SERVERLESS LOAD TESTING FOR REPLAYING TRAFFIC

Yuki Sawa @yukisww
Software Engineer
edmunds.com
github.com/edmunds/shadowreader
SUMMARY

- Challenges of load testing
- How we tried to solve it
- How it solved an incident
- Architecture
HARD PARTS OF LOAD TESTING

- Need real request rates

Traffic count to edmunds.com
Synthetic load test

Load test request rates
Need realistic test URLs

- edmunds.com/used-cars
- edmunds.com/used-hondas
- edmunds.com/ford
- edmunds.com/suv
query=query(makeSlug: String!, model Slug: String!, year: Int!) {
  allVehicles(makeSlug: $makeSlug, modelSlug: $modelSlug, year: $year) {
    models(modelSlug: $modelSlug) {
      modelYears(year: $year) {
        segmentRatings {
          rank
          slugRankedSubmodel
          editorialSegment {
            edmundsTypeCategory
            id
            displayName
            segmentRatings
          rating
          }%
        }%
      }%
    }%
  }%
}
CHALLENGES OF LOAD TESTING

- Distributed load tests
  - Maintenance 
  - Load test configs
  - Boot up scripts
  - CPU/MEM allocation
- Server costs 💰
- Load test takes time to start up
SHADOWREADER

- From a Hackathon in November
  - Used in prod in January
- Replay URLs
- Replay request rate
- Serverless
- AWS Lambda
Blue - Real traffic  Orange - ShadowReader
SHADOWREADER – WHAT IS IT?

- Simulate production traffic in QA
- Pre-prod canary deploys
- Replay peak traffic hour
THE CHRISTMAS EVE MEMORY LEAK

- Memory leak caused high errors rates in prod
- Couldn’t be reproduced in QA
- ShadowReader used to solve it
CHRISTMAS EVE INCIDENT

- December 24th, 2017
  - Memory leak causes high error rates and latency
  - On-call alerted 😭
- Resolved by Docker Orchestration engine (ECS)
1. Memory grows over a course of a few days
2. CPU spikes at the end

Customer response times are increasing the whole time.

Finally at the end availability starts to drop.
EDMUNDS INFRASTRUCTURE

- Docker on ECS
- Canary releases in prod and qa
- AutoScaling

=> Masked memory leak!
INVESTIGATION

- QA doesn’t have a memory leak!
PROD

CPU

Total 34.00%  Kernel 3.00%  User 34.00%

Memory

Usage 380.65 MB  RSS 305.93 MB  Cache 59.94 MB
INVESTIGATION

- Maybe load test can’t recreate it in QA?
- Synthetic load tests in QA
  - Using URLs generated by scripts or by hand
- Static request rates
Introduce ShadowReader

Saw results immediately

Memory usage in QA

Continuous Load Script -
 Doesn't show memory increase

Shadow Reader test in QA-11 -
Reflects memory leak pattern
similarly observed in prod
THE CAUSE

- Point ShadowReader to local
- 400MB of metadata
- Server caching for all users
- Cache was never being used!
THE SOLUTION

Ț Synthetic load test didn’t test with enough URLs/throughput to simulate enough users

Ț Replay traffic

Ț => generated enough unique meta data in the cache
- Disabled server side caching
- Memory nice and even 😊😊
SHADOWREADER ARCHITECTURE

- Tools and AWS Services
- ShadowReader features
  - Replay traffic
- Serverless
- Design and architecture
TOOLS

- Serverless framework
- Python 3
AWS SERVICES

- AWS Lambda
- S3
- Elastic Load Balancers
- CloudWatch Events
REPLAY LOAD TEST

- Parses production access logs and replays it
- Replay request rate and URLs
REPLAY LOAD TEST

- Live replay or Past replay
- Replay headers
- User-Agent or True-Client-IP
SERVERLESS

- Easy to scale
- Provision on demand
- Scale to 50k reqs / minute
- Cheap
- Pay only for what you use
- $1000+ / month → $100 / month
SERVERLESS

- Achieved by
  - High concurrency
  - 100 requests / Worker Lambda
  - 256MB MEM / Lambda
  - No maintenance, fast start up
Test data partitioned into minute intervals

- 1 minute of traffic == list of URLs from that minute
- 1 hour of traffic == 60 jsons
- An array of URLs for each minute of the day
- All URL data stored on S3
OTHER FEATURES

- Plugin system - choose live or past replay
- Support for replaying
  - Application/Elastic Load Balancer
- Ramp traffic by % value
ARCHITECTURE

- 4 Lambdas
- Parser
- Orchestrator
- Master
- Worker
User visits www.edmunds.com/honda

Nginx serves the request then sends access logs to S3

CloudWatch Event triggers Lambdas every minute

S3 Bucket of parsed URLs

Orchestrator Lambda

Parser Lambda

Worker Lambdas

qa.edmunds.com/honda
DEMO

- [github.com/edmunds/shadowreader](https://github.com/edmunds/shadowreader)
- Welcoming contributions 😊
- Replay HAProxy, other LBs, etc.
- Feedback / suggestions welcomed!!
DEMO

- [link](https://github.com/edmunds/shadowreader)
- Serverless framework
  - `npm install serverless`
  - `sls deploy`
DEMO

Serverless: Removing old service artifacts from S3...
(p36) [rc: ...
Serverless: Generated requirements from /Users/rc/repos/sr-past-replay-poc/shadowreader/requirements.txt in /Users/rc/repos/sr-past-replay-poc/shadowreader/.serverless/requirements/requirements.txt...
Serverless: Installing requirements from /Users/rc/repos/sr-past-replay-poc/shadowreader/.serverless/requirements/requirements.txt...
Serverless: Running...
Serverless: Packaging service...
Serverless: Excluding development dependencies...
Serverless: Injecting required Python packages to package...
Serverless: Uploading CloudFormation file to S3...
Serverless: Uploading artifacts...
Serverless: Uploading service sr.zip file to S3 (1.74 MB)...
Serverless: Validating template...
Serverless: Updating Stack...
Serverless: Checking Stack update progress...

Serverless: Stack update finished...

Service Information

service: sr
stage: past-replay
region: us-west-1
stack: sr-past-replay
resources: 14
api keys:
  None
endpoints:
  None
functions:
  orchestrator-past: sr-past-replay-orchestrator-past
  producer: sr-past-replay-producer
  consumer-worker: sr-past-replay-consumer-worker
  consumer master-past: sr-past-replay-consumer-master-past
layers:
  None
Serverless: Removing old service artifacts from S3...

Update available 1.37.1 → 1.38.0
Run npm i serverless to update
CloudWatch Monitoring Details

HTTP fixed response count (Count) Statistic: Sum Time Range: Last 3 Hours Period: 1 Minute

Graph showing HTTP fixed response count over time.
THANK YOU
AND SPECIAL THANKS TO EVERYONE THAT HELPED

CARLOS MACASAET
EMIL NDREU
SHARATH GOWDA
INAYATULLAH BHOLOAT
HABIB KHAN
LELAND SO

DENISE NGAI
ILANA GERSHTEYN
MONICA AIMA
NATALIA HRYSHCHANKOVA
PETER PURWANTO

>github.com/edmunds/shadowreader
@yukisww