Standing on the Shoulders of Giants
Unleashing the Power of Scriptable Load Balancers

Emil Stolarsky
emil@shopify.com

Justin Li
jli@shopify.com
Building Systems is Hard
Typical Web Architecture
Typical Web Architecture
Typical Web Architecture

Edge Tier
- LB
- LB
- LB

Compute Tier
- APP
- APP
- APP
- JOBS
- JOBS
- JOBS

Data Tier
- CACHE
- DB
Typical Web Architecture
Typical Web Architecture
SQL or SQL-like
NoSQL
Proxies
Orchestration
Containers
Frameworks
Edge Tier

- LB
- LB
- LB

Compute Tier

- APP
- APP
- APP
- JOBS
- JOBS
- JOBS

Data Tier

- CACHE
- DB

NGINX

APACHE HTTP SERVER

HAProxy
Just write your own LB!
Scriptable Load Balancers
Layer 1: Edge Tier
- Load Balancer (LB)

Layer 2: Compute Tier
- App
- Jobs

Layer 3: Data Tier
- Cache
- DB

Load Balancer:
- Customized Logic in Scripts
Routing
Routing for services
Service Discovery

Load Balancer

10.0.23.4
10.0.23.7
10.0.23.8
10.0.25.2
Routing for sharding
<table>
<thead>
<tr>
<th>domain</th>
<th>shard_id</th>
</tr>
</thead>
<tbody>
<tr>
<td>store.wikimedia.org</td>
<td>1</td>
</tr>
<tr>
<td>snowdevil.com</td>
<td>1</td>
</tr>
<tr>
<td>shop.teslamotors.com</td>
<td>2</td>
</tr>
<tr>
<td>lakersstore.com</td>
<td>3</td>
</tr>
</tbody>
</table>
{
    "1": "dc2",
    "2": "dc2",
    "3": "dc1",
    ...
    "<shard_id>": "<dc>
}
INTERNET

1.2.3.0/24

DC 1

LB
LB

DC 2

LB
LB
store.wikimedia.org: shard 1
Scriptable Load Balancers @ Shopify
Shopify’s Edge

Edge Tier of DC

- ROUTER
- NGINX
- NGINX
- NGINX

Data Tier
- ZK
- MySQL
Shopify’s Edge

Edge Tier of DC

ROUTER

ECMP

NGINX

Data Tier

ZK

MySQL
Shopify’s Edge

Edge Tier of DC

Router

ECMP

Data Tier

ZK

MySQL

Read Global State

NGINX

NGINX

NGINX
nginScript: NGINX + Javascript

HAProxy: HAProxy + LuaJIT

mod_lua: Apache + LuaJIT
http {
  server {
    listen 8080;

    location / {
      return 200 'Hello, World!';
    }
  }
}
http {
  server {
    listen 8080;

    header_filter_by_lua_block {
      ngx.header['X-Epoch'] = ngx.time()
    }

    location / {
      return 200 'Hello, World!';
    }
  }
}
}
http {
    server {
        listen 8080;
        header_filter_by_lua_file 'epoch.lua';

        location / {
            return 200 'Hello, World!';
        }
    }
}

Undeployed Commits

Previous Deploys

- **Check**
  - **Clean up total_capacity configuration value**
  - **Deploy to prod LB**
  - **Add sleep between aws and chg reloads**
  - **Reload nginx in groups**
Config Management
Development
def test_epoch_header
    start_time = Time.now.to_i

    get "/", host_header: @shop

    assert start_time < response.headers["X-Epoch"].to_i
end
Middleware

- request timing
- rate limiting
- response caching
- render
Service-Level Middleware
Request pausing

Zero-downtime datacenter failovers
The diagram illustrates a network architecture with the following components:

- **INTERNET**
- **LB**: Load Balancer
- **app servers**

### Passive Configuration

- **LB**
- **app servers**

### Active Configuration

- **LB**
- **app servers**

The passive configuration uses a single load balancer to distribute traffic to the app servers, while the active configuration employs two load balancers for redundancy and increased performance.
INTERNET

active

app servers

LB

LB

passive

app servers

LB

LB
Multi-tenant TLS

All stores really should use encryption
Multi-tenant TLS

ame, err = ssl.server_name()

-- look up certificate from cache or sql
ssl.set_der_cert(certificate_chain)
ssl.set_der_priv_key(private_key)
Here Be Dragons
Testing
Staging/Deploys
How to draw an owl

1. Draw some circles

2. Draw the rest of the fucking owl
State Management
Scaling with NGINX
Caching
Throttling
GET /checkout
USER

NGINX

GET /checkout

GET /throttle_page

RAILS
USER

GET /checkout

NGINX

200: throttle page

GET /throttle_page

200: (cached by LB)

RAILS
USER

GET /checkout

200: throttle page

GET /checkout?poll=1

200: throttle page

GET /checkout?poll=1

302: set checkout pass

NGINX

GET /throttle_page

200: (cached by LB)

RAILS

GET /checkout?poll=1

302: /checkout
Just the Beginning
Thank You

@EmilStolarsky
@pushrax