JBoss Community

Clustering in AS 7.0

By Paul Ferraro
& Bela Ban

Agenda

- The Basics
- Configuring applications for clustering
- Clustering subsystems
  - JGroups
  - Infinispan
  - mod_cluster
- Latest Infinispan, JGroups features
- Scaling, cloud readiness

Starting AS7 w/clustering

- Starting servers via managed domain:
  - Use “ha” profile from domain.xml

```
<server-group name="clustered-group" profile="ha">
  <socket-binding-group ref="ha-sockets"/>
</server-group>
```

    - $ ./bin/domain.sh
- Starting multiple standalone servers:
  - Use standalone-ha.xml configuration
    - $ ./bin/standalone.sh -server-config standalone/configuration/standalone-ha.xml

Hey, Where are my clusters?

- All clustering services start on demand and stop when no longer demanded
  - Lifecycle example
    - Deploy app1, starts channel and cache
    - Deploy app2
    - Undeploy app1
    - Undeploy app2, stops cache and channel
- Starting a server with no deployments will not start any channels/caches

Clustered Deployments

- No changes since AS6, mostly...
- Distributed web sessions
  - Add <distributable/> to web.xml
  - Uses “web” cache container, by default
- Clustered Stateful Session Beans
  - @Clustered @Stateful
  - Uses “sfsb” cache container, by default
  - Not yet implemented, coming in 7.1

Clustered Deployments (cont.)

- JPA/Hibernate 2nd level cache
  - Infinispan is default 2nd level cache provider
    - persistence.xml no longer needs to define *hibernate.cache.region.factory_class*
  - Uses “hibernate” cache container, by default
  - Non-clustering profiles use local-cache
    - Provides eviction & expiration support
  - “ha” profiles use clustered caches
    - invalidation-cache for entities/collections

Locating JGroups configuration

- AS5 - AS6
  - server/all/deploy/cluster/jgroups-channelfactory.sar/META-INF/jgroups-channelfactory-stacks.xml
- AS7
  - standalone/configuration/standalone-ha.xml
  - domain/configuration/domain.xml
    - “ha” profile

JGroups Subsystem

```
<subsystem xmlns="urn:jboss:domain:jgroups:1.0" default-stack="udp">
  <stack name="udp">
    <transport type="UDP" socket-binding="jgroups-udp"
               diagnostics-socket-binding="jgroups-diagnostics"/>
    <protocol type="PING"/>
    <protocol type="MERGE2"/>
    <protocol type="FD_SOCK" socket-binding="jgroups-udp-fd"/>
    <protocol type="FD"/>
    <protocol type="VERIFY_SUSPECT"/>
    <protocol type="BARRIER"/>
    <protocol type="pbcast.NAKACK"/>
    <protocol type="UNICAST"/>
    <protocol type="pbcast.STABLE"/>
    <protocol type="VIEW_SYNC"/>
    <protocol type="pbcast.GMS"/>
    <protocol type="UFC"/>
    <protocol type="MFC"/>
    <protocol type="FRAG2"/>
    <protocol type="pbcast.STREAMING_STATE_TRANSFER"/>
    <protocol type="pbcast.FLUSH"/>
  </stack>
  <!-- More stacks -->
</subsystem>
```

General Improvements

- docs/schema/jboss-jgroups.xsd
- Toggle default stack for all clustering services via *default-stack* attribute
- Generic schema grammatically compatible with new protocols and properties
- Externalized socket bindings, thread pools
- All ports registered with socket binding manager

JGroups Socket Bindings

```
<interfaces>
  <interface name="loopback">
    <inet-address value="127.0.0.1"/>
  </interface>
</interfaces>

<socket-binding-group name="clustering-sockets" default-interface="loopback"
                      port-offset="0">
  <socket-binding name="jgroups-udp" port="55200"
          multicast-address="230.0.0.4" multicast-port="45688"/>
  <socket-binding name="jgroups-udp-fd" port="54200"/>
  <socket-binding name="jgroups-diagnostics" port="0"
          multicast-address="224.0.75.75" multicast-port="7500"/>
  <socket-binding name="jgroups-tcp" port="7600"/>
  <socket-binding name="jgroups-tcp-fd" port="57600"/>
  <socket-binding name="jgroups-mping" port="0"
          multicast-address="230.0.0.4" multicast-port="45700"/>
</socket-binding-group>
```

Protocol Stack Customization

- Default properties per protocol
  - https://raw.github.com/jbossas/jboss-as/master/clustering/jgroups/src/main/resources/jgroups-defaults.xml
- Overriding default properties

```
<stack name="udp">
  <transport type="UDP" ...>
    <property name=”enable_bundling”>true</property>
    <property name=”ip_ttl”>0</property>
  </transport>
  <!-- ... -->
</stack>
```

Managing JGroups Threads

```
<subsystem xmlns="urn:jboss:domain:clustering.jgroups:1.0">
  <stack name="udp">
    <transport type="UDP" ...
      default-executor=”jgroups-default”
      oob-executor=”jgroups-oob”
      timer-executor=”jgroups-timer”/>
    <!-- Remaining protocols -->
  </stack>
</subsystem>

<subsystem xmlns="urn:jboss:domain:threads:1.0">
  <queueless-thread-pool name="jgroups-default" blocking="false">
    <core-threads count="20" per-cpu="40"/>
    <max-threads count="200" per-cpu="400"/>
    <keepalive-time time="5" unit="seconds"/>
  </queueless-thread-pool>
  <queueless-thread-pool name=”jgroups-oob” blocking=”false”>
    <!-- ... -->
  </queueless-thread-pool>
  <scheduled-thread-pool name=”jgroups-timer”>
    <!-- ... -->
  </scheduled-thread-pool>
</subsystem>
```

JBoss Community

Locating Infinispan configuration

- AS6
  - server/all/deploy/cluster/infinispan-cache-registry.sar/infinispan-configs.xml
- AS7
  - standalone/configuration/standalone-ha.xml
  - domain/configuration/domain.xml
    - “ha” profile

Infinispan Subsystem

```
<subsystem xmlns="urn:jboss:domain:infinispan:1.0" default-cache-container="cluster">
  <cache-container name="cluster" default-cache="default">
    <alias>ha-partition</alias>
    <replicated-cache name="default" mode="SYNC" batching="true">
      <locking isolation="REPEATABLE_READ"/>
    </replicated-cache>
  </cache-container>
  <cache-container name="web" default-cache="repl">
    <alias>standard-session-cache</alias>
    <replicated-cache name="repl" mode="ASYNC" batching="true">
      <locking isolation="REPEATABLE_READ"/>
      <file-store/>
    </replicated-cache>
    <distributed-cache name="dist" mode="ASYNC" batching="true">
      <locking isolation="REPEATABLE_READ"/>
      <file-store/>
    </distributed-cache>
  </cache-container>
  <!-- ... -->
</subsystem>
```

General Improvements

- docs/schema/jboss-infinispan.xsd
- Toggle default cache for a given cache container via default-cache attribute
- Schema only exposes configuration relevant to a specific cache mode
- Auto JNDI binding of cache containers
  - java:jboss/infinispan/*container-name*
- Externalized thread pools

Customizing Infinispan Caches

- Cache mode
  - <local-cache>
  - <replicated-cache mode=”SYNC|ASYNC”/>
  - <distributed-cache mode=”SYNC|ASYNC” owners=”2”/>
  - <invalidation-cache mode=”SYNC|ASYNC”/>
- Transport
  - <transport stack=”stack-name”/>

Customizing Infinispan Caches

- Eager vs. lazy startup mode
  - <replicated-cache … start=”LAZY|EAGER”>
- JNDI binding
  - <cache-container … jndi-name=”...”>
  - Assumes java:global namespace if unqualified

Managing Infinispan Threads

```
<subsystem xmlns="urn:jboss:domain:clustering.infinispan:1.0">
  <cache-container name="cluster" listener-executor=”infinispan-listener”
      eviction-executor=”infinispan-eviction”
      replication-queue-executor=”infinispan-repl-queue”>
    <transport executor=”infinispan-transport”/>
    <!-- Caches -->
  </cache-container>
</subsystem>

<subsystem xmlns="urn:jboss:domain:threads:1.0">
  <bounded-queue-thread-pool name="infinispan-listener" blocking="true">
    <max-threads count="1" per-cpu="2"/>
    <queue-length count=”100000” per-cpu=”200000”/>
  </bounded-queue-thread-pool>
  <bounded-queue-thread-pool name="infinispan-transport" blocking=”true”>
    <!-- ... -->
  </bounded-queue-thread-pool>
  <scheduled-thread-pool name=”infinispan-eviction”>
    <!-- ... -->
  </scheduled-thread-pool>
  <scheduled-thread-pool name=”infinispan-repl-queue”>
    <!-- ... -->
  </scheduled-thread-pool>
</subsystem>
```

Customizing clustered deployments

- jboss-web.xml

```
<jboss-web>
  <replication-config>
    <cache-name>web.dist</cache-name>
  </replication-config>
</jboss-web>
```

- @Stateful @Clustered @CacheConfig(name=”sfsb.dist”)

Customizing clustered deployments (cont.)

- Interpreting deployment cache name
  - Parsed as ServiceName of cache
    - e.g. “jboss.infinispan.web.dist”
  - Parsed as ServiceName of cache container, assumes default cache
    - e.g. “jboss.infinispan.web”
  - Assumes *jboss.infinispan* base name
    - e.g. “web”, “web.dist”

Customizing clustered deployments (cont.)

- persistence.xml
  - Enabling

```
<property name="hibernate.cache.use_second_level_cache" value="true"/>
<property name="hibernate.cache.use_query_cache" value="true"/>
```

  - Overriding default cache container

```
<property name="hibernate.cache.infinispan.cachemanager"
          value="java:jboss/infinispan/mycontainer"/>
```

  - Overriding individual cache regions

```
<property name="hibernate.cache.infinispan.entity.cfg" value="entity"/>
<property name="hibernate.cache.infinispan.collection.cfg" value="entity"/>
<property name="hibernate.cache.infinispan.query.cfg" value="local-query"/>
<property name="hibernate.cache.infinispan.timestamp.cfg" value="timestamp"/>
```

Using Infinispan directly

- On demand injection of cache container

```
@ManagedBean
public class MyBean<K, V> {
  @Resource(lookup = “java:jboss/infinispan/mycontainer”)
  private org.infinispan.manager.CacheContainer container;
  private org.infinispan.Cache<K, V> cache;

  @PostConstruct
  public void start() {
    this.cache = this.container.getCache();
  }
  // Use cache
}
```

Locating mod_cluster config

- AS6
  - server/all/deploy/mod_cluster.sar/META-INF/mod_cluster-jboss-beans.xml
- AS7
  - standalone/configuration/standalone-ha.xml
  - domain/configuration/domain.xml
    - “ha” profile

mod_cluster Subsystem

```
<subsystem xmlns="urn:jboss:domain:modcluster:1.0">
  <mod-cluster-config advertise-socket="modcluster">
  </mod-cluster-config>
</subsystem>


<socket-binding-group name="clustering-sockets"
                      default-interface="public">
  <socket-binding name="modcluster" multicast-address="224.0.1.105"
                  multicast-port="23364" port="0"/>
</socket-binding-group>
```

General Improvements

- docs/schema/jboss-mod-cluster.xsd
- Much more concise
- No more weeding through class names, component wiring, etc.
- Enumerated load metrics
- Externalized socket bindings

Configuring mod_cluster

- Disabling advertise

```
<subsystem xmlns="urn:jboss:domain:modcluster:1.0">
  <mod-cluster-config proxy-list="192.168.0.1:6666"/>
</subsystem>
```

- Excluding specific contexts

```
<subsystem xmlns="urn:jboss:domain:modcluster:1.0">
  <mod-cluster-config excluded-contexts="ROOT, foo"/>
</subsystem>
```

Configuring mod_cluster (cont.)

- Customizing load metrics
  - Supported types:
    - cpu, mem, heap, sessions, receive-traffic, send-traffic, requests, busyness

```
<subsystem xmlns="urn:jboss:domain:modcluster:1.0">
  <mod-cluster-config advertise-socket="mod_cluster">
    <dynamic-load-provider history="10" decay="2">
      <load-metric type="cpu" weight="2" capacity="1"/>
      <load-metric type="sessions" weight="1" capacity="512"/>
    </dynamic-load-provider>
  </mod-cluster-config>
</subsystem>
```

Caveat: the following slides are not product announcements

New Infinispan Features

- New transactional behavior
  - <transaction mode=”...”/>
    - NON_XA (default)
      - Uses javax.transaction.Synchronization enlistment
      - Allows TransactionManager to use 1PC optimization
      - Translates to faster JPA 2nd level cache
    - NON_DURABLE_XA
      - Uses XAResource w/out recovery
    - FULL_XA
      - Uses XAResource with recovery

New Infinispan Features (cont.)

- storeAsBinary for keys, values, or both
  - Set automatically based on cache mode
    - <local-cache/>
      - storeAsBinary disabled
    - <distributed-cache/>, <replicated-cache/>
      - storeAsBinary enabled for keys and values
    - <invalidation-cache>
      - storeAsBinary enabled for keys only

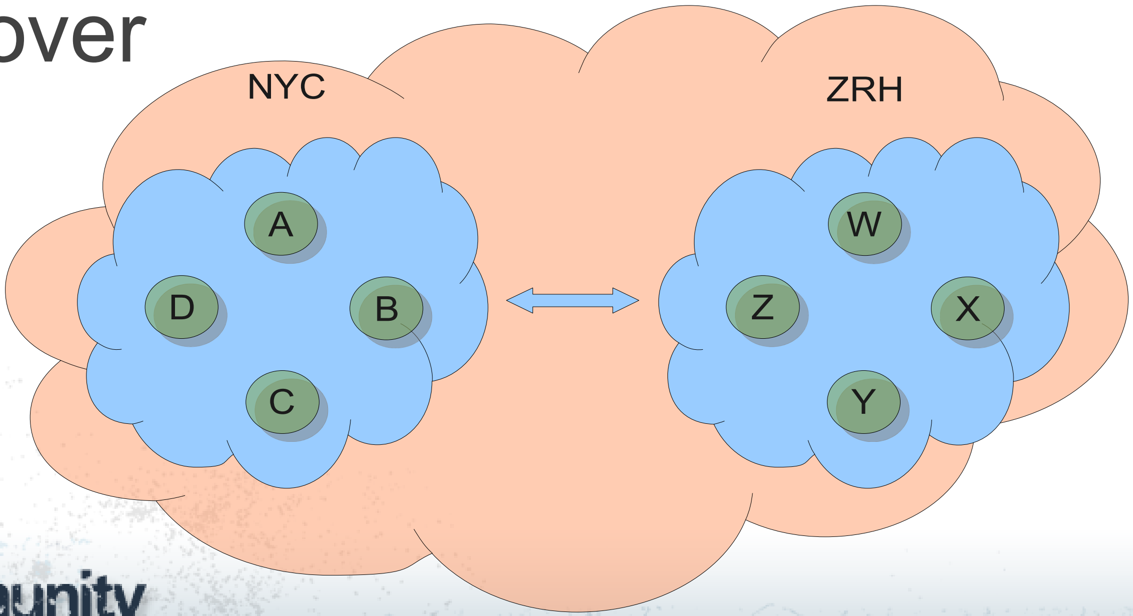
New Infinispan Features (cont.)

- Map/Reduce API
  - Allows large scale computation to be transparently parallelized across cluster
  - Watch Manik's JUDCon presentation

```
Cache<K, V> cache;
Mapper<K, V, Kout, Vout> mapper;
Reducer<Kout, Vout> reducer;
MapReduceTask<K, V, Kout, Vout> task = new MapReduceTask<K, V, Kout, Vout>(cache);
Map<Kout, Vout> results = task.mappedWith(mapper).reducedWith(reducer).execute();
```

New in JGroups: Geographic failover

- Connecting separate clusters into one virtual cluster
- Using a backup cluster for geographic failover



New in JGroups: Ergonomics

- Zero configuration
- Monitor environment and adjust cluster configuration dynamically
  - One-size-fits-all doesn't yield optimal results
  - From small clusters to large clusters
  - From private clusters to clusters in the cloud
  - Take traffic patterns into account
- Ongoing work

New in Jgroups: Daisy Chaining

- Used for cluster traffic when IP multicasting it not available
  - This is the case in the cloud
- Spreads traffic load more evenly across a cluster
  - Decentralized model
  - Higher throughput, slightly higher latency, too...
  - Good for some apps, bad for others
    - Ergonomics

Large Clusters

- Support for large JBoss clusters (100 – 1000 nodes)
- Largest known (JGroups) cluster: 500+ nodes
- Important for Enterprise Data Grid
- Requires configuration changes for optimal performance
  - The goal is to reduce the number of changes through ergonomics

AS7 in the Cloud

- Better integration with OpenShift
- Start cluster nodes via OpenShift Express/ Flex
  - Increase/decrease cluster size based on load policies
- Deploy apps to entire cluster

AS7 in the Cloud (cont.)

- Cluster management
- Support for more clouds
  - JBoss clustering works on all clouds, but we can have even better integration
    - e.g. Rackspace cloud store etc

Links

- JBoss AS: www.jboss.org/jbossas
- JGroups: www.jgroups.org
- Infinispan: www.infinispan.org
- mod_cluster: www.jboss.org/mod_cluster

Links (cont.)

- JBoss World presentations:
  - "Running a JBoss cluster in the cloud"
  - "Geographic failover for JBoss clusters"
- Enterprise Data Grid
- OpenShift: www.openshift.org

Questions?

Comments?

Feedback?