

文章编号: 1005-0523(2009)02-0076-05

基于 ICE 中间件自定义 Session 机制的远程监控系统

程 锦, 李 磊, 杨 诚

(四川大学 计算机学院, 四川 成都 640014)

摘要: 为了满足实时远程监测系统的需求, 服务器需要实时监测客户端的连接状态。为此利用 ICE 中间件的优点设计了一种基于面向对象 ICE 中间件自定义 Session 机制的解决方案, 将其应用在远程监控系统中。并以实验来验证该方案在保证系统的实时性和可靠性方面提供了一种新方法。

关 键 词: ICE 中间件; 自定义 Session; 远程监控系统

中图分类号: TP393

文献标识码: A

在众多实际应用中, 服务器需要实时监控客户端的运行状态, 在无法保证网络通信可靠性的环境中, 可能会出现各种通信异常, 比如: 网络堵塞, 网络中断。所以需要实时监测连接状态。传统的方法是采用基于 Socket 的心跳机制, 但是 Socket 不具有 ICE 中间件的简单而高效的协议。尤其在大型项目中 Socket 通信机制的实现具有较大局限性和复杂度, 比如跨平台和分布式的实现。为此, 我们选择 ICE 中间件作为网络通信平台, 并设计一种自定义的 Session 机制替代传统的 Socket 心跳机制^[1]。

1 ICE 概述

1.1 ICE 简介

ICE (Internet Communications Engine) 是 ZeroC 公司推出的一种面向对象中间件平台, 它具有高效的网络通信能力, 支持分布式的部署管理、消息中间件以及网络计算等^[2]。从根本上说, ICE 为构建面向对象的客户-服务器应用提供了工具、API 和库支持。ICE 应用适合在异构环境中, 并且可以使用多种网络技术进行通信。无论部署环境如何, 这些应用的源码都是可移植的。

1.2 ICE 的优点

(1) 一种高性能、易于使用、面向对象模型和基于分布式系统的中间件。所有的操作调用都使用迟后绑定;

(2) 提供丰富的调用和分派模型, 支持同步和异步的消息传递;

(3) 采用简单的语言映射规则, 与上层的编程语言无关, 客户端和服务端可以分别采用不同的语言;

(4) ICE 适用于异构平台环境中, 可以运行在不同的操作系统和机器架构上, 使用多种网络技术进行通信;

(5) 提供线程安全和内存管理机制;

收稿日期: 2008-12-16

中国知网 <https://www.cnki.net>

作者简介: 程 锦 (1986-), 女, 江西乐平人, 硕士研究生, 主要研究方向为实时软件工程和网络安全。

- (6) 简单而高效的协议,可提高应用程序的性能和减少对网络资源的占用;
- (7) 内建的安全机制,可穿透防火墙和 NAT。可以在不安全的公网上安全传输,而无需要专门的 VPN;
- (8) ICE 拥有内建的对象持久服务,能够轻松地在数据库中存储对象状态^[3]。

2 Session机制的设计

2.1 基本原理

基于 ICE这种高效的网络通信平台,我们可以利用客户端 Session代理定时刷新服务器对象,返回刷新状态或超时信息,以实时判断是否与服务器成功连接。如果返回成功则表示与服务器仍保持连接,否则失败。而在服务器端,分别保存了每个客户端的一个 Session对象,用来处理来该客户端的刷新请求。服务器端响应刷新请求的实现只是更新时间戳,然后马上返回。服务器端会定时取出本地客户端列表的最近时间戳和当前时间比较,如果超过某个时间差则判断该客户端已经断开连接,并且把它从客户端列表上删除。

2.2 客户端设计

按照上述 Session的基本原理,客户端的 Session设计框架图如图 1所示。

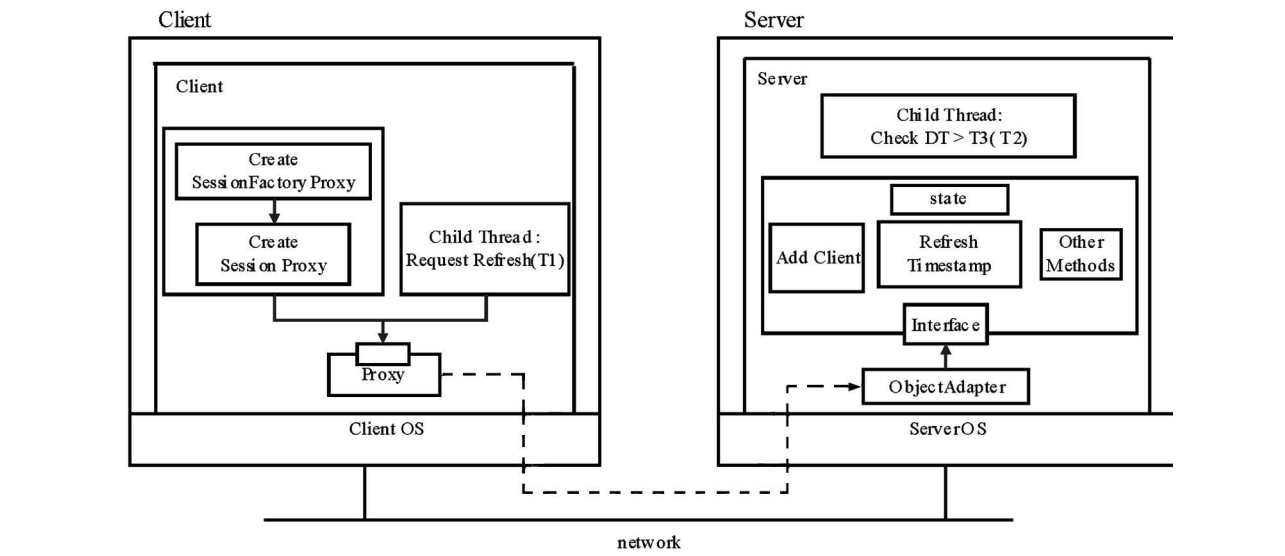


图 1 ICE 的 Session设计框架

首先客户端初始化 ICE运行时环境,读取客户端配置文件创建 SessionFactory Proxy,然后用之创建一个 Session Proxy。创建成功后,客户端继续创建子线程来管理此 Session Proxy。子线程每隔时间 T1请求刷新一次服务器上的时间戳。为了减少服务器的压力,在刷新之前让子线程挂起时间(T1)。T1可以根据具体的实时性要求来设定,在本模型中采用 T1=5 s。这个请求会返回状态值或者被 ICE运行时捕获到超时异常,除返回 TRUE值以外的情况都可判断该客户端与远程服务器断开连接。

2.3 服务器设计

服务器端的框架图如上图 1所示。

服务器端初始化 ICE运行时环境,读取服务器配置分别创建对象适配器(ObjectAdapter)和 SessionFactory Servant并把 SessionFactory Servant加入到对象适配器中。SessionFactory Servant提供了一个 Add Client的服务,用于监听客户端的连接。当有一个客户端连接时,SessionFactory Servant响应 Add client请求,创建一个 Session Servant和 Session Proxy,并把 Session Proxy返回给客户端。Session Servant提供了一个

时间戳刷新服务 Refresh 同时服务器要保存客户端 ID 和该 Session Servant 到一个 Session 列表。服务端创建子线程每隔时间 T_2 检查这个 Session 列表, 如果当前时间和某个客户端的最近时间戳之差 (DT) 大于时间范围 (T_3), 则判断该客户端和服务端断开连接, 并将之从 Session 列表中删除。由于 Session 列表被两个线程访问, 所以要用互斥锁保护, 这里采用了 ICE 的监视器来保护临界区资源, 监视器的锁不用自己手动释放。ICE 支持在临界区内挂起线程, 所以在子线程在刷新 Session 列表的之前, 可以让自己挂起时间 T_2 , 这样在 T_2 时间内就允许客户端请求 Add Client 服务。至于 T_2 T_3 的取值也要根据实时性要求来设定, 本模型中采用 $T_2=1\text{ s}$ $T_3=10\text{ s}$

3 远程监控系统的设计

3.1 系统功能

以某公司的远程设备监控系统为例, 该系统主要负责监控远程设备的运行状态, 采用 ICE 中间件作为网络通信方式。系统应用在局域网中, 采用星型网络拓补结构。系统组成主要有服务器、交换机、客户端和 GRPS Modem。服务器通过交换机和客户端相连, 此外服务器还连接了报警装置和 GPRS Modem。SM 卡置于 GPRS Modem 内, GPRS Modem 再通过 USB 接口与服务器相连。客户端通过数据采集装置与设备相连。如图 2 所示。

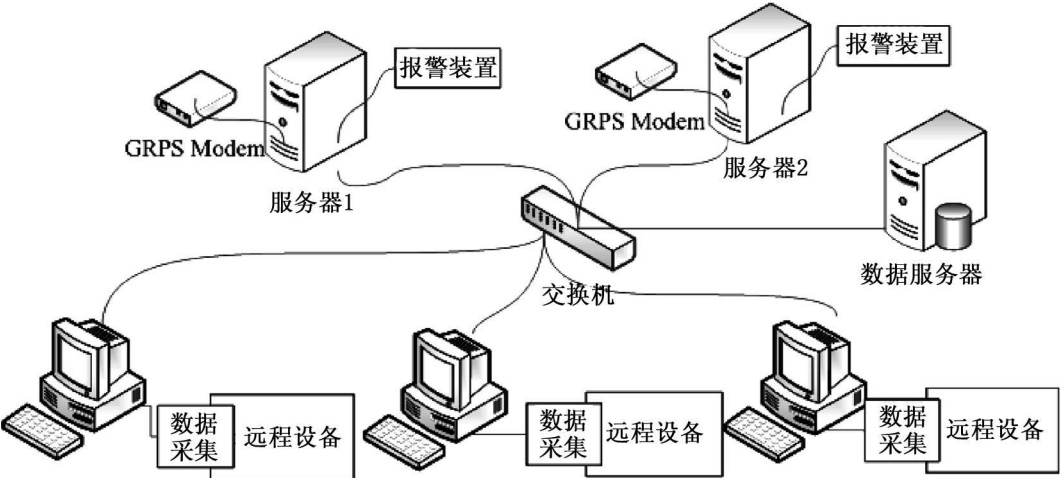


图 2 远程设备监控系统连接示意图

该系统主要功能包括：

- (1) 网络唤醒和命令控制

客户端使用支持 Wake-On-LAN 远程唤醒功能的主板和网卡, 网卡收到特殊信号激活系统开机后, 自启动安装在客户端的监控软件。客户端自动连接服务器后, 服务器就可发送控制命令给客户端以间接控制设备。

- (2) 监视设备的运行状况

实现对远程设备性能、状况的实时监控。远程设备实时上报数据到服务端显示。

- (3) 故障处理

故障处理主要包括断开连接和非断开连接故障, 如果是非断开连接的故障可以根据错误原因和级别错误做出不同处理, 最严重的会产生报警, 并通过 SMS 通知负责人。因远程设备故障或网络故障造成断开服务器端连接的, 则直接以上述最严重的故障处理。

3.2 Session 模块在系统的应用

Session 机制在系统中担当了重要的角色。因为该系统的各种功能都建立在服务器和客户端连接的

情况下。通过 Session可以实时监控与客户端的连接情况,保证监控的实时性。由于系统是在局域网的环境中使用,此处不考虑网络质量问题。

3.3 Session机制测试分析

测试环境采用一台服务器和三台客户端,并在客户端和服务端上通过日志文件记录请求和响应时间,以测试自定义 Session机制的实时性。主要用到客户端发送 Refresh请求的时刻 T_0 服务器响应 Refresh timestamp请求后的时刻 T_s 通过相邻两次 T_s 或 T_c 的差值可以反应出服务器与客户端连接实时性。试验中采用 $T1=5\text{ s}$ $T2=1\text{ s}$ $T3=10\text{ s}$ 以下是几种常见情况下的时间日志分析。

(1) 正常刷新的同步

表 1 正常刷新时间				hh:mm:ss.ms
序号	T_{c1}	T_{c2}	T_{c3}	操作
	T_s	T_s	T_s	
1	16:30:31.781	16:30:39.390	16:30:36.734	正常
	16:30:31.796	16:30:39.406	16:30:36.750	
2	16:30:36.796	16:30:44.406	16:30:41.750	正常
	16:30:36.796	16:30:44.406	16:30:41.750	
3	16:30:41.796	16:30:49.406	16:30:46.750	正常
	16:30:41.796	16:30:49.406	16:30:46.750	
4	16:30:46.812	16:30:54.421	16:30:51.765	正常
	16:30:46.812	16:30:54.421	16:30:51.765	
5	16:30:51.812	16:30:59.421	16:30:56.781	正常
	16:30:51.812	16:30:59.421	16:30:56.765	

结果分析: Session的刷新时间精确到毫秒,可见客户端的请求与服务端端的响应同步,这一点与 ICE中间件的高效通信和局域网环境都有关。

(2) 阻塞和非阻塞事件对 Session的影响

表 2 阻塞时刷新时间				hh:mm:ss.ms
序号	T_{c1}	T_{c2}	T_{c3}	操作
	T_s	T_s	T_s	
1	17:40:29.031	17:40:32.828	17:40:28.468	正常
	17:40:29.046	17:40:32.828	17:40:28.484	
2	17:40:34.062	17:40:37.828	17:40:33.500	向客户端发 送阻塞操作
	17:40:34.062	17:40:37.828	17:40:33.500	
3	17:40:39.062	17:40:42.843	17:40:38.500	正常
	17:40:39.062	17:40:42.843	17:40:38.500	

结果分析:服务器发出对客户端同步和异步操作对 Session机制的实时性都几乎没有影响。由于客户端和服务端都采用了多线程,所以异步的操作只会增加客户端和服务端本地资源的消耗,不会对网络造成阻塞;而同步操作可以通过增加客户端的和服务端 ICE运行时的线程池的线程数目来解决,所以对 Session机制的实时性影响甚少。

(3) 网络中断的情况

表 3 网络中断刷新时间				hh:mm:ss.ms
序号	Tc1	Tc2	Tc3	操作
	Ts	Ts	Ts	
1	17:13:43.812	17:13:42.750	17:13:43.312	正常
	17:13:43.828	17:13:42.750	17:13:43.312	
2	17:13:48.828	17:13:47.765	17:13:48.312	正常
	17:13:48.828	17:13:47.765	17:13:48.312	
3	17:13:53.828	17:13:52.765	17:13:53.328	正常
	17:13:53.828	17:13:52.765	17:13:53.328	
4	17:13:57.703	17:13:57.703	17:13:57.703	断开网络
	17:14:04.343	17:14:03.296	17:14:04.328	

结果分析:因为各种情况造成的网络中断,服务器能在规定的时间内检测到异常。这个响应时间($DTs=Ts4-Ts3$)与客户端的刷新时间间隔($T1$),服务器检查客户端列表的时间间隔($T2$),服务器判断Session失效的范围($T3$)有关。其中 $DTs<T3+T2$ 并且 $T1<T3$ 。所以这个响应时间可以根据不同系统的实时性要求来设置。

4 结语

本文提出了一种基于 ICE中间件的自定义 Session机制,以替代传统的 Socket心跳机制,甚至替代整个 Socket通信平台,并将其应用在远程监控系统中。测试表明,这种机制能够满足用户自定义的实时性要求并且易于实现。由于 ICE是完全可移植的,所以也容易移植到各种平台。

参考文献:

[1] 胡志坤,何多昌,桂卫华,等.基于改进心跳包机制的整流远程监控系统[J].计算机应用,2008,28(2):364—366.
[2] Henning M, et al. Distributed programming with ice ZeroC[EB/OL]. www.zeroc.com/IceManual.pdf, 2008—10—10.
[3] M ichiHenning. A new approach to object-oriented middleware[J]. IEEE Computer Society, 2004, (2): 66—75.

Remote Monitoring System Based on a Custom ized Session
M echanism Using ICE M iddleware

CHENG Jin, LI Lei, YANG Cheng

(School of Computer Science, Sichuan University, Chengdu 640014, China)

Abstract: Monitoring the connection status between server and clients is a common requirement in Real-time Remote Monitoring System. The paper presents a new solution based on object-oriented ICE Middleware aiming at designing a custom ized session mechanism. This solution takes advantage of ICE Middleware in the implementation of Real-time Remote Monitoring System. Experiments show that it is a new way for system to guarantee its Real-time and reliability.

Key words: ICE middleware; custom ized session; remote monitoring system