

实时CORBA rtORB的性能评估*

彭 舰** 刘锦德 谭 浩 张向刚

(电子科技大学计算机科学与工程学院 成都 610054)

【摘要】介绍了一个实时CORBA的原型rtORB,给出了一个对其进行实时性能测试的平台。以延迟和延迟抖动为指标,通过与其他实时CORBA的性能比较,表明rtORB可以支持实时应用。为评估操作系统和通信对实时CORBA的影响,在不同操作系统上进行了测试。针对rtORB在不同操作系统上的测试结果,讨论了操作系统和通信对基于实时CORBA实时应用的影响。

关键词 实时CORBA; 测试平台; 操作系统; 实时通信; 延迟; 抖动

中图分类号 TP311.133 文献标识码 A

Real-Time Performance Evaluation for rtORB

Peng Jian Liu Jinde Tan Hao Zhang Xianggang

(School of Computer Science and Engineering, UEST of China Chengdu 610054)

Abstract Real-time CORBA specification extends the scope of CORBA application to real-time field. A real-time CORBA prototype called rtORB is introduced. Then a test platform is proposed to evaluate real-time performance of rtORB. The delay and delay jitter are selected as test parameters to compare rtORB with other real-time CORBA product. The test results show that rtORB can support real-time application. A series of experiments on various operating systems are conducted to evaluate the influence that operating system and communication system impact on real-time CORBA. The test results are discussed in details and some suggestions directing how to build real-time application based on real-time CORBA are given.

Key words real-time CORBA; test platform; operating system; real-time communication; delay; jitter

1 rtORB简介

实时CORBA规范的推出,将CORBA的应用范围扩展到实时领域。实时CORBA对原有的规范进行了扩展,定义了标准的界面和QoS策略,以允许应用配置和控制处理器资源、内存资源、通信资源等^[1],其应用范围限制在固定优先级的CORBA应用系统中。

实时CORBA的原型rtORB参照实时CORBA规范实现,其结构如图1所示。考虑到实时应用很少使用动态调用的特点(在实时系统中,一般使用静态调用,而动态调用代价太大),因此,rtORB作了一些简化,没有实现动态调用接口(DII)和动态框架接口(DSI)。

rtORB不仅参照实时CORBA规范,而且在设计和实现过程中加入了自己的特色,即通过平台依赖层去

2003年1月2日收稿

* 国防科研基金资助项目

** 男 32岁 在职博士生 讲师 主要从事开发系统和中间件技术方面的研究

掉了ORB与具体操作系统的强耦合关系,使其具有很好的可移植性;加入动态调度服务,使rtORB保留了CORBA开放性的特点^[2];提供显式绑定模型,解决了实时对象间交互的可预测性^[3];在可移植对象适配器中使用不分层的策略,减少优先级反转和非确定性的可能性^[4]。在rtORB的实现过程中,本文采用了多种优化手段,使rtORB具有良好的性能。

为了评估rtORB的实时性和操作系统对rtORB实时性的影响,本文设计并作了一系列的实验,着重描述rtORB实时性的测试方法,详细讨论了操作系统对其的影响。

2 测试平台

为评估rtORB的实时性能,结合文献[5, 6]中的测试方法,构筑了如图2所示的实时性能测试系统。

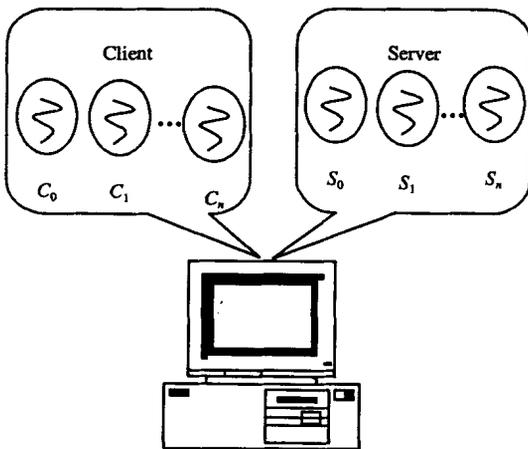


图2 实时性能测试系统

由于当前的rtORB只支持TCP协议,而TCP协议的一个重要特点是不确定性,无法提供可预测的实时行为,会影响rtORB实时性的测试结果,但为可比较的测试提供了一个基础平台。由于测试对象主要是rtORB本身的实时性,为尽量最小化TCP协议带来的优先级反转,可将客户和服务器放置于同一台机器上,各主机的配置如表1所示。

测试系统的配置:服务器进程包含一个具有最高优先级 P_h 的servant S_0 , N 个具有相同低优先级 P_l 的servant S_1, S_2, \dots, S_n 。每个servant对应一个线程,而客户进程包含一组线程 $C_0, C_1, C_2, \dots, C_n$,其中 C_0 的优先级为 P_h ,而其他客户线程 $C_1 \sim C_n$ 的优先级为 P_l , C_i 的请求由 S_i 处理。 C_0 的操作请求频率为20 Hz,即每秒20个双向操作请求。其他客户 $C_1 \sim C_n$ 的请求频率为10 Hz,即每秒10个双向操作请求。所用的客户进程向对应的servant线程发送一个float, servant接收到float后,进行1 000次循环的立方运算,然后返回结果。

当客户进程创建线程时处于阻塞状态,直到所用线程都创建好后才同时开始执行。每个低优先级的客户线程按10 Hz的频率执行4 000次,而高优先级的客户线程直到所用的客户线程结束才结束。

测试的指标为双向操作的延迟和抖动。由于客户和服务器在同一台机器上,通信的代价较小,延迟直接反映了应用的性能。而延迟抖动(高优先级的客户延迟抖动为其所有调用中最长延迟和最短延迟之差,低优先级的客户延迟抖动为所用低优先级客户的调用中最长延迟和最短延迟之差)则反映最坏执行时间状况,这一指标对于调度许多实时应用非常必要。

在理想的实时ORB端系统中,低优先级客户的延迟会随低优先级客户数的增加而上升^[5],这是因为客户数的增加必然会增大系统的负载,从而对资源的竞争更加激烈,但高优先级的客户延迟应保持固定值或增长非常缓慢,延迟抖动越大,对理想的最坏执行时间的预测越困难,即很难进行实时调度。

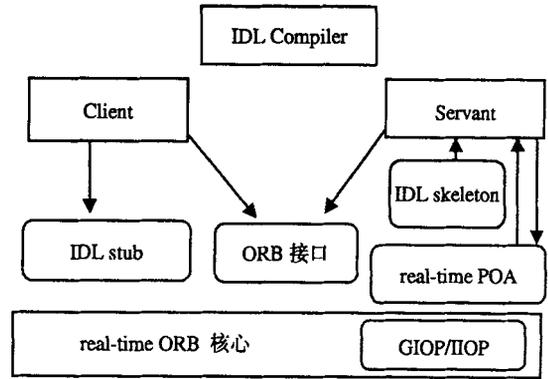


图1 rtORB 的结构

表1 各主机的配置

主机	OS	CPU	主频/ MHz	RAM/ M
Sun	Solaris2.5.1	Ultra-SPARC	167	64
PC-1	NT4.0	PIII	500	64
PC-3	Red Hat 6.2	PIII	500	64
PC-4	RTLinux	PIII	500	64

3 rtORB与TAO比较

因TAO具有很好的实时性能^[5,6],故将实时CORBA TAO作为比较的对象。测试在Solaris上进行,Solaris提供了对实时线程的调度,为分派应用或内核线程执行提供了最坏保证^[7]。另外,Solaris为互斥和按优先级顺序排队/分派线程实现了优先级继承协议,其测试结果如图3和图4所示。

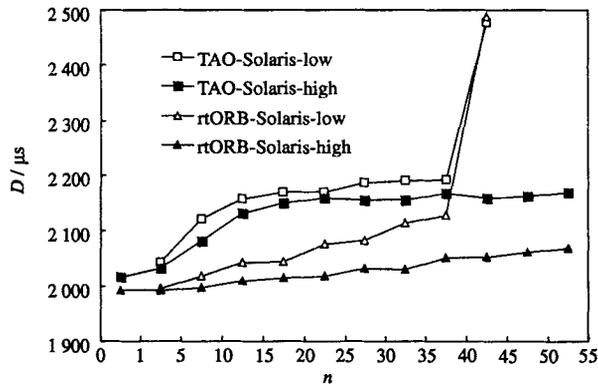


图3 rtORB和TAO延迟的比较

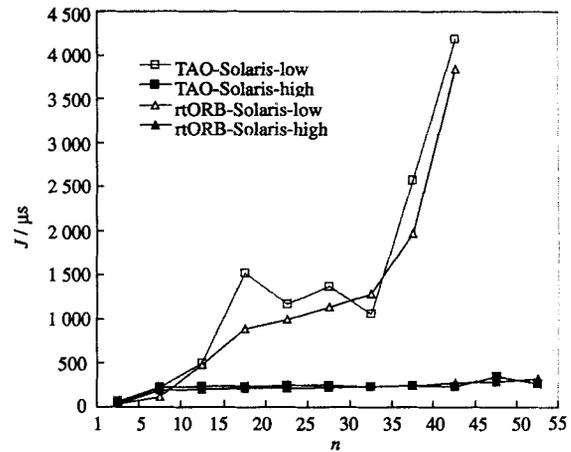


图4 rtORB和TAO抖动的比较

图3给出了随着低优先级客户数 n 的增加,双向操作平均延迟 D 的变化。从高优先级客户的延迟变化看,两者的延迟变化都较小(TAO的延迟变化= $2\ 171$ (低优先级客户数为50时的延迟)- $2\ 017$ (低优先级客户数为0时的延迟)= $54\ \mu\text{s}$,rtORB的延迟变化= $2\ 069$ (低优先级客户数为50时的延迟)- $1\ 993$ (低优先级客户数为0时的延迟)= $76\ \mu\text{s}$),说明rtORB和TAO的高优先级的客户都具有很好的实时性,而TAO的延迟变化更小一点,特别是当低优先级客户数大于50后,TAO的高优先级客户的延迟比较稳定,而rtORB有逐渐增加的趋势。从低优先级客户的延迟变化看,两者的延迟都在逐渐增加。特别是当低优先级客户数大于40后,延迟增加非常明显,到低优先级客户数为45时,延迟已经大于 $10\ 000\ \mu\text{s}$ (数值太大,图中未示出),这是由于系统中的线程数太多,机器内存太小,一部分线程需要被置换出内存空间,放于虚拟空间上,而被置换出内存空间的线程主要是低优先级的线程。因此,低优先级客户的延迟会迅速增大。

延迟抖动 J 的变化如图4所示。从图中可以看出,随着低优先级客户数 n 的增加,rtORB和TAO高优先级客户的抖动的变化较小,说明rtORB和TAO都具有很好的可预测性,能够对其最坏的执行时间作出较为准确的预测。而低优先级客户的延迟抖动相对较大,特别是在优先级客户数大于45时,抖动大于 $100\ 000\ \mu\text{s}$ (数值太大,图中未示出),这是部分线程需要被置换出内存空间所引起。

综上所述,rtORB具有很好的实时性,可以用于实时应用的开发。

4 rtORB在不同操作系统上比较

实时操作系统提供了受优先级控制的访问硬件和软件资源的机制,实时CORBA必须建立在实时操作系统上才能支持实时应用。测试主要是通过rtORB在不同操作系统上的情况,说明实时CORBA必须有实时操作系统的支持,才能体现其实时性。

测试所选的操作系统包括Solaris、Linux(RED HAT)、RTLinux和NT,其中Solaris和RTLinux提供了实时、可抢占、先进先出(FIFO)的线程调度,而NT的实时优先级类可以抢占非实时应用,但其使用轮循方式调度,而不是FIFO^[8]。

Linux是一个类UNIX操作系统,采用以分时调度为主,兼顾一些软实时任务的调度。Linux的分时调度机制与实时任务的低延迟、可预测性要求相悖。实时Linux的设计目标是以Linux内核为基础,在同一个操作系统中既提供实时服务,又支持普通Linux进程的执行,其中,RTLinux是实时Linux的典型代表。RTLinux系统主要是在RED HAT 6.2的系统上,将内核替换为Linux kernel 2.2.14+RTLinux2.2。

RtORB在不同操作系统上的延迟比较如图5所示,图6和图7给出了抖动的变化情况。从图中可以看出,无论是低优先级的客户延迟还是高优先级的客户延迟,NT变化都非常大,而延迟抖动的变化更是不可预测,所以NT的实时性最差,不适合实时应用。在Linux上,当客户数较少时,高优先级的客户延迟变化较小,而此时的低优先级客户的延迟上升就比较快。当低优先级客户数大于35时,情况就不稳定,延迟变化较大,此时的抖动也迅速增大,低优先级客户的抖动大于70 000 μs ,高优先级客户的抖动大于10 000 μs ,这是由于一部分线程被置换出内存,其中包含了高优先级的线程,说明Linux的实时线程在实时响应时间上并不能保证确定性。与Linux相比,RTLinux则有很好的实时性,高优先级的客户延迟在整个过程中保持了很好的稳定性(延迟变化=346(低优先级客户数为50时的延迟)-299(低优先级客户数为0时的延迟)=47 μs)。在系统发生页空间置换时,系统所考虑的是低优先级的客户,因此当低优先级客户数大于35时,低优先级客户的延迟迅速增大,而高优先级客户的延迟仍然保持相对稳定。Solaris也显示了很好的实时性,其高优先级客户的延迟也较小(延迟变化=2 069(低优先级客户数为50时的延迟)-1 993(低优先级客户数为0时的延迟)=76 μs)。从图7的抖动看,RTLinux的高优先级客户的抖动保持相对稳定,且抖动的幅度也较小(抖动变化=80(低优先级客户数为50时的抖动)-44(低优先级客户数为1的抖动)=36 μs)。经比较,Solaris上的高优先级客户的抖动相对大一些(抖动变化=346(低优先级客户数为50时的抖动)-48(低优先级客户数为1的抖动)=298 μs),说明RTLinux的实时性最好。

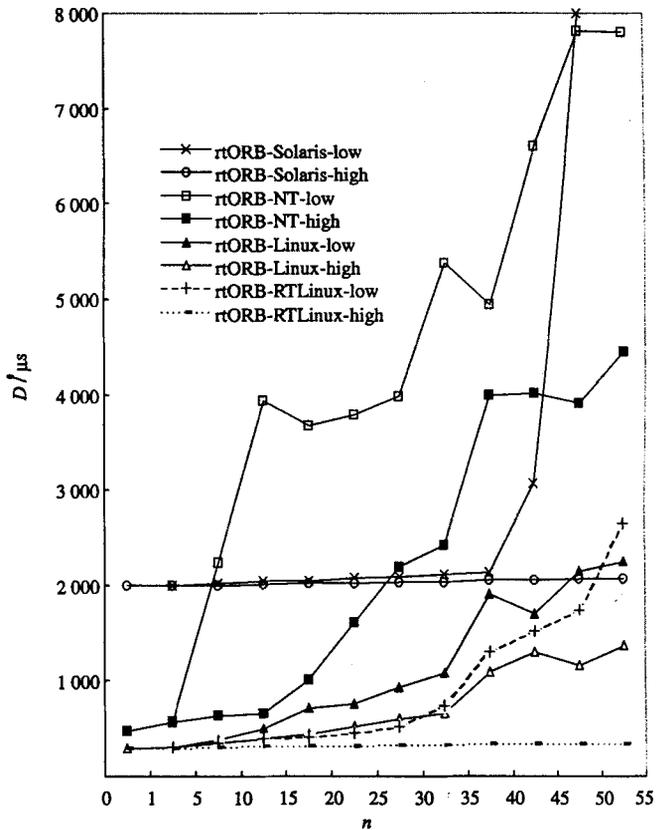


图5 rtORB在不同操作系统上的延迟比较(本地)

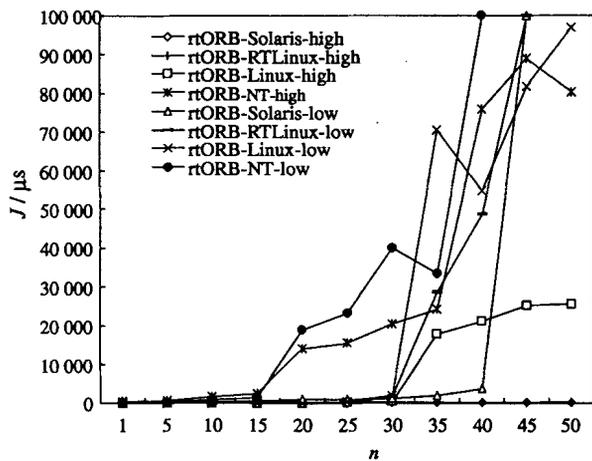


图6 rtORB在不同操作系统上的抖动比较(全貌,本地)

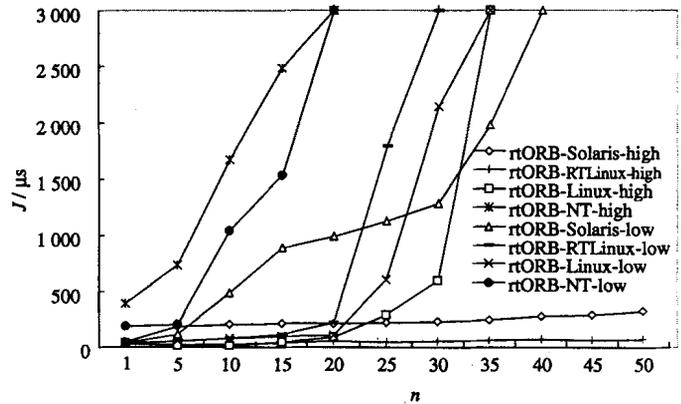


图7 rtORB在不同操作系统上的抖动比较(部分,本地)

客户进程和服务端进程的测试都在同一台机器上进行,主要是把TCP带来的负面影响最小化。为了从反面说明这种测试方法的正确,把rtORB置于NT和RTLinux上,这时客户进程和服务端进程分别置于远程相隔的机器上,测试的方法同前,测试结果如图8和图9所示。从图8可以看出,此时高优先级的客户延迟的变化较大,即使是在RTLinux上。将rtORB置于NT或RTLinux上,没有很大的区别,说明尽管RTLinux具有很好的实时性,但由于网络的非实时性,使rtORB在其上的运行效果与在NT上的运行效果没有本质的区别。从图9

看,在RTLinux上低优先级客户的延迟抖动和高优先级客户的延迟抖动很接近,且变化无法预测。由于客户线程和服务线程分别在各自的机器上,没有发生内存空间置换的情况,所以没有出现图5和图6中延迟和抖动突然变得非常大的情况。

经测试表明,在分布式实时系统中通信的实时性和端系统的实时性同等重要,任何一方面不支持实时都会导致整个系统不支持实时。

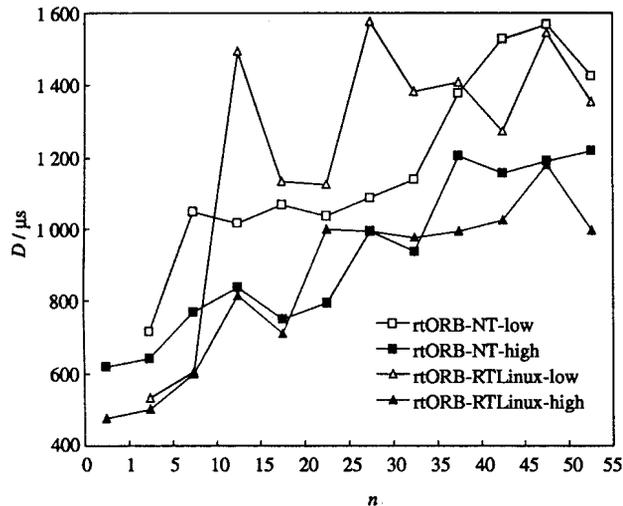


图8 延迟比较(远程)

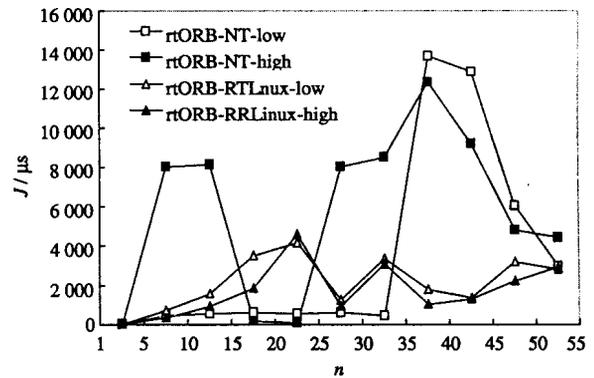


图9 抖动比较(远程)

5 结束语

本文对实时CORBA原型rtORB作了实时性能的评估,并构筑了相应的测试平台。从rtORB与CORBA TAO的实时性能对比中可以看出,rtORB具有良好的实时性,能够支持实时应用。从rtORB在不同操作系统上的测试结果可以看出,基于实时CORBA的实时应用必需运行于实时操作系统和实时通信上才可以保证其实时性。

参考文献

- [1] 骆志刚,刘锦德. 实时CORBA[J]. 计算机科学, 1999, 26(10): 84-87
- [2] Luo Zhigang, Tan Hao, Liu Jinde. A dynamic scheduling service model for real-time CORBA[J]. 电子科技大学学报. 2001, 30(6): 618-624
- [3] 骆志刚,谭浩,刘锦德. 实时CORBA绑定模型的设计和实现[J]. 计算机应用, 2002, 22(2): 30-33
- [4] Irfan P, Douglas C S. An overview of the CORBA portable object adapter[J]. Special Issue on CORBA in the ACM StandardView Magazine, 1999, 6(1): 30-43
- [5] David L, Douglas C S, Sergio F G. An empirical evaluation of OS support for real-time CORBA object request brokers[C]. Proceedings of the Multimedia Computing and Networking 2000 (MMCN00) Conference, ACM, San Jose, CA, 2000: 38-47
- [6] Douglas C S, Sumedh M. Software architectures for reducing priority inversion and non-determinism in real-time object request brokers[J]. Journal of Real-time Systems, 2001, 21(1-2): 77-125
- [7] Khanna S. Realtime scheduling in sunOS 5.0[C]. Proceedings of the USENIX Winter Conference, USENIX Association, 1992. 375-390
- [8] Ramamritham K, Shen C, Gonzales O, et al. Using windows NT for real-time applications: Experimental observations and recommendations[C]. Proceedings of the 4th IEEE Real-Time Technology and Applications Symposium, 1998. 102-111

编辑 徐培红