Redliner

How LinkedIn Determines Its Service Capacity Limit Using Live Traffic

Anant Rao, Susie Xia
Performance @ LinkedIn

Oct 3, 2017
About Us

Anant Rao
Engineering Lead/Manager

Susie Xia
Sr. Software Engineer
About LinkedIn

First Launch: 2003
First Business Lines: Jobs & Subscription
IPO: LNKD 2011
Economic Graph 2014

2016
WHAAAT?
Why is server disproportionate to page view growth?
Why is server disproportionate to page view growth?

So at this rate, how many servers do we need say in 2018?
Why is server disproportionate to page view growth?

So at this rate, how many servers do we need say in 2018?

Are we utilizing resources efficiently?
Why is server disproportionate to page view growth?

So at this rate, how many servers do we need say in 2018?

Are we utilizing resources efficiently?

Can we confidently ramp product features?
Why is server disproportionate to page view growth?

So at this rate, how many servers do we need say in 2018?

Are we utilizing resources efficiently?

Can we confidently ramp product features?

Will the site be up when we have traffic surges?
What Gets Measured Gets Fixed.
Metric of Capacity

Maximum throughput a service can handle without compromising user experience.
Determining Max Throughput
Synthetic Test in Isolation
Synthetic Test in Production
Can we use live traffic?
Real Test in Production
Redliner

A tool that determines service capacity redline beyond which performance health can no longer be guaranteed.
LinkedIn Stack

Browser / App

Frontend Web Services

Mid-Tier Web Services

Mid-Tier Web Services

Mid-Tier Web Services

Backend Data Services

DB

Stateless Services

Stateful Services
Key Components

- Traffic Diversion Layer
- Metrics Collector
- Service Health Analyzer
Stateless Redlining

Service Under Test

Service Instance

Service Instance

Service Health Analyzer

Redliner

Metrics Collector

Proxy / Load Balancer

A/B Testing Framework

Traffic Shift Request

Redline Health Check Request

Pass/Fail

Live Requests from Service Clients

Raw Performance Metrics

Aggregated Metrics
The Inflection Point for QPS Ramp
How to redline stateful services?
High-Level Stateful Architecture

- **Incoming Production Traffic**
  - Traffic diverted for config lookup or scattered/gathered

- **Stateless Routers/Brokers**

  - **Stateful Partition/Shard/Node**
    - 40-59
    - 20-39
    - 0-19

  - **Source Node**
  - **Dark Node** 0-19
Stateful Redlining

A/B Testing Framework

Router / Broker

DB
DB_1
DB_2

Source Node

Dark Node

Traffic Shift Request

Get/Set Traffic Multiplier

Incoming Production Traffic

Redline Health Check Request

Pass/Fail

Service Health Analyzer

Aggregated Metrics

Metrics Collector

Raw Performance Metrics

Raw Performance Metrics
QPS

90pct Latency
I SEE YOU TEST YOUR CODE IN PRODUCTION

I TOO LIKE TO LIVE DANGEROUSLY
How to ensure low impact?
Minimize Impact

1. Health check & real-time monitoring
2. Intelligent ramp algorithms
3. Complete automation
1. Health Check & Monitoring

- Look out for early performance indicators
- Monitor overall data center health
2. Intelligent Ramp Algorithms

- **Slow Ramp (3 hours)**
- **Fast Ramp (45 minutes)**
3. Complete Automation

- Reports potential bottlenecks
- Fast recovery in the event of problems
Recap

• Three key components for Redliner: traffic diversion layer, metric collector and service health analyzer.
• Use dark node for stateful redlining!
• Load testing in production is not that intimidating.
Case Studies
LinkedIn's Multi-Hour Outage
Detecting Throughput Regression
Prevent Throughput Regressions

- Run Redliner test side-by-side on canary and production versions
Surplus Capacity

Server CapEx Trends for Service

Reclaimed unused resources
Takeaways

• Performance testing is most effective in prod
• Leverage APIs, Build your own Redliner.
• Run perf tests everyday – Build a capacity culture.
Do Not Count Servers, Make Servers Count.
thank you!
More Information

- RedLiner: Measuring Service Capacity with Live Production Traffic
  Susie(Jingshu) Xia, Zhenyun Zhuang, Anant Rao, Haricharan Ramachandra, Yi Feng, Ramya Pasumarti, IEEE ICWS 2017
- Service Capacity Measurement by Redlining with Live Production Traffic
  Susie(Jingshu) Xia, Zhenyun Zhuang, Anant Rao, Haricharan Ramachandra, Yi Feng, Ramya Pasumarti, ISPASS 2017
- Capacity Measurement and Planning for NoSQL Databases
  Ramya Pasumarti, Rushin Barot, Susie(Jingshu) Xia, Ang Xu, Haricharan Ramachandra, IEEE ICSC 2017