Advanced Reverse Proxy Load Balancing in Apache HTTP Server 2.2

Jim Jagielski

http://www.jimjag.com/
jim@jaguNET.com
Whew

- That’s a mouthful
About me

• Longtime active contributor (July/Aug 1995)
• Been giving mod_proxy much TLC
• ASF Co-founder
• Other ASF titles as well
• CTO of Covalent Technologies
• Husband, father, all around nice guy
mod_proxy? Wazzat?

- An Apache module
- Implements core proxy capability
- Both forward and reverse proxy
- In general, most people use it for reverse proxy (gateway) functionality
Forward Proxy

- Intent is to “protect” internal clients
Reverse Proxy

- Intent is to “protect” internal Servers
How did we get here?

• A stroll down mod_proxy lane
  – First available in Apache 1.1
    • “Experimental Caching Proxy Server”
  – In Apache 1.2, pretty stable, but just HTTP/1.0
  – In Apache 1.3, much improved with added support for HTTP/1.1
  – In Apache 2.0, break out cache and proxy
What’s new/improved in 2.2

- Large file support
- Graceful stop
- mod_dbd
- mod_filter
- Better Debugging and info
- Caching
- Event MPM
- Authn/Authz
- **Proxy**
Goal for mod_proxy in 2.2
Goal for mod_proxy in 2.2

- Suck less ass
Proxy Improvements

• Becoming a robust but generic proxy implementation

• Support various protocols
  – HTTP, HTTPS, CONNECT, FTP
  – AJP, FastCGI (coming “soonish”)

• Load balancing

• Clustering, failover
AJP? Really?

- Yep, Apache can now talk AJP with Tomcat directly
- mod_proxy_ajp is the magic mojo
- Other proxy improvements make this even more exciting
- mod_jk alternative
But I like mod_jk

- That’s fine, but...
  - Now the config is much easier and more consistent
    - ProxyPass /servlets ajp://tc.example.com:8089
  - Easier when Apache needs to proxy both HTTP and AJP
  - Leverage improvements in proxy module
Load Balancer

- `mod_proxy_balancer.so`
- `mod_proxy` can do native load balancing
  - weight by actual requests
  - weight by traffic
  - `lbfactors`
- LB algo’s are impl as providers
  - easy to add
  - no core code changes required
Providers? Wazzat?

• New feature of Apache 2.x
• Originally used mostly in mod_dav
• Then in caching
• Now in other places too
  – authn / authz
  – mod_proxy
Providers... so what

• Think of providers as providing services
• modules implement providers and register them
• Other modules can then use those providers to implement that “service”
• Cool pizza delivery analogy
Why cool for mod_proxy?

• We mentioned that right now, we balance by traffic and requests
• But what if you want some other method
• You can add that capability with no core code changes to Apache.
• Very flexible
Load Balancer

• Backend connection pooling
  – Available for named workers:
    • eg: `ProxyPass /foo http://bar.example.com`
  – Reusable connection to origin
  – For threaded MPMs, can adjust size of pool (min, max, smax)
  – For prefork: singleton

• Shared data held in scoreboard
Pooling example
Pooling example

<Proxy balancer://foo>
BalancerMember http://www1.example.com:80/ loadfactor=1
BalancerMember http://www3.example.com:80/ loadfactor=1
BalancerMember http://www2.example.com:80/ loadfactor=4 status=+h
ProxySet lbmethod=bytraffic
</Proxy>
Pooling example

<Proxy balancer://foo>
  BalancerMember http://www1.example.com:80/ loadfactor=1
  BalancerMember http://www3.example.com:80/ loadfactor=1
  BalancerMember http://www2.example.com:80/ loadfactor=4 status=+h
  ProxySet lbmethod=bytraffic
</Proxy>

proxy: grabbed scoreboard slot 0 in child 371 for worker http://www1.example.com/
proxy: initialized single connection worker 0 in child 371 for (www1.example.com)
proxy: grabbed scoreboard slot 0 in child 369 for worker http://www1.example.com/
proxy: worker http://www1.example.com/ already initialized
proxy: grabbed scoreboard slot 0 in child 372 for worker http://www1.example.com/
proxy: worker http://www1.example.com/ already initialized
proxy: grabbed scoreboard slot 2 in child 371 for worker http://www3.example.com/
proxy: initialized single connection worker 2 in child 371 for (www3.example.com)
proxy: initialized single connection worker 0 in child 369 for (www1.example.com)
proxy: grabbed scoreboard slot 2 in child 369 for worker http://www3.example.com/
...
proxy: grabbed scoreboard slot 6 in child 369 for worker proxy:reverse
proxy: initialized single connection worker 6 in child 369 for (*)
proxy: grabbed scoreboard slot 6 in child 372 for worker proxy:reverse
proxy: worker proxy:reverse already initialized
proxy: grabbed scoreboard slot 1 in child 369 for worker http://www1.example.com/
proxy: initialized single connection worker 6 in child 372 for (*)
Workers and worker

- Don’t get too confused
- Both the worker MPM and the proxy balancer use the term “worker”
Load Balancer

- **Sticky session support**
  - aka “session affinity”
  - Cookie based
    - stickysession=PHPSESSID
    - stickysession=JSESSIONID
  - Natively easy with Tomcat
  - May require more setup for “simple” HTTP proxying
  - Do you really want/need it?
Load Balancer

• Cluster set with failover
  – Lump backend servers as numbered sets
  – balancer will try lower-valued sets first
  – If no workers are available, will try next set

• Hot standby
<Proxy balancer://foo>
  BalancerMember http://php1:8080/ loadfactor=1
  BalancerMember http://php2:8080/ loadfactor=4
  BalancerMember http://phpbkup:8080/ loadfactor=4 status=+h

  BalancerMember http://offsite1:8080/ lbset=1
  BalancerMember http://offsite2:8080/ lbset=1

  ProxySet lbmethod=bytraffic
</Proxy>

ProxyPass /apps/ balancer://foo/
Embedded Admin

- Allows for real-time
  - Monitoring of stats for each worker
  - Adjustment of balancer and worker params
    - lbset
    - lbmethod
    - route
    - enabled / disabled
    - ...

Leading the Wave of Open Source
Easy setup

<Location /balancer-manager>
  SetHandler balancer-manager
  Order Deny,Allow
  Deny from all
  Allow from 192.168.2.22
</Location>
Load Balancer Manager for localhost

Server Version: Apache/2.2.4-dev (Unix) mod_ssl/2.2.4-dev OpenSSL/0.9.8d DAV/2
Server Built: Nov 2 2006 12:16:28

LoadBalancer Status for **Balancer://foo**

<table>
<thead>
<tr>
<th>StickySession</th>
<th>Timeout</th>
<th>FailoverAttempts</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>byrequests</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Worker URL</th>
<th>Route</th>
<th>RouteRedir</th>
<th>Factor</th>
<th>Set</th>
<th>Status</th>
<th>Elected To</th>
<th>From</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://php1:8080/">http://php1:8080/</a></td>
<td>1</td>
<td>0</td>
<td>Ok</td>
<td>311</td>
<td>125K</td>
<td>446K</td>
<td></td>
</tr>
<tr>
<td><a href="http://php2:8080/">http://php2:8080/</a></td>
<td>4</td>
<td>1</td>
<td>Ok</td>
<td>1232</td>
<td>1322</td>
<td>433K</td>
<td>1743K</td>
</tr>
<tr>
<td><a href="http://phpbkup:8080/">http://phpbkup:8080/</a></td>
<td>4</td>
<td>0</td>
<td>Stby</td>
<td>Ok</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

**Edit balancer settings for balancer://foo**

StickySession Identifier:

Timeout: 0

Failover Attempts: 2

LB Method: bytraffic

Submit
Some tuning params

• For workers:
  – loadfactor
    • normalized load for worker [1]
  – lbset
    • worker cluster number [0]
  – retry
    • retry timeout, in seconds, for non-ready workers [60]
Some tuning params

- For workers - connection pool:
  - min
    - Initial number of connections [0]
  - max
    - Hard maximum number of connections [1|TPC]
  - smax:
    - soft max - keep this number available [max]
  - ttl
    - time to live for connections above smax
Some tuning params

• For workers (cont):
  – timeout
    • Connection timeout on backend [Timeout]
  – flushpackets *
    • Does proxy need to flush data with each chunk of data?
      – on : Yes
      – off : No
      – auto : wait and see
  – flushwait *
    • ms to wait for data before flushing
Some tuning params

• For workers (cont):
  – ping *
    • Ping backend to check for availability; value is time to wait for response
  – status (+/-)
    • D : disabled
    • S : Stopped
    • I : Ignore errors
    • H : Hot standby
    • E : Error
Some tuning params

- For balancers:
  - lbmethod
    - load balancing algo to use [byrequests]

- For both:
  - ProxySet
    - Alternate method to set various params

```bash
ProxySet balancer://foo timeout=10
... ProxyPass / balancer://foo timeout=10
```
Oh yeah

• **ProxyPassMatch**
  – ProxyPass can now take regex’s instead of just “paths”
    • ProxyPassMatch ^(/.*\.[^\s]*\.[^\s]*$) http://backend.example.com$1
  – JkMount migration

• **Shhhh**
  – ProxyPass ~ ^(/.*\.[^\s]*\.[^\s]*$) http://backend.example.com$1

• **mod_rewrite is balancer aware**
Gotcha

- ProxyPassReverse is NOT balancer aware!
- The below will not work:

```xml
<Proxy balancer://foo>
  BalancerMember http://php1:8080/ loadfactor=1
  BalancerMember http://php2:8080/ loadfactor=4
</Proxy>

ProxyPass /apps/ balancer://foo/

ProxyPassReverse /apps balancer://foo/
```
Workaround

• Instead, do this

```xml
<Proxy balancer://foo>
  BalancerMember http://php1:8080/ loadfactor=1
  BalancerMember http://php2:8080/ loadfactor=4
</Proxy>

ProxyPass /apps/ balancer://foo/

ProxyPassReverse /apps http://php1:8080/
ProxyPassReverse /apps http://php2:8080/
```
Putting it all together

```xml
<Proxy balancer://foo>
    BalancerMember http://php1:8080/     loadfactor=1
    BalancerMember http://php2:8080/     loadfactor=4
    BalancerMember http://phpbkup:8080/  loadfactor=4 status=+h
    ProxySet lbmethod=bytraffic
</Proxy>

<Proxy balancer://javaapps>
    BalancerMember ajp://tc1:8089/     loadfactor=1
    BalancerMember ajp://tc2:8089/     loadfactor=4
    ProxySet lbmethod=byrequests
</Proxy>

ProxyPass /apps/ balancer://foo/
ProxyPass /serv/ balancer://javaapps/
ProxyPass /images/ http://images:8080/
```
What’s on the horizon?

- Some additional potential backports
  - FastCGI proxy module
  - HTTP “ping” (OPTIONS *)
- More LB methods
- Enhancing ProxyPassReverse
- Better RFC compliance
- Something completely different...
Thanks!

• Q&A

• Resources:
  – http://httpd.apache.org/
  – dev@httpd.apache.org
  – A certain Open Source support provider