Restcomm SIP Servlets for JBoss

Transparent Failover

Table of Contents

A Restcomm SIP Servlets Server for JBoss cluster does not employ any standby nodes. Typically, proxies and registrars must share the user location table by using a database cluster.

The Restcomm SIP Load Balancer, which is a SIP Call ID-aware load balancer, is used as the intermediary. The SIP Load Balancer forwards stateful transaction requests to cluster nodes based on its provisioning algorithm. The SIP Load Balancer acts as an entry-point to the cluster, and distributes the incoming requests between nodes. It is always advised to use a SIP load balancer or an IP load balancer in a cluster configuration.

This choice of implementation has many benefits:

- There is no need for standby nodes, because the remaining nodes in a degraded cluster automatically and transparently (to the user) take on the load of the failed node. This can be done because both the SIP Load Balancer and SIP Servlet-enabled JBoss Application Servers support mid-call or call setup failover (respectively Established SIP Dialog and Early SIP Dialog).
- There is no need to ensure that requests are directed to the correct node, because in a SIP Servlets-enabled JBoss Application Server (or Restcomm JAIN SLEE server) cluster, any node can serve any request to any User Agent (UA).
- All hardware is in use, reducing costs.
- Maintenance is easier, due to all nodes having nearly-identical configurations.

Restcomm Failover Capabilities

The SIP Stack used by the Restcomm SIP Servlets for JBoss supports two modes:

- `ESTABLISHED SIP DIALOG` failover. This means that failover can occur only on established calls (SIP Dialogs which are in the CONFIRMED state as per RFC 3261) and calls that are in the process of being setup will not be failed over (SIP Dialogs which are in the EARLY state as per RFC 3261).

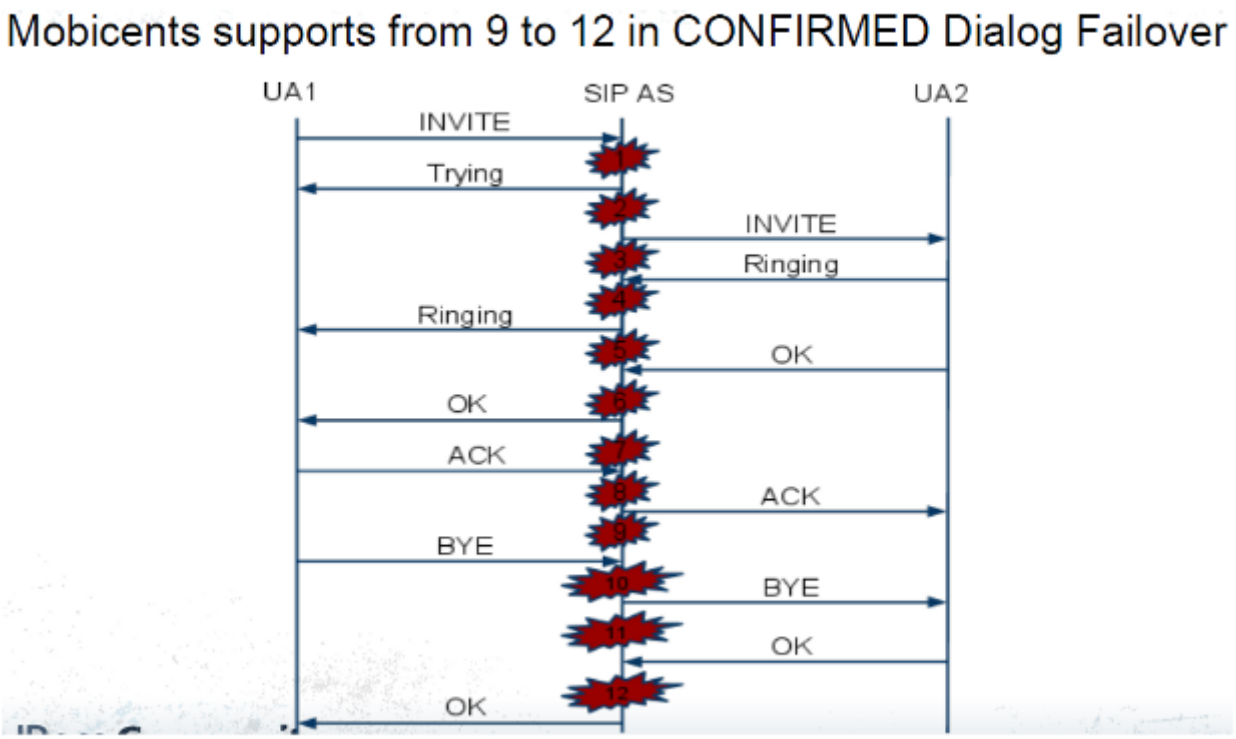

Figure 1. Established Dialog Failover

This is the default configuration.

Optimization Note

To maximize performance it is recommended to use a load balancer with SIP affinity enabled, so that all messages related to the same call go to the same node even though Restcomm Sip Servlets supports the case where SIP transactions for a given SIP Dialog go to different nodes.

- `EARLY SIP DIALOG` failover. This means that failover can occur after an informational response (1xx) is received with a To Tag. For example, calls that are in the process of being setup will be failed over (SIP Dialogs which are in the EARLY state as per RFC 3261).

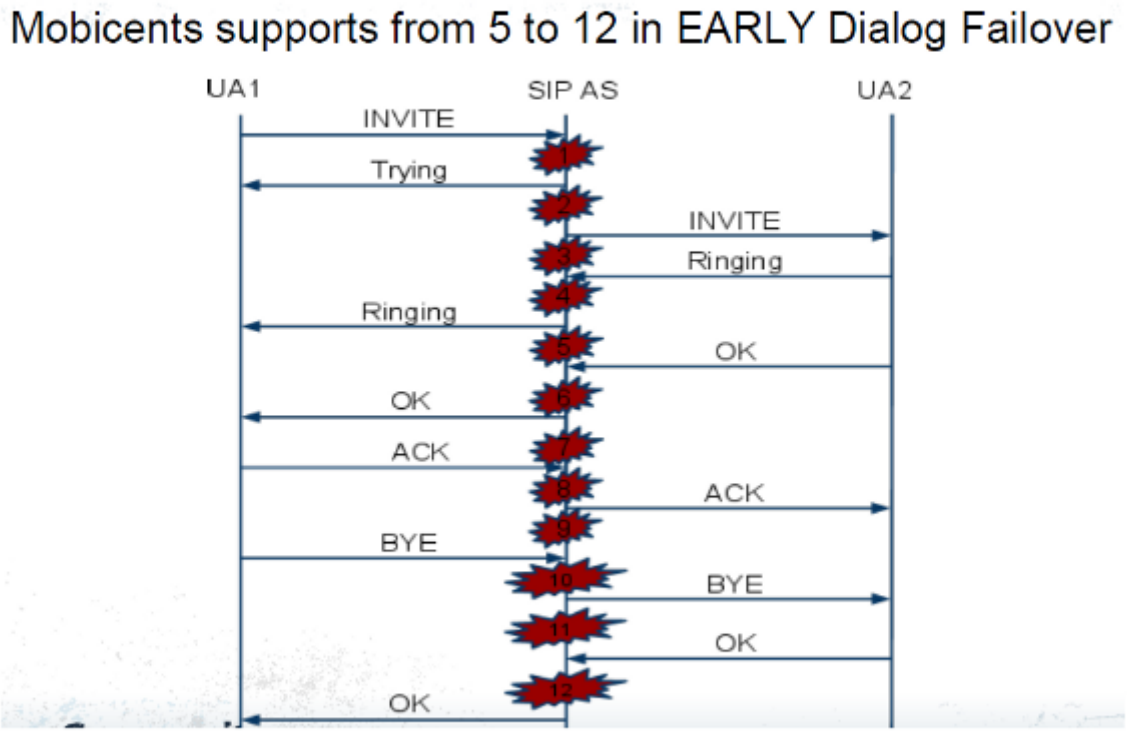

Figure 2. Early Dialog Failover

As Early Dialog failover means replicating some transaction states, it can introduce some overhead in terms of network replication as well. The failover granularity is configurable to best fit applications on their usage and optimize the performance. The following property, `org.mobicents.ha.javax.sip.REPLICATION_STRATEGY=EarlyDialog`, needs to be added to the SIP Stack Stack properties, defined in the external properties file specified by the *sipStackPropertiesFile* attribute as described in [_bsssc_binary_sip_servlets_server_adding_sip_connectors].

Optimization Note

To maximize performance it is recommended to use a Load Balancer with SIP affinity enabled so that all messages related to the same SIP transaction (and even SIP call) go to the same node even though Restcomm Sip Servlets supports the case where SIP transactions for a given SIP Dialog go to different nodes.

Restcomm for JBoss Cluster: Installing, Configuring and Running

There are a number of options you can specify for Restcomm clustering. By default, most of them are configured in the "all" server configuration, which is ready to use. In this chapter we will cover the most common configuration options you might need.

Downloading

Installing

Configuring

Running

Using

Testing

Uninstalling