PRESENTED BY RED HAT

LEARN. NETWORK.
EXPERIENCE OPEN SOURCE.

www.theredhatsummit.com

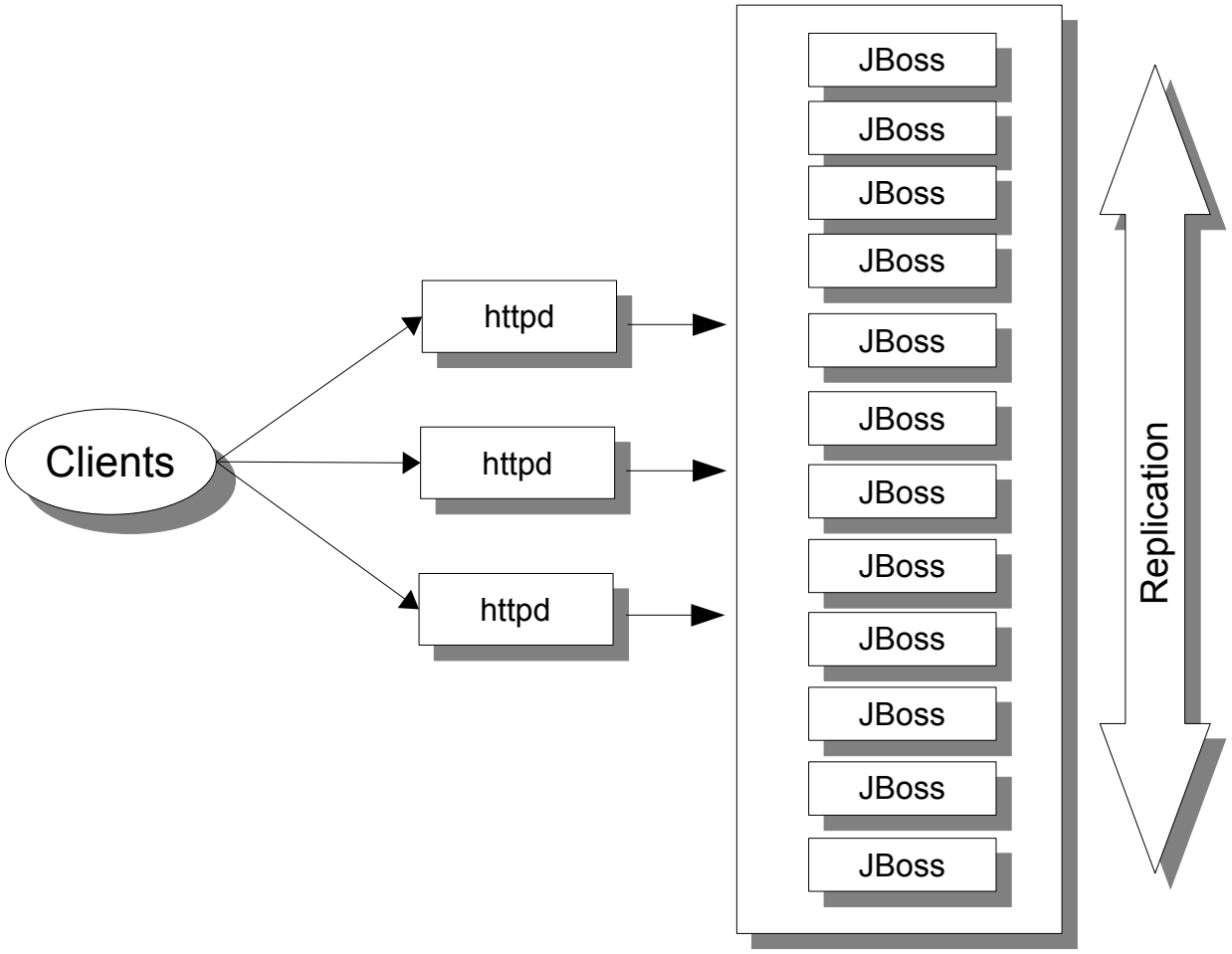
Dynamic clusters with Apache httpd, mod-cluster and JBoss

Bela Ban, Paul Ferraro
Software Engineers, JBoss
June 2010

Agenda

- Scenario:
  - Clustered web applications
  - Apache mod-jk versus mod-cluster
  - Large cluster of JBoss application servers
  - Large number of clients
- Issues
  - Rolling upgrades and static configuration
- mod-cluster
- Demo (mod-cluster domains in a cloud)

Typical mod_jk based setup



mod_jk configuration (httpd side)

worker.properties:

```
worker.jboss1.host=jboss1
worker.jboss2.host=jboss2
worker.jboss3.host=jboss3
…
worker.loadbalancer.type=lb
worker.loadbalancer.balance_workers=jboss1,jboss2,jboss3
```

uriworkermap.properties:

```
/jmx-console/*=loadbalancer
/web-console/*=loadbalancer
/mywebapp/*=loadbalancer
/mynewapp/*=loadbalancer
```


Issues

- #1 Large flat cluster
  - Incompatible upgrades are impossible
    - New software (JBoss, 3rd party, application)
  - Failures (e.g. network partitions) affect many nodes
- #2 Static configuration
  - Config has to be modified on all httpd servers when
    - a host is added or removed
    - a web app is deployed or undeployed
  - Not good in a cloud environment where hosts are dynamically added and removed

Problem #1: large flat clusters

- Most of the down time is not caused by crashes, but by upgrades
- If a new release is binary incompatible, the restarted node won't be able to participate in the cluster
- Incompatibilities are caused by
  - New JBoss version, e.g. 4.2 → 4.3
  - Individual component upgrades (JGroups, Infinispan)
  - Application incompatibilities
  - DB schema changes

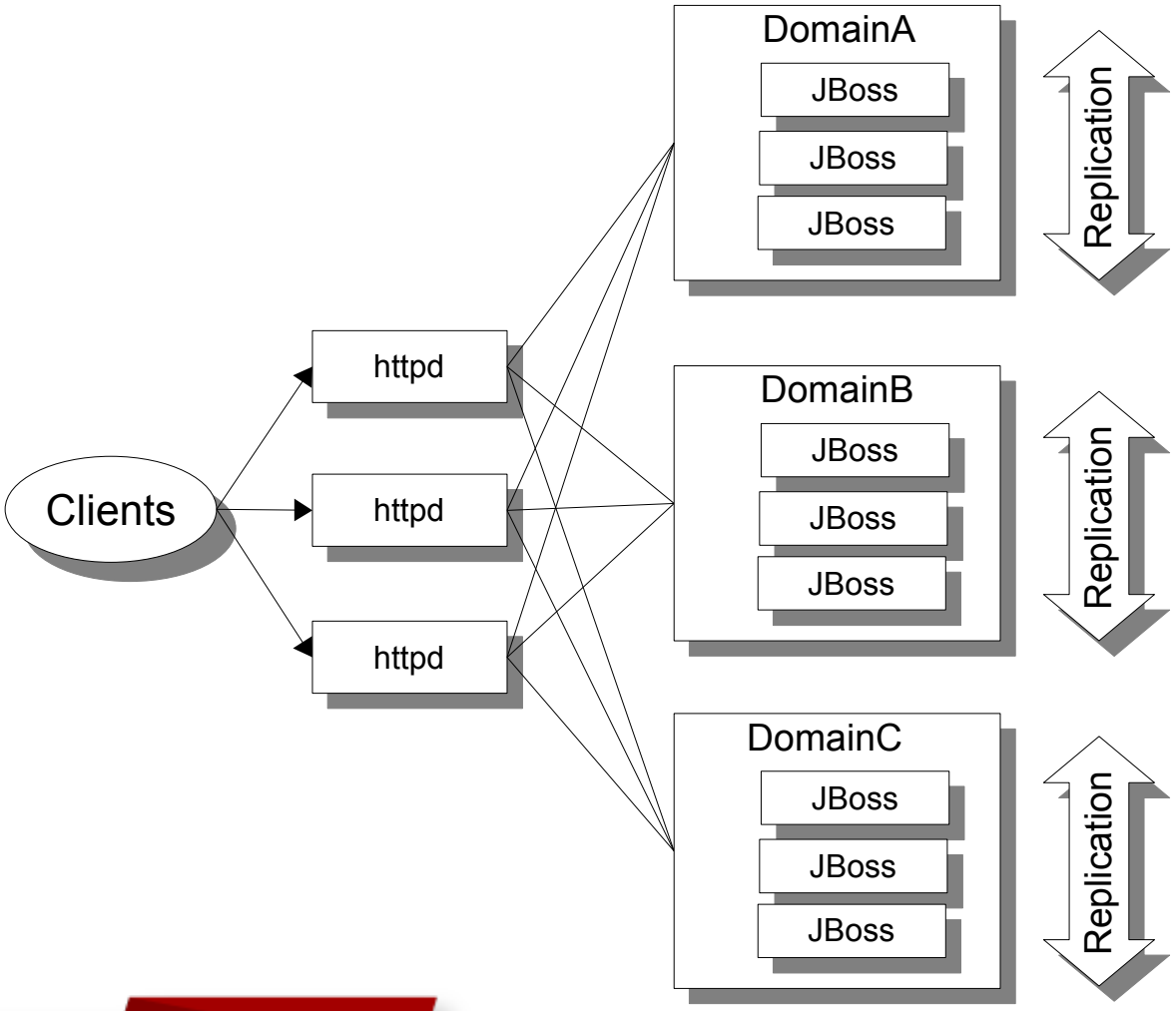
Issues with large flat clusters

- A problem potentially affects many nodes
  - State transfer, rebalancing of state (Infinispan's DIST mode), merge handling
  - RPC's across the cluster might block until a node is marked as crashed
- State
  - We cannot use total replication for scalability reasons
- Communication
  - TCP becomes an issue as a message is sent N-1 times

Solution: mod-jk domains

- Instead of a 1000 node cluster, create 10 *domains* of 100 nodes each. A domain == a cluster
- httpd creates a new session S on a random node N in a random domain D
  - All requests for S go to N in D (if sticky sessions are configured)
  - If N is shut down or crashes, S is directed to another node *within* D
  - Changes to S are replicated only *within* D
  - If there are no more nodes in D, S is lost
    - No replication between domains, a domain is a JBoss cluster

mod-jk domains



Rolling upgrades with domains

- An *entire domain* is upgraded, not an individual node
- Steps
  - *Disable* entire domain (mod-jk /status app on httpd)
  - httpd won't create new sessions on disabled workers
  - Requests for existing sessions will still be forwarded to disabled workers
  - When all sessions on of a given domain have expired we can shut down all workers, upgrade and restart
  - httpd can now create new sessions in the upgraded domain
    - Different domains can have different software versions !

mod_jk domain configuration (httpd side)

worker.properties:

```
worker.jboss1.host=jboss1
worker.jboss1.domain=A
worker.jboss2.host=jboss2
worker.jboss2.domain=A
worker.jboss3.host=jboss3
worker.jboss3.domain=B
worker.jboss4.host=jboss4
worker.jboss4.domain=B
```

- Make sure you separate cluster traffic for A and B !

```
jboss1: ./run.sh -c all -g A -u 232.1.1.1 -m 7500 -Djboss.jvmRoute=jboss1
jboss2: ./run.sh -c all -g A -u 232.1.1.1 -m 7500 -Djboss.jvmRoute=jboss2
jboss3: ./run.sh -c all -g B -u 232.2.2.2 -m 8500 -Djboss.jvmRoute=jboss3
jboss4: ./run.sh -c all -g B -u 232.2.2.2 -m 8500 -Djboss.jvmRoute=jboss4
```

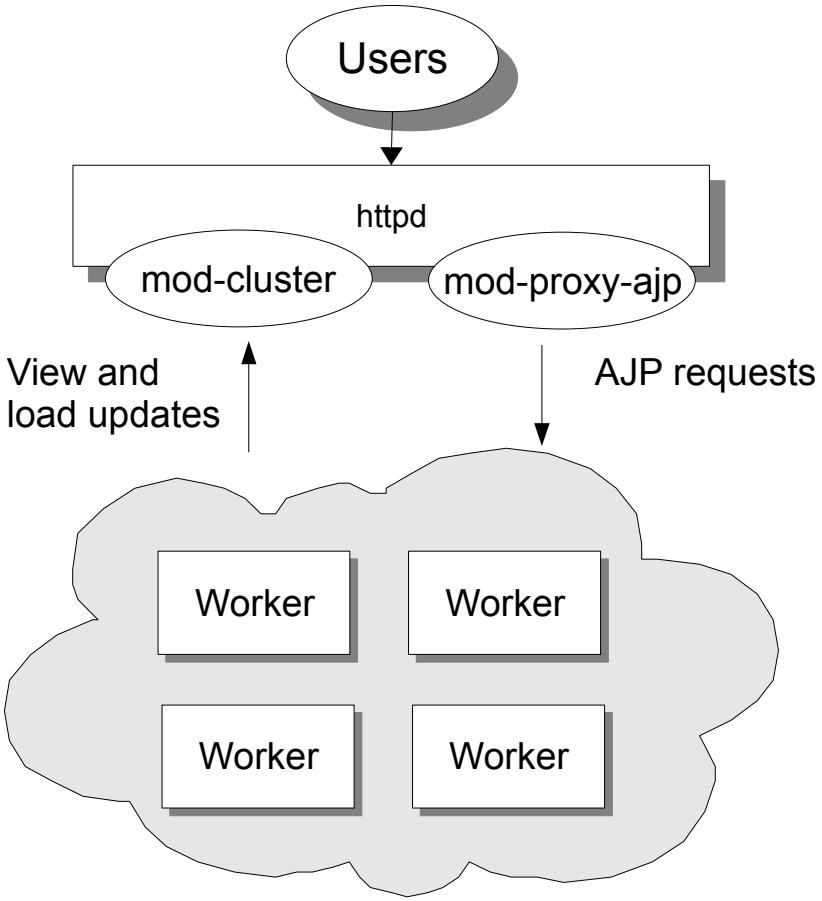

Problem #2: static configuration

- When a worker is added or removed, we need to modify workers.properties or uriworkermap.properties on all httpd servers
  - Tedius for large clusters, or dynamically changing clouds
- Enter **mod-cluster**
  - (Almost) no configuration on the httpd side
  - Works with httpd 2.2.8+, JBoss AS 5+, JBossWeb 2.2.1 and Tomcat 6
  - Based on mod-proxy
  - http://www.jboss.org/mod_cluster

mod-cluster

- Workers register *themselves* with httpd server(s)
  - No more workers.properties
  - Changes in cluster topology are also sent to httpd
- Deployed webapps are automatically registered with httpd, undeployed webapps are un-registered
  - No more uriworkermap.properties
  - No 404s for undeployed webapps
- Workers send their load factors to httpd, so httpd can forward requests based on actual load
  - Load computation is pluggable

mod-cluster architecture



mod_cluster configuration (httpd side)

httpd.conf:

```
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_ajp_module modules/mod_proxy_ajp.so
LoadModule slotmem_module modules/mod_slotmem.so
LoadModule manager_module modules/mod_manager.so
LoadModule proxy_cluster_module modules/mod_proxy_cluster.so

Listen 8000
<VirtualHost *:8000>
  <Directory />
   Order deny,allow
   Allow from all
  </Directory>
  <Location /mod_cluster_manager>
     SetHandler mod_cluster-manager
     Order deny,allow
     Allow from all
  </Location>
</VirtualHost>
```


mod_cluster configuration 'all' (JBossAS 6 M4)

JBOSS/server/all/deploy/mod_cluster.sar/META-INF/mod_cluster-jboss-beans.xml:

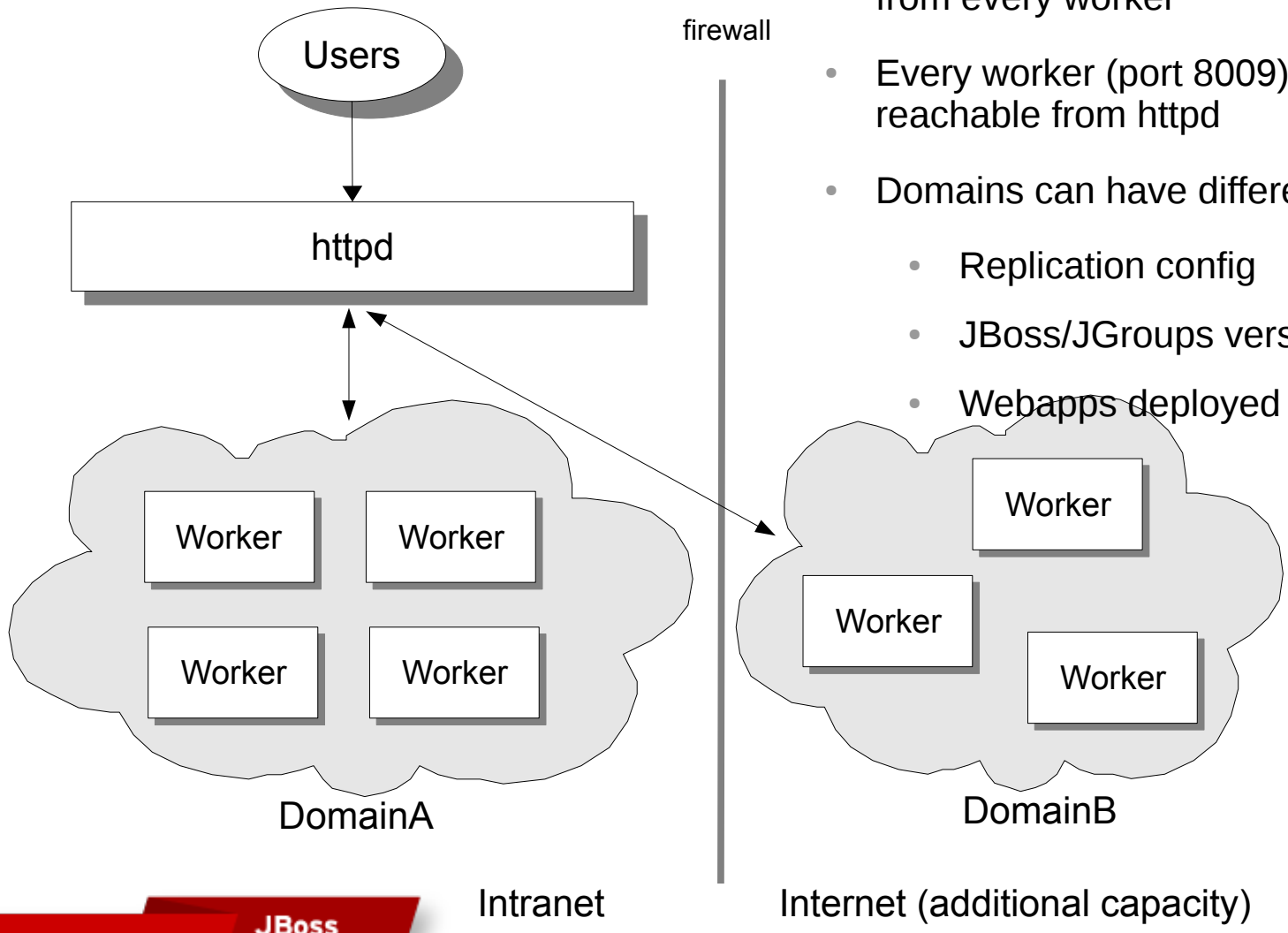
```
<bean name="ModClusterConfig" class="org.jboss.modcluster.config.ha.HAModClusterConfig">
   <property name="proxyList">${jboss.mod_cluster.proxyList:httpd-1:8000,httpd-2:8000}</property>
   <property name="domain">${jboss.domain:DefaultDomain}</property>
   ...
</bean>
```


Starting the JBossAS instances

```
jboss1: ./run.sh -c all -g A -u 232.1.1.1 -m 7500 -Djboss.domain=A -Djboss.jvmRoute=jboss1
jboss2: ./run.sh -c all -g A -u 232.1.1.1 -m 7500 -Djboss.domain=A -Djboss.jvmRoute=jboss2
jboss3: ./run.sh -c all -g B -u 232.2.2.2 -m 8500 -Djboss.domain=B -Djboss.jvmRoute=jboss3
jboss4: ./run.sh -c all -g B -u 232.2.2.2 -m 8500 -Djboss.domain=B -Djboss.jvmRoute=jboss4
```

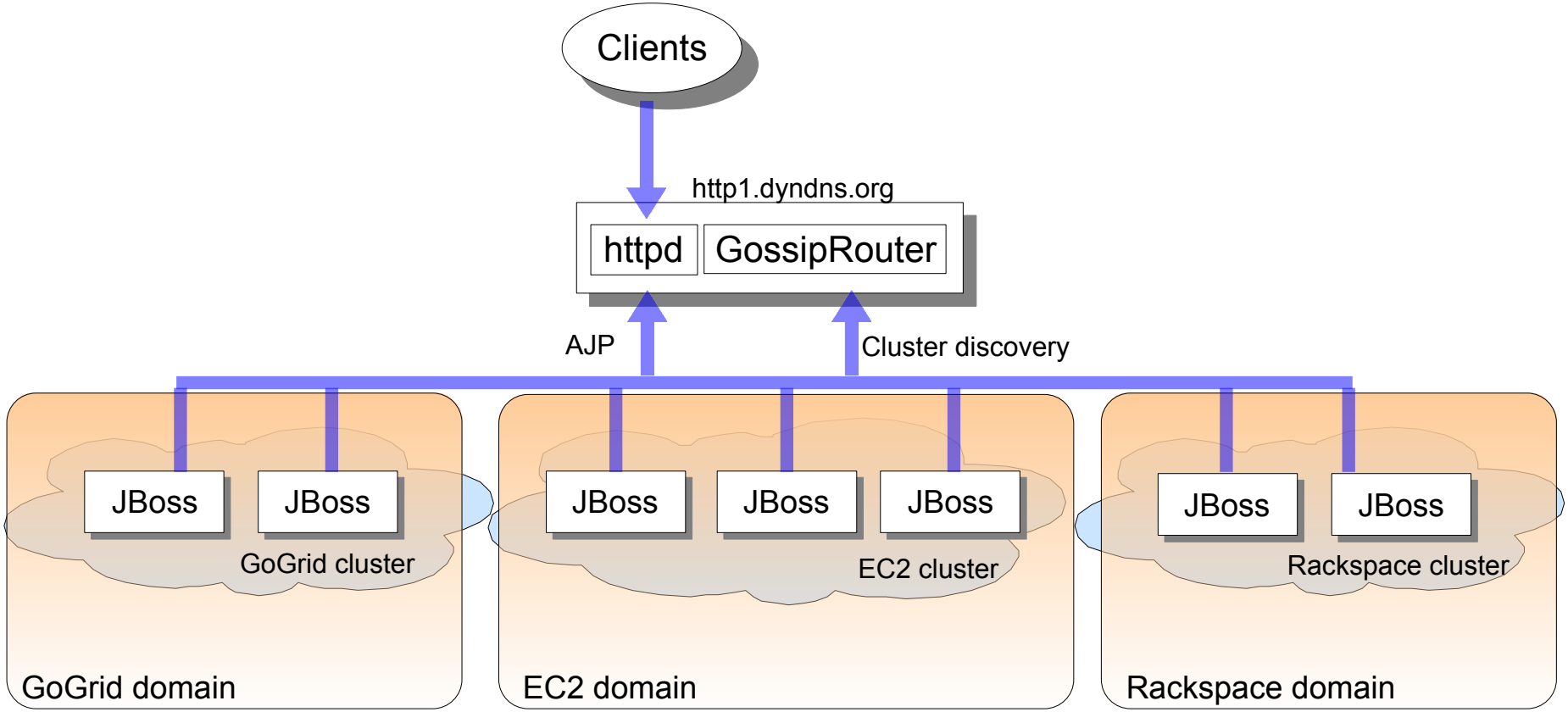
- Domain A has nodes jboss1 and jboss2 and domain B has jboss3 and jboss4
- jboss1 and jboss2 register with httpd under domain A
- jboss3 and jboss4 register with httpd under domain B
- Every server registers the webapps it serves
  - The same app doesn't need to be present on all nodes
  - mod-cluster will only send requests to nodes which have it

Dynamically adding hosts when load increases



- httpd (port 8000) has to be reachable from every worker
- Every worker (port 8009) has to be reachable from httpd
- Domains can have different
  - Replication config
  - JBoss/JGroups versions
  - Webapps deployed (security)

Demo


Clients
http1.dyndns.org
httpd
GossipRouter
AJP
Cluster discovery
JBoss
JBoss
GoGrid cluster
GoGrid domain
JBoss
JBoss
JBoss
EC2 cluster
EC2 domain
JBoss
JBoss
Rackspace cluster
Rackspace domain

Have we solved our 2 problems now ?

- Static configuration: yes
  - We only configure httpd once (adding modules)
  - Every JBoss instance is started with the following props
    - jboss.jvmRoute, jboss.domain, jboss.modcluster.proxyList (not necessary if multicast advertisements are used)
- Rolling upgrades: yes
  - Same as mod-jk, with some added features
    - shutdown-when-drained.sh (in mod-cluster 1.1)
    - Disable, drain and shutdown entire domain (TBD, in JON/JOPR)
    - Start and enable entire domain (JON / JOPR)

Conclusion

- Partition your cluster
  - Divide large clusters into smaller subclusters (domains) to help rolling upgrades
- Use mod-cluster to
  - Dynamically add / remove nodes to / from a cluster
  - Dynamically deploy / undeploy webapps
  - Provide actual load information to help httpd distribute load optimally

Outlook

- Optimize JBoss clusters to run in the cloud
  - (Oftentimes) no multicasting available, large scale
    - Eliminate the TCP N-1 problem
  - Instances often don't have a static IP address
- Use of JON/JOPR to
  - Start, stop, disable and enable an entire cluster
  - Upgrading a domain will be a 3 step process
    - Drain and shut down the cluster
    - Upgrade all nodes of the cluster
    - Start the cluster again
  - View of a cluster topology, vitals, stats

FOLLOW US ON TWITTER

www.twitter.com/redhatsummit

TWEET ABOUT IT

#summitjbw

READ THE BLOG

http://summitblog.redhat.com/