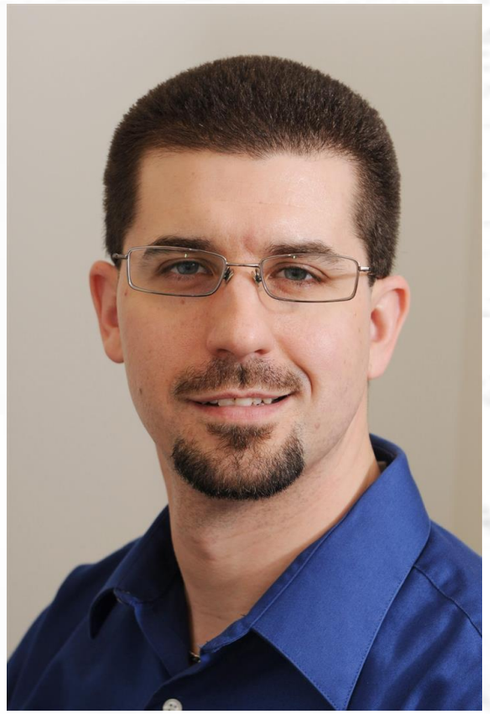
The mod_proxy Cookbook

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.

Between You and Lunch

- 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 - 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
```

One done - Daniel Kulinski - CC BY-NC-SA 2.0 - https://www.flickr.com/photos/didmyself/6530389351

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 /application/.*.do http://backend.local/application/
```

- 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

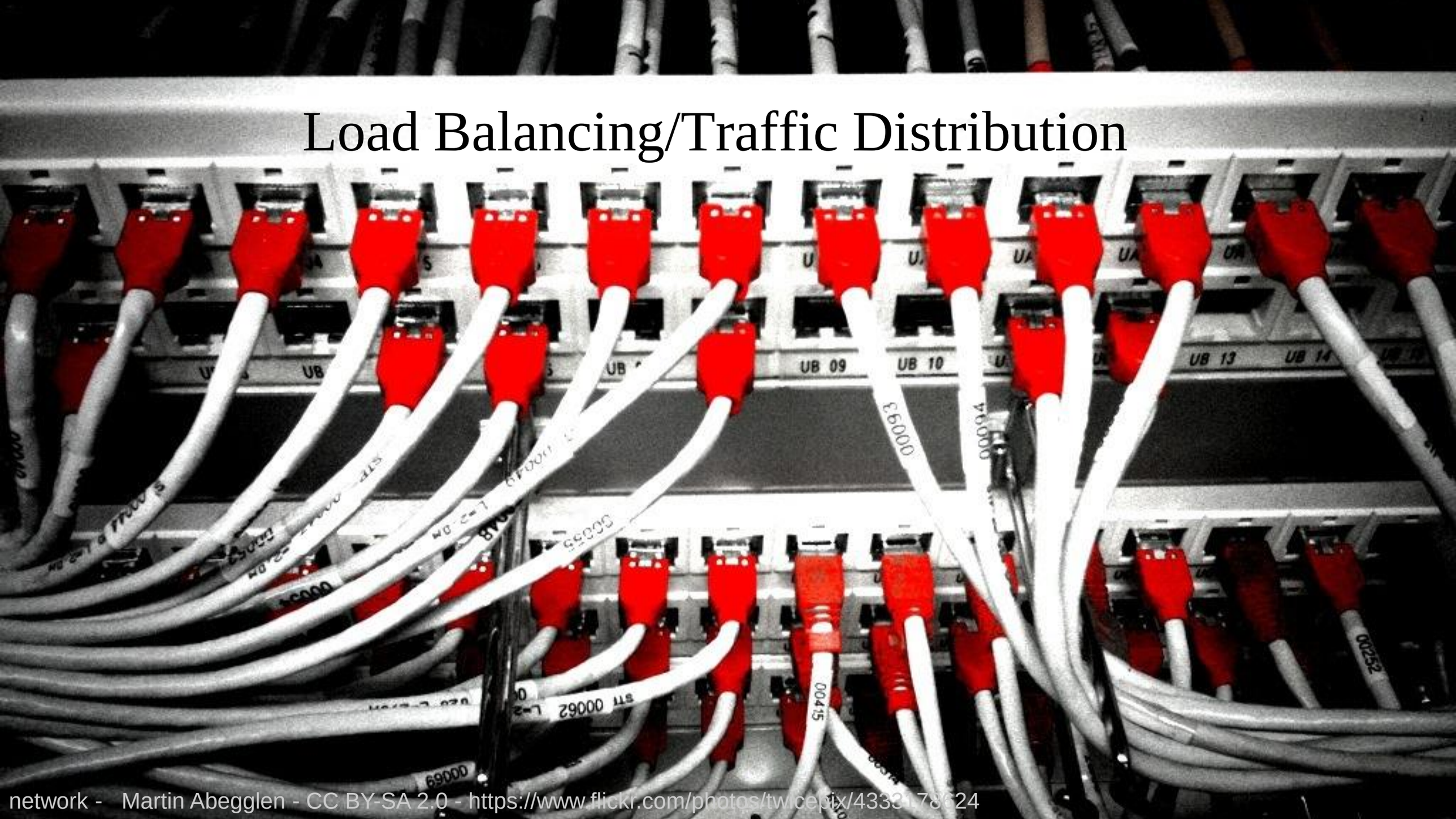
- As a DB (2.4)

```
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)
  - Worker is disabled and will not accept any requests.
- Stopped (S)
  - Worker is administratively stopped.
- 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.
- 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) stickysessionsep 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)
  - ttl 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.

I Don't Feel So Well

- No heath check capability
  - Requires real, live traffic
- Must come up with a way to work around it
- In the future...
  - Scratch your own itch, Daniel!

Mitigating Controls

- 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
https://127.0.0.1:8001			1	0	Init Ok	0	0	0	0	0
https://127.0.0.1:8002			1	0	Init Ok	0	0	0	0	0

Edit balancer settings for balancer://mycluster

LBmethod:	byrequests
Timeout:	0
Failover Attempts:	1
Disable Failover:	On ○ Off ●
Sticky Session:	(Use '-' to delete)
	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
https://127.0.0.1:8001			1	0	Init Ok	0	0	0	0	0
https://127.0.0.1:8002			1	0	Init Ok	0	0	0	0	0

Edit worker settings for https://127.0.0.1:8001

Load factor:	1			
LB Set:	0			
Route:				
Route Redirect:				
Status:	Ignore Errors	Draining Mode	Disabled	Hot Standby
	On ○ Off ●	On ○ Off ●	On ○ Off ●	On ○ Off ●
	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
- Persistance 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/appl
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

Learning the hard way - Ludovic Bertron -CC BY 2.0 - https://www.flickr.com/photos/23912576@N05/ -

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