Application Research of Java Message Service Technology

Xia Yan
Shenyang Institute of Engineering
Shenyang Key Laboratory of Information Security for Power System
Shenyang, China
winston_xy@163.com

***Abstract*-For the message system of an enterprise, it is a common service to deliver message in the distributed system. Message Oriented Middleware becomes critical component in the interoperation. This kind component may integrate into one reliable flexible system. JMS provided by SUN, want to unit the specification of all MOM so that can be visited by Java application. In this text，the main content is the basic concept, composition and implement.**

Keywords- Message service; JMS Interface; JNDI; JMS Provider;MOM

I. INTRODUCTION

News is a kind of communication methods between the software components and the applications, and it is distributed communications of loosely coupled[1]. In a messaging system, client can send messages to others; also can receive the messages from other clients.

In order to realize loosely coupled distributed communications, the sender and receiver of message only request know the format and objective about messages.

Java Message Service (JMS) is a specification which was proposed by SUN and its partner company for the purpose of unifying each kind ofmessage.Oriented middleware system interface． It promotes the reliabifity of asynchronous communication between components in distributed environment ， and call be used for establishing loosely-coupled message system； therefore,it simplifies thedevelopment of enterprise applications．

JMS provides an interface which realize message passing through the bottom of the access, and it provides JMS services to the clients. JMS is a group of Java API, which implements the standard of service by providing a standard application programming interface and service provider interface.

JMS defines the standard for reliable Enterprise Messaging. Enterprise messaging, often also referred to as Messaging Oriented Middleware (MOM), is universally recognized as an essential tool for building enterprise applications. By combining Java technology with enterprise messaging, the JMS API provides a powerful tool for solving enterprise computing problems.

The JMS API improves programmer productivity by defining a common set of messaging concepts and programming strategies that will be supported by all JMS technology-compliant messaging systems[2].

It defines a standard between the clients and the underlying messaging service providers to interact messages. Its specific role is to provide a creating, send, receive and read enterprise information approach to Java program. It contains two models of message, which are PTP (Point to Point) and Pub/Sub (Publish/Subscribe), and these two kinds of models provide reliable information transmission, affairs and news filter, etc.

II. REALIZE MESSAGE SYSTEM BASED ON JMS

JMS is a kind of interface form does not contain a concrete realization method, and we usually use the message oriented middleware which called JMS Provider to realize JMS interface.

With a reliable, asynchronous, loosely coupled, languages and platform-independent way, this kind of software support messages between in distributed applications. Each client connected to a message broker in the messaging system, and the message broker provides services just like creating, sending, receiving and reading the messages.

Because the transmission of messages is based on point-to-point, all of the JMS clients are specified for the client normally. JMS application is not only the application of a defined message but also is a series of client in order to exchange messages. For example, a form of JMS application is to achieve message transmission between the multiple applications.

JMS supports two message types based on PTP and Pub/Sub, which are called PTP Domain and Pub/Sub Domain, and both of them inherit the unified super interface of JMS. The establishment of PTP Domain is based on the message queue.

Each message is sent to a specified queue, and then the client to remove messages from the queue. Pub/Sub Domain defines JMS client how news distribution to the specific topics. Topics can be organized according to layers and in a specific topic within the context of distribution and subscriptions can be further allowed to receive messages.

Connection Factory is the management of a client object which used to create a connection; Connection is an activated connection point to a JMS provider, used to create the Session object; Destination is an object which used to encapsulate the message target; Session is the Single-threaded context used to send and receive messages; Message Consumer is an Object used for receiving messages which create by Session; Message Producer is an Object used for sending messages which create by SessionMessage interface is the root of all the JMS messages.

This interface defines the JMS message header fields, nature and all messages are used in the response method. The

Published by Atlantis Press, Paris, France.

corresponding relationship and operation are showed between the PTP-Domain Interfaces, Pub / Sub-Domain interfaces and the JMS Common Interfaces and in Figure. 1.

In point to point or queuing model, a producer to release a particular queue the message, a consumer reads messages from the queue. In this mode, the producers know the queue of consumers, and send a message to the consumer's queue directly.

Features on this model is only one consumer will get the message, during the consumption of message in the receiver, the producers do not need to be running; similarly, the recipient does not need to be running when the message is sent,so every message successfully processed will be sign by the recipient.

Publisher/Subscriber Domain allows the release of information to a particular message topic. Multiple subscribers may be interested in receiving messages from a particular message topic. In this model, publishers and subscribers do not know each other. This model is like anonymous BBS.

There is a time-dependent properties between the publishers and the subscribers. Publishers need to create a subscription, so that customers can purchase subscriptions. Subscribers must maintain continuous active to receive messages unless the subscriber to establish a durable subscription. In this case, the announced message when the subscriber is not connected will release when the subscriber reconnects.

Using the Java language, JMS provides a way to separate application and transport layer which to provide data. The same set of Java classes can connect to different JMS providers through the information of Providers in JNDI. The group first to use a connection factory class to connect to the queue or topic, and then send or post messages. At the receiving end, customers receive or subscribe to these messages.

III. JMS INTERFACE

A typical JMS application programming interface includes seven types,including ConnectionFactory,

Connection,Destination,MessageConsumer,MessageProducer,Message and Session.

(1) ConnectionFactory object is managed by the user to create a connection to the JMS provider,and we need to create a ConnectionFactory object using JNDI（Java Naming and Directory Interface）. JMS clients access through portable interface connection, this approach is that code does not need to be modified when the underlying implementation changes.

So the JMS clients can be able to find them when the administrator configure the JNDI connection factory name space. According to the message type, the user will use the queue connection factory or topic connection factory.

(2) Connection is the communication link between the message and the application server. When users create and receive connection factory, you can create a connection with the JMS provider. Depending on the connection type, connection allows users to create conversation, to send or receive queue and topic to the target.

(3) Destination is a managed wrapper which including the message target object identifier. Message destination is the place to publish and receive messages, queue or topic. The administrator of JMS create these objects, then the user through JNDI to find them. Similar to the connection factory, the administrator can create two types of goals, one is point to point model of the queue, the other is the publisher / subscriber model for the theme.

(4) MessageConsumer is created by the session object, used to receive messages sent to the target.Session and the Destination object to create the required MessageProducers and MessageConsumers. MessageConsumer can receive a message queue and topic type using the synchronous mode or the asynchronous mode.

(5) MessageProducer is created by the session object, used to send messages to the target. Users can create a target of the sender or create a common sender.MessageProducer send a message in the specified target.

(6)Message is an object used to transfer between consumers and producers, that is sent from one application to another application. A message has three main parts:a message header, a set of message properties and a message body.Message head is necessary, the message properties and message body is optional.

The message header contains information used to identify and find routes for the operation set. All of the messages support the unified field of header. A complete message header will be posted to all JMS clients which receive messages, but will be not sent to non-JMS clients.

Message property contains the message of the additional properties, support providers and users of other compatible. Message property can be seen as an additional header field, and it built-in tools to support the needs of the JMS message property values. It is effective for the message provides an additional, optional header field mechanism.Users can create custom fields and filters, and it can provide a mechanism to add application-specific attributes to messages. Property value is used for the message selection mechanism, and along with any application or other data which message infrastructure required. Names of property must obey the rules of the message selection marker

The body of the message contains the actual data of message. JMS defines several message types used by the body to cover the current major message types. Body of the message allows the user to create many types of messages, such as Textmessage, Mapmessage, Bytesmessage, Streammessage and Objectmessage,and Most message application systems support the filtering and classification of message.

(7)Session represents a single-threaded context for sending and receiving messages.Because sessions are single-threaded, so the message is continuous, meaning that the message is sent in accordance one by one to receive with the order.

Session has the advantage of support services. If the user selects the transaction support, A group of messages will be saved during the session, until the transaction is committed only to send the message. Before committing the transaction, the user can cancel these messages by using the rollback

operation. Session allows the user to create a message producer to send messages, create a message consumer to receive messages.It is mapped on the sub-interface objects and JMS interface objects based on two types in TABLE.1[3].

IV. THE PROCEDURES AND NEWS SERVICE PROVIDER OF CONNECTION INTERMEDIARY ANALYSIS.

Message Service also called the message server. Generally the services for the JMS have mature soft wares which include commercial and free. Such as IBM's Web Sphere MQ is business, and Open JMS is free.

How to connect to a JMS message server is achieved by the JMS administration objects. JMS provides two management objects including Connection Factory and Destination. Management object is created for the JMS client application object and it contains JMS configuration information, and it is also need to complete all the tasks of client.

In the architecture of JMS, role of JNDI (Java Naming and Directory Interface) is to create an initial context which associated with the JMS Connection Factory, then use this context to create a connection based on the message server of JMS.

Usually, the two management objects of JMS are placed in the namespace of JNDI; clients of JMS look up in the JNDI namespace management objects, JMS clients find administered objects in JNDI name space. The realization method of the connection between JMS client and JMS Provider by JNDI is expressed in Figure 2[3].

V. THE ANALYSIS AND APPLICATION OF SERVER CONFIGURATION ON THE PROGRAM OF A TYPICAL JMS CLIENT.

The main role of JNDI server is to provide naming and name query about the client.JNDI system defines the four classes used to implement the name service. The four classes include Contextlmpl, NameParserlmpl, NameService and Namespace.

A typical JMS client applications perform the following steps:

(1)First need to find a ConnectionFactory object and one or more Destination objects using JNDI.

(2)Then Use ConnectionFactory to create a JMS Connection object.

(3)Use the Connection to create one or more of JMS Session objects.

(4)Create the required MessageProducers and MessageConsumers using Session and Destination objects

(5)The last step is inform the Connection to start delivering messages.

If the user is not using a reference, or the user want to change name of the queue or the connection factory, you need to use the tools of platform management which configure the JMS provider. The queue or the connection factory will be created to remain on the server until the server is restarted. Some platforms possibly provide expand API to the client in order to create the queue and the connection factory programmatically.

The basic steps to configure the server on the platform of J2EE SDK as follows:

Start the application server, create a new message queue based on J2EE SDK, Then named message queue which need to meet the requirements of JNDI. For ease of management, we should select a queue which can indicate that it is used by the JMS, just like JMS/ MyTestQueue.

Because the Queue Connection Factory is placed in a rigid sample program, so the program needs to use a different connection factory name and recompile.

Use the server of J2EE admin or client tool to create a ConnectionFactory which name is required meeting the requirements of JNDI, at the same time, we need to List the connection factory and the destination. For example, we can use the j2eeadmin-listJmsFactory and j2eeadmin-listJms_

Destination.

If it is not a platform for the J2EE SDK, you can use the J2EE platform deployment tool to create the message queue and the connection factory.

VI. SUMMARIES

The architecture of JMS can be used as a standard API or the framework of provider. JMS can develop middleware quickly and the portable, message-based Java client service program using existing messaging services middleware.

ACKNOWLEDGMENT

Supported by project of 2009 key laboratory constructing of Shenyang (Project number: 1091244-1-00)

REFERENCES

[1] Paul J. Perrone. Designing enterprise systems with the J2EE Platform. Sun Microsystems, 2009

[2] http://www.oracle.com/technetwork/java/overview-137943.html

[3] Mark Hapner. Java Message Service . Sun Microsystems, 2010

[4] Shaun Terry. Enterprise JMS Programming . John Wiley & Sons, 2010

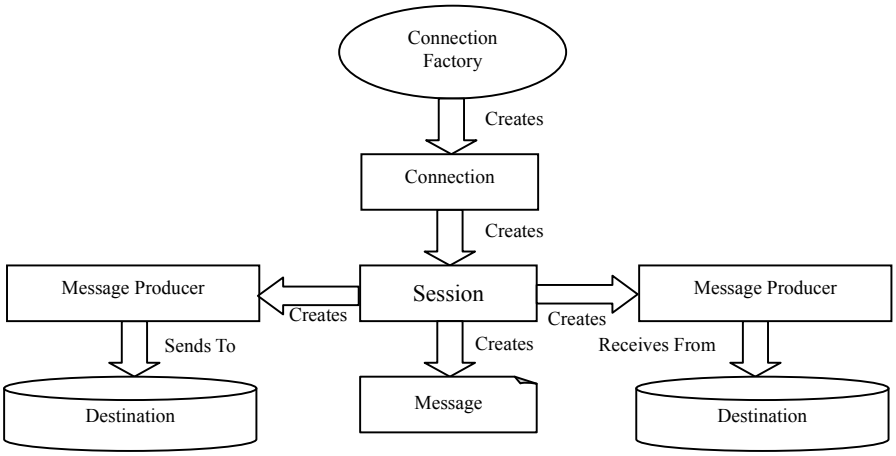

Figure1. The relationship between JMS objects

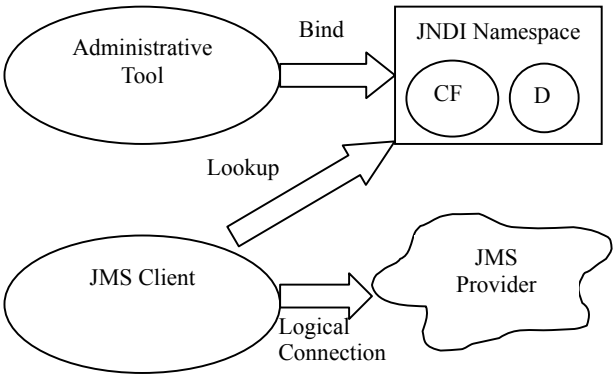

Figure 2. The connection method of JMS Provider creating by JNDI

TABLE I. MAPPING ON THE SUB-INTERFACE OBJECTS AND JMS INTERFACE OBJECTS BASED ON TWO TYPES.

JMS Common Interfaces	**PTP-Domain Interfaces**	**Pub/Sub-Domain interfaces**
Connection Factory	Queue Connection Factory	Topic Connection Factory
Connection	Queue Connection	Topic Connection
Destination	Queue	Topic Connection
Session	Queue Session	Topic Session
Message Producer	Queue Session	Topic Publisher
Message Consumer	Queue Receiver/Browser	Topic Subscriber