Cougaar

Cougaar Agent Architecture Open-Source site

Servlet Tunnel

This page discusses Cougaar’s remote servlet tunnel/redirect support.

The servlet tunnel allows requests for remote servlets to be tunneled through the message transport, instead of using a browser HTTP-Redirect. This provides two benefits:

1. It uses the message transport protocol, firewall, and security code
2. It allows a node to support servlets even if that node can’t run an HTTP server.

Features

For example, say we had:

- host “A” has node “1″ and agent “alpha”
- host “B” has node “2″ and agent “beta”

Without the tunnel, a servlet request from a browser/UI on host “A” to

- http://A:8800/$beta/foo

will send back a browser HTTP-Redirect of:

- http://B:8800/$beta/foo

and the browser/UI will open this remote URL. The HTTP-Redirect is visible in browsers as a change to the Brower URL Location line.

With the tunnel, the request is instead sent through the message transport to the remote node. This tunneling is completely transparent to the browser/UI. In particular, the Browser URL Location will remain as

- http://A:8800/$beta/foo

yet show agent “beta”‘s servlet pages (note the “A” hostname). The console will show “+-+-” output due to the messages being sent & received between the two nodes. The message transport will use the low-level link protocol (e.g. JMS) to take care of any firewall/security issues.

Multiple redirectors can be loaded into a node and will be attempted in their load-order, which (by default) favors the HTTP-Redirect. The user can specify an explicit redirector/tunnel on the URL line, e.g.:

- http://A:8800/$(mts_tunnel)beta/foo

For details, see the javadocs for

- webserver/src/org/cougaar/lib/web/arch/util/PathParser.java

Configuration

By default, a multi-node Cougaar society uses backwards-compatible HTTP-Redirects. The MTS tunnel is enabled but will only be used if
a) the remote node is not running an HTTP server, or
b) the local node is configured to disable remote HTTP-Redirects, or
c) the user explicitly requests the MTS tunnel in the URL.

a)
To start an "embedded" node with no HTTP server that will allow MTS tunnel-based servlet access, set the following system properties:

- -Dorg.cougaar.society.xsl.param.servlet_engine.tomcat=false
- -Dorg.cougaar.society.xsl.param.servlet_redirector.http_redirect=false

The following system properties are the defaults, so you don't need to specify them:

- -Dorg.cougaar.society.xsl.param.servlets=true
- -Dorg.cougaar.society.xsl.param.servlet_engine.mts=true
- -Dorg.cougaar.society.xsl.param.servlet_redirector.mts_tunnel=true

With the above configuration, a remote Tomcat-enabled "gateway" node will be able to access servlets on the above "embedded" node.

b)
To disable HTTP redirects and force the use of the MTS tunnel, even if the remote node is running an HTTP server, set:

- -Dorg.cougaar.society.xsl.param.servlet_redirector.http_redirect=false

This option is useful if HTTP-Redirects are forbidden (e.g. blocked by firewalls).

c)

The MTS tunnel can also be forced on the URL line via URL “options”, without requiring the above -D. For example:

- http://A:8800/$(mts_tunnel)beta/foo

Or, to prevent redirects and display an error page if the specified agent “beta” is not on “A:8800″, use:

- http://A:8800/$()beta/foo

For details, see the javadocs for

- webserver/src/org/cougaar/lib/web/arch/util/PathParser.java

Advanced configuration options

The servlet “engine”s and “redirector”s are pluggable. They are specified in the agent template files:

- configs/common/NodeAgent.xsl
- configs/common/SimpleAgent.xsl

A node can be configured to run any/all of the following “servlet_engine”s:

- “tomcat” : Our standard Tomcat-based HTTP engine
- “micro” : A new, trimmed-down HTTP engine, which is subclassed by the MTS tunnel engine
- “mts” : Receives tunneled MTS servlet requests

and any/all of the following servlet “servlet_redirector”s:

- “http_redirect” : Our standard HTTP-Redirect to a remote host:port
- “http_tunnel” : A proof-of-concept URL connection tunnel, disabled by default
- “mts_tunnel” : Sends tunneled MTS servlet requests

When a node is created, it registers its available servlet engines in the naming service. This tells a remote node how to redirect/tunnel requests to agents on that remote node.