1 程序功能：实现群聊，类似于 QQ 群聊天。

2 编程语言：Java

3 基于网络（Socket）实现

群聊中的服务器不是必须的，但加入服务器可以简化实现，程序中采用了 C/S（客户端/服务器）模式。客户端将消息发送给服务器，服务器接收到客户端的消息后将其分发给所有客户端（包括消息的发送者），实现群聊功能。

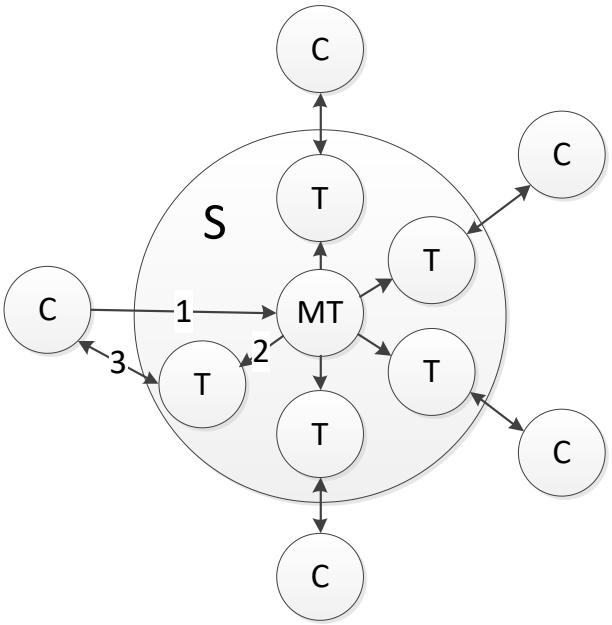

图 1 C/S 模式系统结构

C 表示客户端，S 表示服务器，MT 表示服务器主线程，用于处理客户端的初次连接请求，T 表示服务器中接收、分发客户端消息的线程，与客户端一一对应。客户端和服务器之间的通信过程在后面叙述。

3.1 客户端（Client）的实现：

3.1.1 图形界面

如图 1 所示，界面中包括 1 个 JFrame：用来容纳其它控件，记为"frame"，2 个 JTextArea：其中一个用来展示历史消息，记为"history"，另一个用来编辑需要发送的消息，记为"current"，1 个 JButton：用来发送消息，记为"send"。

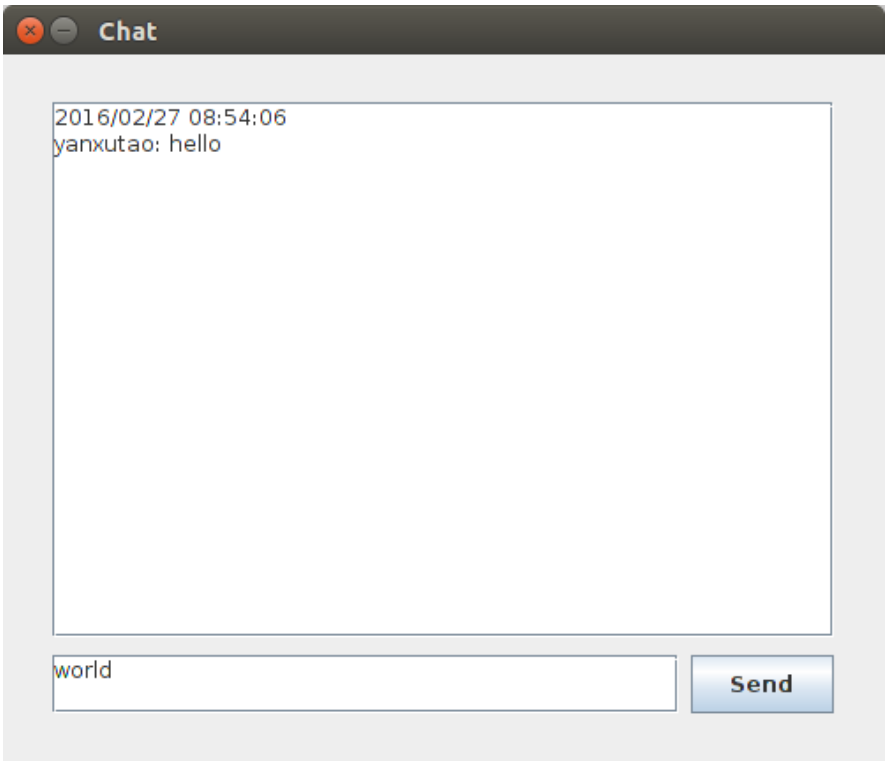

图 2 客户端图形界面

3.1.2 发送消息

用户点击发送按钮"send"时触发一个事件，在事件的响应代码中将"current"中的内容发送给服务器。

3.1.2 接收消息

客户端接收服务器发给自己的消息，将消息内容添加到"history"中。

3.2 服务器(Server)的实现

服务器端采用“一客户一线程”的模式。客户端启动之后向服务器发起连接请求，服务器主线程新建一个线程来处理该客户端的所有请求。主线程只负责处理客户端的初次连接请求，不参与任何消息的接收和分发。当前所有连接的 Socket 信息存储在一个 ArrayList 中，所有的线程都可以访问这个 ArrayList。当某个线程接收到自己的客户端发来的消息时，该线程将消息发给所有正在连接的客户端。

这样做的原因是：简单；效率较高，不需要在每次发送/接收消息时都建立 TCP 连接，TCP 建立连接时的三次握手比较耗时。

可能出现的问题是：存在单点故障，服务器可能会当掉，当客户端非常多时，会耗尽服务器的线程资源，达到系统允许的上线，导致新来的客户端无法连接到服务器。

4 基于中间件（Ice）实现

群聊可以用发布-订阅（publish/subscribe）模型来表示，每个客户端对群里的消息感兴趣。这里比较特殊的是客户端既是发布者，又是订阅者。

IceStorm 可以为 Ice 应用提供发布/订阅服务。程序基于 IceStorm 实现，实现过程中参考了 ice-demos 中的 clock 程序。

IceStorm 可以看作是服务器，所以只需要编写客户端即可。

客户端(Client)的实现：图形界面部分同上；客户端启动之后，首先将自己注册为 IceStorm 中"chat"主题的订阅者和发布者，然后初始化图形界面，最后等待发送/接收消息。

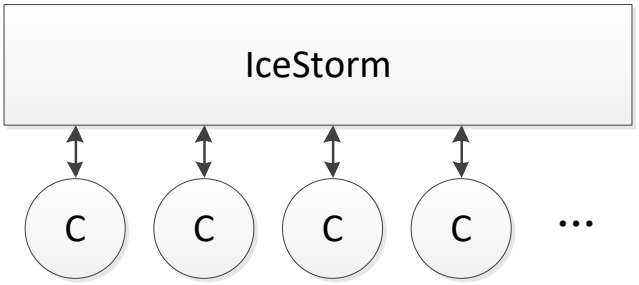

图 3 基于 IceStorm 实现的系统结构

5 编程中遇到的问题和解决方法

5.1 用 Socket 实现时如何确定消息的边界

程序中设计了一个简单的应用层协议。程序中的客户端和服务器以行（hang，"\n"）为单位来接收消息。收到"END_OF_MESSAGE"这行消息代表上一条消息已经发送完毕，可以进行下一步的处理了：分发给正在连接的客户端（服务器中的线程）或者呈现给用户（客户端）。客户端断开连接时向服务器发送"END_OF_CHAT"消息，服务器中对应这个客户端的线程收到这条消息后将该客户端的连接信息从 ArrayList 中删除，最后线程正常返回。

5.2 关于逻辑时钟

因为两种实现中都存在单点：Socket 实现中处理连接请求的主线程，Ice 实现中的 IceStorm，所以添加逻辑时钟比较简单：处理连接请求的主线程或者 IceStorm 给收到的消息添加时间戳即可。但这样做存在一些问题：客户端收到消息的时间有可能早于这条消息的时间戳，即客户端收到一条“未来的消息”。另外，在群聊这种应用场景中，所有消息之间严格的先后关系并不是特别重要，加上网络传输延迟是可变的，所以这种先后关系很难甚至无法确定。所以在群聊这种应用场景中不需要考虑逻辑时钟的概念，消息的时间以客户端收到消息的时间为准，这样做既符合常理又容易实现。

5.3 关于单点故障和可扩展性

单点故障和可扩展性之间的关系是：如果存在单点故障，那么可扩展性就不会太好，单点的存在势必会影响到程序的可扩展性。基于 Socket 的实现中没有考虑单点故障的问题。IceStorm 通过主从复制（master-slave replication）的手段提供了一个高可用（HA）模式。

5.4 关于分布式对象中间件

分布式对象中间件屏蔽了不同计算机系统之间的异构性和底层的网络通信

细节，使得应用程序可以像调用本地对象一样去调用远程对象。

6 程序运行截图

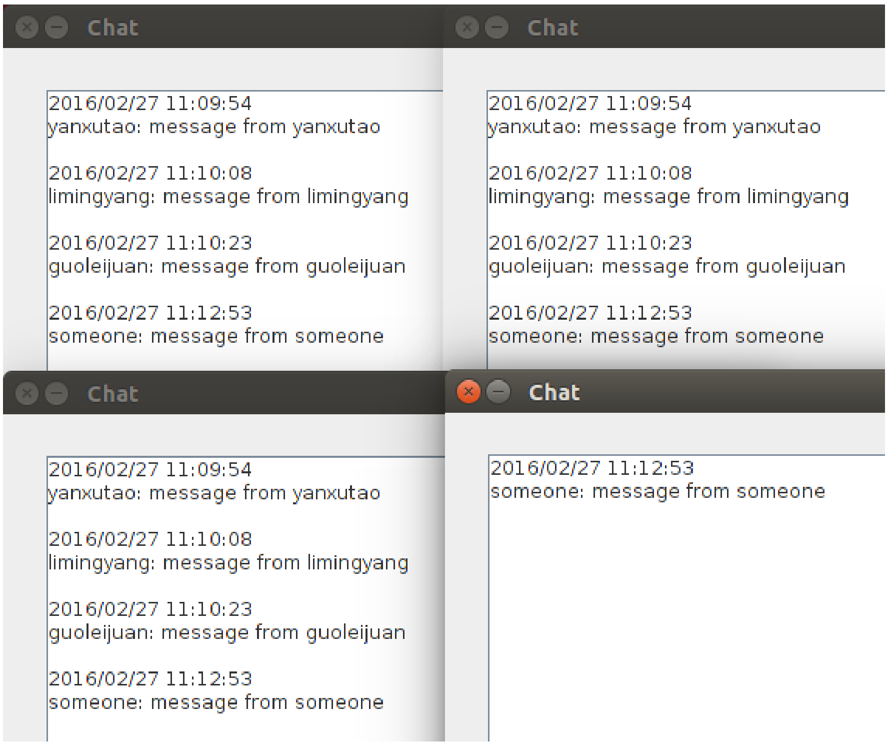

图 4 程序运行截图

上图中 4 个客户端是在同一台机器上运行的。基于 Socket 的实现中将服务器的 IP 地址（127.0.0.1）硬编码在了代码中，更为合理的做法是使用配置文件，以便更改服务器的 IP 地址。基于 IceStorm 的实现中使用了配置文件（config.chat）。