RMI-P4

Harsimrankaur
PDMCEW, Bahadurgarh

Abstract: SAP is one of the leading providers of business software. Its product portfolio for enterprise application software is organized around the various key offerings. The P4 protocol is an SAP proprietary protocol that facilitates communication between remote objects from different namespaces (possibly different hosts). It is related to the Remote Method Invocation (RMI) and Common Object Request Broker Architecture (CORBA) technologies, and its implementation combines features of both of them. In this paper P4 features, Network Configuration and uses has been discussed.

1. Introduction

The Java Remote Method Invocation (RMI) system allows an object running in one Java virtual machine to invoke methods on an object running in another Java virtual machine. RMI provides for remote communication between programs written in the Java programming language. The P4 protocol also has some specific features that make it well-suited to the SAP NetWeaver Application Server Java (AS Java) cluster architecture and provides transparent, robust failover and load balancing mechanisms. The P4 protocol is designed exclusively to support Java-to-Java remote communication. It greatly simplifies and significantly speeds up the input-output operations for transferring objects. The protocol uses the standard Java Serialization techniques to transfer the objects, thereby providing full compatibility with all Java types. This also reduces the volume of data being transferred through the network by avoiding the requirement to align the data types

The P4 protocol employs the concept of the Broker architecture defined by CORBA. This allows for the functional extension of the protocol, by adding information that is transmitted as an independent part of each remote call, a transport layer, and so on, on top of it. The P4 protocol is designed to work in both standalone and cluster environments. It is well-suited to the J2EE Engine cluster architecture and provides transparent, robust failover and load balancing mechanisms.

The P4 protocol is designed to work in both standalone and cluster environments. RMI-P4 refers to the development framework that is used to develop distributed remote Java objects applications on theAS Java. It is based entirely on the P4 protocol. The framework and the protocol functions are logically concentrated in the P4 Provider Service. The RMI-P4 system provides the functions that meet the

requirements of applications based on the Distributed Object Model. The distributed applications consist of two parts - a client part and a server part. RMI-P4 enables the mechanism by which client and server parts communicate, which includes:

- Locating the remote object
- Communication with the remote object
- Loading classes of objects that are transmitted in the communication process

The RMI-P4 system provides the following functions:

- A standard mechanism for the communication between remote objects using stubs and skeletons. To simplify development, the stubs and skeletons are now transparently generated by the RMI-P4 system at runtime.
- Effective garbage collection: The P4 Provider Service has a reference-counting garbage collection mechanism which automatically removes objects that are not referenced by any clients.
- Load balancing: Client RMI-P4 requests are load balanced at the point where the client creates an InitialContext and obtains a reference to the remote server-side object using it.
- Transparent failover for clustered remote objects: RMI-P4 implements a mechanism for migrating instances of clustered remote objects to other available server processes within the AS Java cluster in case a server process crashes.
- Reliable network connections between remote clients and servers, supporting different connectivity options, such as encryption (SSL) and connections via SAP router.

2. Using P4 Protocol

A secure P4 communication can be set up using both SSL and HTTPS protocols as underlying transport layers.

a. Procedure

In the client code, obtain the InitialContext to connect to the remote object using the following properties:

1. Specify the port for the secure connection with the provider URL property.
2. Specify the underlying transport layer you want to use. You use the TransportLayerQueueproperty with value SSL for P4 over an SSL connection, or HTTPS for P4 over an HTTPS connection.

b. Viewing Remote Protocols

1. From the *Instance* drop-down list, choose the instance of the

server process that need to monitor.

2. From the *Cluster Node* drop-down list, specify the server process.
3. Choose the *Protocols* tab.
4. In the *Protocol list* screen area, find all the remote protocols (P4; IIOP; Telnet), and the number of threads reserved for each one of them.
5. To view all the remote objects exported on the server process for a particular protocol, select the protocol name.

- In the *Remote Objects* screen area, find information about the Java class name of the remote object, the key under which it is registered in the protocol, and whether or not it is redirectable (can be restored after a server crash).
- In the *Remote Object Details* screen area, find more protocol-specific information about a selected remote object, such as remote interfaces implemented by the remote object, and so on.

c. Viewing Connections

1. To view information about the connections currently opened from/to the specified server process, choose the *Connections* tab.

2. The *Connections* screen area displays details such as connection ID, IP address, port number and the communication protocol.

3. In the *Calls* screen area, you can view information about the calls currently processed in the cluster.

3.Network Configuration for RMI-P4

The RMI-P4 provides reliable network connections between remote clients and servers if the following requirements are met:

- **Firewalls/proxies**

P4 must be able to establish direct TCP connections to the configured P4 ports of the cluster. P4 does not automatically reestablish broken connections, so if the firewall is configured to close established connections after a timeout, it breaks the communication. If the message server is used for load balancing, connections to its HTTP/HTTPS port must also be possible. Only standalone P4 applications that use the HTTPS as the connection type support proxies. The proxy must be configured to allow SSL connections to the specified P4 SSL host/port.

- **NAT**

NAT is supported with the following limitations: Load balancing with the message server cannot be used unless it is also accessible through NAT, the DNS on the local side is configured to resolve the ICM host names to the configured addresses for NAT access, and the port numbers have to be the same. Connections to other

instances from the same cluster that are also behind NAT cannot be opened automatically when stubs are created because the correct NAT IP address for them is not known. The limitations for private network addresses also apply when NAT is used.

- **Loopback addresses**

Loopback addresses such as 127.0.0.1, 127.0.0.2 and so on are not used with RMI-P4 as they are invalid on the remote side. The machine should have a valid IP address and the host name must not resolve locally to a loopback address

- **Private network addresses (rfc1918)**

Private network addresses are generally supported by RMI-P4 but connections between machines from two different local networks that use the same private network address space are not supported.two machines have real IP addresses that they can use to connect with each other over the Internet but are also part of two separate private networks with a conflicting address space.

- **Security limitations**

For security reasons, only a single connection between two participants is allowed. Subsequent connection attempts are rejected. If two machines can access each other via several IP addresses, you have to configure the applications to use the same address.

- **SAP router**

Only standalone P4 applications support SAP router connections. The user must pass a valid SAP router string which is typically in the InitalContext properties.

4. RMI-P4 Specific InitialContext Properties

When the InitialContext in remote client code is constructed, specify the properties with which the context environment is initialized. In addition to the standard properties defined by the javax.naming.Context interface (for example theINITIAL_CONTEXT_FACTORY,PROVIDER_URL,SECURITY_CREDENTIALS, and so on), use properties that are specific to the RMI-P4 implementation on the AS Java. These properties allow to adapt the behavior of the RMI-P4 according to specific application scenario.

- InitializeConnectionTimeout:To specify the timeout for initializing new connections. If the connection cannot be initialized for the specified period, an error message is returned to the client.The value of the property is specified in milliseconds.If the value is 0, then no timeout is set and the call waits until the connection is initialized
- RuntimeConnectionTimeout:
  The value of this property is specified in milliseconds. A value of 0 means no such

timeout is set and therefore, the client never checks whether the remote side is available

- TransportLayerQueue: The property can take values none, ssl, https or SAPRouter.
- EnableNAT:To enable/disable support for client-server P4 connections through NAT gateways.
- CallTimeout : The value is specified in milliseconds. A value of 0 means no timeout is set.
- sap.p4.remote_classloading: Remote RMI-P4 class loading is a function that allows server-side classes that are not present in the class path of the client that makes the remote call to be loaded. This function is available only after the client has already looked up a remote object.
- HTTP_Host: This property is used only in cases when you use P4 over SSL as the transport layer for RMI-P4 communication, which passes through an HTTP proxy. The property specifies the host name of the HTTP proxy.
- HTTP_Port: This property is used only in cases when you use P4 over SSL as the transport layer for RMI-P4 communication which passes through an HTTP proxy. The property specifies the port the HTTP proxy listens to for client connections.
- SAPRouter: This property specifies that SAP Router is used for the service connection. The host and port obtained from the message server are appended at the end of the string, so the routers must be configured to allow connections to every access point that is published in the message server
- SAPRouterPassword: Password required to access the service if the router requires one.

5. Conclusion

SAP software is used to manage many core business processes and data. As a result, it is critical for all organizations to manage the life cycle of user access to the SAP applications while adhering to security and risk compliance requirements.RMI-P4 implements a efficient SAP proprietary protocol mechanism for migrating instances of clustered remote objects to other available server processes within the AS Java cluster.

6. References:

1. Breg, F., Diwan, S., Villacis, J., Balasubramanian, J., Akman, E., and Gannon, D. 1998. “Java RMIperformance and object model interoperability: Experiments with Java/HPC++ distributed components”. In *Proceedings of the ACM 1998 Workshop on Java for High-Performance Network*

Computing (Santa Barbara, CA), ACM, New York, NY.

2. Java RMI Documentation : http://java.sun.com/j2se/1.4.2/docs/guide/rmi
3. R. van Nieuwpoort, J. Maassen, H. E. Bal, T. Kielmann, and R. Veldema. Wide-area parallel computing in Java. In *Proc. of ACM 1999 Java Grande Conference*, pages 8–14, San Francisco, Calif., June 1999.
4. S. Microsystems. Java Remote Method Invocation – Distributed Computing for Java. White Paper, 1998.
5. IBM , Integrating IBM Tivoli Security and SAP Solutions, Red paper , 2000