

Beskrivelse og resultater af test af SOSI-GW hos Trifork

Indledning

Som en del af leverancen af SOSI-GW gennemføres en kvalitetssikring. I praksis består denne af en række tests udført af Trifork. Dette dokument beskriver hvordan testene er udført og hvad der kan aflæses af måleresultater fra dem.

Omgivelser og værktøj

Der benyttes følgende produkter til testene:

- SOSI-GW testklient, der kalder SOSI-GW services og kalder en test-service gennem SOSI-GW proxy-metoden.
 - Testklienten kan konfigureres gennem Jmeters GUI, så den kan benyttes til flere forskellige slags test.
 - Navne på SOSI-GW hosts, der skal være del af samme SOSI-GW cluster.
 - Størrelsen på svaret fra testservicen.
 - Ventetiden i testservicen, til at simulere lang svartid fra servicen.
 - Mængden af ID-kort den holder fast i, i sin cache.
 - Hvor ofte den skal gennemføre logout/login på eksisterende ID-kort i cachen
 - Hvor ofte den skal erstatte et ID-kort med et nyt, uden at kalde logout på det gamle.
 - Testklienten vælger ved hvert kald til SOSI-GW en tilfældig sosigw-server. Hvis kaldet fejler med socket-niveau fejl, prøver den de andre efter tur. Hvis det ikke lykkes på nogen af dem, fejler kaldet. Dette simulerer brug af en loadbalancer, der ikke vedligeholder nogen tilstand for klienterne og som sørger for at benytte en anden server når den valgte server ikke svarer. Bemærk at dette valg stresser intra-cluster kommunikationen rigtig meget. Den er ikke altid hurtig nok til at et login mod en server er propageret til samtlige andre servere, inden klienten forsøger at benytte proxy mod en anden server. Klienten vil dermed få implicit login i denne situation, hvilket den opfatter som en fejl. For at modvirke dette kan SOSI-GW konfigureres til at vente et stykke tid, før den svarer med implicit-login, mens den venter på at idkortet ankommer. I testresultaterne kan det ses dels som fejl-procent, når det ikke når frem, og dels som øget svartid. I virkeligheden vil denne situation kun opstå, hvis man benytter klienter, der fuldautomatisk underskriver id-kortet på brugerens vegne på ganske få millisekunder. (Det ville have været let at løse i testklienten, hvis ikke implicit-login havde eksisteret: man kunne bare prøve igen, eller prøve den næste server. Men implicit login nulstiller ens ID-kort, så det skal underskrives igen.)
- Jakarta JMeter v2.3.1 som test-driver og analyseværktøj for nogle af

resultaterne

- Java Sampler med førnævnte testklient.
- HTTP Request kan benyttes til at stresse dele af browser-baseret login, men ikke hele forløbet, da den ikke kan udføre den applet, der udgør den reelle login. Den kan dog benyttes til at belaste serveren svarende til belastningen af klienter, der ikke tidligere har downloadet applet'en fra serveren.
- Trifork P4 v4.0.1 som måleværktøj til at analysere fordelingen af svartiden internt i SOSI-GW. Som kravene til SOSI-GW er beskrevet, er det nødvendigt at skelne mellem tid brugt i SOSI-GW og tid brugt i test-servicen. (Som resultaterne er faldet ud, har denne del ikke været nødvendig og er undladt.)
- Trifork T4 v4.1.29 som omgivelse for SOSI-GW. Denne foretrækkes frem for Tomcat på grund af den indbyggede monitorering af memory og tråde.
- Windows XP på forskelligt bestykkede maskiner:
 - sosigw01: AMD Sempron 2800+ @1.60GHz 1.0GB RAM.
 - Trifork T4 med sosigw-testservice.war.
 - PostgreSQL med den globale audit-log database.
 - sosigw02, sosigw03, sosigw04: AMD Sempron 2400+ @1.66GHz 1.5GB RAM
 - SOSI-GW cluster medlemmer
 - Alternativt: Trifork T4 med sosigw-testservice.war.
 - sosigw05, sosigw06: AMD Sempron 2400+ @1.66GHz 1.5GB RAM
 - JMeter testklienter.
 - Alternativt: SOSI-GW cluster medlemmer
- Mellem testmaskinerne er der et 100 Mbit switched netværk.

Alle tests foretages med et cluster-setup, da det er den anbefalede og forventede driftform. Der er også en større del af systemet involveret i testen på denne måde.

Relation til krav

[Krav 26] til SOSI-GW udtaler sig om svartider for proxy-metoden, og da det er den brugerne vil bemærke svartiden af, hvis den overstiger kravene, er det denne, der testes og måles på. Der udføres også kald til de øvrige services på SOSI-GW, men tiderne for disse måles ikke. De indgår dog som belastning på maskinen og påvirker dermed throughput for proxy-metoden.

[Krav 25] og [krav 27] har svartidskrav på 5 hhv. 10 sekunder. Disse er så høje, samtidig med at testscenarierne er mere komplekse, at det er mere fordelagtigt at foretage tests af dem med håndkraft.

Skaleringstest

Skaleringstesten foretages ved at måle responstiden i klienten ved forskellige grader af belastning, og verificere at denne ikke stiger mere end proportionalt

med belastningen. Det totale throughput ved de forskellige belastningsgrader måles.

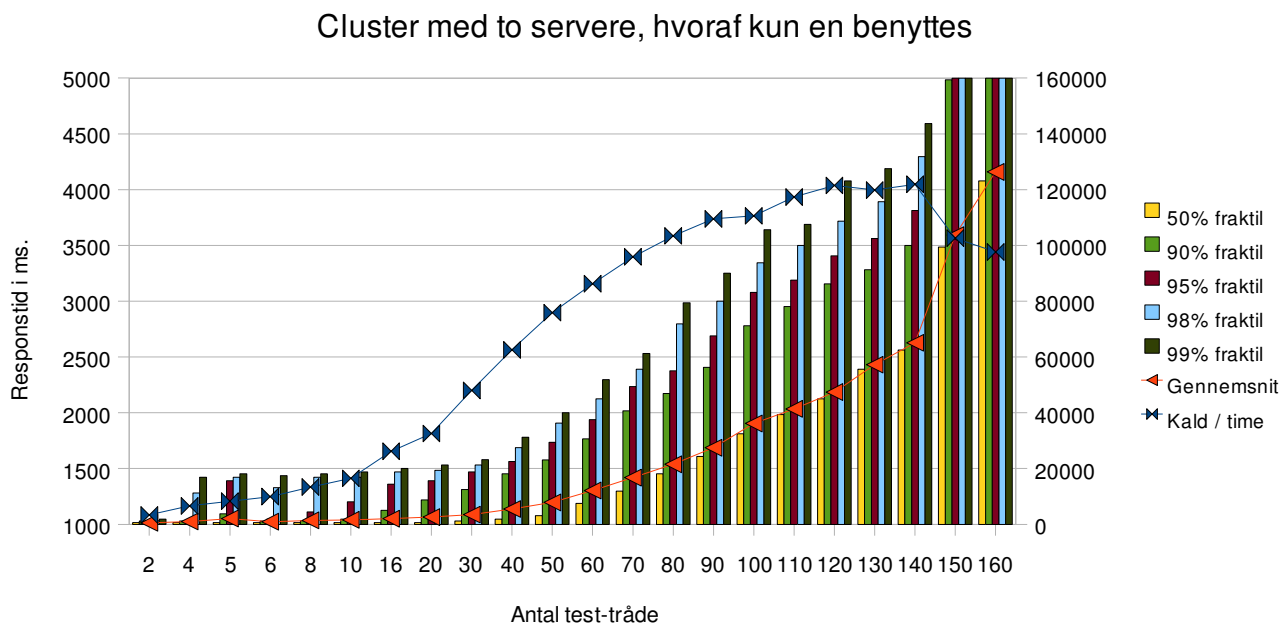
Testen er afviklet med følgende parametre:

- Clusteret består af de to medlemmer, men der kaldes kun mod et medlem: sosigw02.
- Der benyttes to maskiner til at kalde med: sosigw05, sosigw06.
- Varierende antal tråde
- Der ventes 1 sekunder imellem hvert kald til proxy.
- Test-servicen venter 1 sekunder før den svarer med et svar, som er en streng på 4096 tegn, (pakket ind i SOAP-transport, så netværkstransportmængden er noget større).
- Hver tråd har 1 ID-kort, og gennemfører logout+login ved 5% af kaldene til proxy.

Under testen registreres responstiden, som den ses af testklienten. Den måles altså inklusiv al netværkstransport og det ene sekunds forsinkelse, som test-servicen lægger på.

Resultater

Antal tråde	Kald / time	Gennemsn it	50% fraktil	90% fraktil	95% fraktil	98% fraktil	99% fraktil
2	3377	1014	1015	1031	1031	1032	1047
4	6602	1025	1016	1031	1032	1281	1422
5	8294	1052	1016	1094	1390	1422	1453
6	9922	1023	1015	1031	1046	1328	1437
8	13363	1035	1016	1032	1110	1421	1453
10	16481	1040	1016	1047	1203	1422	1469
16	26179	1051	1016	1125	1359	1469	1500
20	32573	1065	1016	1219	1391	1485	1531
30	48017	1087	1031	1313	1469	1532	1579
40	62546	1137	1047	1453	1563	1687	1781
50	75881	1196	1078	1578	1734	1906	2000
60	86242	1303	1187	1766	1938	2125	2297
70	95933	1419	1297	2016	2234	2390	2531
80	103421	1538	1453	2172	2375	2797	2985
90	109505	1687	1609	2407	2688	3000	3250
100	110599	1904	1813	2781	3078	3344	3641
110	117367	2035	1984	2953	3188	3500	3688
120	121522	2185	2125	3156	3406	3718	4079
130	119844	2431	2391	3282	3562	3891	4187
140	121918	2627	2563	3500	3813	4297	4593
150	102600	3592	3485	4984	5406	6234	7265
160	97654	4159	4078	5391	5891	7703	8719



Målingerne for de små belastninger skæmmes af at JMeter benytter en timer, der på Windows kun har en opløsning på ca. 15ms. Bemærk også at målingerne indeholder ventetiden i test-servicen på 1000 ms, samt test-servicens øvrige tid, samt al netværkstitid. Det betyder at der skal trækkes 1000 ms fra alle målinger for at finde frem til tiden brugt i SOSI-GW, inkl. netværkstitid.

Relativt til [Krav 26] er det meget fine tal. Kravet omtaler gennemsnit- og fraktil-tider for proxy-kaldet ved gennemførelse af 10.000 proxy-kald pr. time, fraregnet al netværkstitid.

- Gennemsnit < 500ms. Dette er til fulde opfyldt for den krævede belastning på 10.000 kald pr. time, med en værdi på 35ms for 13.363 kald pr. time.
- 95 % fraktilen < 1 sekund. Ved 13.363 kald pr. time nåede den op på 110 ms.
- 99 % fraktilen < 4 sekunder. Ved 13.363 kald pr. time nåede den op på 453 ms.

Samlet set er der rigeligt skalerbarhed til at nå målene i krav-specifikationen. Svartidskravene er opfyldt op til 86.242 kald pr. time, hvor 95% fraktilen nærmer sig 1 sekund. Omkring 120.000 kald pr. time er det maksimale, der kan presses igennem denne testserver, men over 90.000 kald pr. time vokser svartiderne ud over det acceptable.

Loadbalancing

Loadbalancing aftesttes i et miljø med 2 gateway servere. Dette foretages ved at gentage skaleringstesten med 2 aktive servere i clusteret. Det verificeres at overbelastningspunktet øges i forhold til et miljø bestående af kun én server. Det må forventes, at gevinsten ved at tilføje en server vil være mindre end liniær.

Testen er afviklet med samme parametre som skaleringstesten, undtagen:

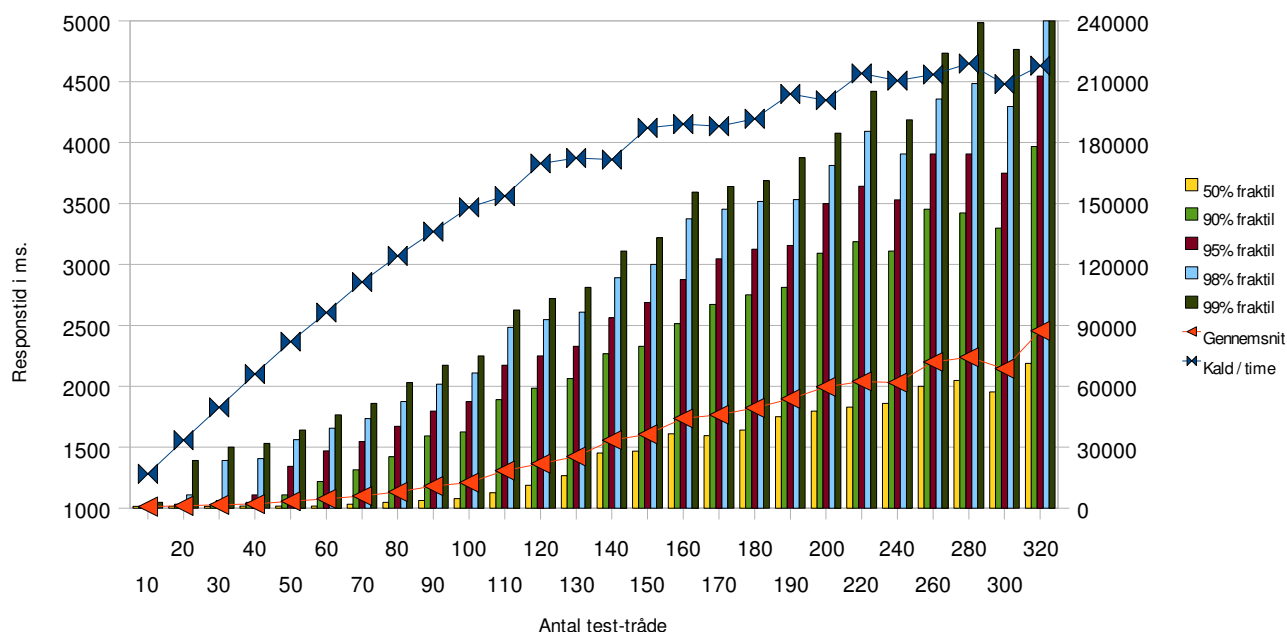
- Clusteret består af de to medlemmer, der begge benyttes til kald.

Vi ønsker her at afprøve om det er muligt at gennemføre flere proxy-kald med lavere svartider ved at benytte flere servere i clusteret. Bemærk at der er de samme medlemmer af clusteret som før, og at det derfor udelukkende er kaldsmønstret, der er ændret. Clustermedlemmerne har samme overhead ved at holde hinanden orienteret om den fælles tilstand som i skaleringstesten, omend trafikken nu går begge veje.

Resultater

Antal tråde	Kald / time	Gennemsn it	50% fraktil	90% fraktil	95% fraktil	98% fraktil	99% fraktil	Fejl rate
10	16884	1012	1015	1016	1031	1032	1047	0.00%
20	33444	1019	1015	1031	1032	1109	1390	0.00%
30	49572	1027	1015	1032	1063	1390	1500	0.00%
40	66067	1033	1016	1047	1109	1406	1531	0.04%
50	81994	1057	1016	1109	1343	1562	1641	0.00%
60	96300	1076	1016	1218	1469	1656	1765	0.01%
70	111398	1099	1031	1313	1547	1735	1859	0.00%
80	124222	1132	1046	1422	1672	1875	2031	0.01%
90	136188	1181	1062	1593	1797	2016	2172	0.00%
100	148176	1210	1078	1625	1875	2109	2250	0.02%
110	153626	1308	1125	1890	2172	2484	2625	0.10%
120	169733	1364	1188	1984	2250	2547	2719	0.00%
130	172411	1424	1266	2063	2328	2609	2812	0.00%
140	171634	1558	1453	2266	2563	2890	3109	0.15%
150	187315	1605	1468	2329	2687	3000	3219	0.06%
160	189101	1736	1609	2515	2875	3375	3594	0.08%
170	188042	1766	1594	2672	3047	3453	3640	0.31%
180	191729	1823	1641	2750	3125	3516	3688	0.41%
190	203998	1896	1750	2812	3156	3532	3875	0.10%
200	200873	1995	1797	3093	3500	3813	4078	0.29%
220	214056	2040	1828	3187	3641	4094	4422	0.65%
240	210528	2031	1859	3109	3531	3906	4187	1.17%
260	213538	2200	2000	3453	3906	4359	4734	1.29%
280	218916	2239	2047	3422	3906	4485	4985	1.43%
300	208807	2144	1953	3297	3750	4297	4766	2.53%
320	217930	2455	2188	3969	4547	5078	5437	2.19%

Cluster med to servere, der begge benyttes



Bemærk at grafen her og i skaleringstest-afsnittet har forskellige skalaer på de begge akser og dermed skal sammenlignes med varsomhed. Ved omkring 150.000 kald pr. time rammer 95% fraktilen grænsen på 1 sekund. Det er en klar forbedring over de ca. 90.000 fra enkelt-server testen. Det er muligt at gennemføre omkring 210.000 kald pr. time i denne konfiguration, med acceptable svartider, der dog ikke lever op til kravet til 95% fraktilen.

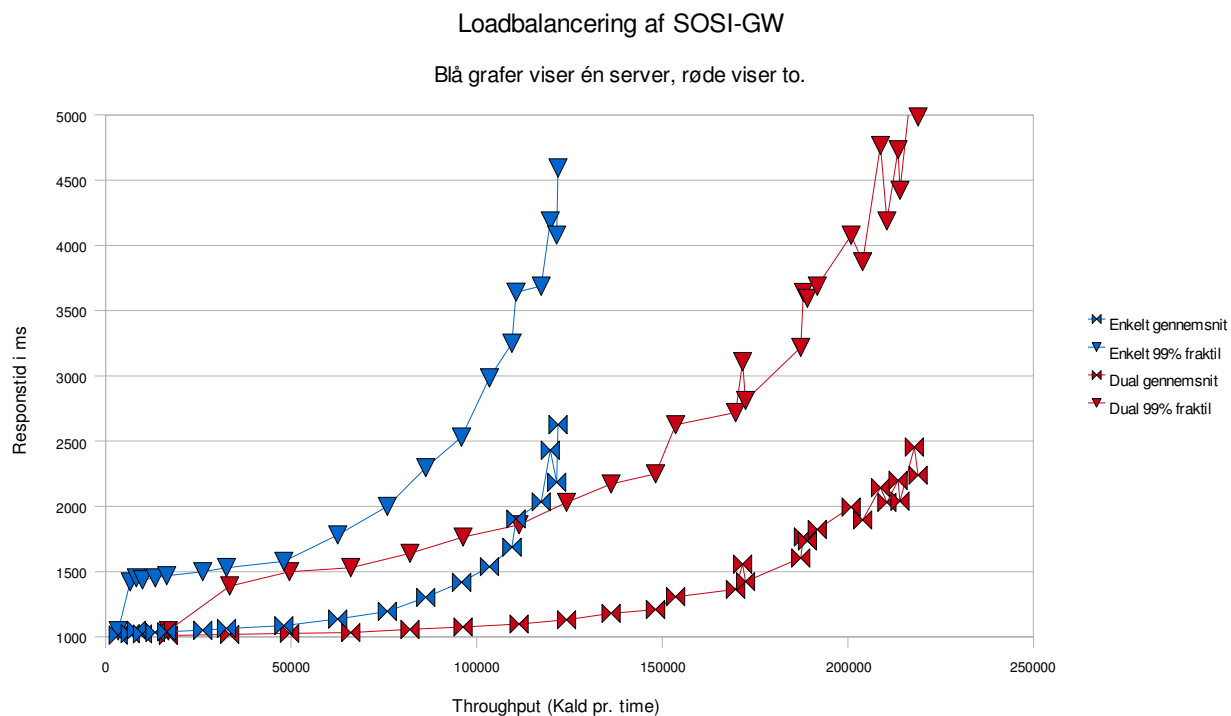
Ved belastning over 150.000 kald pr. time begynder der at opstå fejl, som det ses af søjlen Fejl rate . Denne angiver hvor mange kald, der fejlede ud af det totale antal, der er forsøgt gennemført. Fejlene består i, at der rammes en anden server efter et login, og at denne endnu ikke kender id-kortet, hvorfor den svarer implicit login.

Sammenligning med skaleringstesten

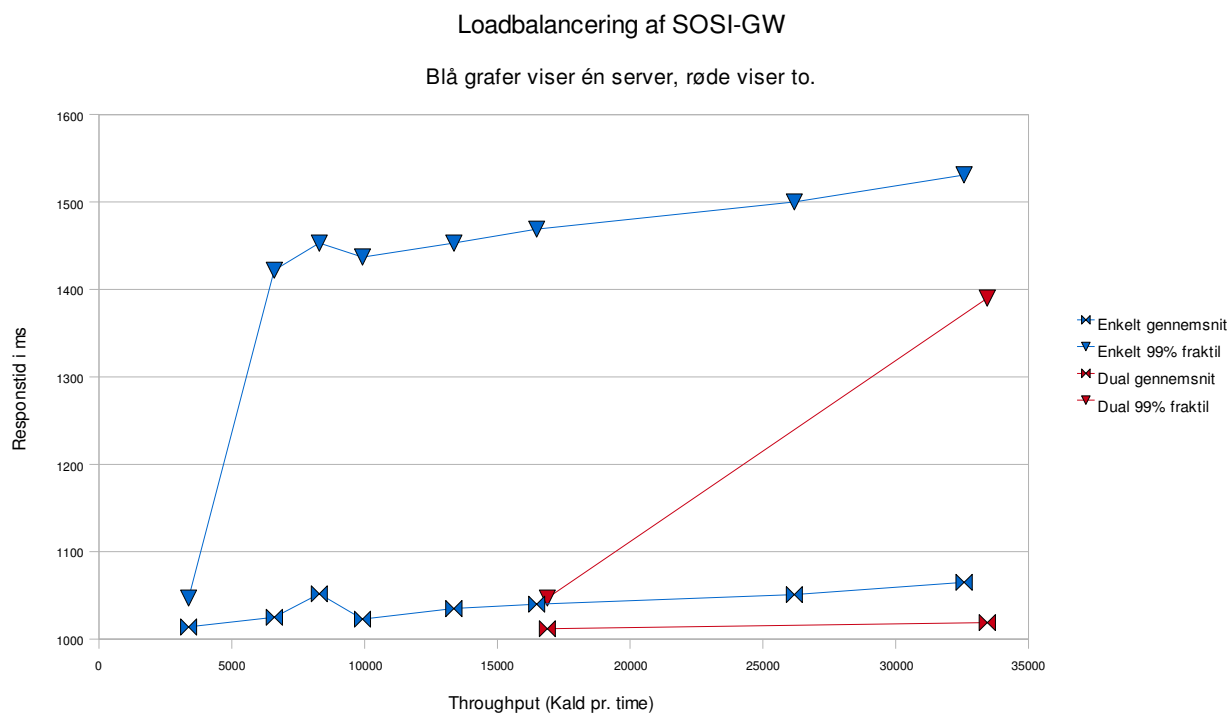
Vi sammenligner her resultaterne af skaleringstesten for 60 tråde med loadbalanceringstesten for 60 tråde. De 60 tråde er den højest målte belastning, hvor svartiderne overholdes i skaleringstesten.

Antal aktive servere	Kald / time	Gennemsnit	50% fraktil	90% fraktil	95% fraktil	98% fraktil	99% fraktil
1	86242	1303	1187	1766	1938	2125	2297
2	96300	1076	1016	1218	1469	1656	1765

Det samme antal tråde når at gennemføre flere kald på grund af den gennemgående lavere svartid. Gennemsnitssvartiden falder fra 303 ms til 76 ms, mens 95% fraktilen halveres fra 938 ms til 469 ms. Der altså en markant forbedring i svartid for brugerne ved at benytte to servere frem for en.



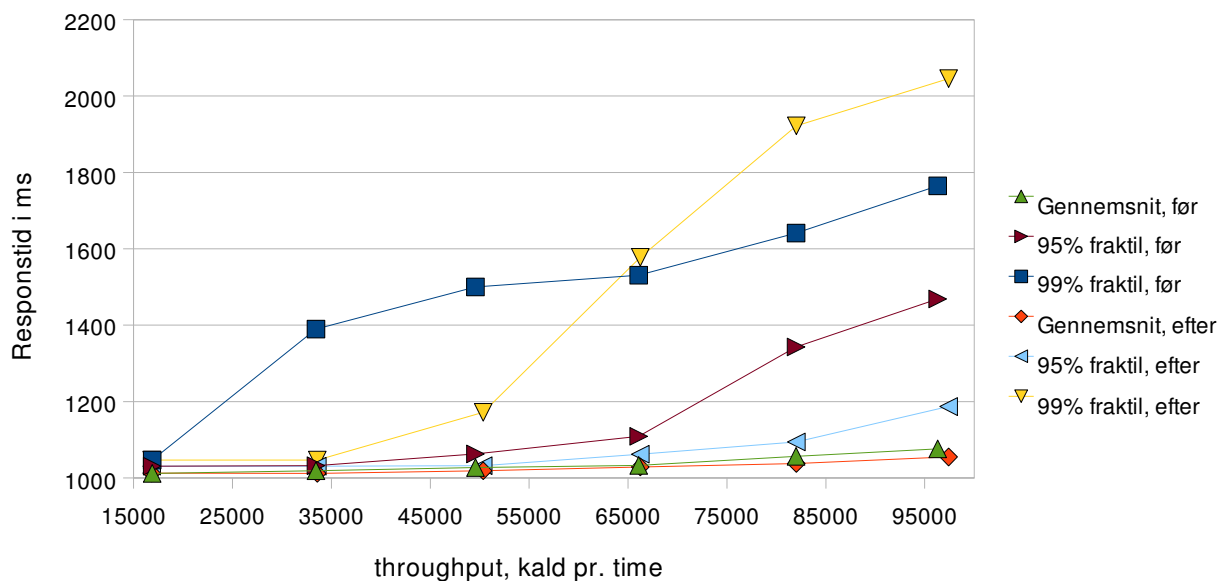
Denne graf viser hvordan svartiderne udvikler sig i skaleringstesten og loadbalanceringstesten. Det ses at det er muligt at have væsentligt større belastning med rimelige svartider ved to servere i forhold til en. I den følgende graf ses det at der også ved lavere belastninger er en gevinst ved at benytte to servere frem for en, idet det er muligt at holde 99% fraktilen meget lavere.



Load/stresstest

I forlængelse af skaleringstesten fastslås den belastningsgrad hvor throughput ikke længere stiger som funktion af antal afsendte requests. Det verificeres, at overbelastning ikke fører til fejlbehandlede requests samt at systemet efterfølgende kommer sig uden nogen blivende degradering. I praksis udføres denne test ved at udføre loadbalanceringstesten med noget højere belastning end den viste sig at kunne bære, og derefter udføre samme tests lave belastninger. Derefter sammenlignes resultatet med resultatet fra loadbalanceringstesten og det verificeres at der ikke forekommer fejl, samt at svartiden falder i takt med belastningen, til samme niveau som før overbelastningen.

Svartider før og efter stress test



Grafen viser resultatet at gennemføre den samme test før og efter en kørsel med markant overbelastning. Der ses stort set uændrede gennemsnitlige svartider og throughput, mens 99% fraktilerne varierer lidt mere kaotisk, uden at der er nogen markant forskel. 95% fraktilen er lidt lavere efter stress-perioden end før, men ikke markant. Konklusionen er at der ikke er nogen blivende eftervirkning af en stress-periode at spore.

Endurancetest

Endurancetest foretages ved en længerevarende test, hvor testklienten konfigureres til en belastningsgrad, der ligger et stykke under det overbelastningspunkt, der er identificeret under load/stresstesten. Under endurancetest overvåges memoryforbrug, antal tråde, og CPU forbrug.

Testen er afviklet med følgende parametre, der er valgt til at ramme den beskrevne belastning i [krav 26], som er 10.000 proxy-kald i timen (166 i minuttet), samtidig med 1000 login-kald i timen. (17 i minuttet)

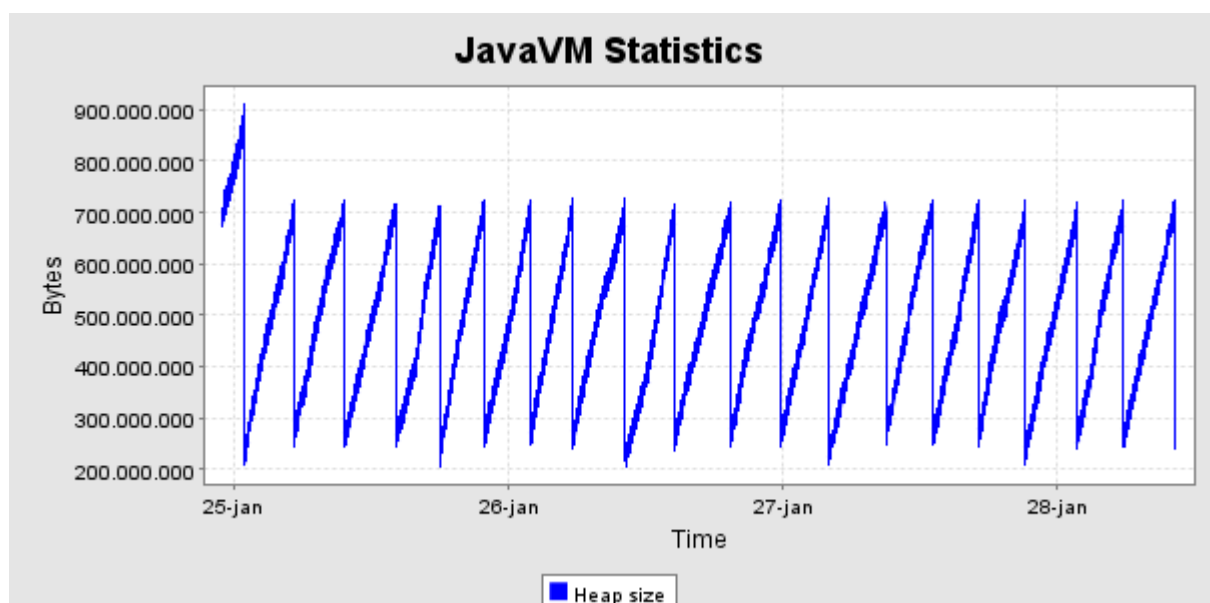
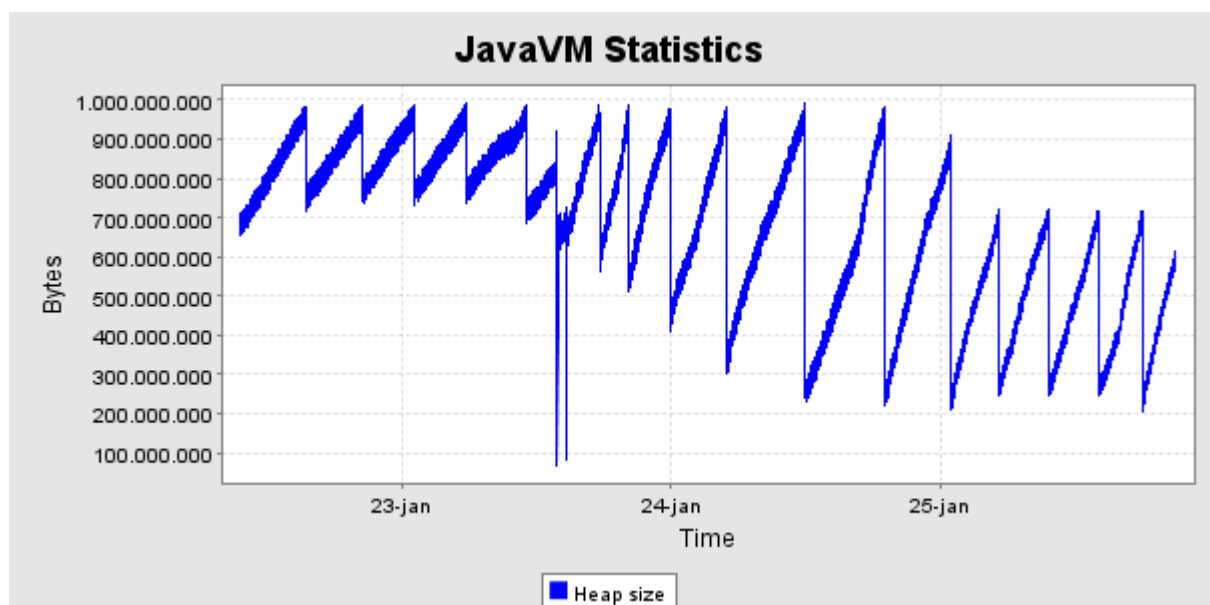
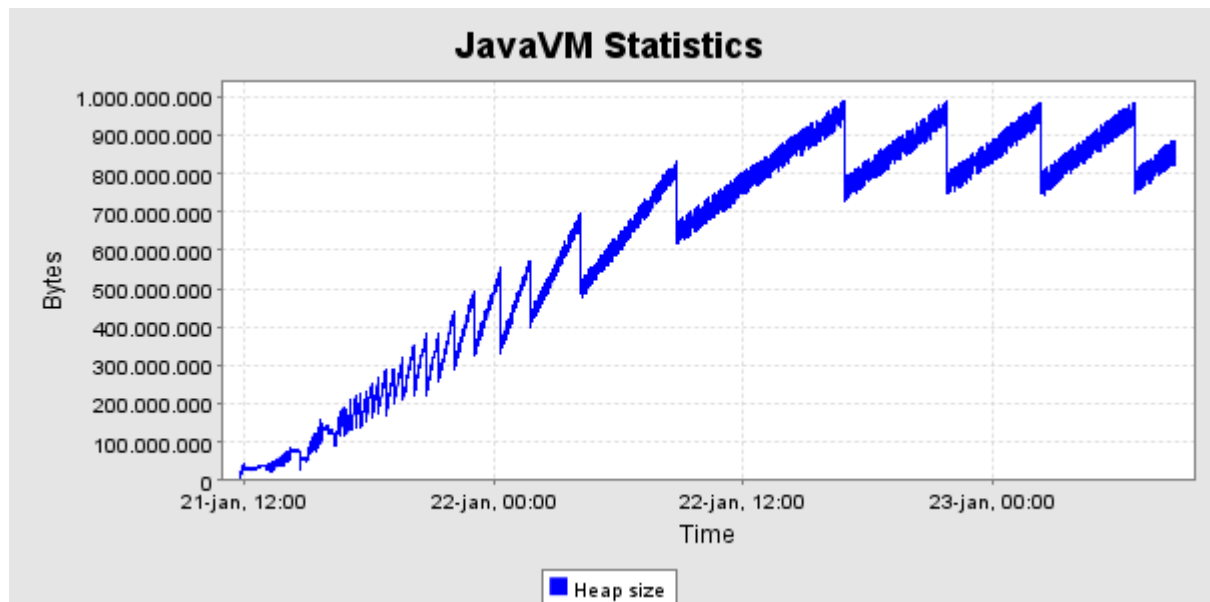
- 100 tråde
- Der ventes 8 sekunder mellem kaldene.
- Test-servicen venter 10 sekunder før den svarer med et svar, som er en

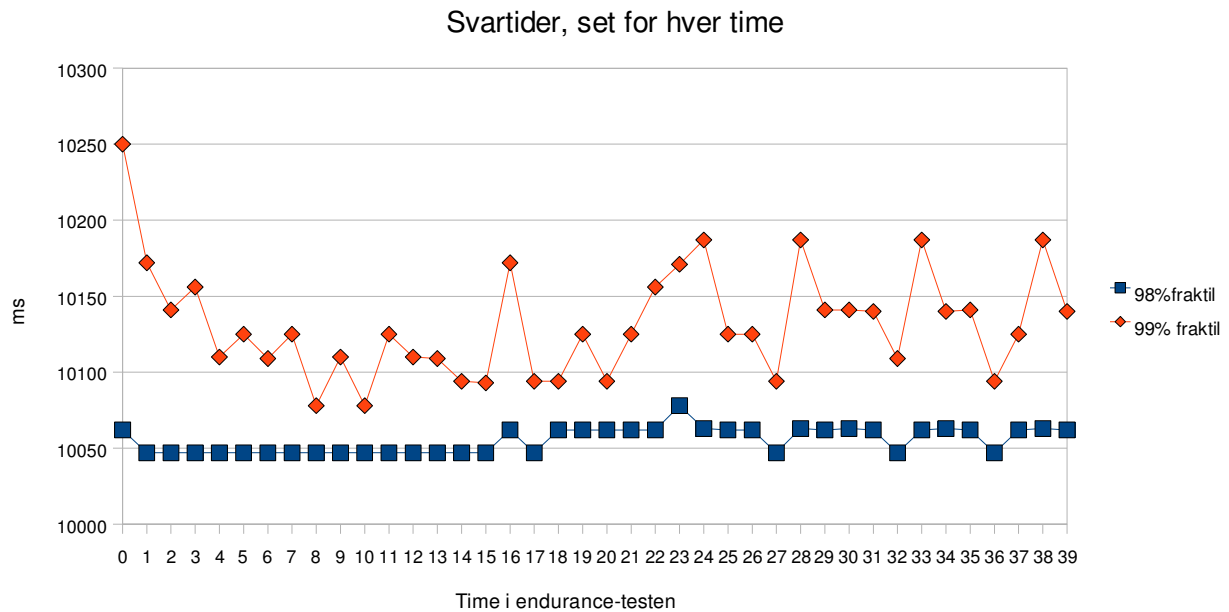
streng på 4096 tegn, pakket ind i SOAP-transport. Der sendes også en 4096 tegns streng med hvert kald.

- Hver tråd ventes af tage lidt over 10 sekunder i alt på kaldene. Med i alt 18 sekunder pr kald kan en tråd nå $3600/18=200$ kald i timen. Med 100 tråde ventes dermed op mod $100*200=20.000$ kald i timen. I praksis ventes mellem 19.000 og 20.000
- Hver tråd har et ID-kort. Hver tråd laver login *uden at kalde logout på det gamle kort* ved 5% af kaldene til proxy. Dette har den effekt at der i løbet af 24 timers kørsel skabes omkring $5\% * 20.000 * 24 = 24.000$ id-kort i cachen på SOSI-GW. Derefter vil de ældste af dem blive fjernet af cachen selv, efterhånden som de overskrider deres maksimale levetid på 24 timer.
- Testen køres i minimum 24 timer. De første 24-25 timer forventes resourceforbruget at vokse støt på grund af den tiltagende mængde id-kort i cachen, hvorefter det bør stabilisere sig. Der bør ikke være nogen nævneværdig ændring i svartiderne hen over perioden, omend der ventes en lille effekt af at have så mange ID-kort.
- Der forventes et memoryforbrug på omkring 32kb pr. id-kort, altså i alt omkring 768 MB heap til id-kort. Tallet er anslået ved at måle gennemsnitsstørrelsen af et ID-kort med Trifork P4. Det forholdsvis store forbrug skyldes at en id-kort instans, som den er i SEAL, indeholder en DOMrepræsentation, der fylder temmelig meget. Oven i det kommer at den også gemmes serialiseret som XML-streng, hvad der fylder omkring 6kb, af hensyn til at kunne sende den til andre medlemmer i clusteret. Man kunne nøjes med at have den serialiserede udgave af hensyn til plads, men det er ikke en tids-effektiv måde at opbevare den på, da det så skulle parses ved hvert proxy-kald.
- For at validere at forældelse af gamle kort fungerer, køres i fortsættelse en test, hvor der foretages meget færre logins og ellers benyttes samme parametre. I praksis er valgt at lave login i 1% af kaldene, og samtidig foretage logout på det gamle nameID ved gen-login, så der ikke ophobes ret mange id-kort. Efter et døgn bør der være 100 id-kort i cachen, samt oplysninger om $1\% * 20.000 * 24 = 4.800$ brugte nameID'er med status REVOKED.
- For at teste cluster-delen mest muligt, benyttes 4 servere i clusteret i denne test. Der er kun behov for én maskine til test-servicen og én til at drive testen med denne lave belastning.

Resultater

Der er kørt med den ovennævnte høje belastning i 40 timer, og derefter en lille uges tid med den lavere belastning. Her er grafer og memoryforbrug og svartider i denne periode. Svartiderne er for perioden med den høje belastning.





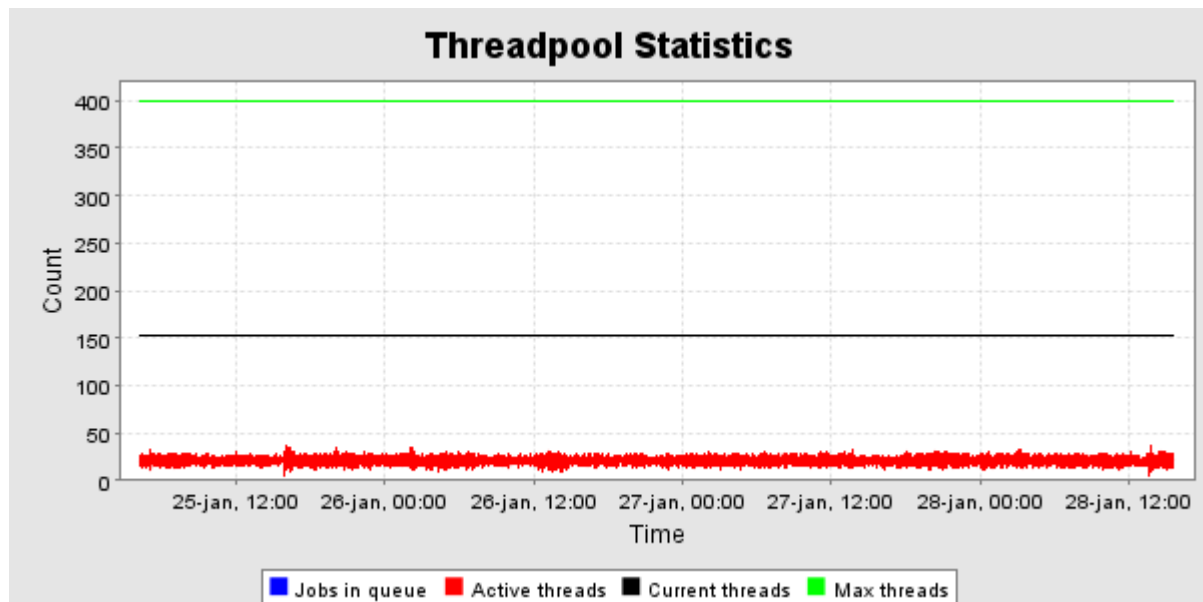
Der ses det forventede forløb i heap-udviklingen, nemlig en kraftig stigning de første 24 timer og derefter en kurve hvis bund er vandret, med den nøjagtighed det kan afgøres her. Der er gennemført 797513 kald i alt i løbet af de 40 timer, svarende til 19938 kald pr. time i gennemsnit. Heraf gav de 3 fejl tilbage til klienten, og i alle tre tilfælde er der tale om at testklienten ikke kunne gennemføre socketforbindelsen til SOSI-GW-servicen.

For at teste synkroniseringsmekanismen startes en ekstra server efter ca. 41 timers test. Denne server modtager tilstanden fra en af de eksisterende. Den modtog 122MB data i løbet af 37 sekunder, og det bestod af 22561 id-kort. Afsendelsen af disse data belastede den afsendende server med ca. 20% cpu, så den i alt var omkring 25% belastet.

Efterfølgende har clusteret af nu 4 servere fået lov at køre i en lille uges tid for at se heap-grafen for en længere periode. Den har et uproblematisk forløb.

CPU-forbruget er under testen næsten ikke målbart. Windos taskmanager viser det typisk som 2-5%.

Trådpoolen i T4 er sat til max 400 http-tråde. Som det ses af følgende graf er der højst benyttet 150 i VM'ens levetid, og under endurance-testen er der omkring 20-30 i brug i gennemsnit. Dette svarer med forventningerne, da der er 100 testtråde, hvis kald skal fordeles over 4 servere.



Rullende opgradering

Der af testes rullende opgradering i et clustered miljø. Det vil bestå i redeploy af applikation og deraf følgende automatisk auto-genoptagelse i clusteret. Dette foretages samtidig med at testklienten vedligeholder en stabil belastning af clusteret. Til dette formål anvendes samme test-belastning som til endurance-testen, dog uden at den har kørt længe i forvejen, så mængden af ID-kort i cachen er ikke helt så stort. Der forventes en stigning i svartiderne imens dette foregår, da den nye server skal have overført den samlede tilstand fra de andre, før den kan svare korrekt. Der ventes en lille mængde fejl-svar ved denne handling, da det ikke helt kan undgås at der kommer fejl fra transport-laget, når servere kommer og går på denne måde. Der bør ikke forekomme fejl fra selve SOSI-GW, mens det er muligt at web-containeren f.eks. lukker forbindelsen midt i en besked. Det er vigtigt at lade undlade at opgradere mere end et medlem af clusteret ad gangen, da de ellers kan komme i en situation, hvor de nye kan blive enige om at de er i sync med verden i en periode, selv om de endnu ikke har modtaget alt fra de gamle .

Til denne test køres fortsat med 4 medlemmer i clusteret, for at få udført flest mulige kombinationer af tilstand mellem dem. På T4 fungerer et redeploy ved at den nye applikation deployes, hvorefter systemcontaineren genstartes hvorved applikationen genstartes og alle klaser loades fra den nye version. Den gamle version kører fortsat i de tråde, der var i gang, indtil de kommer i threadpoolen eller dør af sig selv. Der testes over flere omgange ved at deploye til to af de fire på skift. Der deployes kun på en ad gangen, som beskrevet i installationsvejledningen. Den maskine, hvis heap-forbrug overvåges og vises i grafen over dette, er ikke med blandt dem, der røres, da det ville ødelægge dens mulighed for at vise en eventuel leak.

Resultater

Der er deployet 4 gange, uden at det har givet anledning til andre fejl for testklienten end de forventede connection reset , der skyldes at den forsvinder midt i et request. Det ses i loggen, at den nye node tager omkring et minut i alt

om at komme i luften, heraf går omkring 20-50 sekunder med at modtage den delte tilstand og resten med at lytte til tilstrækkeligt mange pings fra de andre til at kende deres systemtids-offset fra den lokale tid.

Fail over

Der testes at de resterende servere i et cluster på korrekt vis kan forsætte håndteringen af indkommende requests hvis en server tages ud af drift uden varsel (fx ved at afbryde netværksforbindelsen til den pågældende server). Denne test adskiller sig ikke væsentligt fra Rullende opgradering, da forskellen alene består i de mere brutale måder forbindelserne mellem klient og server samt serverne imellem brydes. Da der ikke er en avanceret loadbalancer foran test-klienterne, forventes det at de får lidt længere svartid ud af selv at skulle prøve den manglende server. Der forventes også her en lille mængde fejl-svar idet den forsvinder fra nettet.

Resultater

To af maskinerne er konfigureret til at have fast IP-adresse, så det er muligt at trække deres netværksstik ud, uden at de mister deres logiske netværksinterface. Der er gennemført test ved at der flere gange er trukket netstikket ud af disse to maskiner, både hver for sig om samtidigt, i perioder mellem 1 og 30 sekunder ad gangen. Der er ikke observeret nogen fejl i testklienten i denne forbindelse, ud over de forventede connection reset og connection timeout, som alle håndteres af failover delen af test-klienten. Ved brug af en loadbalancer, der laver retry ved den slags tcp-niveau-fejl og ved HTTP 404 fejl, vil det være muligt at køre videre uden at opleve fejl i klienterne.

Konklusion

Resultatet af det samlede testforløb er tilfredsstillende. Der blev fundet og rettet fejl i begyndelsen af forløbet, hvorefter alle tests kunne gennemføres med fine svartider, både i forhold til kravene og i forhold til hvad man kunne forvente af det forholdsvis gamle og langsomme hardware, der er benyttet til testene.