

Förväntad väntetid hos en server

Maryam Varfan (maryam@kth.se)
Johanna Svenningsson (mea@nada.kth.se)
Gilbert Netzer (noname@aon.at)
Johan Husman (husman@kth.se)

7 april 2003

1 Inledning

Projektet syftade till att vi skulle få en ökad förståelse för hur en M/M/1 simulator fungerar, samt hur man använder en sådan för att undersöka beteendet hos en server och hur resultaten ska analyseras.

Detta genomfördes genom att vi modifierade en existerande simulator och använde den för att analysera logfiler för en webserver.

2 Experiment

Experimenten genomfördes genom att vi först modifierade den c-fil som ingick i projektbeskrivningen. Vi la till några få rader för att den skulle tolka loggen och även ge oss resultat för kölängd, maximal väntetid, genomsnittlig väntetid och antalet kunder som väntat mer än 0.5 sekunder.

Dessa tester genomfördes först på de första 5000 raderna i logfilen och därefter på de 5000 efterföljande raderna. Detta gjordes med två olika hastigheter, först för en webserver som tog 0.006 sekunder per kilobyte (167 kb/sekund) och sedan för en server som tog 0.003 sekunder per kilobyte (333 kb/sekund).

Efter det genomfördes ett serie av simulationer där serverns hastighet stegvis ökades från 0.001 sekunder per kilobyte upp till 0.4 sekunder per kilobyte. Dessa simuleringar använde sig av raderna 5000-10000 i logfilen. Valet av dessa rader berodde på att syftet inte var att undersöka serverns beteende vid extrema frågor, utan att få en uppskattning av förhållandet mellan serverns hastighet och väntetider och kölängd. Resultat visas i figur 1 och 2.

Resultaten finns med i appendix.

3 Analys

När testerna genomfördes på de första 5000 raderna med den långsamma servern blev väntetiderna förhållandevis långa (över en sekund genomsnittlig kötid). Detta berodde på att det tidigt kom en fråga om en mycket stor fil och denna fråga blockerade därefter servern en längre period. Under den här perioden byggdes kön upp och väntetiderna blev mycket långa (maximal väntetid blev 175 minuter).

Genom att öka hastigheten hos servern blev kön visserligen kortare, men problemet fanns fortfarande kvar.

När samma tester kördes på efterföljande 5000 rader från logfilen var väntetiderna avsevärt mindre (genomsnittlig kötid låg för den långsamma servern på 0.053 sekunder och maximal väntetid låg på ca. 10 minuter). Detta berodde på att kön inte byggdes upp på samma sätt, då det inte fanns några liknande, stora frågor.

Avslutande simuleringar med stegvis ökande hastighet visar att systemet är väldigt känsligt för serverns hastighet relativt hastigheten av ankommande frågor. Kön byggs snabbt upp och väntetiderna växer snabbare än exponentiell. Resultatet beror troligtvis på att en stor fråga kan blockera servern för en relativt lång period under vilken kön byggs upp. När servern besvarat frågan och tar nästa fråga ur kön har nästa stora fråga redan hunnit läggas in i kön. Detta leder till att den existerar en kö under långa perioder. Det är också tydligt att servern fungerar bra vid en låg last, men har avsevärt mycket sämre prestanda under medel till hög belastning.

En möjlig förbättring är att använda sig av en server som kan besvara flera frågor samtidigt, så att en stor fråga inte kan blockera servern helt under den tidsperiod det tar att besvara frågan. En annan möjlighet är att byta kö-policy, så att mindre frågor får högre prioritet än stora, vilket skulle reducera kölängd och väntetider.

Från teorin är det känt att shortes-job-next policyn ger kortast möjliga väntetider, men den kan leda till att stora frågor aldrig blir besvarade vid hög last. Detta beror på att det hela tiden hinner komma in nya, snabbare besvarade frågor innan servern hinner börja besvara de stora frågorna. Emellertid är detta inte nödvändigtvis ett problem om systemet är överbelastat, eftersom en shortest-job-next policy skulle leda till att fler frågor blev besvarade inom ett rimligt tidsintervall. Ett exempel är om det bara kommer in en enda stor fråga, men många små. Shortest-job-next skulle i det fallet kunna leda till att bara en fråga inte blir besvarad inom ett rimligt tidsspann. First-in-first-served skulle däremot kunna leda till att ett mycket stort antal frågor blir kraftigt fördröjda.

4 Sammanfattning

De körda simulationerna visade att de simulerade webservern fungerade bra vid låg belastning, men att stora problem kunde uppstå med den aktuella belastningen. Vidare hanterade servern frågor om stora filer mycket dåligt, vilket ledde till att servern blev helt överlastad i några av simulationerna.

Ett sätt att hantera problemet då det kommer ett fåtal, stora frågor är att köra flera webbservrar som turas om att svara på frågorna, alternativt låta servern behandla de inkommande frågorna parallellt. Det går också att ändra köpolicy till exempelvis shortest-job-next. Med det indata som gavs i början av logfilen skulle samtliga av dessa förslag ge avsevärt mycket bättre resultat än att köpa en dubbelt så snabb server.

5 Vem gjorde vad?

Modifiering av koden samt körningar för de 5000 första frågorna gjordes av Gilbert och Johan. Maryam och Johanna skrev rapporten och körde simulationen av de efterföljande 5000 raderna. Gilbert genomförde simulationerna med stegvis ökande serverhastighet.

6 Möjliga förbättringar

Anvisningarna för projektet var mycket otydliga. Det var svårt att avgöra vad syftet med projektet var och att hitta nödvändiga filer, såsom logfilen, och den givna koden var bitvis svårläst.

Projektet var också för litet för att göras i grupper om fyra och det var svårt att se några konstruktiva utökningar av projektuppgiften i dess nuvarande form. Att skriva hela simulatören från början hade antagligen varit mer givande och hade passat bättre för fyra personer.

Appendix — resultat

Långsam server — 5000 första

Average delay in queue 1.065 seconds
Average number in queue 0.273
Server Utilization 0.054
Time simulation ended 19513.506
Maximum queue length 29
Maximum time in queue 175.941
Customers waiting longer than 0.5: 165

Snabb server — 5000 första

Average delay in queue 0.332 seconds
Average number in queue 0.085
Server Utilization 0.028
Time simulation ended 19513.506
Maximum queue length 26
Maximum time in queue 87.749
Customers waiting longer than 0.5: 72

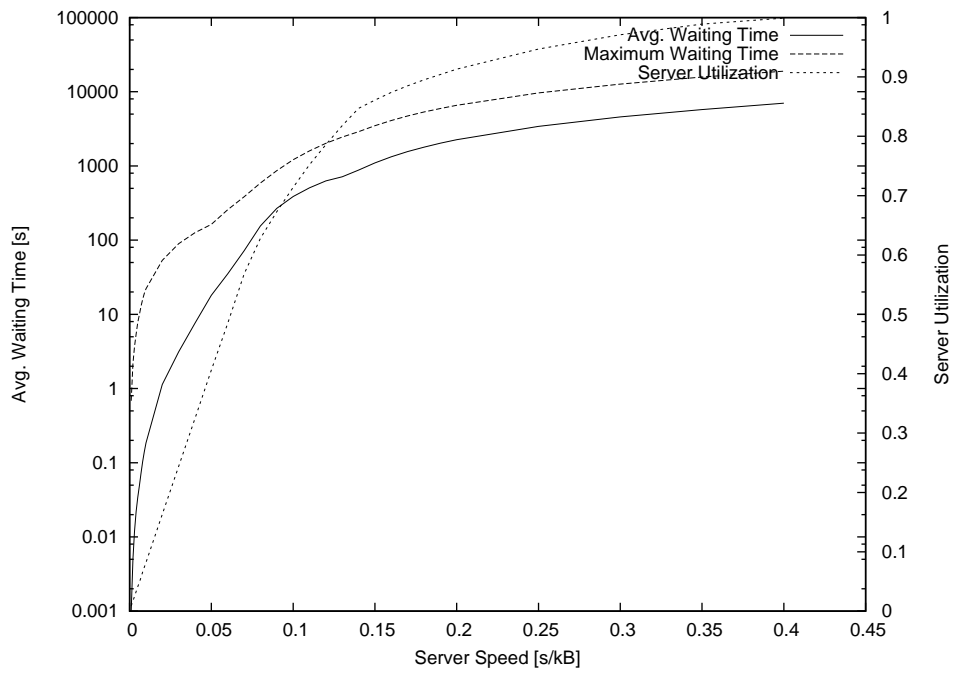
Långsam server — efterföljande 5000

Average delay in queue 0.053 seconds
Average number in queue 0.032
Server Utilization 0.055
Time simulation ended 8413.066
Maximum queue length 17
Maximum time in queue 10.068
Customers waiting longer than 0.5: 106

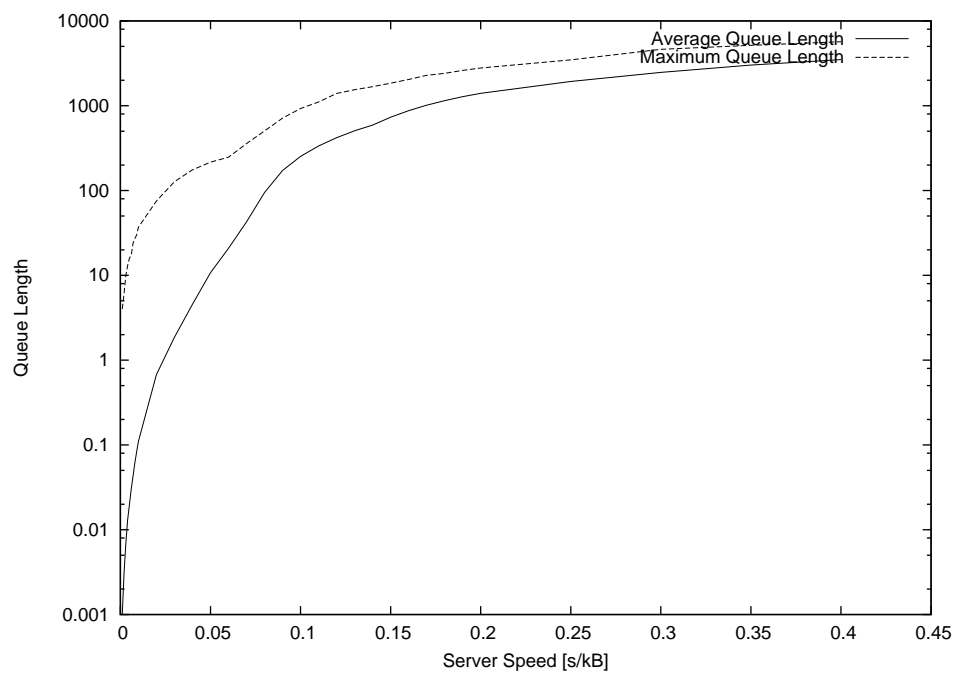
Snabb server - efterföljande 5000

Average delay in queue 0.013 seconds
Average number in queue 0.008
Server Utilization 0.030
Time simulation ended 8412.798
Maximum queue length 10
Maximum time in queue 3.719
Customers waiting longer than 0.5: 33

Noggrann Granskning — efterföljande 5000



Figur 1: Väntetider och belastning av en server relativt serverns hastighet (rad 5000-10000 ur logfilen)



Figur 2: Kölangder för en server relativt serverns hastighet (rad 5000-10000 ur logfilen)