlang and Poisson distribution, Integrable Differential, <http://adrian.idv.hk/doku.php/notepad/math/erlang>
9. Erlang B Calculator, <http://www.erlang.com/calculator/erlb/>
10. Extended Erlang B Calculator, <http://www.erlang.com/calculator/exeb/>
11. Erlang C Calculator, <http://www.math.vu.nl/~koole/ccmath/ErlangC/>
12. ERLANG C, <http://www.ccmath.com/index.php/nl/online-calculators.html>
13. Erlang C queue model calculator, <http://owenduffy.net/traffic/erlangc.htm>
14. CC Modeler Lite, Erlang C Calculator for Call Centers & Help Desks, <http://www.kooltoolz.com/ccm.htm>