1. **各个课题所需传输的消息类型**

**具体每个课题所需传输消息类型采用的方式如下(代码见最后)：**

①普通文件类型(dat、xml、cfg、mp4、jpg等数据格式文件)

②字符串信息(简单的消息标识等)

③大型文件(数据量很大的时候) 注：需配置FTP服务器(见FTP服务器搭建)

# 1.1 课题１

## 态势数据/目标检测结果数据

文件类型②

## AIS数据

文件类型②

## 光电平台图像数据

简单消息标识③

# 1.2 课题２

## 地波雷达目标探测结果数据

文件类型(cfg、dat)②

## 地波雷达系统状态

简单消息标识③

# 1.3 课题３

## 无人船实时报文

位置信息、状态信息、目标或障碍物信息③

## AIS数据

文件类型(多条写入文件一起发送)②

字符串类型(单条发送)③

## 光电平台图像视频

简单消息标识③

JPG图片文件和MP4视频文件②

## 无人船系统状态

简单消息标识③

# 1.4 课题５

## AIS数据

文件类型(多条写入文件一起发送)②

Str字符串类型(单条发送)③

# 传输三类消息示例代码：

# 2.1 普通文件类型

**2.1.1 生产者**

package p2p.producer;

import org.apache.activemq.ActiveMQConnectionFactory;

import javax.jms.\*;

import javax.swing.\*;

import java.io.\*;

/\*\*

\* @ClassName StreamProducer

\* @Description 二进制文件传输

\* @Author Anhua

\* @Date 2019/9/14 21:25

\* @Version 1.0

\*/

public class StreamProducer {

// 默认的连接用户名

private static final String USERNAME= "admin";

// 默认的连接密码

private static final String PASSWORD= "admin";

// 默认的连接地址

private static final String BROKE\_URL="tcp://192.168.217.129:61616";

//集群地址

private static final String BROKEFAILOVER\_URL="failover:(tcp://192.168.217.132:61616,tcp://192.168.217.133:61616)?randomize=false";

//连接工厂

private static ConnectionFactory connectionFactor = null;

//连接

private static Connection connection = null;

//会话

private static Session session = null;

//目的地

private static Destination destination = null;

//生产者

private static MessageProducer messageProducer = null;

//消息

private static StreamMessage streamMessage = null;

public static void main(String[] args) {

// 选择文件

JFileChooser fileChooser = new JFileChooser();

fileChooser.setDialogTitle("请选择要传送的文件");

if (fileChooser.showOpenDialog(null) != JFileChooser.APPROVE\_OPTION) {

return;

}

File file = fileChooser.getSelectedFile();

connectionFactor = new ActiveMQConnectionFactory(USERNAME, PASSWORD, BROKEFAILOVER\_URL);

try {

connection = connectionFactor.createConnection();

connection.start();

session = connection.createSession(Boolean.FALSE, Session.AUTO\_ACKNOWLEDGE);

destination = session.createQueue("stream\_queue");

messageProducer = session.createProducer(destination);

//设置优先级

messageProducer.setPriority(9);

//设置持久化模式

messageProducer.setDeliveryMode(DeliveryMode.PERSISTENT);

streamMessage = session.createStreamMessage();

streamMessage.setStringProperty("FILE\_NAME",file.getName());

streamMessage.setStringProperty("FILE\_SIZE",String.valueOf(file.length()));

InputStream in = new FileInputStream(file);

ByteArrayOutputStream bos = new ByteArrayOutputStream();

byte[] buffer = new byte[1024\*128\*2\*2\*2];

int c = -1;

while((c = in.read(buffer)) > 0){

bos.write(buffer);

System.out.println("send: " + c);

}

streamMessage.writeBytes(bos.toByteArray());

messageProducer.send(streamMessage);

System.out.println("发送完成！");

bos.close();

in.close();

messageProducer.close();

session.close();

connection.close();

} catch (Exception e) {

e.printStackTrace();

}

}

}

**2.1.2 消费者**

package p2p.consumer;

import org.apache.activemq.ActiveMQConnectionFactory;

import javax.jms.\*;

import javax.swing.\*;

import java.io.\*;

/\*\*

\* @ClassName StreamConsumer

\* @Description 接收bytes类型数据

\* @Author Anhua

\* @Date 2019/9/14 21:27

\* @Version 1.0

\*/

public class StreamConsumer {

// 默认的连接用户名

private static final String USERNAME= "admin";

// 默认的连接密码

private static final String PASSWORD= "admin";

// 默认的连接地址

private static final String BROKE\_URL="tcp://192.168.217.129:61616";

//集群地址

private static final String BROKE\_FAILOVER\_URL="failover:(tcp://192.168.217.132:61616,tcp://192.168.217.133:61616)?randomize=false";

//连接工厂

private static ConnectionFactory connectionFactor = null;

//连接

private static Connection connection = null;

//会话

private static Session session = null;

//目的地

private static Destination destination = null;

//生产者

private static MessageConsumer messageConsumer = null;

//消息

private static StreamMessage streamMessage = null;

public static void main(String[] args) {

connectionFactor = new ActiveMQConnectionFactory(USERNAME, PASSWORD, BROKE\_FAILOVER\_URL);

try {

connection = connectionFactor.createConnection();

connection.start();

session = connection.createSession(Boolean.FALSE, Session.AUTO\_ACKNOWLEDGE);

destination = session.createQueue("stream\_queue");

messageConsumer = session.createConsumer(destination);

messageConsumer.setMessageListener(

new MessageListener() {

@Override

public void onMessage(Message message) {

try {

streamMessage = (StreamMessage) message;

byte[] bytes = new byte[Integer.valueOf(streamMessage.getStringProperty("FILE\_SIZE"))];

streamMessage.readBytes(bytes);

File file = new File("E:\\"+streamMessage.getStringProperty("FILE\_NAME"));

FileOutputStream fileOutputStream = null;

if(!file.exists()){

file.createNewFile();//如果文件不存在，就创建该文件

fileOutputStream = new FileOutputStream(file);//首次写入获取

}else{

//如果文件已存在，那么就在文件末尾追加写入

fileOutputStream = new FileOutputStream(file,true);//这里构造方法多了一个参数true,表示在文件末尾追加写入

}

fileOutputStream.write(bytes);

fileOutputStream.flush();

fileOutputStream.close();

System.out.println("ID:" + streamMessage.getJMSMessageID());

System.out.println("原始ID" + streamMessage.getStringProperty("originalMsgId"));

System.out.println(System.currentTimeMillis()-streamMessage.getJMSTimestamp());

} catch (Exception e) {

e.printStackTrace();

}

}

}

);

}catch (Exception e){

e.printStackTrace();

}

}

}

# 2.2 字符串信息

**2.2.1 生产者**

package p2p.producer;

import org.apache.activemq.ActiveMQConnectionFactory;

import javax.jms.\*;

/\*\*

\* @ClassName TextProducer

\* @Description 点对点text类型消息生产者

\* @Author Anhua

\* @Date 2019/9/14 21:26

\* @Version 1.0

\*/

public class TextProducer {

// 默认的连接用户名

private static final String USERNAME= "admin";

// 默认的连接密码

private static final String PASSWORD= "admin";

// 默认的连接地址

private static final String BROKE\_URL="tcp://192.168.217.129:61616";

//集群地址

//tcp://localhost:61616?jms.blobTransferPolicy.defaultUploadUrl=http://localhost:8161/fileserver/

private static final String BROKEFAILOVER\_URL="failover:(tcp://192.168.217.132:61616,tcp://192.168.217.133:61616)?randomize=false";

//连接工厂

private static ConnectionFactory connectionFactor = null;

//连接

private static Connection connection = null;

//会话

private static Session session = null;

//目的地

private static Destination destination = null;

//生产者

private static MessageProducer messageProducer = null;

//消息

private static TextMessage textMessage = null;

public static void main(String[] args) {

connectionFactor = new ActiveMQConnectionFactory(USERNAME, PASSWORD, BROKEFAILOVER\_URL);

try {

connection = connectionFactor.createConnection();

connection.start();

session = connection.createSession(Boolean.FALSE, Session.AUTO\_ACKNOWLEDGE);

destination = session.createQueue("text\_queue");

messageProducer = session.createProducer(destination);

//设置优先级

messageProducer.setPriority(9);

//设置持久化模式

messageProducer.setDeliveryMode(DeliveryMode.PERSISTENT);

textMessage = session.createTextMessage();

textMessage.setText("我是消息内容");

messageProducer.send(textMessage);

messageProducer.close();

session.close();

connection.close();

} catch (JMSException e) {

e.printStackTrace();

}

}

}

**2.2.2 消费者**

package p2p.consumer;

import org.apache.activemq.ActiveMQConnectionFactory;

import javax.jms.\*;

/\*\*

\* @ClassName TextConsumer

\* @Description text类型消息消费者

\* @Author Anhua

\* @Date 2019/9/14 21:28

\* @Version 1.0

\*/

public class TextConsumer {

// 默认的连接用户名

private static final String USERNAME= "admin";

// 默认的连接密码

private static final String PASSWORD= "admin";

// 默认的连接地址

private static final String BROKE\_URL="tcp://192.168.217.129:61616";

//集群地址

private static final String BROKE\_FAILOVER\_URL="failover:(tcp://192.168.217.132:61616,tcp://192.168.217.133:61616)?randomize=false";

//连接工厂

private static ConnectionFactory connectionFactor = null;

//连接

private static Connection connection = null;

//会话

private static Session session = null;

//目的地

private static Destination destination = null;

//生产者

private static MessageConsumer messageConsumer = null;

//消息

private static TextMessage textMessage = null;

public static void main(String[] args) {

connectionFactor = new ActiveMQConnectionFactory(USERNAME, PASSWORD, BROKE\_FAILOVER\_URL);

try {

connection = connectionFactor.createConnection();

connection.start();

session = connection.createSession(Boolean.FALSE, Session.AUTO\_ACKNOWLEDGE);

destination = session.createQueue("text\_queue");

messageConsumer = session.createConsumer(destination);

messageConsumer.setMessageListener(

new MessageListener() {

public void onMessage(Message message) {

try {

TextMessage textMessage = (TextMessage) message;

System.out.println("收到的消息：" + textMessage.getText());

System.out.println("ID:" + textMessage.getJMSMessageID());

System.out.println("原始ID" + textMessage.getStringProperty("originalMsgId"));

System.out.println(System.currentTimeMillis()-textMessage.getJMSTimestamp());

} catch (JMSException e) {

e.printStackTrace();

}

}

}

);

}catch (Exception e){

e.printStackTrace();

}

}

}

# 2.3 大型文件

**2.3.1 生产者**

package p2p.producer;

import org.apache.activemq.ActiveMQConnectionFactory;

import org.apache.activemq.ActiveMQSession;

import org.apache.activemq.command.ActiveMQBlobMessage;

import javax.jms.\*;

import javax.swing.\*;

import java.io.File;

/\*\*

\* @ClassName BlobConsumer

\* @Description 大二进制文件生产者

\* @Author Liu Ketao

\* @Date 2019/9/14 21:25

\* @Version 1.0

\*/

public class BlobProducer {

public static void main(String[] args) {

// 选择文件

JFileChooser fileChooser = new JFileChooser();

fileChooser.setDialogTitle("请选择要传送的文件");

if (fileChooser.showOpenDialog(null) != JFileChooser.APPROVE\_OPTION) {

return;

}

File file = fileChooser.getSelectedFile();

// 获取 ConnectionFactory

ConnectionFactory connectionFactory = new ActiveMQConnectionFactory(

"tcp://192.168.217.132:61616?jms.blobTransferPolicy.defaultUploadUrl=ftp://anhua2015:0327@192.168.217.132/XXX");

try {

// 创建 Connection

Connection connection = connectionFactory.createConnection();

connection.start();

// 创建 Session

ActiveMQSession session = (ActiveMQSession) connection.createSession(false, Session.AUTO\_ACKNOWLEDGE);

// 创建 Destination

Destination destination = session.createQueue("File.Transport");

// 创建 Producer

MessageProducer producer = session.createProducer(destination);

// 设置持久性的话，文件也可以先缓存下来，接收端离线再连接也可以收到文件,非持久性为NON\_PERSISTENT

producer.setDeliveryMode(DeliveryMode.PERSISTENT);

// 构造 BlobMessage，用来传输文件

ActiveMQBlobMessage blobMessage = (ActiveMQBlobMessage) session.createBlobMessage(file);

blobMessage.setStringProperty("FILE.NAME", file.getName());

blobMessage.setLongProperty("FILE.SIZE", file.length());

blobMessage.setDeletedByBroker(true);

System.out.println("开始发送文件：" + file.getName() + "，文件大小：" + file.length() + " 字节");

// 发送文件

producer.send(blobMessage);

System.out.println("完成文件发送：" + file.getName());

producer.close();

session.close();

// 不关闭 Connection, 程序则不退出

connection.close();

}catch (Exception e){

e.printStackTrace();

}

}

}

**2.3.2 消费者**

package p2p.consumer;

import org.apache.activemq.ActiveMQConnectionFactory;

import org.apache.activemq.BlobMessage;

import org.apache.activemq.command.ActiveMQBlobMessage;

import org.apache.commons.net.ftp.FTPClient;

import org.apache.commons.net.ftp.FTPReply;

import javax.jms.\*;

import javax.swing.\*;

import java.io.\*;

import java.net.MalformedURLException;

/\*\*

\* @ClassName BlobConsumer

\* @Description TODO

\* @Author Anhua

\* @Date 2019/9/14 21:27

\* @Version 1.0

\*/

public class BlobConsumer {

//ftp服务器地址

public static String FTP\_HOST\_NAME = "192.168.217.132";

//ftp服务器端口号默认为21

public static Integer FTP\_PORT = 21 ;

//ftp登录账号

public static String FTP\_USER\_NAME = "anhua2015";

//ftp登录密码

public static String FTP\_PASSWORD = "0327";

public static FTPClient ftpClient = null;

public static void main(String[] args) {

// 获取 ConnectionFactory

ConnectionFactory connectionFactory = new ActiveMQConnectionFactory("failover:(tcp://192.168.217.132:61616" +

",tcp://192.168.217.133:61616)?randomize=false");

try {

// 创建 Connection

Connection connection = connectionFactory.createConnection();

connection.start();

// 创建 Session

Session session = connection.createSession(false, Session.CLIENT\_ACKNOWLEDGE);

// 创建 Destination

Destination destination = session.createQueue("File.Transport");

// 创建 Consumer

MessageConsumer consumer = session.createConsumer(destination);

// 注册消息监听器，当消息到达时被触发并处理消息

consumer.setMessageListener(new MessageListener() {

// 监听器中处理消息

@Override

public void onMessage(Message message) {

if (message instanceof BlobMessage) {

ActiveMQBlobMessage blobMessage = (ActiveMQBlobMessage) message;

try {

String fileName = blobMessage.getStringProperty("FILE.NAME");

System.out.println("文件接收请求处理：" + fileName + "，文件大小：" + blobMessage.getLongProperty("FILE.SIZE")

+ " 字节");

JFileChooser fileChooser = new JFileChooser();

fileChooser.setDialogTitle("请指定文件保存位置");

fileChooser.setSelectedFile(new File(fileName));

if (fileChooser.showSaveDialog(null) == JFileChooser.APPROVE\_OPTION) {

File file = fileChooser.getSelectedFile();

OutputStream os = new FileOutputStream(file);

System.out.println("开始接收文件：" + fileName);

InputStream inputStream = blobMessage.getInputStream();

// 写文件，你也可以使用其他方式

byte[] buff = new byte[256];

int len = 0;

while ((len = inputStream.read(buff)) > 0) {

os.write(buff, 0, len);

}

//删除ftp服务器文件

//blobMessage.deleteFile();

ftpOption(blobMessage);

blobMessage.acknowledge();

os.close();

System.out.println("完成文件接收：" + fileName);

}

} catch (Exception e) {

e.printStackTrace();

}

}

}

});

}catch (Exception e){

e.printStackTrace();

}

}

/\*\*

\* 删除ftp服务器上的文件

\*/

public static void ftpOption(ActiveMQBlobMessage blobMessage) {

ftpClient = new FTPClient();

ftpClient.setControlEncoding("utf-8");

try {

System.out.println("connecting...ftp服务器:"+FTP\_HOST\_NAME+":"+FTP\_PORT);

//连接ftp服务器

ftpClient.connect(FTP\_HOST\_NAME, FTP\_PORT);

//登录ftp服务器

ftpClient.login(FTP\_USER\_NAME, FTP\_PASSWORD);

//是否成功登录服务器

int replyCode = ftpClient.getReplyCode();

if(!FTPReply.isPositiveCompletion(replyCode)){

System.out.println("connect failed...ftp服务器:"+FTP\_HOST\_NAME+":"+FTP\_PORT);

}

System.out.println("connect successfu...ftp服务器:"+FTP\_USER\_NAME+":"+FTP\_PORT);

//切换FTP目录

ftpClient.changeWorkingDirectory("opt/anhua2015");

String[] str = blobMessage.getURL().toString().split("/");

ftpClient.dele(str[str.length-1]);

ftpClient.logout();

if(ftpClient.isConnected()){

ftpClient.disconnect();

}

}catch (Exception e) {

e.printStackTrace();

}

}

}

# FTP服务搭建

## 3.1 匿名服务搭建

**3.1.1 ftp服务器配置**

vi /etc/vsftpd/vsftpd.conf，具体内容如下：

anonymous\_enable=YES

anon\_upload\_enable=YES

anon\_mkdir\_write\_enable=YES

**3.1.2 ftp目录的权限设置**

默认情况下，ftp的根目录为/var/ftp，为了安全，这个目录默认不允许设置为777权限，否则ftp将无法访问。但是我们要匿名上传文件，需要“other”用户的写权限，正确的做法：

在/var/ftp中建立一个upload（名子自己起吧）文件夹，将个文件夹权限设置为777（视具体需要自己设），在upload这个文件夹中，匿名用户可以上传文件、创建文件夹、删除文件等。

vsftpd增强了安全检查，如果用户被限定在了其主目录下，则该用户的主目录不能再具有写权限了。可以用命令chmod a-w /var/ftp 去除用户主目录的写权限。

一般至此，便实现vsftpd匿名用户的上传下载了。如果还不行，就是下面的问题。

**3.1.3**

SELinux(Security-Enhanced Linux) 是美国国家安全局（NAS）对于强制访问控制的实现，是 Linux上最杰出的新安全子系统。NSA是在Linux社区的帮助下开发了一种访问控制体系，在这种访问控制体系的限制下，进程只能访问那些在他的任务中所需要文件。SELinux 默认安装在 Fedora 和 Red Hat Enterprise Linux 上，也可以作为其他发行版上容易安装的包得到。

方法1：用命令setenforce 0，无需重启。（setenforce的格式：setenforce [ Enforcing | Permissive | 1 | 0 ]）

方法2：在lilo或者grub的启动参数中增加：selinux=0,也可以关闭selinux。

使用getenforce查看当前selinux是否正在运行。不关闭selinux，就要设置selinux的ftp权限。

（1）使用getsebool -a | grep ftp查看ftp相关设置状态，我们要将allow\_ftpd\_anon\_write设为on。

（2）使用setsebool -P 进行设置。例：setsebool -P allow\_ftpd\_anon\_write=on。

或使用togglesebool进行bool值取反，例如togglesebool　allow\_ftpd\_anon\_write。

（3）修改selinux安全上下文，先介绍两个命令：

命令1、ls -Zps -Zid -Z# 分别可以看到文件,进程和用户的SELinux属性

命令2、#chcon 改变SELinux安全上下文

chcon -u [user] 对象 -r [role] -t [type] -R 递归 --reference 源文件 目标文件 # 复制安全上下文。

使用方法：

步骤1、ls -Zd /var/ftp/upload/ 通常会看到：

drwxr-xr-x ftp root system\_u:object\_r:public\_content\_t /var/ftp/upload/

步骤2、chcon -R -t public\_content\_rw\_t /var/ftp/upload/

步骤3、ls -Zd /var/ftp/upload/ 如果看到如下信息就OK了：

drwxr-xr-x ftp root system\_u:object\_r:public\_content\_rw\_t /var/ftp/upload/

最后还是重启下selinux和vsftpd吧，不重启其实也没关系。重新登录到ftp上，应该就能解决问题了。

另外，selinux的图形界面可由system-config-selinux命令进入。

## 3.2 Vsftpd.conf参数说明

# 禁止匿名用户anonymous登录

anonymous\_enable=NO

# 允许本地用户登录

local\_enable=YES

local\_root=/data/wwwroot/

# 让登录的用户有写权限(上传，删除)

write\_enable=YES

# 默认umask

local\_umask=022

# 把传输记录的日志保存到/var/log/vsftpd.log

dual\_log\_enable=YES

dirmessage\_enable=YES

xferlog\_enable=YES

xferlog\_file=/var/log/vsftpd.log

xferlog\_std\_format=NO

# 允许ASCII模式上传

ascii\_upload\_enable=YES

# 允许ASCII模式下载

ascii\_download\_enable=YES

# 使用20号端口传输数据

connect\_from\_port\_20=YES

connect\_timeout=60

accept\_timeout=60

idle\_session\_timeout=300

max\_clients=2000

max\_per\_ip=0

# 欢迎标语

ftpd\_banner=Welcome to use my test ftp server.

#限制用户到指定目录

allow\_writeable\_chroot=YES

chroot\_local\_user=yes

#chroot\_list\_enable=YES

#chroot\_list\_file=/etc/vsftpd/chroot\_list

use\_localtime=YES

listen=YES

pam\_service\_name=vsftpd

anon\_max\_rate=50000

local\_max\_rate=200000

#主动模式开：需要防火墙放开20和21号端口

#port\_enable=yes

#被动模式开：需要防火墙放开21号端口和高位端口（）

#pasv\_enable=YES

#pasv\_min\_port=30000

#pasv\_max\_port=35000

#指定被动模式监听ip

#pasv\_address=172.16.1.131

## 3.3 配置本地用户

创建用户

登录Linux主机后，运行命令：”useradd ftpname -s /sbin/nologin “。该账户路径默认指向/home/ftpadmin目录；如果需要将用户指向其他目录，请运行命令：useradd ftpname-s /sbin/nologin –d /www(其他目录) 或者执行 adduser -d /data/ -g ftp -s /sbin/nologin ftpname

【修改用户指向目录运行命令：useradd -d /home/www ftpname】

设置ftpadmin用户密码，运行命令：”passwd ftpadmin” ; 输入两次密码，匹配成功后，就设置好了ftpadmin用户的密码了。

创建chroot\_list文件，添加新创建的用户。

修改用户家目录权限，/home/用户名 755，/home/用户名/pub 777

修改vsftpd.conf文件

修改local\_umask=002

将匿名模式关掉，然后追加如下内容。

# 允许ASCII模式上传

ascii\_upload\_enable=YES

# # 允许ASCII模式下载

ascii\_download\_enable=YES

check\_shell=NO

seccomp\_sandbox=NO

connect\_timeout=60

accept\_timeout=60

idle\_session\_timeout=300

max\_clients=2000

max\_per\_ip=0

allow\_writeable\_chroot=YES

#是否将所有用户限制在主目录

chroot\_local\_user=YES

#是否启动限制用户的名单

chroot\_list\_enable=YES

chroot\_list\_file=/etc/vsftpd/chroot\_list

#允许user\_list用户登录

userlist\_deny=NO

userlist\_enable=YES

local\_root=/home/用户名/pub

修改selinux

setenforce 0 #暂时让SELinux进入Permissive模式

getsebool -a | grep ftpd 判断查看权限

setsebool -P tftp\_home\_dir 1

setsebool -P allow\_ftpd\_full\_access 1

setenforce 1 #进入Enforcing模式

修改/etc/pam.d/vsftpd文件

注释掉auth required pam\_shells.so