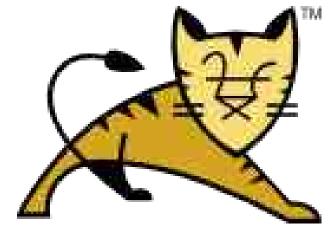


Intro to Load-Balancing Tomcat with httpd and mod_jk



Christopher Schultz
Chief Technology Officer
Total Child Health, Inc.

* Slides available on the Linux Foundation / ApacheCon2015 web site and at http://people.apache.org/~schultz/ApacheCon NA 2015/Load-balancing with mod_jk.odp

Intro to Load-Balancing Tomcat with httpd and mod_jk

- Covering
 - Load balancing
- Not covering
 - Clustering*

^{*} See Mark's 3-part presentation(s) today starting at 10:00 in this room

Tomcat



- Tomcat as a web server
 - Capable
 - HTTP, HTTPS, WebSocket, NIO
 - Virtual hosting, CGI, URL-rewriting
 - Authentication (RDBMS, LDAP, file)
 - Styled directory listings
 - Arbitrary data filtering
 - Fast
 - Static throughput can be comparable to httpd*

^{*} See Jean-Frederic's presentation today at 15:15 in this room

Tomcat



- Tomcat as an application server
 - Java servlets, JSPs
 - Sky is the limit



Image credit: Stan Shebs CC BY-SA 3.0 via Wikimedia Commons

Tomcat



- Tomcat as an application server
 - Java servlets, JSPs
 - Sky is the limit*



Image credit: Stan Shebs CC BY-SA 3.0 via Wikimedia Commons

^{*} Okay, heap size is the limit



More memory

- More memory
- More deployed applications
 - without complex URLs





- More memory
- More deployed applications
 - without complex URLs
- Better fault-tolerance
 - fail-over









- More memory
- More deployed applications
 - without complex URLs
- Better fault-tolerance
 - fail-over
- Easier maintenance
 - bring-down a server without bringing down a service

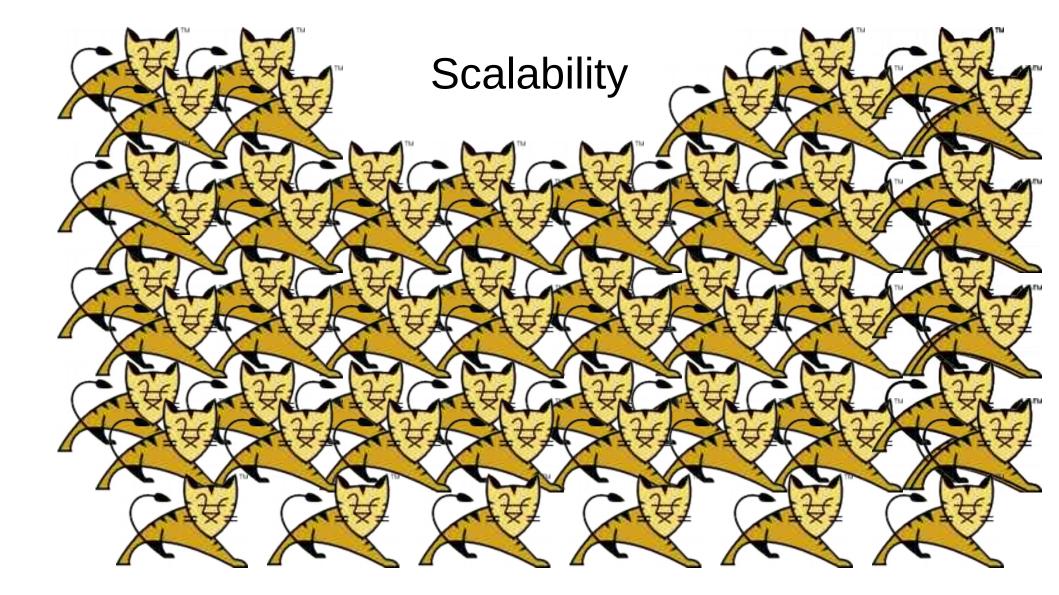












Load Balancing



- Client sees a single "service"
- "Server" is really an army of servers
- This army runs behind a façade: the load-balancer (lb)
- The load-balancer is also called a reverse proxy*

^{*} Because *forward proxy* was already taken

Balancing versus Clustering

- Balancing is basic
 - Route incoming requests
 - Pushes bytes around
- *Clustering** is complex
 - Relies on balancing as a building block
 - Configuration
 - Communication
 - Replication

^{*} See Mark's 3-part presentation(s) today starting at 10:00 in this room

Reverse Proxying

- Necessary components
 - 1. Reverse proxy (or proxies) (lb)
 - 2. Proxied / balanced back-end nodes (servers)
 - 3. A protocol to connect the two
 - HTTP(S)/1.1
 - AJP/13 (Apache JServ Protocol 1.3)

Reverse Proxying

- Choice of load-balancers
 - Hardware
 - F5/BIGIP, Cisco, Barracuda, etc.
 - Software
 - Apache httpd
 - lighttpd
 - NGINX
 - Squid
 - Varnish

Proxy Protocols

- HTTP
 - Easy to configure
 - Easy to debug
 - Supports TLS delivery (HTTPS)
 - Wide support

Proxy Protocols

- Apache JServ Protocol
 - Binary protocol that tunnels HTTP
 - Designed to forward SSL client state to the back-end node
 - Uses mnemonics for often-used headers, etc. offers a kind of compression to improve performance

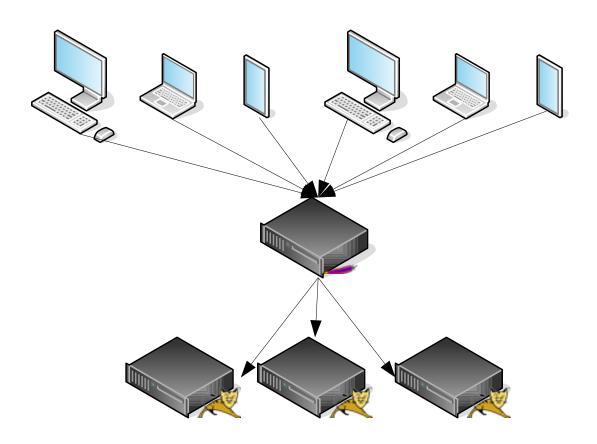
^{*} http://tomcat.apache.org/connectors-doc/ajp/ajpv13a.html

Apache httpd



- Using HTTP
 - mod_proxy_http
- Using AJP13
 - mod_proxy_ajp
 - mod_jk

Reverse Proxying



mod_jk

- Longer history than mod_proxy_ajp
- More expressive configuration, more options
- Default configuration does more
- Not a default module in any httpd version :(



Configuring mod_jk

- Workers
 - Definition of a connection
 - One worker per Tomcat instance
 - Building block for other configuration
 - Used to map requests to a particular place
- Mounts
 - Associate a URL pattern with a worker

workers.properties

```
worker.list=myworker
worker.myworker.host=localhost
worker.myworker.port=8009
worker.myworker.type=ajp13
```

httpd.conf

```
JkMount /examples/* myworker
```

Tomcat's conf/server.xml

```
<Connector port="8009"
protocol="AJP/1.3" />
```

- Most of this is default configuration
 - Tomcat's default server.xml
 - AJP connector on port 8009
 - mod_jk's default worker
 - host=localhost
 - port=8009
 - type=ajp13

Point a client at http://host/examples/



Load-balancing examples

Small changes to workers.properties

```
worker.list=1b
worker.lb.type=lb
worker.lb.balance workers=myworker, other
worker.myworker.host=localhost
worker.myworker.port=8009
worker.myworker.type=ajp13
worker.other.host=otherhost
worker.other.port=8009
worker.other.type=ajp13
```

Load Balancing examples

Small change to httpd.conf
 JkMount /examples/* lb

Load-balancing examples

- Deploy examples webapp to "other" server
- All is well

Load-balancing examples

- Deploy examples webapp to "other" server
- All is well

... until you try to run the "Sessions Example"

Session Tracking

Sessions

- Maintained using cookie or URL parameter
- Tied to a single back-end node
- Load-balancer needs to know which node to use

Session Tracking Techniques

- No session tracking
 - Complete chaos
- Allow nodes to negotiate
 - Clustering

Session Tracking Techniques

- "Sticky" sessions
 - In-memory registry
 - Doesn't scale well
 - Can get out of sync
 - Another cookie
 - NODE=node01
 - Can get out of sync
 - Encode node identity in the session id

Sticky Sessions in mod_jk

- Sticky sessions are the mod jk default!
- Must tell Tomcat about it's role
 - Small change to Tomcat configuration

```
<Engine name="Catalina"
    defaultHost="localhost"
    jvmRoute="myworker">
```

Configuration for second node:

```
<Engine name="Catalina"
    defaultHost="localhost"
    jvmRoute="other">
```

Load Balancing examples

• Sessions example is feeling much better, now

Sessions	Example
Session II	D: C4B83B3342D0C07CF4C296109E8874ED.othe
Created: 5	Sat Mar 21 08:16:18 EDT 2015
Last Acce	ssed: Sat Mar 21 08:16:22 EDT 2015
The follo	wing data is in your session:
=	
Name of	Session Attribute:
Value of	Session Attribute:
Submit Qu	erv

Load Balancing with mod_jk

- Define workers
 - Individual or balanced
- Map URLs to workers
 - Lots of options
- Configure Tomcat
 - Don't forget to set jvmRoute if you'll be using (sticky) sessions

Monitoring mod_jk

- How is mod_jk feeling*?
- Are the workers all working?
- What does the load distribution look like?
- Are there any failures?

^{*} Come to my presentation at 14:15 today for monitoring Tomcat itself.

- How is mod_jk feeling?
- Are the workers all working?
- What does the load distribution look like?
- Are there any failures?

mod_jk has a special status worker

Configure the status worker

```
worker.list=status*
worker.status.type=status
```

Mount the worker on a URL

JkMount /jk-status status

^{*} The worker.list directive can be specified multiple times



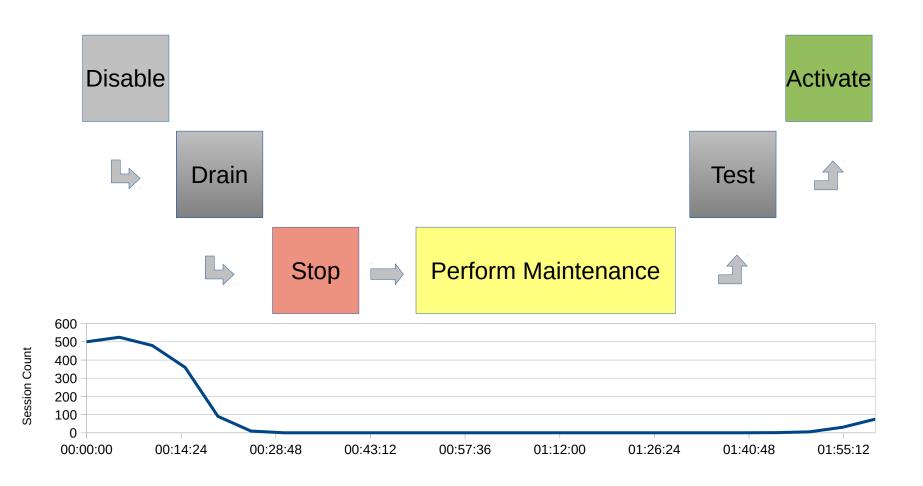
• Also snoop on load-balancer members

Balance	Member	s [Hide	1																					
Name	Туре Н	Iostnam	ne A	dd	ress	:Por	t Source						Prepos Timeou						100	Max Pack Size	cet [Hide]			
mywork	er ajp13 lo	calhost	12	27.0	0.0.1	:821	5 undefin	cd 0			0	į.	0	0		2	9	0	0	65536				
other	ajp13 lo	calhost	13	27.0	0.0.1	:822	5 undefin	cđ 0			0	ý	0	0		2	Ü	0	0	8192				
	Name	Act S	tate	D	FM	v	Acc	Sess	Err	CE	RE	,	Wr		Rd		Busy	MaxBus	y Con	MaxCon	Route	RR Cd Rs	LR L	E
[SIEIR]	myworker	ACT	K	0	1 1	0 1	2 (0/sec)	1 (0/sec)	0	0	0 7	2K	(6 /sec)	18K	(15)	(sec)	0	1	2	2 :	myworker	0/0	1159	
[SIEIR]	other	ACT C	K	0	1 1	0.2	5 (0/sec)	0 (0/sec)	0	0	0 1	6K ((13 /sec)	33K	(28)	(sec)	0	1	2	2 0	other	0/0	1159	

Node Maintenance

- Crash
- Application upgrade
- System / package upgrade
- DR testing

Node Maintenance



	Name	Act 5	State 1	D F	M	v	Acc	Sess	Err	CE I	RE	Wr	1	Rd	Busy	MaxBusy	Con	MaxCon	Route	RR Cd Rs	LR LE
ISER!	myworker	ACT (OK (0 1	1	0	12 (0/sec)	1 (0/sec)	0	0 0	1	7.2K (6 /sec)	18K (15 /sec)	0	1	2	2	myworker	0/0 1	159
ISIER1	other	ACT (OK (0 1	1	0	26 (0/sec) (0 (0/sec)	0	0 0	1	16K (13 /sec)	33K (28 /sec)	0	1	2	2	other	0/0 1	159

Edit worker	settings for mywo	orker	
Balancin	g related settings	AJP setti	ngs
Activation:		Hostname:	localhost
Active	0	Port:	8215
Disabled	0	Connection Pool Timeout:	0
Stopped	0	Ping Timcout:	10000

- New clients are sent to active nodes
- Existing client sessions continue to be valid
- Disabled node continues to serve these clients
- Usage profile means draining can take a long time

- Some clients keep coming back
- Session tracking strategy strikes again!
 - Client is assigned to myworker node; session times out
 - Node myworker is disabled
 - Client does not close browser
 - Client visits your service with old session cookie value
 - Cookie still ties the client to the disabled server
 - mod jk doesn't know any better

How do we get these clients to stop coming back?



- How do we get these clients to stop coming back?
- LoadBalancerDrainingFilter / LoadBalancerDrainingValve

```
<filter>
    <filter-name>loadBalancerDrainingFilter</filter-name>
        <filter-class>LoadBalancerDrainingFilter</filter-class>
</filter>
<filter-mapping>
        <filter-name>loadBalancerDrainingFilter</filter-name>
        <url-pattern>/*</url-pattern>
</filter-mapping>
```

- Client sends session cookie to server
- mod jk respects session hint, sends worker attribute ACTIVATION=DIS
- LoadBalancerDrainingFilter
 - sees invalid session
 - sees ACTIVATION=DIS
 - strips jsessionid,
 - expires cookie
 - redirects client to same URL
 - mod_jk chooses an active node

Node Maintenance - Stop

	Name	Act St	ate I	F	M V	Acc	Sess	Err	CER	Œ	Wr		Rd	Busy	MaxBusy	Con	MaxCon	Route	RR Cd Rs I	RI	LE
ISERI :	myworker	ACT OF	(0	1	0	12 (0/sec)	1 (0/sec)	0	0 0		7.2K (6/sec)	18K	(15 /sec)	0	1	2	2	myworker	0/0 1	159	
[SIEIR]	other	ACT OF	(0	1	0	26 (0/sec)	0 (0/sec)	0	0 0		16K (13 /sec)	33K	(28 /sec)	0	1	2	2	other	0/0 1	159	

Edit worker	settings for mywo	orker	
Balancin	g related settings	AJP setti	ngs
Activation:		Hostname:	localhost
Active	0	Port:	8215
Disabled	0	Connection Pool Timeout:	0
Stopped	0	Ping Timeout:	10000

Node Maintenance - Test

- Test the upgraded web application
- How do we access the target node?
 - Bypass load balancer (mynode.domain.ext)
 - Through load balancer (www.domain.ext)
 - http://www.domain.ext/examples/;jsessionid=00.myworker
- Target node is disabled

Node Maintenance - Test

```
<filter>
  [...]
  <init-param>
    <param-name>ignore-cookie-name/param-name>
    <param-value>lbdf.ignore</param-value>
  </init-param>
  <init-param>
    <param-name>ignore-cookie-value</param-name>
    <param-value>true</param-value>
  </init-param>
</filter>
```

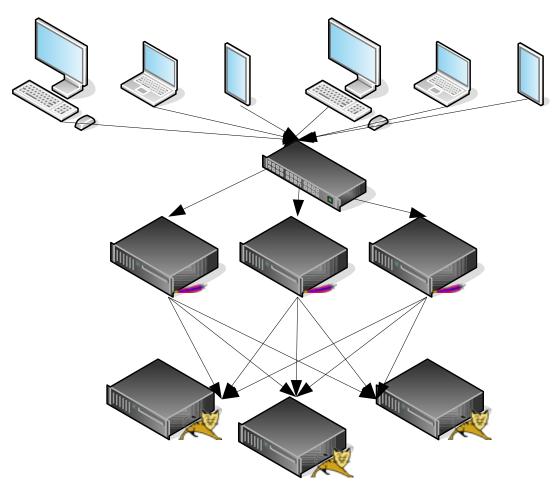
Node Maintenance - Test

- Use browser dev tools to create cookie
 - lbdf.ignore=true
- mod jk respects session hint
- LoadBalancerDrainingFilter
 - sees invalid session
 - sees ACTIVATION=DIS
 - sees valid "ignore" cookie
 - allows access to the disabled node

	Name	Act St	ate I	F	M V	Acc	Sess	Err	CER	Œ	Wr		Rd	Busy	MaxBusy	Con	MaxCon	Route	RR Cd Rs I	RI	LE
ISERI :	myworker	ACT OF	(0	1	0	12 (0/sec)	1 (0/sec)	0	0 0		7.2K (6/sec)	18K	(15 /sec)	0	1	2	2	myworker	0/0 1	159	
[SIEIR]	other	ACT OF	(0	1	0	26 (0/sec)	0 (0/sec)	0	0 0		16K (13 /sec)	33K	(28 /sec)	0	1	2	2	other	0/0 1	159	

Edit worker	settings for mywo	orker	
Balancin	g related settings	AJP setti	ngs
Activation:		Hostname:	localhost
Active	0	Port:	8215
Disabled	0	Connection Pool Timeout:	0
Stopped	0	Ping Timeout:	10000

Reverse Proxying



- N web servers
 - T web server threads (or processes)
- M Tomcat servers
- Web servers must be prepared

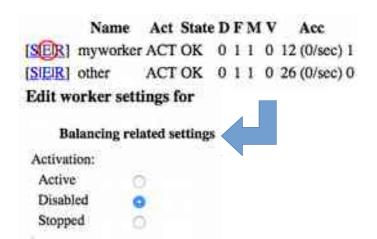
- N web servers
 - T web server threads (or processes)
- M Tomcat servers
- Web servers must be prepared
 - T * M connections

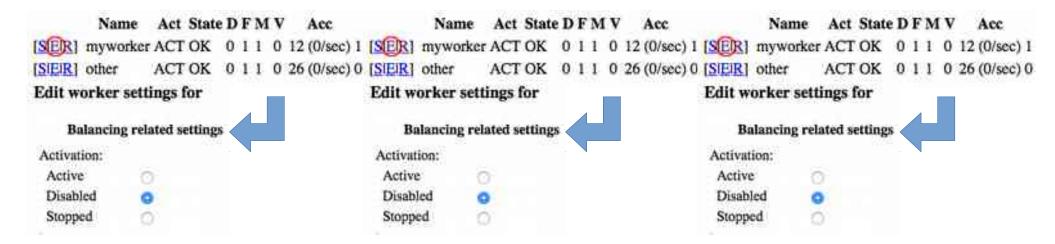
- N web servers
 - T web server threads (or processes)
- M Tomcat servers
- Web servers must be prepared
 - T * M connections
- Tomcat nodes must be prepared

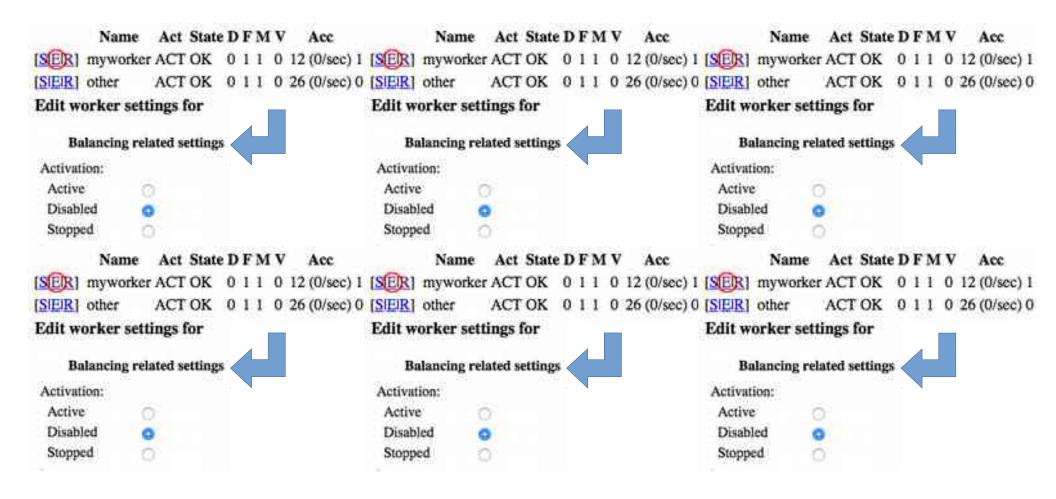
- N web servers
 - T web server threads (or processes)
- M Tomcat servers
- Web servers must be prepared
 - T * M connections
- Tomcat nodes must be prepared
 - N * T connections
- 3 * 256 = 768 connections

- Resource exhaustion
 - Threads (processes)
 - File handles

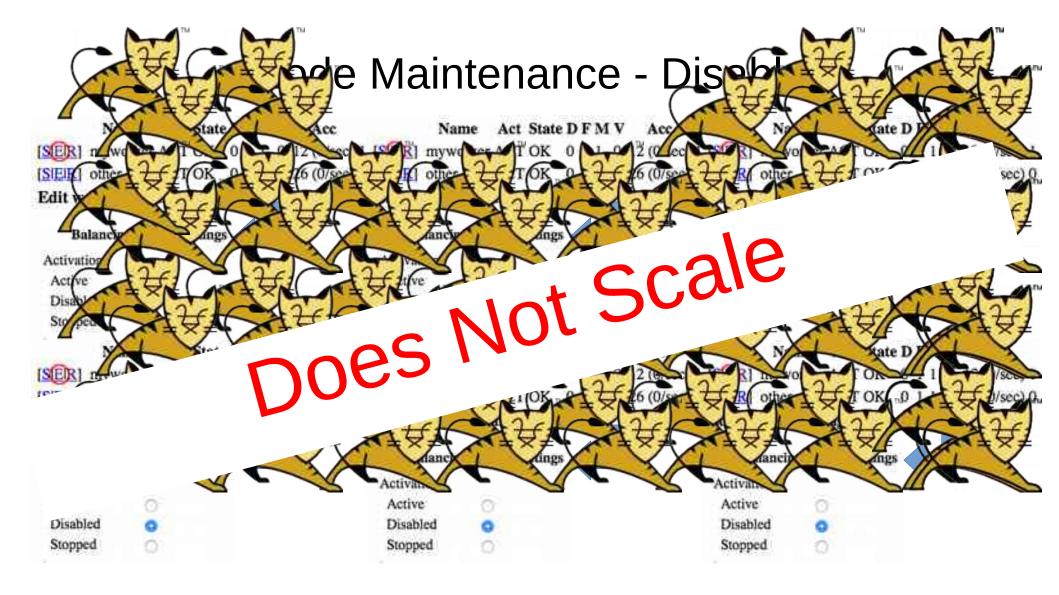
- Resource exhaustion
 - Threads (processes)
 - File handles
- Resource Management
 - httpd
 - Use event/worker/NT MPM with limited mod jk connection pool size
 - Pre-fork will always use MaxClients[2.2]/MaxRequestWorkers[2.4]
 - Tomcat
 - Use the Tomcat NIO or NIO2 connector











• Script this!

```
$ mod_jk.py -b lb -w myworker -u activation=DIS
+ Updating localhost
   Updating load-balancer lb worker myworker
+ localhost (mod_jk/1.2.41-dev)
- lb
   - myworker
   activation=DIS
```

mod_jk.py can be found at https://wiki.apache.org/tomcat/tools/mod_jk.py

Multi-web server example

```
$ mod_jk.py -b lb -w myworker -u activation=ACT
+ Updating web-1
   Updating load-balancer lb worker myworker
+ Updating web-2
   Updating load-balancer lb worker myworker
+ Updating web-3
   Updating load-balancer lb worker myworker
[...]
```

Resources

- LoadBalancerDrainingFilter
 - http://people.apache.org/~schultz/lbdf/
- LoadBalancerDrainingValve
 - Landing in trunk, soon
- mod_jk.py
 - https://wiki.apache.org/tomcat/tools/mod_jk.py

Questions

^{*} Slides available on the Linux Foundation / ApacheCon2015 web site and at http://people.apache.org/~schultz/ApacheCon NA 2015/Load-balancing with mod_jk.odp