

Performance rapport - Kommunernes Gateway

Author: Henrik Thuesen
Date: 16-Jan-2014 19:51
URL: <https://www.nspop.dk/display/NSPP/Performance+rapport+-+Kommunernes+Gateway>

Table of Contents

1		3
2	Kommunernes gateway	4
3	Antagelser og forbehold	5
4	Testen	6
4.1	Afvikling	6
4.2	Testplan	6
4.3	Fordeling	6
5	Målingerne	8
5.1	Throughput	8
5.2	Miljø	8
6	Konklusion	9
7	Forbedringer	10
8	Splunk søgninger	11

1

- Kommunernes gateway
- Antagelser og forbehold
- Testen
 - Afvikling
 - Testplan
 - Fordeling
- Målingerne
 - Throughput
 - Miljø
- Konklusion
- Forbedringer
- Splunk søgninger

2 Kommunernes gateway

Kommunernes Gateway er SOSI Gateway opsat med partitionering i et 4 søjle miljø, som kommunerne skal anvende på samme måde som regionerne anvender SOSI Gateway på dNSP. I den forbindelse er der foretaget målinger af Throughput (TP) på et staging miljø opsat af Netic. Målingerne er foretaget i løbet af november måned 2013 på SOSI-GW version 1.2.1.

SOSI Gateway udstiller et antal services til at få udstedt og cachet et id-kort samt en service til at kalde andre services gennem. Disse services bliver alle anvendt af performance testen.

3 Antagelser og forbehold

Performancetesten afspejler ikke produktionlignende forhold, da disse pt. ikke er kendt. Derfor laves der antagelser og der må ligeledes tages forbehold for disse i resultatet.

- **Søjler:** Modsat produktionsmiljøet som består af 4 søjler så består staging miljøet kun af 2 søjler. Resultaten fra testen kan der kun bruges som udgangspunkt for et TP til det endelige produktionsmiljø. Adskillige faktorer gør at en lineær beregning ikke er tilstrækkelig; bla. synkronisering i cachen påvirke tallet nedadgående. Det er dog sandsynligt at TP vil øges med en faktor der er tæt på proportionalt så længe der er tale om få søjler.
- **Fordelinger:** Fordelingen af kald mellem de forskellige services i SOSI Gateway er udvalgt på baggrund af en uges målinger foretaget i produktion. Fordelingen afhænger af hvordan de systemer der anvender dNSP'erne er udviklet, og en anderledes fordeling kan forekomme når nye systemer i kommunerne begynder at anvende SOSI Gateway.
- **Certifikat:** Det samme certifikat er brugt til alle forespørgelser mod SOSI Gateway. Dette vurderes ikke at gøre nogen forskel for gatewayen, da testen varierer `NameId` og dermed simulerer flere anvendere. Det giver en positiv påvirkning til STS'en, idet den cacher data baseret på CPR nummer og CVR nummer, og dermed kun skal slå disse op en gang. Dette vil dog også påvirke resultatet.
- **Partition:** Alle kald til SOSI Gateway har anvendt den samme partition. Dette vurderes ikke at påvirke testresultaten, da gatewayen udelukkende anvender partitionen-identifikationen som en del af nøglen til cachen.
- **Lokalt netværk:** Alle kald til SOSI Gateway sker via et netværk der er betydeligt hurtigere end hvad man kan forvente kommunernes systemer har adgang til. Dette vil samlet set betyde en længere svartid pr forespørgelse, men burde ikke påvirke TP synderligt.
- **SSL:** I produktion skal kommunernes systemer tilgå SOSI Gateway via HTTPS, men i performance testen er der brugt HTTP for at minimere omkostningerne ved opsætning og afvikling af testen.
- **Proxy endpoint:** I forbindelse med udviklingen af performance testen er der også udviklet en simpel service som kaldes gennem gatewayen. Denne service modtager requestet og sender det retur byte for byte. I et rigtig produktionssetup vil størrelsen af request og reponse være meget forskellige alt efter hvilke endpoints, der kaldes. Dette kan ligeledes påvirke TP.

4 Testen

Trifork, som er leverandør til SOSI-GW, har tidligere udført en tests, hvoraf der også fremkom et tal for TP. Denne tests har dog andre forudsætninger, og blev udført i 2008 på andet hardware. Testen kan findes [her](#).

Performance testen afvikles i flere tempi. Først køres med et antal tråde på et antal noder, og laves flere kørsler, hvor antallet af tråde og noder sættes op. Testen stopper, når den beregnede TP ikke længere bliver højere. Dette kan også ses på nogle af gaferne.

4.1 Afvikling

Performance testen består af en række JMeter testplaner, samt scripts, der afvikler den valgte performancetest inkrementelt indtil det endelige throughput er fundet. For hver iteration øges enten antallet af tråde eller antallet af noder indtil det målte throughput ikke længere vokser tilsvarende.

Følgende tests er blevet afviklet:

Id	Søjler	Fordeling	Gennemløb
20131118_144601	2	morning	20
20131120_144454	2	midday	20
20131120_170431	1	midday	100
20131122_130117	2	midday	100
20131126_134451	1	morning	100
20131126_152518	2	morning	100

Gennemløb beskrives under afsnittet, *Fordeling*.

4.2 Testplan

Testplaner anvendt i denne performance test: `gw`

Testplanen `gw` består af en del der lægger et id-kort i cachen og kalder proxy endpointet et antal gange samt en del der lægger et id-kort i cachen og tjekker status via metoden `GetValidIdCard`. De to blokke afvikles i tilfældig orden for hvert gennemløb.

4.3 Fordeling

Fordelinger anvendt i denne performance test: `morning`, `midday`.

Fordelingen `morning` foretager 6 proxy kald, et enkelt status kald samt 7 signeringer af id-kort for hvert gennemløb. Hvert gennemløb gentages et antal gange.

Fordelingen `midday` foretager 24 proxy kald, et enkelt status kald samt 7 signeringer af id-kort for hvert gennemløb. Hvert gennemløb gentages et antal gange.

Disse fordelinger er fremkommet vha. analyse af produktionslog gennem splunk. Der er forsøgt at finde fordelinger, der skulle svare til den "normale" brug af komponenten.

5 Målingerne

5.1 Throughput

De kørsler af performance testen har givet de TP der kan ses i tabellen nedenunder. Uder over TP vises også hvor mange tråde og noder der skulle til for at opnå dette TP.

Id	Throughput	Tråde	Noder
20131118_144601	69.13	14	2
20131120_144454	127.64	14	2
20131120_170431	113.32	7	2
20131122_130117	116.55	7	2
20131126_134451	61.92	14	2
20131126_152518	59.12	14	2

5.2 Miljø

- **CPU:** De to søjler i miljøet ser begge ud til at have betydelig CPU resources tilovers under alle 3 test. Målingerne i Splunk viser at de to søjler kun sjældent bruger mere end 50% af de CPU resourcer der er til rådighed.
- **Memory:** Under testene har begge søjler mellem 400 Mb og 1 Gb fri Java Heap space.

6 Konklusion

Efter kørslen af de to første tests med 20 gentagelser, valgte vi at køre testene igen med 100 gentagelser, da det var sandsynligt at rampup påvirkede resultatet uforbeholdent meget.

Med de målinger der er gennemført, kan der nu konkluderes, at et 2 søjlesystem med fordelingen fra spidsbelastningsperioden i NSP (morgen) kan håndtere ca 60 forespørgelser pr sekund. Det giver **216.000 forespørgelser i timen**. Komponenten og miljøet vil maksimalt kunne håndtere dette, dog vil komponenten være i stand til at modtage flere og vente med at behandle disse. Denne kø-funktionalitet er indbygget i JBoss. Funktionaliteten vil naturligvis give anledning til en længere svartid.

En måling fra torsdag den 14. nov 2013 i produktion viser at SOSI Gateway maksimalt modtager omkring 11.000 forespørgelser pr time. Den udvalgte dag har det samme antal forespørgelser som de andre hverdage i perioden 23/10-22/11, og giver derfor et repræsentativt billede.

Ifht. til en udrulning til 98 kommuner, betyder dette at hver kommune gennemsnitlig kun må have 1/5 så meget trafik som en region for at være under angivne TP. Her er stadig kun medregnet 2 søjler.

7 Forbedringer

- **JBoss:** Opsætningen af JBoss er aldrig blevet gennemgået med et performance mæssigt synspunkt. En analyse af konfigurationsmuligheder i JVM'en, som f.eks. Heap size, PermGenSpace size og Garbage Collection Algorithm samt JBoss, som f.eks. antal HTTP tråde, ville med stor sandsynlighed kunne give muligheden for et bedre TP.
- **Seal.Java:** SOSI Gateways services til håndtering af id-kort er opbygget således at input først parses op til JAX-WS objekter, derefter opbygges et Seal.Java id-kort objekt som så serialiseres til et W3 DOM træ. Det betyder at der foretages unødvendig parsing. Det er værd at se på en mulig optimering af denne algoritme.
- **Cachen:** SOSI Gateway har en cache der synkroniserer på tværs af søjlerne. Denne kode er central for svartiden og i sidste ende Throughput. Implementationen heraf er udviklet sammen med komponenten, og har ikke i større udstrækning været analyseret i forbindelse med performance.
- **Miljø:** Jo flere søjler der er i et miljø, jo mere skal SOSI Gateway synkronisere, det ville derfor være en god ide at splitte miljøet op i to segmenter der hver servicerer halvdelen af kommunerne.
- **STS:** Under performance testen af STS blev det konstateret at Luna boksen bidrager med en meget stor andel til STS'ens svar tid. SOSI-GW vil dermed ligeledes blive ramt af denne forøgede svartid, og det kunne være fornuftig at lave en yderligere kørsel med en STS, der ikke bruger en Luna boks. Dette vil måske også forklare at TP ikke forøges ved 2 søjler.

8 Splunk søgninger

Heap:

```
index=nspstage host=cstag-kgw0* sourcetype="jmx" mbean_property_type="Memory" | c
```

CPU:

```
index=nspstage sourcetype=vmstat host=cstag-kgw0* | multikv | timechart span=1m :
```

Requests:

```
index=nspstage host=cstag-kgw0* sourcetype="json_auto_timestamp" path="/sosigw*"
```

Name	Version	Date
20131122_130117.png	1	2013-11-26 11:30
20131122_130117_heap.png	2	2013-11-26 13:41
20131122_130117_reqs.png	1	2013-11-26 11:38
20131126_134451.png	1	2013-12-09 15:06
20131126_134451_heap.png	1	2013-12-09 15:05
20131126_134451_reqs.png	1	2013-12-09 15:08
20131126_152518.png	1	2013-12-09 15:10
20131126_152518_heap.png	1	2013-12-09 15:09
20131126_152518_reqs.png	1	2013-12-09 15:11