Traffic Ops
Golang Rewrite

Dewayne Richardson
dewrich@apache.org
Traffic Ops Golang Incremental Rewrite App #729

https://github.com/apache/incubator-trafficcontrol/pull/729
Overview

- Golang Proxy Sequences
- Feature Parity Progress
- How the code base is being affected
- Testing
- Anatomy of a Handler
  - how to build one (if time)
Why the Golang Rewrite?

• Prefer Evolution over Big Bang
  
  • (Strangler Pattern - Martin Fowler)

• Reuse of deployment features
  
  • (rpms, configs, Puppet, Ansible)

• Performance
Why the Golang Rewrite?

• Tools
  • (Gogland, goconvey, and of course vim-go)
  • Emacs users, you’re on your own!

• Innovation
Sequence Diagrams
Golang Proxy with Golang Route

1. Client
2. Golang Proxy
   - /login request
   - Mojo Cookie requests Golang Proxy Route /servers
3. Mojo Login
   - Authentication Request (via LDAP or DB)
   - Mojo Cookie
   - Validates cookie and user role (against DB) if good then forwards request to Golang Handler
4. Golang Handler
   - Processes Request
5. Traffic Ops DB

Golang Proxy

Client

Mojo Login

Golang Handler

Traffic Ops DB
502 Bad Gateway
The server was acting as a **gateway** or proxy and received an invalid response from the upstream server.
Feature Parity Progress

- 11 of ~300 endpoints (handlers)
- Github Project
  - [https://github.com/apache/incubator-trafficcontrol/projects/2](https://github.com/apache/incubator-trafficcontrol/projects/2)
Traffic Ops v2.1 Features

Mojo/Perl Libraries

<table>
<thead>
<tr>
<th>UI/APIs</th>
<th>Extensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Routes</td>
<td></td>
</tr>
<tr>
<td>Cookies</td>
<td>RpmS</td>
</tr>
<tr>
<td>Ldap</td>
<td></td>
</tr>
<tr>
<td>Docs</td>
<td></td>
</tr>
<tr>
<td>Traffic Ops v2.1</td>
<td>UtilityS</td>
</tr>
</tbody>
</table>

HTTP Headers

<table>
<thead>
<tr>
<th>Gzip</th>
<th>Log4perl</th>
<th>COrS</th>
</tr>
</thead>
</table>

SSL

<table>
<thead>
<tr>
<th>Riak</th>
<th>Configs</th>
<th>Db Drivers</th>
</tr>
</thead>
</table>

Influxdb

<table>
<thead>
<tr>
<th>Email</th>
<th></th>
<th></th>
</tr>
</thead>
</table>
Traffic Ops v2.2 Features

- **TBD**
- * External Dependency
- Reused

**Golang**
- APIs
- Extensions

**HTTP Headers**
- routes
- cookies

**Proxying**
- docs
- * db drivers

**Extensions**
- rpms
- gzip
- CORS
- SSL
- log
- configs

**External Dependency**
- *Riak

**Reused**
- ldap
- Influxdb
- Email
Traffic Ops v2.x

APIs

Common API Packages

- rpms
- HTTP Headers
- cookies
- CORS
- SSL
- log
- configs
- *Riak
- * db drivers
- Idap
- Influxdb
- Email
- proxies
- docs
- Extenssions/Microservices?
How is the code base being affected?
package tc

type ASNsResponse struct {
    Response []ASN `json:"response"`
}

type ASN struct {
    ASN int `json:"asn" db:"asn"
    Cachegroup string `json:"cachegroup" db:"cachegroup"
    CachegroupID int `json:"cachegroupId" db:"cachegroup_id"
    ID int `json:"id" db:"id"
    LastUpdated Time `json:"lastUpdated" db:"last_updated"
}
Traffic Control
Golang with Mojo Components

- Traffic Monitor
- Traffic Stats
- Traffic Ops Client
- Traffic Ops
Traffic Ops Golang
Components
“Future State”

Traffic Router?
Traffic Ops Client
Traffic Monitor
Traffic Ops Client
Traffic Stats
Traffic Ops

structs
structs
structs
Traffic Ops Golang

Components

“Future State”

Traffic Router

Traffic Monitor

Traffic Stats

Traffic Ops Client

Integration Tests

Traffic Ops

/incubator-trafficcontrol/lib/go-tc

 structs
Config Files

/opt/traffic_ops/app/conf/cdn.conf

/opt/traffic_ops/app/conf/production/database.conf

/opt/traffic_ops/app/conf/production/influxdb.conf

/opt/traffic_ops/app/conf/production/riak.conf

/opt/traffic_ops/app/conf/production/log4perl.conf
Mojolicious formatted cdn.conf

```perl
{
    hypnotoad => {
        listen => [
            'https://[:]:443?cert=/etc/pki/tls/certs/localhost.crt&key=/etc/pki/tls/private/localhost.key&verify=0x00&ciphers=AES128-GCM-SHA256:HIGH:!RC4:!MD5:!aNULL:!EDH:!ED',
            user => 'trafops',
            group => 'trafops',
            heartbeat_timeout => 20,
            pid_file => '/var/run/traffic_ops.pid',
            workers => 96
        ],
        cors => {
            access_control_allow_origin => '*'
        },
        to => {
            base_url => 'http://localhost:3000',
            email_from => 'no-reply@traffic-ops-domain.com'
        },
        portal => {
            base_url => 'http://localhost:8080',
            email_from => 'no-reply@traffic-portal-domain.com'
        },
        secrets => [ 'mONKEYDOMONKEYSEE.' ],
        geniso => {
            iso_root_path => '/opt/traffic_ops/app/public',
        },
        inactivity_timeout => 60
    };
}
JSON formatted cdn.conf

```json
{
    "hypsnotoad": {
        "listen": ["https://[::]:60443?cert=/etc/pki/tls/certs/localhost.crt&key=/etc/pki/tls/private/localhost.key&verify=0x00&ciphers=AES128-GCM-SHA256:HIGH:!RC4:!MD5:!aNULL:!ED"],
        "user": "trafops",
        "group": "trafops",
        "heartbeat_timeout": 20,
        "pid_file": "/var/run/traffic_ops.pid",
        "workers": 12
    },
    "traffic_ops_golang": {
        "port": "443",
        "proxy_timeout": 60,
        "proxy_keep_alive": 60,
        "proxy_ttls_timeout": 60,
        "proxy_read_header_timeout": 60,
        "read_timeout": 60,
        "read_header_timeout": 60,
        "write_timeout": 60,
        "idle_timeout": 60,
        "log_location_error": "/var/log/traffic_ops/error.log",
        "log_location_warning": "/var/log/traffic_ops/access.log"
    },
    "cors": {
        "access_control_allow_origin": "*
    },
    "to": {
        "base_url": "http://localhost:3000",
        "email_from": "no-reply@traffic-ops-domain.com",
        "no_account_found_msg": "A Traffic Ops user account is required for access. Please contact your Traffic Ops user administrator."
    },
    "portal": {
        "base_url": "http://localhost:8080/!#/",
        "email_from": "no-reply@traffic-portal-domain.com",
        "pass_reset_path": "user",
        "user_register_path": "user"
    },
    "secrets": ["mONKEYDOmONKEYSEE.
    ],
    "geniso": {
        "iso_root_path": "/opt/traffic_ops/app/public"
    }
}
```
Benefits of Testing

• Makes the Process Agile

• Quality of Code

• Finds Software Bugs Early

• Facilitates Changes and Simplifies Integration

https://dzone.com/articles/top-8-benefits-of-unit-testing
Benefits of Testing

• Provides Documentation

• Debugging Process

• Design

• Reduces Cost

https://dzone.com/articles/top-8-benefits-of-unit-testing
How to Test a Golang API Handler

```
func getX() {}  
```

```
func TestGetX() {}  
```

Integration Testing is End-to-End
https://cwiki.apache.org/confluence/display/TC/API+Guidelines
Anatomy of a Golang Handler