Cesta k performance testům produkční infrastruktury

Jan Hroten • 22. 11. 2022

CZ.NIC | SPRÁVCE DOMÉNY CZ

Úvod

Testování na produkci je v IT světě zaklínadlo, které spolehlivě vyklidí místnost, a zároveň příslib spousty historek, kterými se bavíme po konferencích u piva.

Když je tato „produkce“ zároveň registr národní domény .cz, bude určitě o zábavu postaráno.

Pokusím se tedy dnes odpovědět mimo jiné na:

- Proč?
- Jak?
- Co z toho?
- Je to vůbec potřeba?


STARECAT.COM
WHAT COULD
POSSIBLY GO WRONG?

Proč to děláme?

- Máme monitoring a alerting produkčních zdrojů - pouze reaguje na nastalé situace
- Máme integrační testy, ruční i automatizované FE a BE testy každého releasu
- Díky píli SQA oddělení pravidelně provádíme performance testy v TestEnvu
- ... bohužel kvůli rozdílům mezi TestEnvem a produkcí nemají velkou výpovědní hodnotu - mimo srovnávání verzí

Narážíme na limity?

- Občas se k limitům některých systémů přiblížíme
- Primárně z důvodů (mis)konfigurace
- Např. služba `secretary`, kde javový blackbox generuje a podepisuje PDF soubory o stovkách stránek
- Vše se samozřejmě vyřešilo, avšak reaktivně a ne proaktivně

Potřebujeme více dat

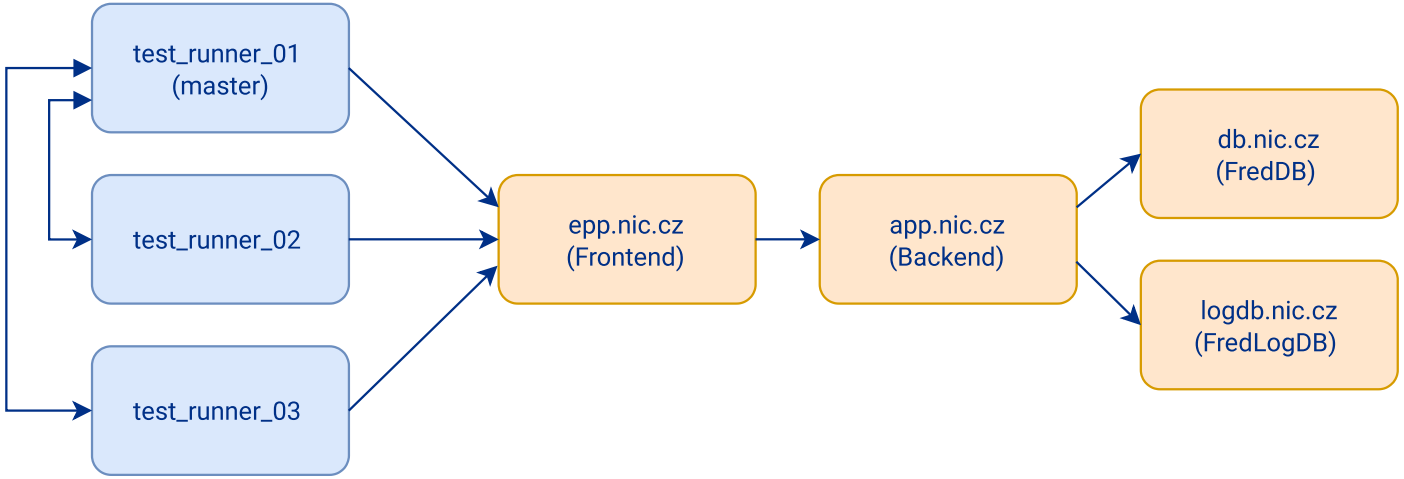
- TestEnv je virtualizované prostředí, servery sdílí zdroje
- Postavit testovací lab?
  - Obrovská investice zdrojů - vlastně další produkční lokalita
  - Je potřeba udržovat srovnané s produkcí, HW, SW i konfigurace
  - Je potřeba testovat, že služby fungují - produkční data?
  - Tudy ne!
- Je potřeba testovat přímo na produkci

Příprava produkčního prostředí

- Ideální řešení - izolovat jednu ze tří produkčních lokalit
- Síť je bohužel jeden L2 segment, nelze izolovat na úrovni sítě
- Izolace je potřeba dosáhnout konfigurací služeb
- Používáme VIP adresy a keepalived. Máme jistotu, že provoz jde pouze na main servery
  - Toto jsme si ověřili pomocí `tcpdumpu` síťového provozu
- Vše se připravuje v rámci `ansible playbooku`, konfiguračních změn je hodně
- Stále preferovaná metoda pro další postup - ale časově velmi náročná

Snížení scope

- Chceme nějaké výsledky!
- Co můžeme udělat bez úplné izolace lokality?
  - Read-only EPP přístupy, konkrétně simulace ”Boje o domény”
  - Prerekvizity: Ohlášená odstávka, vypnutí ostatního provozu do registru - minimalizace šumu



Příprava dat a zdrojů

- Domény - v našem vlastnictví, s dostatečnou platností
- Testovací registrátoři - tentokrát 10, omezení na počet session.
  - Registrátoři založeni jako `internal` - novinka v registru pro monitoring a testy
  - Registrátoři byli automatizovaně vytvořeni pouze pro účely testování
- Testovací scénáře
- Testovací runnery
  - Servery z obměny hlavního DNS stacku
  - Load je primárně na CPU, potřebujeme co nejvíce vláken
  - Do budoucna budeme mít tuto infrastrukturu připravenou a předinstalovanou v racku v naší centrále

Příprava samotných testů

- Locust
  - https://locust.io
  - Propracovaný open-source nástroj pro zátěžový test
  - Psaný v `pythonu`, rozšiřitelný o vlastní implementaci
  - Hotové řešení pro HTTP/HTTPS generování zátěže
  - Konfigurovatelnost běhu, škálovatelnost zátěže
  - Generování provozu z jednoho PC nebo distribuovaně z více strojů
- Vše připraveno a odladěno v TestEnvu
- Včetně infrastruktury pro distribuovaný běh

Monitoring

- Naše standardní tools:
  - NMS - sample rate 1 za 5 minut, lze měnit jen globálně
  - Icinga - alerting, součástí testů bylo i sledování alertů v zátěži
  - PGWatch - sample rate 1 za minutu, velmi užitečný nástroj na vizualizaci zátěže DB
- atop + toolchain pro vizualizaci dat:
  - Sample rate 1 za 5 vteřin, sbíráno do souboru na filesystemu
  - Otestováno v testu i produkci - neznatelný vliv na výkon serveru
  - cca 20MB dat za 3 hodiny odstávky
- Vizualizace atop → awk → influxDB → Grafana

  https://github.com/heinzz-da-ketchup/atop-graph

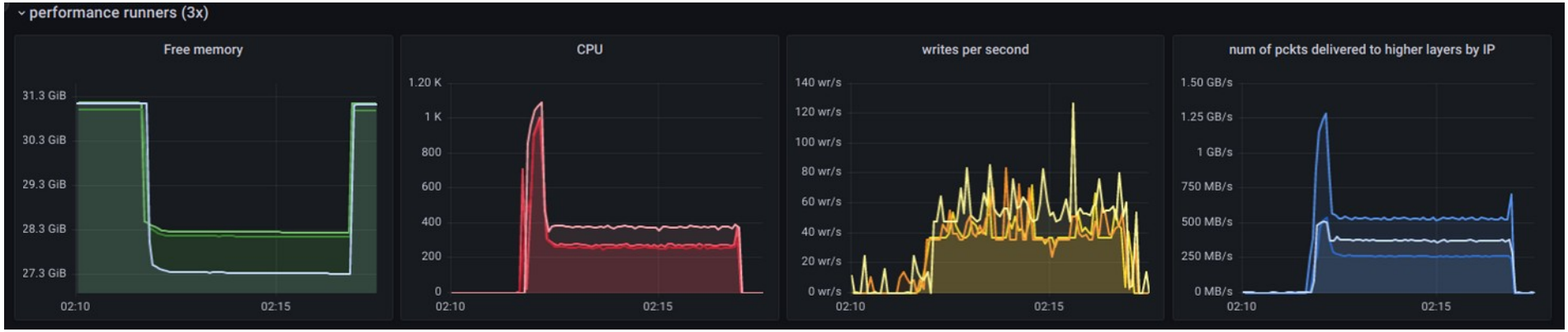
Testování

- V rámci nočního odstávkového okna
- Cca 30 minut příprava, 2,5 hod. testování, 30 minut obnovení provozu, pak verifikace
- Je možnost částečné automatizace pomocí `ansible`
- Nejdříve ”smoke” testy pro ověření, že vše funguje i na produkci
- Postupné přidávání zátěže až po dosažení limitů infrastruktury/testujících serverů
- Z časových důvodů jsme neprošli všechny scénáře. Už během testování nás navíc napadaly další

Testovací scénáře

- Distribuovaný režim na třech serverech
  - Locust master - řízení běhu, centrální sběr dat a vyhodnocení statistik
  - 3x locust runner - běh testů, každá session spouštěná ve vlastním vlákně
- Doba běhu 5 až 10 minut (stabilní provoz lokality)
- Postupně zvyšování počtu paralelních session - 30 až 400
- Nejvyšší zátěž při běhu s 200 sessions, při pokusu o 400 pak aktivních jen cca 120
- V rámci každé session vysílal registrátor requesty nejrychleji, jak to bylo možné, neodpovídá tedy běžnému provozu

Grafy


performance runners (3x)
Free memory
31.3 GiB
30.3 GiB
29.3 GiB
28.3 GiB
27.3 GiB
02:10
02:15
CPU
1.20 K
1 K
800
600
400
200
0
02:10
02:15
writes per second
140 wr/s
120 wr/s
100 wr/s
80 wr/s
60 wr/s
40 wr/s
20 wr/s
0 wr/s
02:10
02:15
num of pckts delivered to higher layers by IP
1.50 GB/s
1.25 GB/s
1 GB/s
750 MB/s
500 MB/s
250 MB/s
0 MB/s
02:10
02:15


DB LOGDB
Free memory
1.56 GiB
1.37 GiB
1.17 GiB
1000 MiB
02:10
02:15
CPU
1 K
800
600
400
200
0
02:10
02:15
writes per second
600 wr/s
400 wr/s
200 wr/s
0 wr/s
02:10
02:15
num of pckts delivered to higher layers by IP
15 GB/s
12.5 GB/s
10 GB/s
7.50 GB/s
5 GB/s
2.50 GB/s
0 MB/s
02:10
02:15

Vyhodnocení

- Testy proběhly během připraveného okna bez významných nečekaných událostí
- Vygenerovali jsme cca 20× běžný provoz, 5× peak provoz
- Jedná se pouze o EPP RO requesty, bez zohlednění ostatních druhů provozu není směrodatné
- Na první pohled je úzké hrdlo I/O na logdb
  - při další analýze to ale zdaleka není tak jednoznačné
  - jen 1% I/O Wait, jen 30 spojení do db - pod limitem
- Žádný z produkčních serverů se nedostal na své HW limity
- Potřebujeme víc dat =)

Odhalené problémy

- Redis cache
  - Python knihovna neuměla pracovat s clusterem, kterému chybí node
  - Opraveno v rámci bugfix releasu - Při odstávce lokality se zároveň odpojí redis node
  - Naučilo nás hodně o fungování redis clusteru
- Keepalived a routované privátní adresy
  - Nečekané chování, stále probíhá analýza

Závěr

- Nejdůležitější výstup - umíme to, nic jsme nerozbili a můžeme pokračovat dál
- Během vyhodnocování dat jsme začali připravovat další testovací scénáře
- Je třeba posílit testovací stroje, připravujeme HW pro trojnásobnou zátěž
- Budeme dál pracovat na scénářích pro ostrovní lokalitu
  - To nám umožní více se přiblížit reálnému provozu
  - Více služeb najednou, ”destruktivní” změny v registru

Poděkování

- Zdeňek Brůna, za motivaci z pozice vedoucího Technického oddělení
- Jaroslav Benkovský za tým sys-admins
- Aleš Jarolímek za SQA
- Jiří Šádek zastupující vývojáře registru

Děkuji vám za pozornost!

Jan Hroten

CZ.NIC | SPRÁVCE DOMÉNY CZ