GlassFish v2
AS/MQ integration changes

Sivakumar Thyagarajan

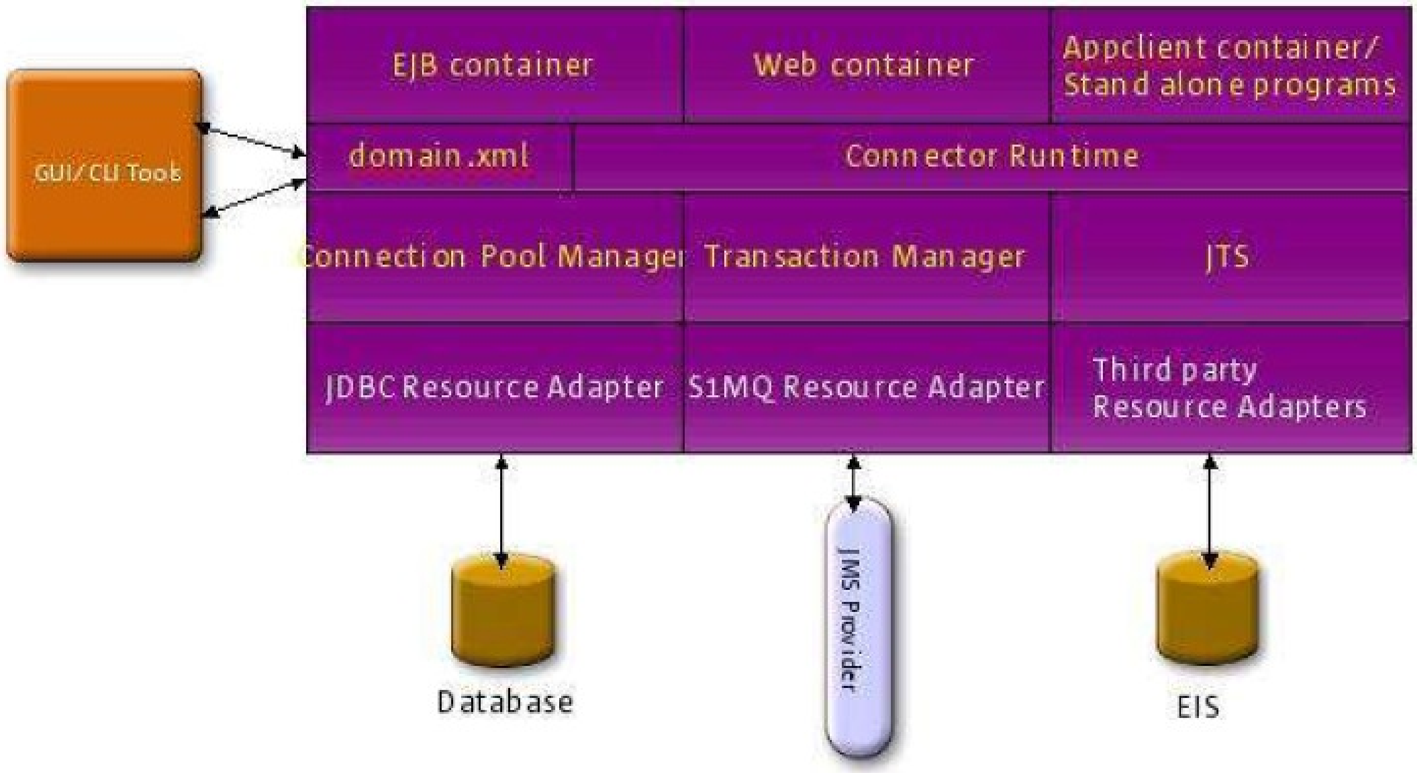
Presentation to AS user experience meeting 27th Sep 2006

http://glassfish.dev.java.net

Agenda

- Overview
- 8.x EE availability support
- GlassFish v1 [AS 9.0] updates
- GlassFish v2 planned features
  - JMS availability – how to configure
- Generic RA for JMS – brief overview
- Resources

AS integration architecture



AS/MQ integration overview

- Connectors 1.5 specification support in GlassFish
  - Message provider pluggability
  - Inbound communication to message endpoints (MDBs)
  - Outbound communication to MoM products
- JMS integration via System Resource Adapter – *jmsra*
  - Generic RA for JMS is the option for 3rd party MoM products
- *jmsra*
  - provided by the Project Open MQ team http://mq.dev.java.net
  - Bi-directional Connectors 1.5 resource adapter

8.x EE support

- Support for MQ broker clustering in SE/EE
- Master broker to track and synchronize configuration changes across MQ cluster instances
- Load balanced message inflow
  - Support for load-balancing durable topic subscriptions
  - No code change for consuming components – transparent
- MQ RA supports connection pooling of outbound connections

jms-service element

- jms-service in domain.xml
  - allows an administrator to configure the integration between an application server cluster or server instance and a MQ broker instance or a cluster.
  - jms-service defines 1...* jms-hosts
  - default JMS host
  - additional reconnect properties

AS/MQ integration modes

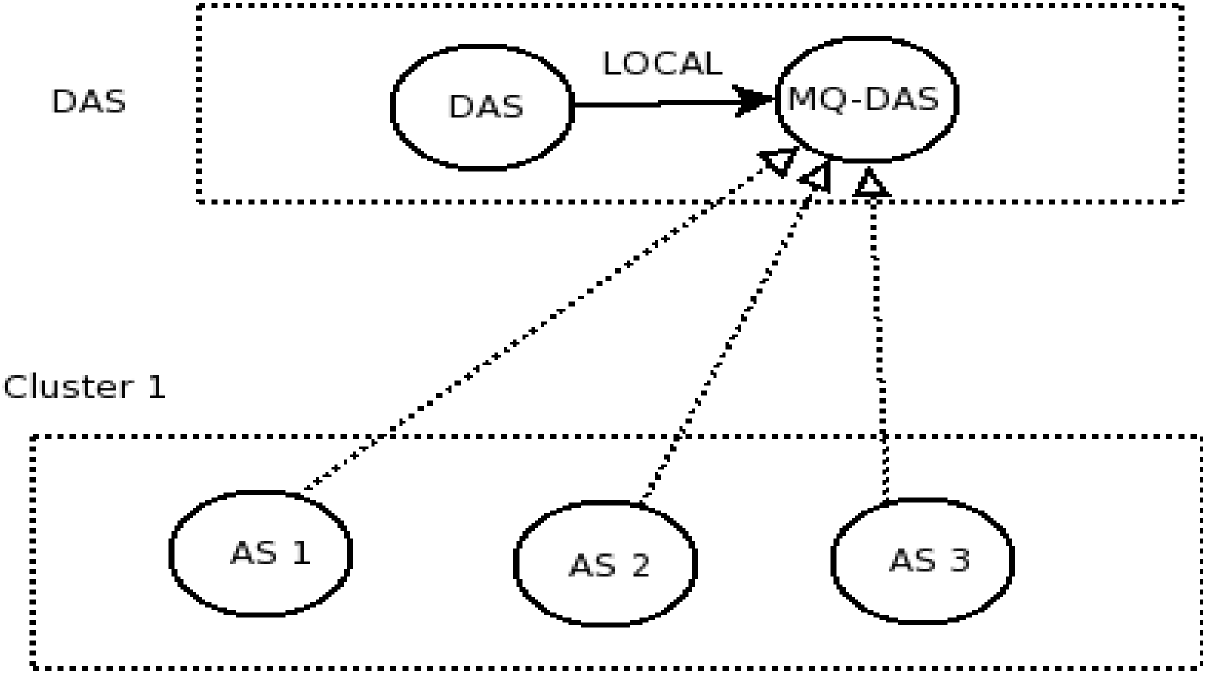
- LOCAL
  - AS manages MQ broker's lifecycle
  - out-of-process
  - default in AS 8.x PE and AS 8.1 EE DAS
- REMOTE
  - AS does not manage MQ broker's lifecycle
  - MQ administrator starts/stops MQ broker
  - recommended production mode for AS 8.1 EE cluster instances.

MQ LB in previous releases

- Message delivery load balancing to Topic subscriptions in MDB
  - transparent to application
  - *inClusteredContainer* and *GroupName* set in the RA by AS
- Inbound message delivery to Queue endpoint consumers are also load balanced
- Destinations – *maxActiveNumConsumers* is transparently set by the AS

AS 8.x out-of-the-box cluster setup

- DAS – LOCAL broker
- Cluster instances connect to DAS' broker by default



GlassFish V1 AS/MQ updates

- MQ-RA based lifecycle control
- Inprocess MQ
  - EMBEDDED as a jms-service type
  - AS lazy init defers broker startup until first use
  - reduces process count for AS operation
  - reduction in memory footprint, robustness *

GlassFish v2 planned features

- Inprocess MQ
  - enhance EMBEDDED mode by short-circuiting network operations
- Sticky connection balancing
- Handling "auto-clustering" for non-HA AS/MQ clusters.
- High availability[HA] cluster
  - providing a HA JMS runtime environment to JMS application components deployed in AS EE.

Enhanced EMBEDDED mode

- Was also referred to as DIRECT
- Bypass the networking stack for JMS operations when the application server and the JMS broker is co-located in the same VM
- Performance optimization
- Enables all JMS operations within a co-located setup to not incur the overhead of inter-socket communication
- Goal : a more performant system for low-end deployments.

GF V2 out-of-the-box cluster setup

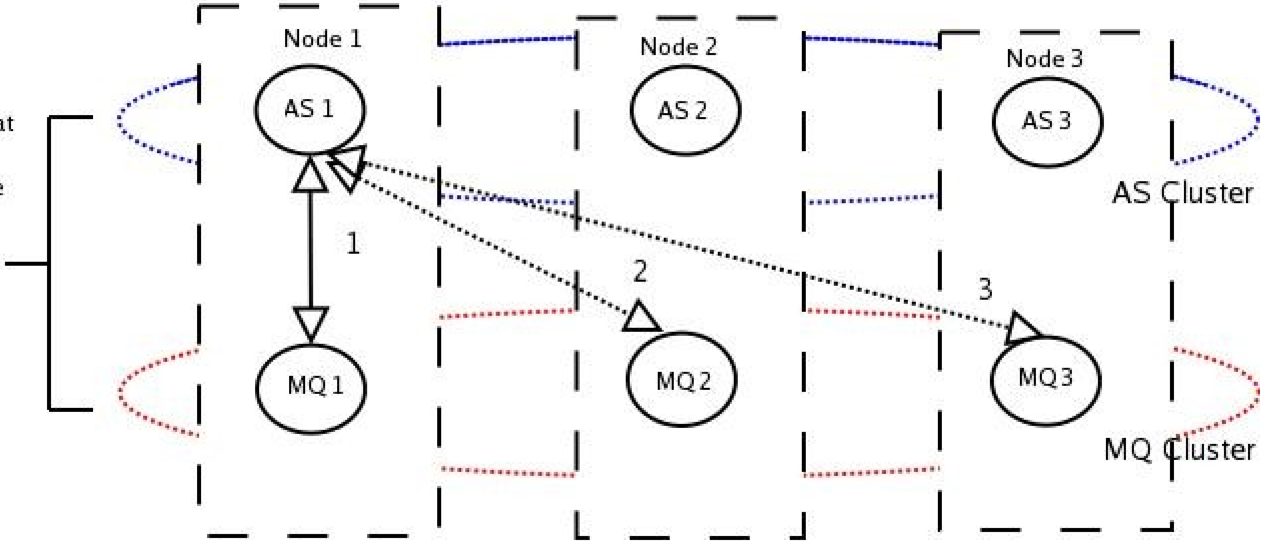
- Earlier default OOTB configuration
  - not performant
  - single point of failure
- New setup
  - DAS – EMBEDDED broker
  - Cluster instances – 1:1 non-HA AS/MQ broker clustered instances

Sticky connection balancing

Sticky loadbalancing for jms-service configured as LOCAL

For LOCAL integration mode MQ address list is set such that the co-located broker instance is the first entry in the list.

Delivery and consumption of messages would be in most cases be LOCAL, except when there is a failure in the co-located broker instance.



Sticky connection balancing

- PRIORITY load balancing
- Enhanced to provide sticky behavior
- *addressList* modified for each server instance to point to co-located LOCAL broker
- co-located production and consumption of messages
- similar to the EJB/Web load-balancing sticky schemes
- ensures equitable load distribution across the MQ broker cluster

Non-HA AS/MQ clusters

- “auto-clustering”
  - LOCAL co-located non-HA AS/MQ clusters
- Setting the master broker
  - first AS broker instance in the cluster

Availability options

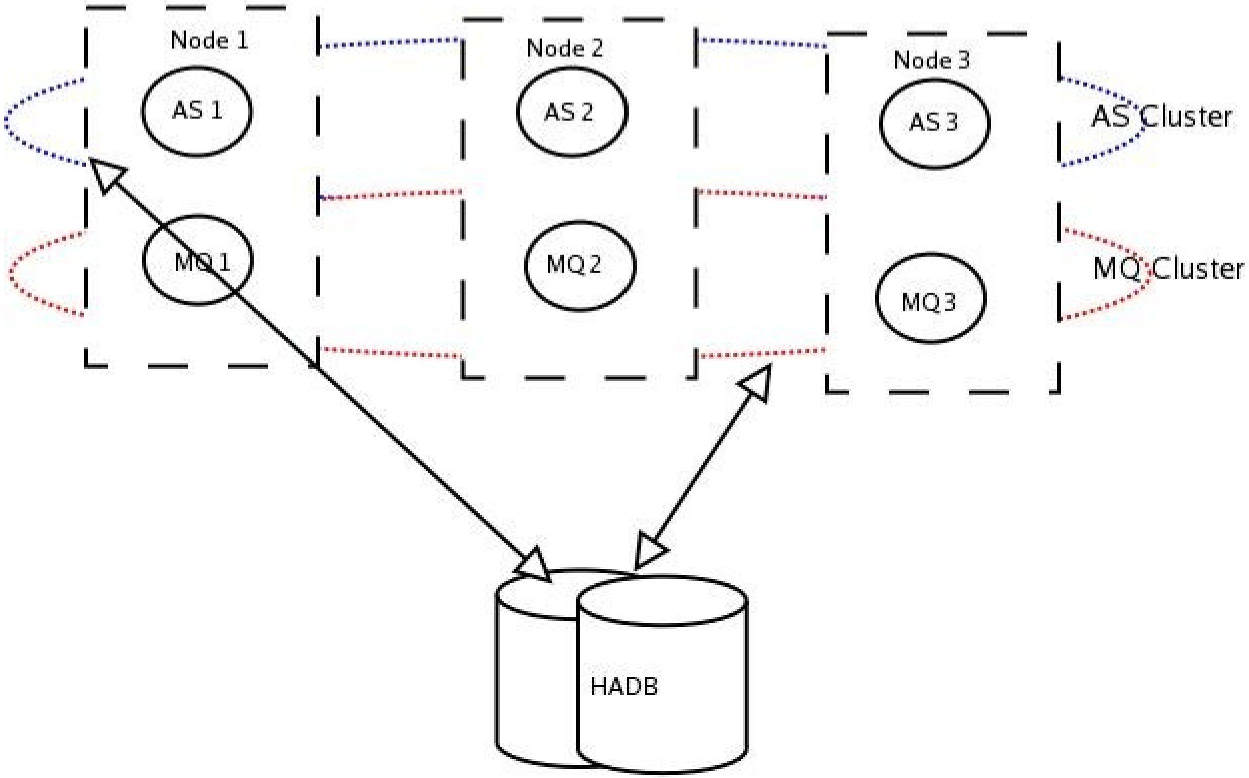
- Two forms of availability
  - *service-availability*
    - Connection load-balancing
    - Users not concerned about non-availability of some messages for a period of time
    - MQ 3.6 cluster model via connection failover
  - *data-availability*
    - stringent availability requirements
    - persistent messages and service availability
    - MQ 4.1 HA-cluster model

MQ 4.1 HA-Cluster type

- MQ broker cluster instances
  - peer-to-peer
  - share a common persistence store [HADB]
  - automatically be able to detect instance failure and perform a takeover of the failed broker's persistent messages
  - no loss of transacted persistent messages to a Queue or a durable topic subscription
  - Master broker is not required

Sharing HADB configuration

A HADB installation shared by AS and MQ clusters

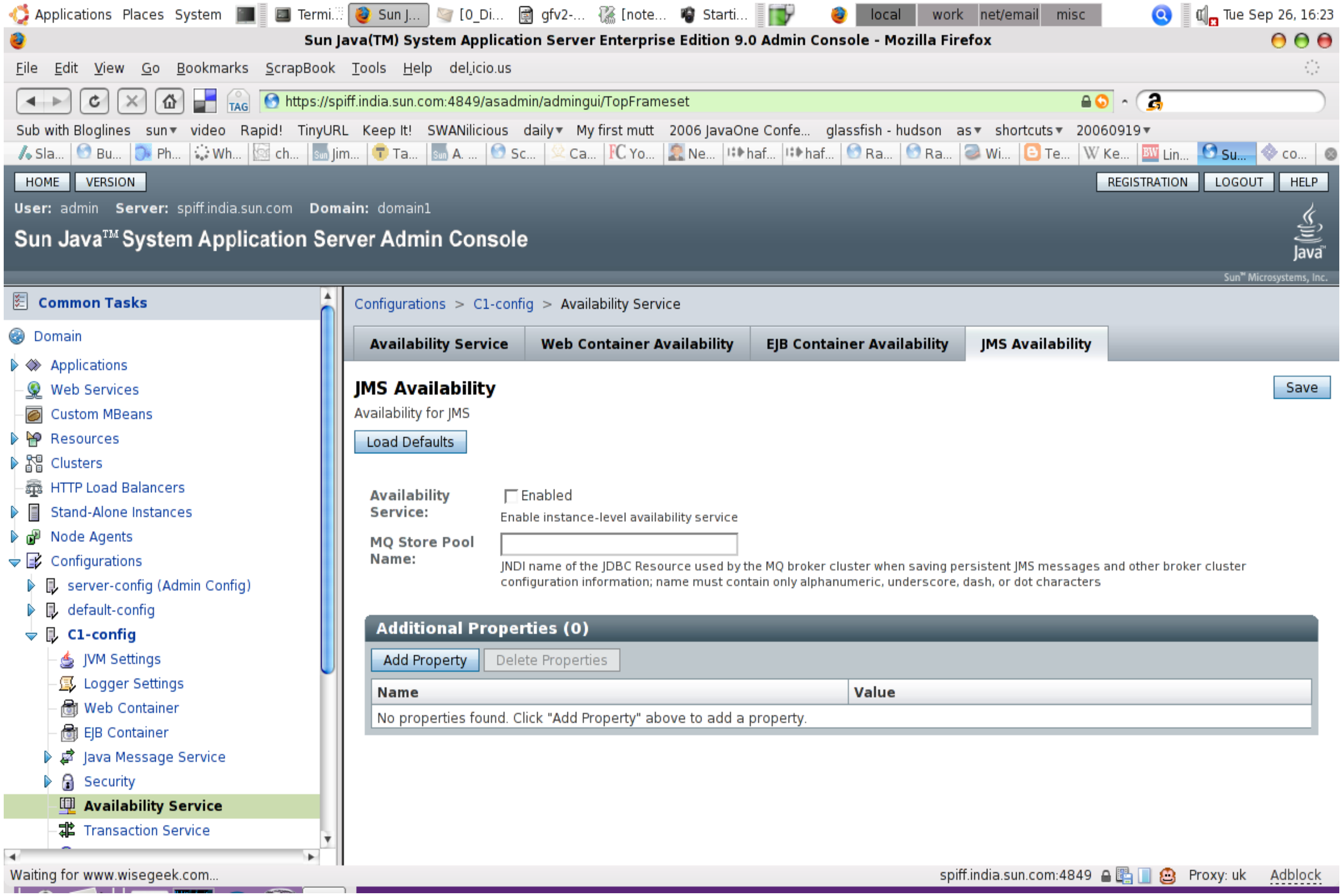


How to configure JMS availability 1/2

- asadmin CLI
  - *asadmin configure-ha-cluster*
    - [today jms-availability needs to be turned on explicitly after this, but this would be fixed]
  - restart cluster
    - MQ broker instance automatically started as a HA-Cluster type.
  - deploy app.

How to configure JMS availability 2/2

- asadmin GUI



Configurations > C1-config > Availability Service

Availability Service | Web Container Availability | EJB Container Availability | JMS Availability

JMS Availability

Availability for JMS

Availability Service: □ Enabled
Enable instance-level availability service

MQ Store Pool Name:
JNDI name of the JDBC Resource used by the MQ broker cluster when saving persistent JMS messages and other broker cluster configuration information; name must contain only alphanumeric, underscore, dash, or dot characters

Additional Properties (0)

Name	Value
No properties found. Click "Add Property" above to add a property.	

HA configuration usecases

- LOCAL shared HADB
  - default when *configure-ha-cluster* is used
  - AS/MQ share the same HADB nodes
- LOCAL non-shared HADB
  - configure custom HADB store using HADB tools
  - creates a corr. jdbc-connection-pool
  - modifies availability-service>jms-availability>mq-store-pool-name
- REMOTE
  - configures custom HADB store using HADB tools
  - configures MQ broker cluster to use custom HADB store

Generic RA for JMS

- Integrate SJSAS with 3rd party Message Oriented Middleware(MOM) products
- Integration with Tibco EMS, Seebeyond, Websphere MQ, ActiveMQ, Sonic MQ, Sun Java System MQ
- Features
  - > Two ways integration - JNDI or javabean
  - > Bi-directional messaging support.
  - > Distributed tx support (XA) in both directions of messaging.
  - > Concurrent Message Delivery
  - > Message Redelivery to MDBs
  - > JMX Monitoring, LB message delivery - Topic

Resources

- AS/MQ integration one pager for GlassFish v2 http://glassfishwiki.org/gfwiki/attach/OnePagersOrFunctionalSpecs/as-mq-integration-gfv2.txt
- Generic RA for JMS integration in GlassFish V2 onepager http://www.glassfishwiki.org/gfwiki/attach/OnePagersOrFunctionalSpecs/genericra-onepager.txt
- Integration Technologies page at GlassFish https://glassfish.dev.java.net/javaee5/integration-tech/glassfish_integration_technologies.html
- Project Open Message Queue http://mq.dev.java.net
- Generic RA for JMS http://genericjmsra.dev.java.net

Q & A

GlassFish v2
AS/MQ integration changes

Sivakumar Thyagarajan

sivakumart@dev.java.net

Presentation to AS user experience
meeting 27th Sep 2006

http://blogs.sun.com/sivakumart