Does your load test have you covered?

Chanchal Gupta, Simon Wong
SRE @ Edmunds
High Level Architecture at Edmunds

- Microservices
- Environments
Building Observability

- Basic Observability
  - Access Logs
  - Correlation
  - Analysis
Why Load Test?

- Identify problems before production
- Reduce failure costs
Challenges

- Gaining visibility
- Optimize quality of load test traffic for page types and microservices
What makes a good load test?

- High quality of load test traffic
  - Breadth
  - Depth

- Optimized
  - Proportional to Production
  - Low variability
How we gained visibility

- What visibility did we have?
- Processes we built to gain visibility
Visibility Gained

Edmunds pages types
- brand_landing
- home_page
- advice_buying_article_index
- advice_leasing_article
- advice_maintenance_article
- model_update_consumer_review
- new_cars_index
- model_car_inventory_vin_detail

100% load of production traffic

Scale is not linear (Log)
Visibility Gained: Breadth Problem

- Many page types had no coverage at all.
- Unable to test features on these page types.
- We now have direct line of sight into improvement.
- Pages with no load on QA were still functional tested.

73% of our page types did not have any coverage.
Visibility Gained: Depth Problem

- Continuous load on QA was not balanced
- Need to first bring everything closer to production level
- Fine tune later

- Many page types had excessive load that’s wasteful
- A lot of page types did not get enough load
Visibility Gained: Microservices Load Quality

- Upstream Micro Services also had breadth and depth issues on QA's continuous load
Optimizing Load Traffic Breadth

- Analyzed production traffic to identify where our coverage gaps were
- Examined the traffic to upstream services triggered by our web app to generate the proper URLs

Page types without QA coverage went from 73% to 17%
Optimizing Load Traffic Depth

- Analyzed production and upstream micro services traffic to create a model that generated the optimal # of requests each URL needed
- Brought most of our page types much closer to production level load
Prioritizing on Top Key Pages

- # of top key page types with high quality QA traffic
  - Previous: 0% page types
  - After: 81% page types
Optimizing Load Traffic Upstream Services

- Upstream services did not improve after rebalancing Web App.
- Decouple the traffic going to our Web App and its upstream services.
Optimizing Load Traffic Upstream Services

- Augmented continuous load for upstream services to provide test confidence
- Improved breadth and depth for upstream services
Optimize Load For Efficiency

- What is the minimum amount of continuous load to maintain the same level of confidence?
- Perform the analysis with historical data.
Optimize Continuous Load For Efficiency

- Previous QA traffic distribution was very different from PROD

- Current QA traffic after configuration update to improve quality of traffic is now much closer to PROD distributions
Optimize Continuous Load For Efficiency

- Result of lowering continuous load by 2/3.
- Reduced the # of H/W nodes for efficiency and cost saving.
Next Steps: Drift Prevention and Automation

- Alert on gaps in breadth or depth coverage of our load test.
- Keep configurations up to date.
Questions

Contact:

https://www.linkedin.com/in/chanchalg/

https://www.linkedin.com/in/simonjaywong/