概述

以ActiveMQ + Log4j + Spring的技术组合，实现基于消息队列的统一日志服务。
参考：Spring+Log4j+ActiveMQ实现远程记录日志——实战+分析

与参考文章的比较

1. 更新了技术的版本
   e.g. Spring升级到4.2.0，ActiveMQ升级到5.13.2
2. 更新了依赖
   e.g. 使用activemq-client 5.13.2替换activemq-core 5.7.0，并取消了多余的spring-jms依赖
3. 精简了配置
   e.g. 去掉spring.xml中的jmsTemplate
4. 其他略述

前提

为理解文章的内容，你可能需要先了解下面的知识：

1. 了解基于Maven的项目结构
2. 下载并运行ActiveMQ
3. 了解log4j基于properties配置的简单用法
4. 了解基于Spring-webmvc的框架的搭建

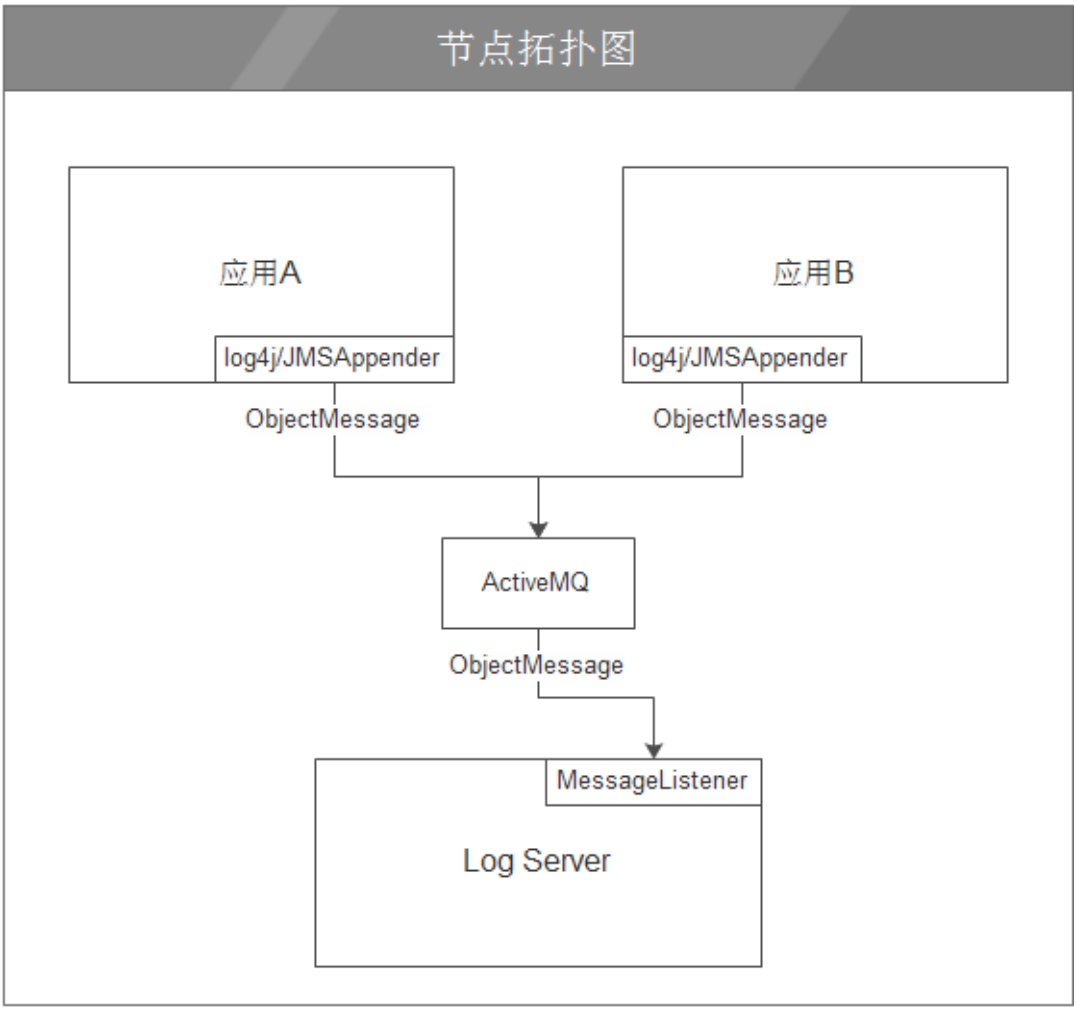
当然，这只是建议.....

技术版本

1. ActiveMQ - 5.13.2
2. Log4j - 1.2.17
3. Spring - 4.2.4

结构图

1.节点拓扑图



说明：

1. 应用系统基于log4j规范，通过JMSAppender将日志发送到ActiveMQ
2. Log Server向ActiveMQ订阅消息，并指定MessageListener的实现来接收ActiveMQ发布的消息

实现

1.Log Server

你可以从amqlog-server拿到源代码。

1.1.文件目录结构

```
pom.xml
src/main/webapp/
 |---- WEB-INF/
          |---- web.xml
 |---- index.jsp # 忽略
src/main/resources/
 |---- spring-beans.xml
 |---- topic.properties # 集中管理修改概率比较高的属性配置
src/main/java/
```

```
    |---- cn.sinobest.asj.logserver
              |---- LogListener.java # 接收并输出log message
```

1.2.文件内容

1.2.1. pom.xml

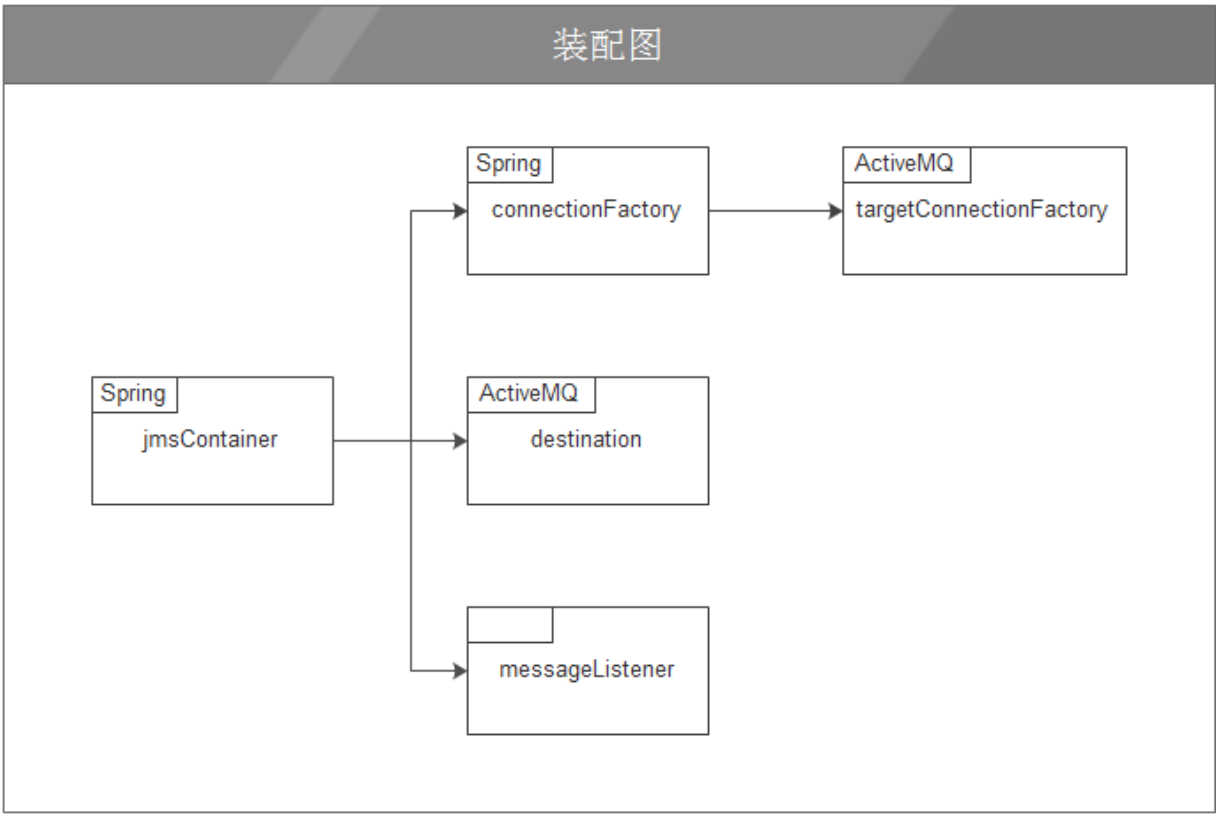
```
<project xmlns="http://maven.apache.org/POM/4.0.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
 <modelVersion>4.0.0</modelVersion>
 <groupId>cn.sinobest.asj</groupId>
 <artifactId>log-servler</artifactId>
 <packaging>war</packaging>
 <version>0.0.1-SNAPSHOT</version>
 <name>amqlog-servler Maven Webapp</name>
 <url>http://maven.apache.org</url>
 <description>日志服务器，从ActiveMQ订阅主题，从而获取相关的日志数据</description>
 <dependencies>
  <dependency>
   <groupId>junit</groupId>
   <artifactId>junit</artifactId>
   <version>3.8.1</version>
   <scope>test</scope>
  </dependency>
  <!-- use to import spring-webmvc framework -->
  <dependency>
   <groupId>org.springframework</groupId>
   <artifactId>spring-web</artifactId>
   <version>4.2.4.RELEASE</version>
  </dependency>
  <dependency>
   <groupId>org.springframework</groupId>
   <artifactId>spring-jms</artifactId>
   <version>4.2.4.RELEASE</version>
  </dependency>
  <!-- use to subscribe topic message from ActiveMQ -->
  <dependency>
   <groupId>org.apache.activemq</groupId>
   <artifactId>activemq-client</artifactId>
   <version>5.13.2</version>
  </dependency>
  <!-- use to extract log content from message -->
  <dependency>
   <groupId>log4j</groupId>
   <artifactId>log4j</artifactId>
   <version>1.2.17</version>
  </dependency>
 </dependencies>
 <build>
  <finalName>amqlog-servler</finalName>
 </build>
</project>
```

1.2.2. web.xml

```
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns="http://java.sun.com/xml/ns/javaee"
 xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
 xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
 id="WebApp_ID" version="3.0" metadata-complete="false">
 <display-name>Archetype Created Web Application</display-name>
 <context-param>
  <param-name>contextConfigLocation</param-name>
  <param-value>
    classpath:spring-beans.xml
   </param-value>
 </context-param>
 <listener>
  <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
 </listener>
</web-app>
```

1.2.3. spring-beans.xml

装配图



说明：左上角标识由谁提供具体的实现，没有标识的由自己提供实现。

内容

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:context="http://www.springframework.org/schema/context"
 xsi:schemaLocation="http://www.springframework.org/schema/beans
  http://www.springframework.org/schema/beans/spring-beans-4.2.xsd
  http://www.springframework.org/schema/context
  http://www.springframework.org/schema/context/spring-context-4.2.xsd">
 <context:property-placeholder location="classpath:topic.properties" />

 <bean id="targetConnectionFactory" class="org.apache.activemq.ActiveMQConnectionFactory">
  <property name="brokerURL" value="${topic.brokerURL}" />
  <!-- add trusted packages. see http://activemq.apache.org/objectmessage.html -->
  <property name="trustedPackages">
   <list>
    <value>org.apache.log4j.spi</value>
   </list>
  </property>
 </bean>
 <bean id="connectionFactory"
  class="org.springframework.jms.connection.SingleConnectionFactory">
  <property name="targetConnectionFactory" ref="targetConnectionFactory" />
 </bean>

 <bean id="destination" class="org.apache.activemq.command.ActiveMQTopic">
  <constructor-arg name="name" value="${topic.topicName}" />
 </bean>

 <!-- define the message-listener to receive and dipose log data. -->
 <bean id="messageListener" class="cn.sinobest.asj.logserver.LogListener" />

 <bean id="jmsContainer"
  class="org.springframework.jms.listener.DefaultMessageListenerContainer">
  <property name="connectionFactory" ref="connectionFactory" />
  <property name="destination" ref="destination" />
  <property name="messageListener" ref="messageListener" />
 </bean>
</beans>
```

说明：

在targetConnectionFactory的属性中，指定了trustedPackages。ActiveMQ自5.12.2版本之后，强制用户指定一份可信任的packages白名单，以对付ObjectMessage存在的安全漏洞。具体内容可参考：http://activemq.apache.org/objectmessage.html。

1.2.4. topic.properties

```
topic.brokerURL=tcp://localhost:61616
topic.topicName=demo
```

注意：brokerURL的值必须和ActiveMQ的监听地址一致。

1.2.5. LogListener.java

```
package cn.sinobest.asj.logserver;

import javax.jms.JMSException;
import javax.jms.Message;
import javax.jms.MessageListener;

import org.apache.activemq.command.ActiveMQObjectMessage;
import org.apache.log4j.spi.LoggingEvent;

public class LogListener implements MessageListener {

 private static final String TEMPLATE = "[%-5s] %s";

 public void onMessage(Message message) {
  try {
   // extract LoggingEvent from message
   // you must set org.apache.log4j.spi into the trusted packages list
   // see spring-beans.xml in classpath
   LoggingEvent event = (LoggingEvent) ((ActiveMQObjectMessage) message)
     .getObject();
   String content = String.format(TEMPLATE, event.getLevel()
     .toString(), event.getMessage().toString());
   System.out.println(content);
  } catch (JMSException e) {
   e.printStackTrace();
  }
 }
}
```

说明：这里的LoggingEvent来自package org.apache.log4j.spi，该package在spring-beans.xml的白名单中。

2.Log Client

log client模拟一般的应用系统。该应用系统有日志存储的需要，将日志发送给ActiveMQ而不用关心日志最终的存储方式。这里仅用一个简单的JavaSE project来模拟，但是已经足够提供完整的核心代码。
你可以从amqlog-client拿到源代码。

2.1.文件目录结构

```
pom.xml
src/main/resources/
 |---- log4j.properties # 配置日志输出地点，及ActiveMQ的相关参数
 |---- jndi.properties # 配置topic
src/main/java/
 |---- cn.sinobest.asj.logclient
           |---- LogProducer.java # 生成并输出日志
```

2.2.文件内容

2.2.1. pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
 <modelVersion>4.0.0</modelVersion>
 <groupId>cn.sinobest.asj</groupId>
 <artifactId>amqlog-client</artifactId>
 <version>0.0.1-SNAPSHOT</version>
 <name>Simple app to send log to ActiveMQ</name>
 <description>模拟一般的应用系统，通过log4j发送日志到ActiveMQ</description>
 <dependencies>
  <!-- use to write log -->
  <dependency>
   <groupId>log4j</groupId>
   <artifactId>log4j</artifactId>
   <version>1.2.17</version>
  </dependency>
  <dependency>
   <groupId>commons-logging</groupId>
   <artifactId>commons-logging</artifactId>
   <version>1.1.1</version>
  </dependency>
  <!-- use to import class org.apache.activemq.jndi.ActiveMQInitialContextFactory
   to write log to ActiveMQ -->
  <dependency>
   <groupId>org.apache.activemq</groupId>
   <artifactId>activemq-client</artifactId>
   <version>5.13.2</version>
  </dependency>
 </dependencies>
</project>
```

2.2.2. log4j.properties

```
# define the stand out appender
log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern=[%-5p] %-d{yyyy-MM-dd HH:mm:ss z}%n        %m%n%n

# define the jms appender
log4j.appender.jms=org.apache.log4j.net.JMSAppender
log4j.appender.jms.InitialContextFactoryName=org.apache.activemq.jndi.ActiveMQInitialContextFactory
log4j.appender.jms.ProviderURL=tcp://localhost:61616
# TopicBindingName可以自由配置，只需要确保提供对应的jndi属性即可
log4j.appender.jms.TopicBindingName=topicName
# TopicConnectionFactoryBindingName目前不能自由配置
```

```
log4j.appender.jms.TopicConnectionFactoryBindingName=ConnectionFactory

# define the logger
log4j.rootLogger=INFO, stdout, jms
```

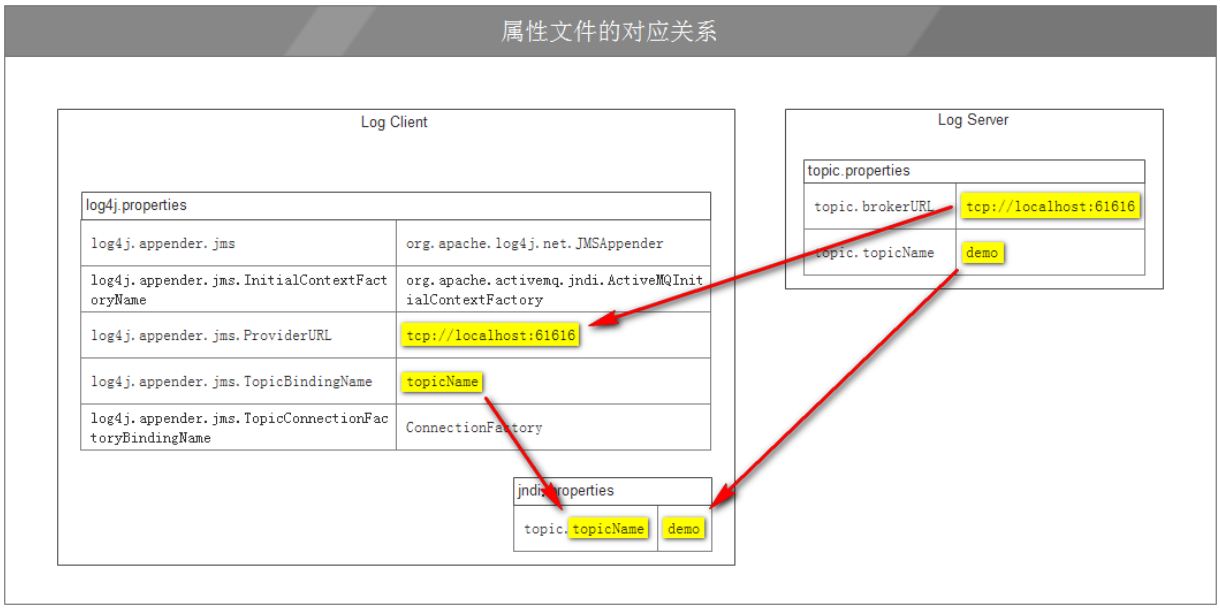
注意：log4j.appender.jms.ProviderURL的值必须和ActiveMQ的监听地址一致。

2.2.3. jndi.properties

```
topic.topicName=demo
```

注意：key的后半部分（topicName）必须与log4j.properties中的log4j.appender.jms.TopicBindingName一致。

属性间的对应关系



2.2.4. LogProducer.java

```
package cn.sinobest.asj.logclient;

import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;

public class LogProducer {
 private static final Log log = LogFactory.getLog(LogProducer.class);

 /**
  * @param args
  */
 public static void main(String[] args) {
  log.debug("this is a debug message.");
  log.info("this is a info message.");
  log.warn("this is a warn message.");
  log.error("this is a error message");
```

```
17.      System.exit(0);
18.     }
19.
20.    }
```

说明：debug的内容不会发送到ActiveMQ。

测试

1. 启动ActiveMQ
   cd到ActiveMQ的解压缩目录，在cmd执行bin\activemq start
2. 部署Log Server到Tomcat并启动
3. 运行Log Client的LogProducer main方法
4. Log Server的Console会有：

```
49 [INFO ] this is a info message.
50 [WARN ] this is a warn message.
51 [ERROR] this is a error message
```