Apache HTTP Server
Load-Balancing with Apache HTTPD 2.2 and later

Erik Abele
www.eatc.de
About Me

• Working internationally as IT Consultant
• Areas: Administration & Operations
• Working on and with Open Source
  • Apache Software Foundation
  • Other projects (Ruby, Zenoss, ...)
• Living in South Germany
Content

• Basics
• Strategies
• Examples
• Questions
Basics
Why Load-Balancing?

- Optimizing Resource Utilization
- Maximizing Quality of Service
- Increasing Reliability & Availability
- Improving Security
- Lowering Administration Risks
Approaches

• DNS-Based Load-Balancing
• Round Robin
• Geographical Scheduling
• Layer 4 Load-Balancing
• Layer 7 Load-Balancing
• Hardware-/Software-Based Load-Balancers
DNS-Based Balancing

- Clients resolve FQDN
- Response with several IPs or chosen randomly or e.g. according to proximity
Layer 4 Balancing

- Transport Layer
- Deals with TCP connections
- Routes pure packets
- NAT with port and transaction awareness
Layer 7 Balancing

- Application Layer
- Deals with HTTP requests
- Forwards whole requests
- Gateway / Reverse Proxy
Layer 4-7 Balancing

- Combination: routing packets but aware of HTTP content
- Load-Balancer acts as a multilayer switch
- Basic schema always the same
Hardware-Based LBs

- Mostly real routers/switches with ML capabilities
- **Pros**
  - Very fast
- **Cons**
  - Expensive
  - Proprietary
- **Examples:** Cisco LoadDirector, F5 Big/ip, Barracuda
Software-Based LBs

- Software-based appliances or just packages
- Pros
  - Cheaper and also available as Open Source
  - Simpler to configure and often more flexible
- Cons
  - Not as fast as hardware based solutions
- Examples: Pen, Pound, Squid, Linux VS, Zeus ZXTM
Balancer Features 1

- TCP Offloading
- TCP Buffering
- SSL Offloading & Acceleration
- HTTP Caching
- HTTP Compression
- Client Authentication
- Attack Protection & Firewalling
Balancer Features 2

- Content-Aware Load-Balancing
- Request Rate Shaping
- Bandwidth Shaping
- Traffic Valuation & Prioritization
- Programmatic Traffic Manipulation
Balancing Algorithms

- Random Choice
- Round Robin
- Weighted Round Robin
  - Request counting algorithm
  - Weighted traffic counting algorithm
- Based on other factors (load, response time, ...)

Balancing Algorithms
Strategies
Gateway

- “Reverse Proxy” - the frontend:
- Apache HTTPD with mod_proxy and mod_proxy_balancer as well as one of:
  - mod_proxy_http
  - mod_proxy_ajp
Application Servers

- A pool of servers - the backend:
  - Apache HTTP with PHP, Perl, ...
  - Apache Tomcat, JBoss, Glassfish, ...
  - Zope, CherryPy, Mongrel, WebRick
  - ...

Focus: Scalability

- A single load-balancer in front of a pool of application servers
Persistence

- “Session Stickiness”

- A problem when not able to use a shared data store for the application (e.g. when using multiple physical locations where replication becomes problematic)

- Alternative: using cookies as data store
Focus: Resilience

- Two or more load-balancers in front of a pool of application servers
Failover

- Active/Active
  - Both load-balancers are actively serving requests
- Active/Passive
  - Only one load-balancer is actively serving requests, the other one takes over in case of failure
- There needs to be a way to detect failures
Focus: Globalization

- Several groups of load-balancers in more than one physical location
Location Selection

- DNS-Based
- Round Robin
- Based on geographical proximity
- Based on utilization of location
- FQDN-Based (e.g. wwwN.example.com)
Examples
HTTPD as Gateway

• Required Modules:
  • mod_proxy
  • mod_proxy_balancer
  • mod_proxy_http / mod_proxy_ajp
  • mod_proxy_ftp
HTTP Backends

• Requires extension module: mod_proxy_http

• Usage:
  BalancerMember http://1.1.1.1:8080 ...
AJP Backends

- Requires extension module: `mod_proxy_ajp`
- Usage:
  `BalancerMember ajp://1.1.1.1: 8009 ...`
Basic Configuration

<VirtualHost *:80>
  ServerName www.example.com
  ...
  ProxyRequests Off
  ProxyVia Off
  ProxyErrorOverride On
  ProxyPreserveHost On
  ProxyTimeout 30

  ProxyPass / balancer://app_farm lbmethod=byrequests maxattempts=3 \
            nofailover=Off stickysession=SID

  <Proxy balancer://app_farm>
    BalancerMember http://1.1.1.1:8080 smax=15 max=50 lbfactor=2
    BalancerMember http://1.1.1.2:8080 smax=15 max=50 lbfactor=2
    BalancerMember http://1.1.1.3:8080 smax=10 max=25 lbfactor=1
  </Proxy>
</VirtualHost>
Connection Pooling

- **min**
  Minimum number of connections that will always be open to the backend server

- **max**
  Hard Maximum number of connections that will be allowed to the backend server

- **smax & ttl**
  Soft Maximum number of connections that will be created on demand, TTL for connect’s above smax
Parameters 1

- **lbmethod**
  The load-balancing scheduler method to use, e.g.
  - weighted request counting
  - weighted traffic counting
- **lbset**
  Assigns a specific cluster set to members
Parameters 2

- loadfactor
  Defines the normalized weighted load applied to this balancer member, e.g.
  - loadfactor=1
  - loadfactor=2 <- twice as much req’s / IO

- status
  Defines the initial state of this member
<VirtualHost *:80>
    ServerName www.example.com
    ...

    ProxyPass / balancer://app_farm lbmethod=byrequests maxattempts=3 \
       nofailover=Off stickysession=SID

    <Proxy balancer://app_farm>
        BalancerMember http://1.1.1.1:8080 ...
        BalancerMember http://1.1.1.2:8080 ...
        BalancerMember http://1.1.1.3:8080 ... status=+H
    </Proxy>
</VirtualHost>
Sticky Sessions

- Supported by stickysession flag:
  stickysession=PHPSESSID
  stickysession=JSESSIONID|jsessionid
Environment Variables

- Generated by mod_proxy
  - X-Forwarded-For
  - X-Forwarded-Host
  - X-Forwarded-Server

- Influence mod_proxy
  - SetEnv force-proxy-request-1.0 1
  - SetEnv proxy-nokeepalive 1
ProxyPass / balancer://app_farm ...

ProxyPassReverse / http://1.1.1.1:8080/
ProxyPassReverse / http://1.1.1.2:8080/

<Proxy balancer://app_farm>
 BalancerMember http://1.1.1.1:8080 ...
 BalancerMember http://1.1.1.2:8080 ...
</Proxy>
URL Rewriting

ProxyPass ~ ^/+(.*)$ balancer://app_farm/$1 ...

[or]

ProxyPassMatch ^/+(.*)$ balancer://app_farm/$1 ...

[or]

RewriteEngine On

RewriteCond %{REQUEST_URI} ^/.balancer
RewriteRule ^(.*)$ - [L]

RewriteRule ^/+(.*)$ balancer://app_farm/$1 [P,L]

ProxySet balancer://app_farm lbmethod=bytraffic
Using Headers

<VirtualHost *:443>
  ServerName www.example.com
  ...
  RequestHeader set Front-End-Https "On"
  ...
  ProxyPass / balancer://app_farm lbmethod=byrequests maxattempts=3 \ 
                 nofailover=Off stickysession=SID

  <Proxy balancer://app_farm>
   BalancerMember http://1.1.1.1:8080 smax=15 max=50 lbfactor=2
  ...
  </Proxy>
</VirtualHost>
Filtering Content

• mod_mime, e.g.
  AddOutputFilter INCLUDES .shtml

• mod_filter

• mod_proxy_html
We can use mod_deflate as usually:
AddOutputFilterByType DEFLATE text/html
Offloading SSL

- Simply configure SSL on the frontend server as you would do without a balancer.
Caching

<VirtualHost *:80>
    ServerName www.example.com
    ...
    CacheEnable mem /
    CacheEnable disk /
    CacheDefaultExpire 1800
    CacheMaxExpire 3600
    ...

    ProxyPass / balancer://app_farm lbmethod=byrequests maxattempts=3 \
   nofailover=Off stickysession=SID

    <Proxy balancer://app_farm>
       BalancerMember http://1.1.1.1:8080 smax=15 max=50 lbfactor=2
        ...
    </Proxy>
</VirtualHost>
Failover

- Automatically done
- if a member is in-error the next one is tried
- Can be switched off if backend servers do not support session replication
Balancer Management

- Requires mod_status
- Allows for real-time monitoring
- Allows for dynamic updates of parameters of balancer members
  - enable/disable balancer members
  - specify balance factors, methods, sets
### Load Balancer Manager for localhost

Server Version: Apache/2.2.8 (Unix)
Server Built: Apr 9 2008 14:12:12

#### LoadBalancer Status for balancer://app_cluster

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

<table>
<thead>
<tr>
<th>Worker URL</th>
<th>Route</th>
<th>RouteRedir</th>
<th>Factor</th>
<th>Set Status</th>
<th>Elected To</th>
<th>From</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://192.168.1.1">http://192.168.1.1</a></td>
<td>10</td>
<td>0</td>
<td>Ok</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><a href="http://192.168.1.2">http://192.168.1.2</a></td>
<td>10</td>
<td>0</td>
<td>Ok</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

#### LoadBalancer Status for balancer://app_farm

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

#### Load worker settings for http://192.168.1.1

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Load factor:</td>
<td>10</td>
</tr>
<tr>
<td>LB Set:</td>
<td>0</td>
</tr>
<tr>
<td>Route:</td>
<td></td>
</tr>
<tr>
<td>Route Redirect:</td>
<td></td>
</tr>
<tr>
<td>Status:</td>
<td>Disabled:</td>
</tr>
</tbody>
</table>

Submit
Example Configuration

```
<Location /.balancer>
  SetHandler balancer-manager

  AuthType Digest
  AuthName "Balancer Manager"
  AuthDigestDomain /.balancer
  AuthDigestProvider file
  AuthUserFile /www/etc/httpd/balancer.passwd

  Require user operations
</Location>

...

ProxyPass /.balancer !
ProxyPass / balancer://app_farm ...
```
Questions?