

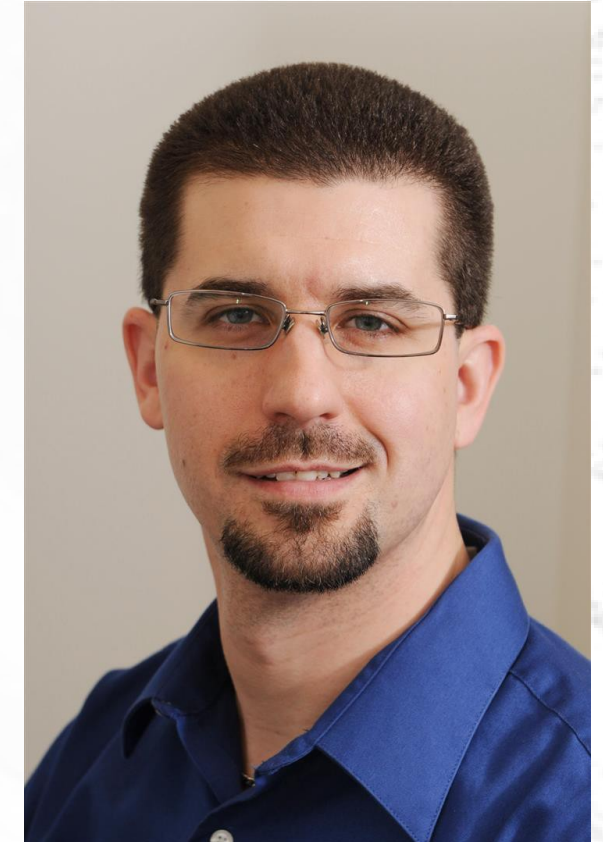
# The mod\_proxy Cookbook

A collection of proxy recipes to  
suit your discerning palate

Daniel Ruggeri

# Who is This Guy?

- About Daniel Ruggeri
  - Infrastructure guy with a love for code
  - DRuggeri <at> apache.org
- Standard Disclaimer
  - I'm speaking personally and not on behalf of my employer. The examples and comments are my personal opinions and should not be considered the official practices or positions of MasterCard.



# Things To Talk About

- About this presentation
  - Not just mod\_proxy
  - Know thine application
- Warning – eye charts ahead!
  - Examples may be hard to read
  - Included for completeness
- Download this presentation!
  - <http://people.apache.org/~druggeri/presentations/proxyCookbook.odp>



# What's New and Hot?



# Newness – HTTP/2

- HTTP/2 (aka H2 - RFC7540) support
  - Negotiated via ALPN in TLS extension
  - Requires libnghttp2, too
- In 2.4 now – “Experimental” state

```
LoadModule http2_module modules/mod_http2.so
```

```
LoadModule proxy_http2_module modules/mod_proxy_http2.so
```

```
ProxyPass /newness/ h2://mybackend.bitnebula.com/newness/
```

# Newness - Monitoring

- Proactive
  - Probe backends for health
  - w00t REALLY hard!

```
<Proxy balancer://myCluster>
```

```
    BalancerMember http://alpha:8080 hcmethod=TCP hcinterval=5
```

```
    BalancerMember http://beta:8080 hcmethod=TCP hcinterval=5
```

```
</Proxy>
```

# Newness - websockets

- WebSocket (RFC6455) support
  - Full duplex socket
  - Upgraded connection via HTTP/1.1

```
LoadModule proxy_wstunnel_module modules/mod_proxy_wstunnel.so
```

```
ProxyPass /ws2/ ws://echo.websocket.org/
```

# Newness - UDS

- Unix Domain Socket
  - Local connection only
  - A socket without all that TCP stuff
  - Pipe separator

```
ProxyPass / unix:/var/run/superApp.sock|http://localhost/
```



# Newness - mod\_proxy\_express

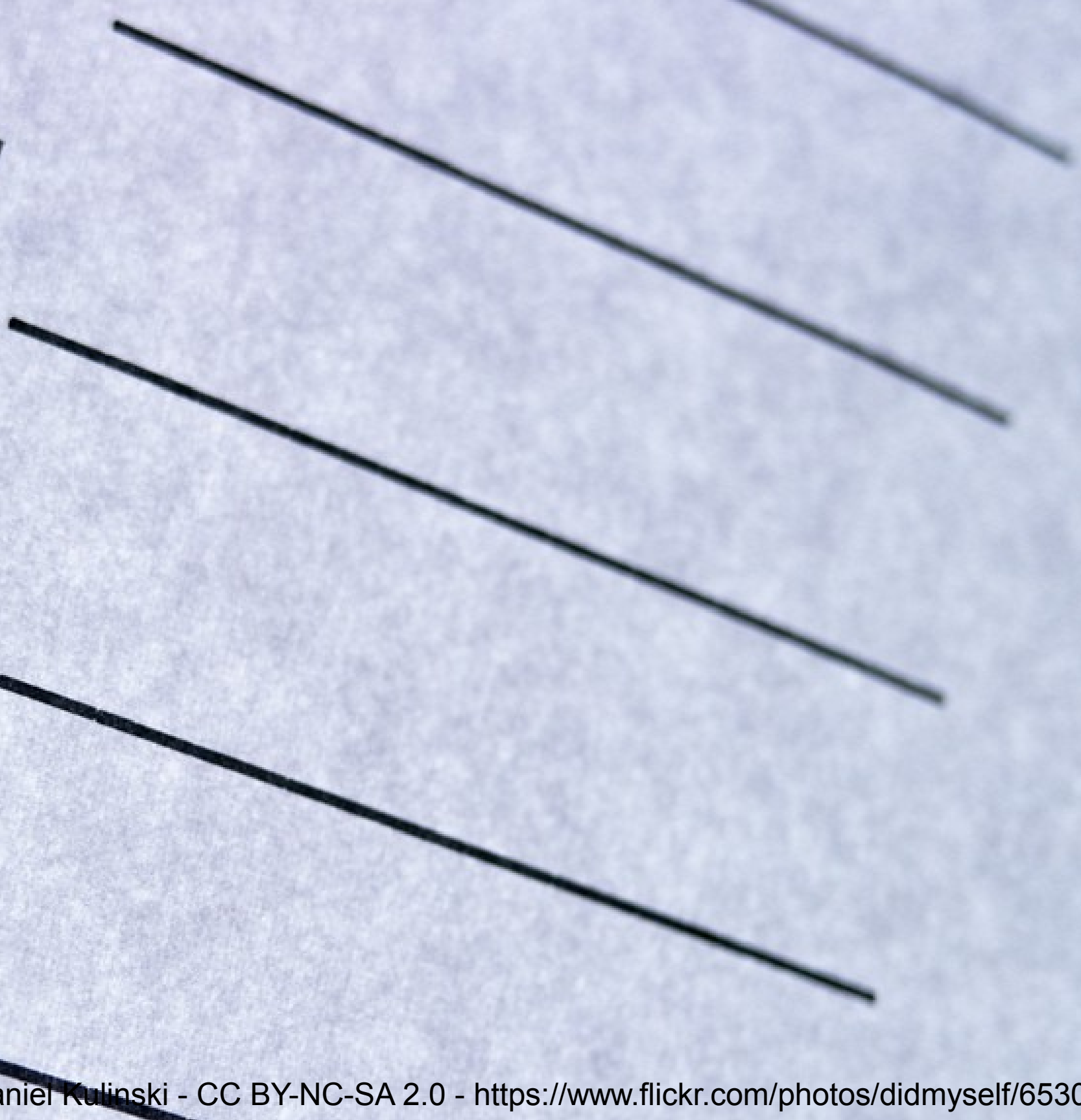
- Express
  - Mass name-based, switch-like proxying
  - Target server selection is driven by DBM file

## DBM file:

```
www.homeawayfromhome.com    http://10.0.1.25  
login.homeawayfromhome.com  http://10.0.2.15
```

## Config file:

```
ProxyExpressEnable on  
ProxyExpressDBMFile /path/to/mapfile
```



# How to Be a Good Proxy

- Connection Marshaling/Protocol Enforcement
- Load Balancing/Session Stickiness
- Connection Pooling/TCP and SSL Offload
- Failover/Health Monitoring
- Dynamic Modification
- Traffic shaping/Caching/Compression
- Attack Mitigation (Security)



# Connection Marshaling/Protocol Enforcement





# Playing Traffic Cop

- Separates clients and servers
- The difference between forward and reverse proxy
  - What does the client know?
- Forward proxy
  - mod\_proxy\_connect for SSL
- Reverse proxy uses mod\_proxy\_(ajp|http|ftp|scgi|fcgi|wstunnel)
  - mod\_ssl and SSLProxyEngine for SSL

# Forward Proxy Example

- **WARNING:** Do not proceed until you know how to lock this down!

```
LoadModule proxy_connect_module modules/mod_proxy_connect.so
```

```
<VirtualHost 10.1.2.3:8888>
```

```
ProxyRequests On
```

```
<Proxy *>
```

```
Require ip 192.168
```

```
</Proxy>
```

```
</VirtualHost>
```



# Reverse Proxy Examples

- In a Location block

```
<Location /application>
```

```
    ProxyPass http://backend.local/application
```

```
</Location>
```

- Standalone ProxyPass directive

```
ProxyPass /application http://backend.local/application
```

```
ProxyPassReverse /application http://backend.local/application
```

# Reverse Proxy Examples

- As a ProxyPassMatch

```
ProxyPassMatch ^/([^/]+\do)$ http://backend.local/app/$1
```

- In the Rewrite engine

```
RewriteCond %{HTTP_COOKIE} TOP_SECRET_ACCESS
```

```
RewriteRule ^/admin/(.*) http://backend.local/admin/ [P]
```

# Reverse Proxy Examples

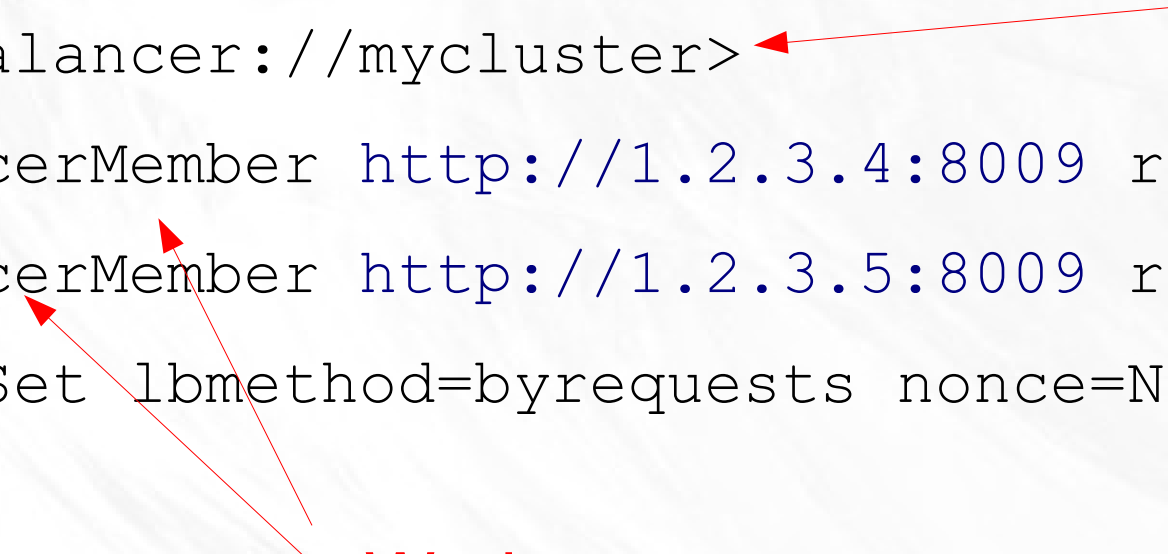
- As a Balancer

```
<Proxy balancer://mycluster>
    BalancerMember http://1.2.3.4:8009 route=Mercury
    BalancerMember http://1.2.3.5:8009 route=Venus
    ProxySet lbmethod=byrequests nonce=None stickysession=JSESSIONID
</Proxy>

ProxyPass /myApp/ balancer://mycluster/myApp/
```

**Balancer**

**Workers**





# Reverse Proxy Examples

- **As a DB (2.4)**

`ProxyExpressEnable on`

`ProxyExpressDBMFile /path/to/mapfile`

- **As a Handler (2.4.10+)**

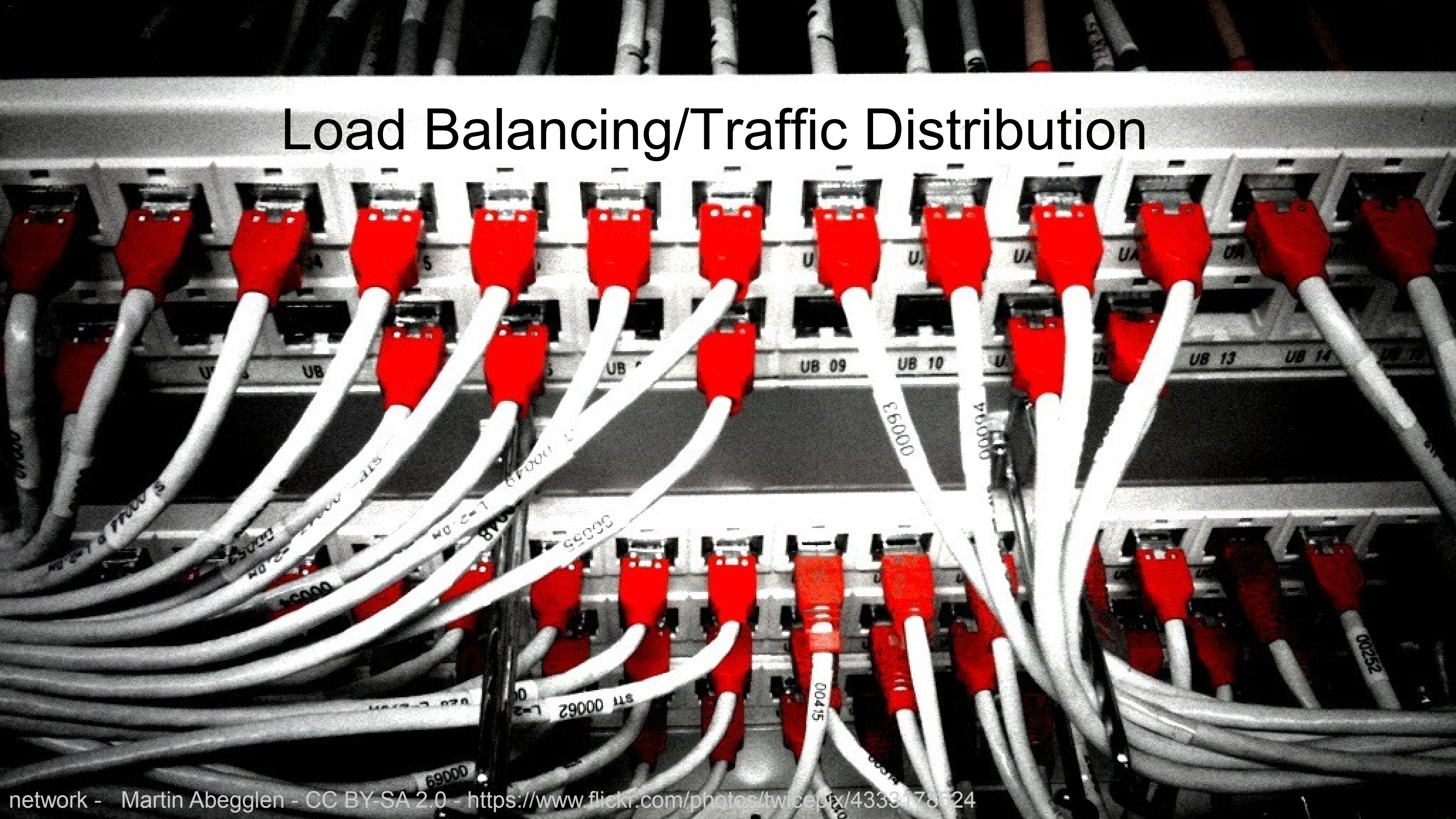
`<FilesMatch \.php$>`

`# Unix sockets require 2.4.7 or later`

`SetHandler "proxy:unix:/path/to/app.sock|fcgi://localhost/"`

`</FilesMatch>`

# Load Balancing/Traffic Distribution



# Load Distribution

- byrequests
  - Perform balancing based solely on requests served
- bytraffic
  - Perform balancing by byte count (in response body) served
- bybusyness
  - Perform balancing based on how many pending requests exist for a backend
- heartbeat
  - Perform balancing based on What mod\_heartbeat tells us
- ???
  - Some rumblings of what is coming



# Load Distribution

- Asymmetric distribution
  - loadfactor option for BalancerMember
  - higher number == higher load
- +H option for hot-standby
  - Disables worker until others are unavailable
  - Don't forget lbset as another option
- Selective proxying using ! and ordering
  - Do not proxy certain paths

# Example: Weighting

```
<Proxy balancer://mycluster>
```

```
    BalancerMember http://1.2.3.4:8009 loadfactor=2
```

```
    BalancerMember http://1.2.3.5:8009 smax=10 loadfactor=2
```

```
    #Less powerful server - fewer requests
```

```
    BalancerMember http://1.2.3.6:8009 smax=1 loadfactor=1
```

```
</Proxy>
```

```
ProxyPass / balancer://mycluster/ stickysession=JSESSIONID
```

# Example: Hot Standby

```
<Proxy balancer://hotcluster>
```

```
    BalancerMember http://1.2.3.4:8009
```

```
    BalancerMember http://1.2.3.5:8009
```

```
    #Hot standby
```

```
    BalancerMember http://1.2.3.6:8009 status=+H
```

```
    ProxySet lbmethod=bytraffic
```

```
</Proxy>
```

```
ProxyPass / balancer://hotcluster/
```

# Example: Selective Proxying

```
<Proxy balancer://AppCluster1>
```

```
    BalancerMember http://1.2.3.4:8009
```

```
    BalancerMember http://1.2.3.5:8009
```

```
</Proxy>
```

```
<Proxy balancer://AppCluster2>
```

```
    BalancerMember http://9.8.7.6:8080
```

```
    BalancerMember http://9.8.7.5:8080
```

```
</Proxy>
```

```
ProxyPass /static/ !
```

```
ProxyPass /applicationA/ balancer://AppCluster1/
```

```
ProxyPass /applicationB/ balancer://AppCluster2/
```

```
ProxyPass / balancer://hotcluster/
```

# Worker Statuses

- Disabled (D) / Stopped (S)
  - Worker is administratively disabled and will not accept any requests.
- Ignore Errors (I)
  - Will always be considered available.
- Hot Standby (H)
  - Will only be used if no other viable workers are available.
- Error (E)
  - Will not be used due to error.
- Drain (N)
  - Will only accept existing sticky sessions for its route.
- Check ©
  - Worker is disabled due to health check
- Redirect\*
  - New requests without sessions will go here.



# Sticky Sessions





# Session Persistence

- Session replication can be expensive
- Built-in (as designed)
  - mod\_proxy\_balancer includes facilities to do this
  - Not always compatible or easy
- Roll your own
  - Use the built-in functions but tweak to your liking
- Route parameter comes into play

# A Sticky Matter

- Many different formats for session identifiers based on backend.
  - Cookies, URLs, formats, etc
- You have to know a lot
  - Name of the cookie
  - Values contained
- Built-in is not 100% compatible.
  - (2.2) Requires dot or semicolon as a delimiter
  - (2.4) sticky session sep can be anything

# Universal Sticky!!!

```
LoadModule headers_module modules/mod_headers.so
```

```
<Proxy balancer://DanielCluster>
```

```
    BalancerMember http://1.2.3.4:8009 route=mercury
```

```
    BalancerMember http://1.2.3.5:8009 route=venus
```

```
    ProxySet stickysession=DanielsApp_STICKY
```

```
</Proxy>
```

```
Header add Set-Cookie "DanielsApp_STICKY=sticky.%"  
{BALANCER_WORKER_ROUTE}e;path=/;" env=BALANCER_ROUTE_CHANGED
```

```
ProxyPass /daniel/ balancer://DanielCluster/daniel/
```



# Connection Pooling/TCP and SSL Offload





# Get in the Pool

- So easy it is almost automatic
- Parameters
  - **max** hard maximum
  - **smax** soft maximum (aggressive TTL cleanup)
  - **ttd** time allowed to be idle
- Other parameters come into play
- Complications...
  - TCP/HTTP Keepalive

# Example: Connection Pooling

```
<Proxy balancer://myCluster>
```

```
    BalancerMember http://1.2.3.4:8009 smax=7 max=10 ttl=10
```

```
    BalancerMember http://1.2.3.5:8009 smax=7 max=10 ttl=10
```

```
</Proxy>
```

```
ProxyPass / balancer://myCluster/
```

# Leave the Tough Stuff to Me

- Funnel all traffic into the pipeline
  - Many requests <-> one backend connection
  - keepalive is a beautiful thing
- SSL benefits as well
  - HTTPS to HTTPD
  - Can run HTTP or HTTPS to backend
- Either will be more efficient!
- Node.js use case



# Failover/Health Monitoring





# Failure Detection

- Failover capability for connection only
  - Connection errors fail over to next backend seamlessly.
- SSL errors go back to user.
  - ... and are taken out of service as of 2.2.18.
- Hung/slow backend errors go back to user.
  - ... but can be taken out of service as of 2.2.25/2.4.5 with failontimeout.



# Built-in Monitoring

- Able to probe backends on an interval as of 2.4.21
- TCP
  - Simple port probe
- HTTP
  - GET/HEAD/OPTIONS
  - Evaluate results against expressions (match = OK)
- Separate workers from the balancer
  - Parallel execution, too!

# Example: Built-in Monitoring

```
ProxyHCExpr nofail {hc('body') !~ /Failure/}
ProxyHCTemplate aliveAndWell hcmethod=GET hcuri=/monitor.do \
    hcinterval=60 hcpasses=2 hcfails=3 hcexpr=nofail

<Proxy balancer://myCluster>
    BalancerMember http://alpha:8080 hctemplate=aliveAndWell
    BalancerMember http://beta:8080 hctemplate=aliveAndWell
</Proxy>

ProxyPass / balancer://myCluster/
```

# Live Traffic Monitoring

- connectiontimeout
  - Sets the number of seconds to wait for a TCP connection.
- ProxyTimeout and failontimeout
  - Fail faster and mark the backend out of service
  - Warning - this may be bad for you
- Failonstatus
  - Mark a backend out of service if a specific HTTP status code is found
- Monitoring
  - Create external monitoring to force traffic through HTTPD.



# Dynamic Modification



# Doing the Shuffle

- BalancerManager is how one modifies members.
  - Good selection of parameters
- Balancer
  - sticky identifier, timeout, failover, failover attempts, lbmethod
  - Workers can be added if growth is set
  - Workers can not be removed
- Worker
  - loadfactor, lbset, route, redirect
  - ignore errors, draining, disabled, hot standby
- Be safe out there

# Lay Thine Eyes Upon It!

## Load Balancer Manager for 127.0.0.1

Server Version: Apache/2.4.12 (Unix) OpenSSL/1.0.1e PHP/5.3.8

Server Built: Mar 30 2015 10:47:14

Balancer changes will NOT be persisted on restart.

Balancers are inherited from main server.

ProxyPass settings are inherited from main server.

### LoadBalancer Status for [balancer://mycluster](#) [pa24444cc\_mycluster]

MaxMembers	StickySession	DisableFailover	Timeout	FailoverAttempts	Method	Path	Active
2 [2 Used]	(None)	Off	0	1	byrequests	/test/	Yes

Worker URL	Route	RouteRedir	Factor	Set	Status	Elected	Busy	Load	To	From
<a href="https://127.0.0.1:8001">https://127.0.0.1:8001</a>			1	0	Init Ok	0	0	0	0	0
<a href="https://127.0.0.1:8002">https://127.0.0.1:8002</a>			1	0	Init Ok	0	0	0	0	0

### Edit balancer settings for balancer://mycluster

LBmethod:	<input type="text" value="byrequests"/>
Timeout:	<input type="text" value="0"/>
Failover Attempts:	<input type="text" value="1"/>
Disable Failover:	On <input type="radio"/> Off <input checked="" type="radio"/>
Sticky Session:	<input type="text" value=""/> (Use '-' to delete)
<input type="button" value="Submit"/>	

## Load Balancer Manager for 127.0.0.1

Server Version: Apache/2.4.12 (Unix) OpenSSL/1.0.1e PHP/5.3.8

Server Built: Mar 30 2015 10:47:14

Balancer changes will NOT be persisted on restart.

Balancers are inherited from main server.

ProxyPass settings are inherited from main server.

### LoadBalancer Status for [balancer://mycluster](#) [pa24444cc\_mycluster]

MaxMembers	StickySession	DisableFailover	Timeout	FailoverAttempts	Method	Path	Active
2 [2 Used]	(None)	Off	0	1	byrequests	/test/	Yes

Worker URL	Route	RouteRedir	Factor	Set	Status	Elected	Busy	Load	To	From
<a href="https://127.0.0.1:8001">https://127.0.0.1:8001</a>			1	0	Init Ok	0	0	0	0	0
<a href="https://127.0.0.1:8002">https://127.0.0.1:8002</a>			1	0	Init Ok	0	0	0	0	0

### Edit worker settings for https://127.0.0.1:8001

Load factor:	<input type="text" value="1"/>												
LB Set:	<input type="text" value="0"/>												
Route:	<input type="text" value=""/>												
Route Redirect:	<input type="text" value=""/>												
Status:	<table><tr><td>Ignore Errors</td><td>Draining Mode</td><td>Disabled</td><td>Hot Standby</td></tr><tr><td>On <input type="radio"/></td><td>On <input type="radio"/></td><td>On <input type="radio"/></td><td>On <input type="radio"/></td></tr><tr><td>Off <input checked="" type="radio"/></td><td>Off <input checked="" type="radio"/></td><td>Off <input checked="" type="radio"/></td><td>Off <input checked="" type="radio"/></td></tr></table>	Ignore Errors	Draining Mode	Disabled	Hot Standby	On <input type="radio"/>	On <input type="radio"/>	On <input type="radio"/>	On <input type="radio"/>	Off <input checked="" type="radio"/>	Off <input checked="" type="radio"/>	Off <input checked="" type="radio"/>	Off <input checked="" type="radio"/>
Ignore Errors	Draining Mode	Disabled	Hot Standby										
On <input type="radio"/>	On <input type="radio"/>	On <input type="radio"/>	On <input type="radio"/>										
Off <input checked="" type="radio"/>	Off <input checked="" type="radio"/>	Off <input checked="" type="radio"/>	Off <input checked="" type="radio"/>										
<input type="button" value="Submit"/>													

# Balancer Manager Errata

- Nonce usage
  - Set the nonce or use "None" for scripting
- XML output
  - Useful for machines
- REST-like (todo)
  - b, w and nonce parameters part of URL
- Persistence over restart
  - (2.4.4) Will write state before shutdown
- Be careful out there



# Shaping





# Traffic Tweaking

- Caching via mod\_cache
  - Too much to cover here.
- Compress via mod\_deflate
- Shape via...
  - mod\_proxy\_html, mod\_headers, mod\_rewrite, mod\_substitute, mod\_sed
  - mod\_env/mod\_setenvif, mod\_expires, mod\_\*filter
- Watch with mod\_dumpio
- The sky is the limit!

# Example: Traffic Shaping

```
ProxyPass /app balancer://myCluster/app
```

```
ProxyPassReverse /app balancer://myCluster/app
```

```
<Location /app>
```

```
    AddOutputFilterByType SUBSTITUTE text/html
```

```
    Substitute "s|http://127.0.0.1:7004|http://mypage|n"
```

```
    RequestHeader set environment production
```

```
    AddOutputFilterByType DEFLATE text/html text/xml
```

```
</Location>
```



# Security



# Not in MY House...

- Security modules
  - mod\_noloris, mod\_security, etc
- Separation
  - Tiered approach
  - Standards enforcement
- Filtering/Blocking/Restricting
  - Allow from certain hosts
  - Authn/Authz modules
  - The sky is (still) the limit!



Questions?

Scenarios?

Feedback: druggeri <at> apache.org

Download me:

<http://people.apache.org/~druggeri/presentations/proxyCookbook.odp>