BÀI THỰC HÀNH
MÔN HỌC: HỆ PHÂN TÁN
CHƯƠNG 2: Kiến trúc

1. Nội dung

Ở bài thực hành này chúng ta sẽ làm về 2 mô hình kiến trúc JMS và DDS. Mục đích của bài thực hành sẽ giúp các bạn nắm vững và hiểu hơn lý thuyết của 2 khái niệm này.

2. Điều kiện

2.1. Kiến thức

Sử dụng thành thạo hđh Unix
Các kiến thức về mô hình Publish/Subscribe đã học trên lớp lý thuyết.
Kỹ năng lập trình Java và C++

2.2. Phần cứng

Máy tính cài hđh Ubuntu

2.3. Phần mềm

Máy phải có cài JDK 8.0 trở lên.

3. Các bước thực hành

3.1. JMS

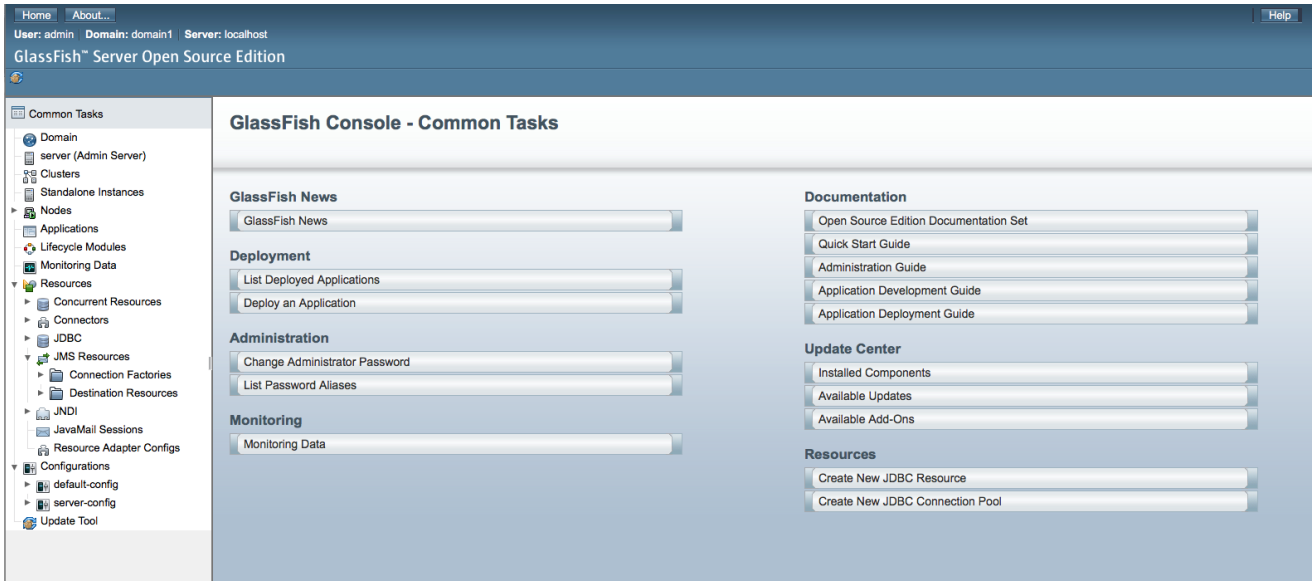
Chúng ta biết JMS hỗ trợ 2 mô hình là Point-to-Point và Publish/Subscribe. Ở bài thực hành này chúng ta sẽ tập trung vào mô hình P/S.
Đầu tiên chúng ta phải cài đặt một application server. Chúng ta sẽ chọn một server nguồn mở là glassfish.

Cài đặt server glassfish:

- Download về tại địa chỉ http://download.java.net/glassfish/4.1.1/release/glassfish-4.1.1.zip
- Giải nén ra thư mục glassfish4.
- Khởi động glassfish bằng lệnh

```
glassfish4/bin/asadmin start-domain
```

Lúc này server glassfish đã chạy một domain là domain1. Ngoài ra glassfish còn hỗ trợ giao diện web trên cổng 4848. Các bạn mở trình duyệt và vào địa chỉ http://localhost:4848
Các bạn sẽ thấy giao diện web như hình dưới đây. Hãy chú ý vào phần JMS Resources, đó là phần chúng ta phải tạo Connection Factories và Destination resources.



Câu hỏi 1: Giải thích vai trò của application server glassfish.

Tạo 2 JNDI

Bước tiếp theo chúng ta phải tạo 2 JNDI là *myTopicConnectionFactory* và *myTopic*.

Thông thường có thể làm bằng giao diện web, tuy nhiên làm theo cách này rất hay bị lỗi. Vì vậy khuyến khích tạo 2 JNDI bằng cách gõ lệnh. Chú ý, các lệnh được gõ sau khi vào thư mục `glassfish4/bin/` và gõ lệnh `./asadmin`

Tạo resource Connection Factory

```
asadmin>create-jms-resource --restype
javax.jms.TopicConnectionFactory
```

Sau đó bạn sẽ được hỏi tên của jndi, gõ là *myTopicConnectionFactory*

```
Enter the value for the jndi_name
operand>myTopicConnectionFactory
```

Tạo resource Destination:

```
asadmin> create-jms-resource --restype javax.jms.Topic
```

Tương tự, khi được hỏi jndi name thì gõ vào là *myTopic*

Vào giao diện web và kiểm tra xem 2 jndi đã được tạo hay chưa.

Câu hỏi 2: Tại sao lại phải tạo 2 JNDI như trên?

Tạo chương trình Sender và Receiver.

Bước này sẽ là lập trình bằng ngôn ngữ Java, khuyến khích chạy chương trình bằng IDE Eclipse.

Mở Eclipse, tạo 1 project chung, đặt tên là JMSTopicProject.

Chú ý, cần phải add thêm các thư viện sau vào project:

- *gf-client.jar*: lấy trong thư mục `glassfish4/glassfish/lib`
- *javax.jms.jar*: có thể tải về từ Internet.

Tạo 3 file đại diện cho 3 lớp sau: MySender.java, MyReceiver.java, và MyListener.java
Các đoạn mã nguồn cho 3 file trên như sau:

File: MySender.java

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import javax.naming.*;
import javax.jms.*;

public class MySender {
   public static void main(String[] args) {
      try
      {   //Create and start connection
         InitialContext ctx=new InitialContext();
         TopicConnectionFactory f=(TopicConnectionFactory)ctx.lookup("myTopic
ConnectionFactory");
         TopicConnection con=f.createTopicConnection();
         con.start();
         //2) create queue session
         TopicSession ses=con.createTopicSession(false, Session.AUTO_ACKNOW
LEDGE);
         //3) get the Topic object
         Topic t=(Topic)ctx.lookup("myTopic");
         //4)create TopicPublisher object
         TopicPublisher publisher=ses.createPublisher(t);
         //5) create TextMessage object
         TextMessage msg=ses.createTextMessage();

         //6) write message
         BufferedReader b=new BufferedReader(new InputStreamReader(Syste
m.in));
         while(true)
         {
            System.out.println("Enter Msg, end to terminate:");
            String s=b.readLine();
            if (s.equals("end"))
               break;
            msg.setText(s);
            //7) send message
            publisher.publish(msg);
            System.out.println("Message successfully sent.");
         }
         //8) connection close
         con.close();
      }catch(Exception e){System.out.println(e);}
   }
}
```

File: MyReceiver.java

```
import javax.jms.*;
import javax.naming.InitialContext;
```

```
public class MyReceiver {
   public static void main(String[] args) {
      try {
         //1) Create and start connection
         InitialContext ctx=new InitialContext();
         TopicConnectionFactory f=(TopicConnectionFactory)ctx.lookup("myTopic
ConnectionFactory");
         TopicConnection con=f.createTopicConnection();
         con.start();
         //2) create topic session
         TopicSession ses=con.createTopicSession(false, Session.AUTO_ACKNOW
LEDGE);
         //3) get the Topic object
         Topic t=(Topic)ctx.lookup("myTopic");
         //4)create TopicSubscriber
         TopicSubscriber receiver=ses.createSubscriber(t);

         //5) create listener object
         MyListener listener=new MyListener();

         //6) register the listener object with subscriber
         receiver.setMessageListener(listener);

         System.out.println("Subscriber1 is ready, waiting for messages...");
         System.out.println("press Ctrl+c to shutdown...");
         while(true){
            Thread.sleep(1000);
         }
      }catch(Exception e){System.out.println(e);}
   }

}
```

File: MyListener.java

```
import javax.jms.*;
public class MyListener implements MessageListener {

   public void onMessage(Message m) {
      try{
      TextMessage msg=(TextMessage)m;

      System.out.println("following message is received:"+msg.getText());
      }catch(JMSException e){System.out.println(e);}
   }
}
```

Câu hỏi 3: Sau khi chạy thử chương trình Sender và Receiver, vận dụng lý thuyết kiến trúc hướng sự kiện đã học trên lớp để giải thích cơ chế chuyền và nhận thông điệp của Sender và Receiver.

3.2. DDS

Ở phần này chúng ta sẽ thực hành để tìm hiểu cách vận hành của mô hình DDS. Cụ thể, chúng ta sẽ cài đặt chương trình nguồn mở OpenDDS.

Cài đặt OpenDDS
Máy của bạn trước khi cài OpenDDS phải cài 3 chương trình sau:

- C++ compiler
- GNU Make
- Perl

Download file .tar.gz phiên bản mới nhất từ link sau:
http://download.ociweb.com/OpenDDS/

Giải nén file bằng lệnh
```
tar -xvzf OpenDDS-3.8.tar.gz
```

Vào thư mục vừa giải nén, gõ 2 lệnh sau để cài đặt:
```
./configure
make
```

Gõ lệnh sau để cài đặt các thông số đường dẫn môi trường:
```
source setenv.sh
```

Sau đó vào thư mục
```
cd OpenDDS-3.8/tests/DCPS/Messenger/
```

Tạo và soạn nội dung file rtps.ini như sau:
[common]
DCPSGlobalTransportConfig=$file
DCPSDefaultDiscovery=DEFAULT_RTPS
[transport/the_rtps_transport]
transport_type=rtps_udp

Sau đó khởi động subscriber:
```
./subscriber -DCPSConfigFile rtps.ini
```

Mở một tag khác để chạy publisher:
(chú ý, vẫn phải chạy lệnh source setenv.sh ở tab mới)
Sau đó vào thư mục như trên và chạy publisher:
```
./publisher -DCPSConfigFile rtps.ini
```

Câu hỏi 4: So sánh JMS và DDS.

4. Kết luận