基于CORBA技术的高可用舰船信息处理系统*

CORBA-based High Available Marine Information Management System

韩剑辉[1]，许镇琳[1]，邓万禧[2]，王刚[2]，刘晓光[2]

HAN Jian-hui[1], XU Zhen-lin[1], DENG Wan-xi[2], Wang Gang[2], Liu Xiao-guang[2]

（1．天津大学电气与自动化工程学院，天津 300072；2．南开-百度联合实验室，南开大学信息学院，天津 300071）

（1. School of Electrical Engineering & Automation, Tianjin University, Tianjin 300072, China;

2. Nankai-Baidu Joint Lab, College of Information Technical Science, NanKai University, Tianjin 300071,China）

摘 要：舰船信息处理面临着大数据量、复杂应用环境和平台异构化等趋势，本文设计了基于 CORBA 的高可用舰船信息处理系统。利用 CORBA 所具有的平台及语言无关的特点，屏蔽平台间差异，实现了异构平台间通信。并运用 Oracle 数据库的 Advanced Replication 功能实现了系统的双机热备和负载均衡，提高了系统的可用性。

Abstract: More and more marine information management systems face a huge mass of data, complex environment and heterogeneous platforms. In this paper, a CORBA-based high available marine information management system is developed. This system hides the difference among platforms and achieves communication between heterogeneous nodes using CORBA's feature of platform and language independent. Advanced Replication in Oracle database is also used to provide hot standby and load balance. This method improves availability effectively.

关键词：CORBA；分布式对象；舰船信息管理；双机热备；负载均衡

Keywords: CORBA; distributed object; marine information management; hot standby; load balance

中图分类号：TP393　　**文献标识码**：A

1. 引言

随着分布式对象技术的发展，为异构数据源集成提供了新的途径。CORBA（Common Object Request Broker Architecture）是由 OMG 提出的应用软件体系结构和对象技术规范之一，其核心是一套标准的规范、接口和协议，以支持异构分布应用程序间的互操作性及独立于平台和编程语言的对象重用[1]。在 CORBA 系统中，所有的应用程序都封装成为对象，其接口定义了对象可提供的操作，客户方只须知道目标对象接口，就可获得目标对象提供的服务。对于舰船分布式导航信息系统而言，当前面临着海量数据处理、复杂应用环境和平台异构性等趋势。通过 CORBA，我们可以尽可能地减少当系统功能要求变更时，需对不同平台系统实现的修改，屏蔽平台间的差异，实现智能舰船各种系统在异构平台间的通信。

* **基金项目：**“十一五”国家科技支撑计划项目（2006BAG03B04）；“863”计划课题(2008AA01Z401)；教育部博士点基金(20070055054)；天津市科技发展计划（08JCYBJC13000）

作者简介：韩剑辉，男，天津大学电气与自动化工程学院博士生，研究方向管理信息系统；许镇琳，男，天津大学电气与自动化工程学院教授，研究方向管理信息系统；邓万禧，男，南开大学信息学院硕士生，研究方向分布式系统；王刚，男，南开大学信息学院副教授，CCF 会员（E20-0006617M），研究方向海量存储，刘晓光，男，南开大学信息学院副教授，研究方向并行计算。

通迅地址：300071 天津市卫津路 94 号南开大学信息学院王刚；Tel：022-23504780；E-mail：wgzwp@163.com

2. 相关研究

分布式系统的研究已经有较长时间。在早期，研究人员只将注意力集中在互连互通和互操作这类功能性研究上。提出了一些底层透明的通信机制，如 RPC、ORB 等。近几年，随着计算机技术、通信技术的迅猛发展，大量要求高性能的应用不断涌现，比如网络多媒体应用、航空航天应用等。研究人员意识到，必须关注分布式系统的性能问题，如实时性能、可信度性能和动态适应性等。目前分布式系统一个主要的研究方向是：在保证分布式系统互操作的基础上，如何满足分布式系统的种种高性能的要求。

分布式系统的应用之一是分布式数据库系统。它是由若干个节点集合而成。节点在通讯网络中联接在一起。每个节点都是一个独立的数据库系统，它们都拥有各自的数据库、中央处理机、终端，以及各自的局部数据库管理系统。因此分布式数据库系统可以看作是一系列独立数据库系统的联合。它们在逻辑上属于同一系统，但在物理结构上是分散的。分布式数据库技术是分布式计算的一个重要组成部分，允许数据在多个服务器端共享。采用分布式数据库技术，一个本地服务器可以存取不同物理地点的远程服务器上的数据；也可以使所有的服务器均可以持有数据的副本，这样分布式系统中的所有服务器均可进行本地存取。设计一个分布式计算解决方案首先需要考虑的问题就是应用的完整性、复杂性、性能和可用性以及响应时间等，同时还需要考虑对于不同的应用需求是采用实时存取远程数据还是采用延迟存取远程数据。对数据复制来讲就是采取实时更新复制方案还是延迟数据复制方案。

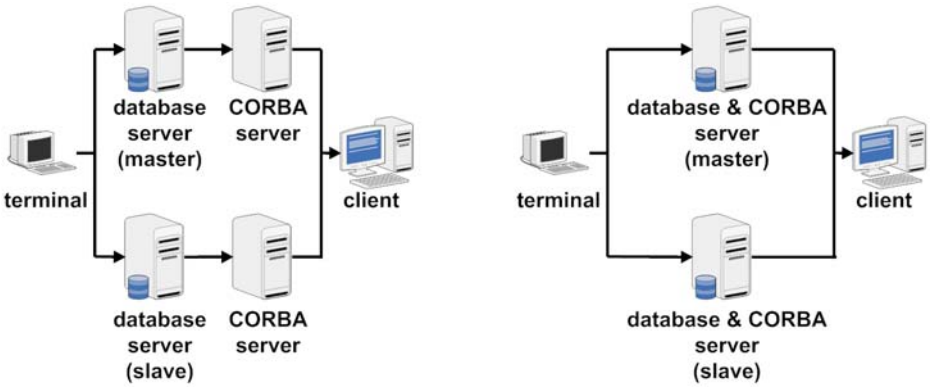

图 1　舰船信息管理系统体系结构

3. 系统体系结构

在舰船的高可用信息处理系统中，我们使用基于 CORBA 的分布式数据库，这种结构如上所述可以提高系统数据完整性、可用性及响应速度。在此结构中，CORBA 服务作为数据库的前端，根据数据库结构构造相应的 CORBA 对象。对象使用 IDL（Interface Definition Language）描述并生成相应的数据结构。客户端通过 IDL Stub 与 CORBA 对象通信，由 CORBA 对象实现与服务器端 CORBA 对象的交互。服务器端的 CORBA 对象通过服务器端程序访问数据库，构造结果 CORBA 对象返回给客户端。

我们设置了一个模拟终端，模拟产生数据源，来测试系统性能。它模拟真实系统里面的诸如雷达、测控设备等各种终端，生成数据传送到数据库服务器上。在主数据库和备份数据库之间采用 oracle 的高级复制功能，采取对等方式的设置，设置主从数据库可以同步方式或者异步方式更新数据，以保证数据一致性。主、从数据库的设置可以极大地提高数据可靠性和系统的可用性，在主数据库服务器宕机的情况下，备份服务器负责接收模拟终端产生的数据以及从 CORBA 服务器所发出的客户端请求事件。

对于 CORBA 服务器，我们设计了两种方案。第一种方案如图 1 左图所示，系统中有两台物理服务器，其上运行 CORBA 服务器软件，完成对应的两台 Oracle 数据库服务器和客户端间的交互。如果两条链路中不同位置发生故障，系统正常服务将终止。一种解决方案是在 CORBA 服务器和数据库服务器间建立交叉连接。另一种方案如图 1 右图所示：将 CORBA 服务器和数据库服务器部署到了同一台物理服

务器上，数据库服务器同时承担 CORBA 服务器的功能，避免了方案一存在的问题，但是产生的代价也显而易见，就是数据库服务器的压力也相对应的增大。

第二种设计方案下客户端和服务器的交互过程如下：客户端 CORBA 程序发出请求（封装为 CORBA 对象）；CORBA 服务器接收到 CORBA 对象，解析出请求；CORBA 分析请求，与 Oracle 数据库服务器交互，执行数据库操作；CORBA 服务器接收到数据库服务器返回的结果，封装为 CORBA 对象，返回给客户端；客户端解包 CORBA 对象，获得执行结果。服务器端向客户端发出请求的过程与上述过程类似。由于主数据库服务器和从数据库服务器之间建立了高级复制连接，对主服务器的更新请求都会同步到从服务器，从而保证主、从数据库的数据一致性。

4. 双机备份与故障处理

4.1 Oracle分布式数据库技术

Oracle 复制技术主要分为两种：同步复制，和异步复制[5]。采用同步复制，复制数据在任何时间在任何复制节点均保持一致。如果复制环境中的任何一个节点的复制数据发生了更新操作，这种变化会立刻反映到其他所有的复制节点。采用异步复制策略，所有复制节点的数据可能在一定时间内是不同步的。如果复制环境中的其中的一个节点的复制数据发生了更新操作，这种改变将在不同的事务中被传播和应用到其他所有复制节点。这些不同的事务间可以间隔，由系统的配置决定。在舰船信息管理系统中，根据不同的负载情况，我们可以适时地在异步复制和同步复制策略间切换。

另外 Oracle 高级复制支持“失效接管”，即，通过将主系统数据复制到从系统，可以起到当主系统崩溃时，利用从系统的数据副本接管主系统任务的作用。如果主系统出现故障，业务依旧可以通过访问复制数据库正常运行。同时，Oracle 也针对失效接管提供了另外两个可选的解决方案：Oracle 并行服务器和备用数据库配置。用户可以根据吞吐量、可用性、事务损失的可能性和其他一些如数据一致性、方法的局限性等参考指标来权衡选择合适的接管方案。

4.2 Oracle高级复制框架

多主体复制方案（Multi-Master）支持全表在各个主节点间的对称复制，允许所有主节点对主表都有更新操作的权利。任何一个主节点上的复制表的更新都会被传播并被直接应用到其他所有主表。一个主节点出现问题，不会对其他主节点之间变化的传播造成影响。多主复制采用一种称为“延迟远程过程调用”（deferred remote procedure calls RPCs）的机制作为主要的传播和应用变化的机制。在传播变化时，如果其中的一个远端系统没有准备好，传播变化的延迟远程过程调用（RPCs）就会保存在其本地队列中，等到系统准备好以后再执行。由于舰船系统的业务要求，我们采用多主体站点分布方式，其内在的数据库应用模型可以理解为备份数据库形式，主、从节点数据库都可以作为 master 数据库，一个数据库的数据更新可以马上通过事务更新到另一个 master 数据库。

4.3 故障处理与任务接管

对于第 3 节所述两种系统架构方案，可以分别设计不同的数据库复制方案来实现故障处理与任务接管。对于第一种实现方案，即数据库服务器和 CORBA 服务器分别在两台主机上，我们通过配置 Oracle 的失效接管功能来实现。如果主数据库服务器宕机，来自于 CORBA 服务器的请求都会通过任务接管机制转移到从服务器上。而对于第二种方案，因为数据库服务器和 CORBA 服务器都在一台主机上，如果一台主机宕机，则数据库和 CORBA 服务将会全部中断，所以需要配置客户端实现 CORBA 服务连接转移间接实现。我们在配置文件写入相应的服务器地址，如果一台主机服务器宕机，CORBA 失效，则转移连接另一台 CORBA 服务器，通过 CORBA 的服务转移来实现故障处理和任务接管。

4.4 系统可用性

对于上述双机备份系统结构，其可用性模型可用 Markov 链描述[6]。通过计算 Markov 稳态概率，可得如下系统可用性公式：

$$MTTDL = \frac{\mu}{1(1+1)\lambda^2} = \frac{MTBF^2}{2MTTR} \qquad \text{（公式 1）}$$

其中 MTBF 表示节点平均故障间隔时间，λ=1/MTBF。MTTR 表示节点平均故障恢复时间，μ=1/MTTR。MTTDL 表示系统平均数据丢失时间，或者说是系统平均崩溃时间。假定节点 MTBF 为 1 万小时，MTTR 为 24 小时，则系统平均崩溃时间可达 200 万小时以上。可见双机备份方案极大地提高了系统的数据可靠性和可用性。对于需要长时间稳定运行的舰船信息处理系统，这具有非常重要的意义。

5. 负载均衡

5.1 负载均衡算法

传统的双机备份系统，从机只是主机的热备份。只有在主机崩溃的情况下，才接管业务，而在正常状态下是不承担业务处理任务的。但如第 4 节所述，我们设计的双机备份系统中，主、从数据库服务器在正常状态下均可接收、处理用户请求，两者互为备份，这就大大提高了设备的利用率，提高了系统负载能力。为了获得最优的系统负载能力，需要有好的负载均衡算法，将用户请求均匀分配给主、从节点。分布式系统负载均衡算法主要有两种：全局轮转算法和随机算法[2, 4, 7]。全局轮转算法是指维护一个全局可用服务器队列，队列的每个成员都具有相同的地位，算法简单地在这组成员中顺序轮转选择，将用户请求依次分配给这些成员。轮转法的结果是可预知的，每个结点被选择的机会均等，因此很容易计算出结点的负载分布。轮转法比较适用于集群中所有结点的处理能力和性能均相同的情况，在实际应用中，一般将它与其他简单方法联合使用时比较有效。其缺点是需要一个全局索引指向当前任务分配目标服务器，此全局索引会成为系统瓶颈，其等效率函数（算法伸缩性的评价指标）为 $O(p^2\log p)$ [7]。随机算法依赖于相应的随机函数，在一个可用服务器队列里，队列的每个成员都具有相同的地位，通过随机数在这组成员中随机选择任务分配目标。由于随机算法在一定程度上具有盲目性，在某一时刻选中的恰好是重载节点，而其余轻载节点却处于空闲中，从而造成系统整体性能和吞吐率下降。但由于任务分配通过随机数完成，不需要更新全局索引，所以其伸缩性要由于全局轮转算法，等效率函数为 $O(p\log p)$ [7]。因而，对于大规模分布式系统，随机负载均衡算法往往会得到更好的实际性能。就我们的系统模型来说，系统规模不大，目前只有主、从两个服务器，少量用户节点，随机负载均衡算法伸缩性上的优势得不到体现。哪种算法性能更优，需要实验分析。为此，我们进行了仿真实验。

5.2 实验环境

我们采取两台主机模拟实际环境，其配置下图：

	CPU	内存	硬盘
主机一	AMD Sempron 2500+	1GB DDR	Seagate 160G
主机二	Intel Celeron CPU 3.06GHz	512M DDR	Seagate 80G

表 1 主机配置

主机一操作系统为 Windows XP sp2, 主机二上操作系统为 Ubuntu 7.10.1。在两台主机上分别安装 CORBA 服务，客户端程序数量分别为两个，这四个客户端程序按照上述的平衡策略对服务器进行访问。

5.3 客户端和服务器模拟程序介绍

客户端模拟程序按照泊松分布发出访问请求，服务器选择执行相应的负载均衡算法，由每个请求的加权值决定。服务器模拟程序会记录每个客户端请求到达的时间，然后根据请求的加权值生成请求处理时间，最后把这些信息保存到待处理的请求队列中。服务器模拟程序从待请求队列中取出请求，并根据请求所需处理时间睡眠一定时间，即模拟了请求的处理过程，当线程醒来后会把前一个请求的处理信息（请求到达时间和处理结束时的时间）记录在日志中。分析这个日志就能够得到请求的延迟情况，从而对两种方案进行评价。

5.4 实验和结果分析

轮询法模拟实验：两个主机分别开启两个客户端，按照轮询方法选择服务器。每发送一个请求后要睡眠，睡眠时间符合泊松分布，数学期望值取 5，请求大小取随机值。而随机法模拟实验产生的请求随机选取服务器，其他参数选择与轮询法相同。根据两台服务器上的请求延迟时间得出图 5，横坐标表示

请求到达的时间（相对值），纵坐标表示请求的延迟时间（相对值），其中红色的两条线是客户端随机选择服务器时，两个服务器处理请求的延迟曲线，蓝色的两条线是客户端轮巡选择服务器是，两个服务器处理请求延迟曲线。由图 2 可见，对于我们的系统，由于规模较小，轮询方法比随机方法获得更小的服务器负载波动。因为两个服务器的负载量也很近似，表明轮询方法能够提供更好的服务质量。因此，在我们的系统实现中，采用了轮询算法。

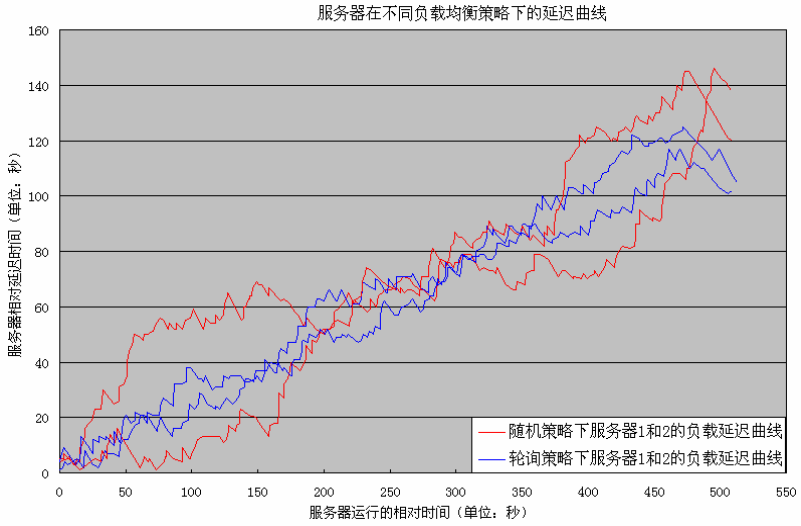

图 2　不同负载均衡策略性能对比

6. 总结

本文论述了一个高可用舰船信息处理系统的设计与实现。该系统的主要特点是：采用 CORBA 技术实现异构平台的协同工作；利用 Oracle 数据库的高级复制功能实现了双机备份，可达到极好的数据可靠性和系统可用性；采用全局轮转负载均衡算法，实现了主、从服务器的负载均衡，优化了系统的负载能力。下一步的工作可以从以下两方面着手：首先，Oracle 数据库的同步复制技术的可靠性需要实践的检验，而且在服务器规模扩大后，这种复制技术的效率不高，可以考虑使用异步复制技术实现数据库同步；其次，负载均衡算法只实现了轮转算法和随机算法，这两种算法在服务器规模扩大后效果也会降低，可以考虑实现自适应负载平衡算法以支持大规模服务器集群系统。

参考文献

[1] 齐向明，侯迪，齐勇，等.一种 CORBA 组件应用服务器的体系结构[J]. 西安交通大学学报，2000，34(4): 37-41.

[2] Arūnas Andriulaitis, Analysis of CORBA Load Balancing Strategies [J]. INFORMACINĖS TECHNOLOGIJOS IR VALDYMAS, 2004, 3(32): 75-79.

[3] 彭舰，基于 CORBA 的分布式系统中实时－容错性的研究－－分布式系统中动态调度的设计与实现[PhD Thesis][D]。电子科技大学，2004.

[4] 程仁贵，杨圣云，CORBA 环境下的动态负载平衡研究[C]. 中国计算机学会网络与数据通信学术会议，2002.

[5] 黄奕华，林晓敏，Oracle 数据库的备份及恢复技术的研究与应用[J]. 办公自动化，2007，8(2)。

[6] G. Gibson and D. Patterson, Designing Disk Arrays for High Data Reliability[J]. Journal of Parallel and Distributed Computing, 1993, 17(1-2):pp. 4-27.

[7] Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar, Introduction to Parallel Computing, Second Edition[M]. Addison Wesley, 2003.