

CORBA 分布式系统中网络分割协议可行度分析

陈雅琳^{1*}, 黄宏光¹, 李燕斌²

(1. 四川大学 电气信息学院, 成都 610065; 2. 中国电子科技集团公司 第十研究所, 成都 610036)

(*通信作者电子邮箱 chenyl_66@163.com)

摘要:对象管理组织(OMG)颁布的容错 CORBA 通过对象冗余的方式实现容错。但容错 CORBA(FT-CORBA)没有对网络分割的问题提供解决方法,即在网络分割的情况下,由于分割子网之间无法传递最新的复制对象状态,导致网络处于不断重复获取复制对象状态的阶段,操作不能继续执行,从而大大降低了系统的可执行性。通过在 CORBA 中间件中添加附件,在原有的容错机制上,根据对象一致性的要求分级,对不同等级的对象一致性对象采取不同的容错措施来提高系统的可执行性。并对网络分割为 3 个子网和 10 个子网的情况下进行了可行度分析验证。当网络分割情况不严重,即使所有子网都没有包含大部分网络节点时,改进的容错机制仍有较好的可执行度。当网络分割成的子网数比较多,网络的可行度与单个子网包含整个分布式系统节点数的多少有关,如果没有一个子网包含分布式系统中大部分节点时,网络的可行度仍然非常低,但仍优于改进前分布式系统的可行度。整个改进机制添加在中件附件中,无需改变原有的 ORB 代码。

关键词:公共对象请求代理体系结构;容错机制;对象冗余;网络分割;复制对象;分布式系统

中图分类号: TN915.41 **文献标志码:** A

Analysis on availability of partition-aware CORBA-based distributed system

CHEN Yalin^{1*}, HUANG Hongguang¹, LI Yanbin²

(1. School of Electrical Engineering and Information, Sichuan University, Chengdu Sichuan 610065, China;

2. Southwest China Institute of Electronic Technology, China Electronics Technology Group Corporation, Chengdu Sichuan 610036, China)

Abstract: Fault-Tolerant Common Object Request Broker Architecture (FT-CORBA) released by Object Management Group (OMG) provides fault-tolerance by replication. However, one of the main drawbacks of the FT-CORBA is that it does not support network partition, which limits the availability of the system severely since it makes the system access the states of adequate replications again and again. By adding appendix to the ORB middleware, classifying the requirement of the replication's uniformity into two levels, and acting according to the different levels, distributed system can better perform during partition failure. Considering the situation that network has been partitioned into 3 and 10 subnets: when there are three subnets, improved replication mechanism has pretty good performance no matter how many nodes the subnets contain; when the network is divided into ten subnets, the availability of the distributed system is mostly dependent on the number of the nodes that a single subnet contains. If no subnet contains most nodes of the distributed system, the availability also stays low but still higher than before. Since solution is applied to an off-the-set ORB, there is no need to have ORB's source code modified.

Key words: Common Object Request Broker Architecture (CORBA); fault tolerance; entity redundancy; network partition; object replication; distributed system

0 引言

当今越来越多的应用由于其规模和复杂性,需要在分布式系统中完成,而许多分布式应用都需要系统提供容错机制^[1],例如航空交通管制、网上支付应用等,若没有容错措施,系统中任何一个环节的故障都可能造成巨大损失。而分布式系统各节点使用不同的操作平台和不同的通信协议,容错机制在系统不同层次上实现,对于每一个新的操作平台都要重新研究相应的容错机制,这样就大大降低了系统的可扩展性,并极大地增加了开发难度和开发成本。

公共对象请求代理体系结构(Common Object Request Broker Architecture, CORBA)^[2]是由对象管理组织(Object Management Group, OMG)提出的基于分布式对象的中间件规范,该规范中的接口定义语言(Interface Description Language, IDL)描述了客户对象请求和对对象执行接口,完整地定义了接

口并说明每个操作参数。IDL 实现了组件间交互无需考虑操作平台和通信协议的不同。将分布式系统与 CORBA 结合^[3-4],并将容错机制交由 CORBA 实现能很好地解决上述问题。

容错 CORBA(Fault Tolerant CORBA, FT-CORBA)介绍了一种基于冗余^[5]的容错机制,但没有考虑网络分割的情况,当网络出现故障,形成两个或多个相互隔离的子网,各子网对象状态因为不能互通所以不能保持一致,从而大大限制了操作的执行。因此,研究一种新的能够应对网络分割情况的 FT-CORBA 就显得非常必要了。

1 体系结构

1.1 CORBA 体系结构及特性

客户端通过接入对象引用(Object Reference)、了解对象类型和希望实现的操作来执行请求^[6]。客户端可静态(访问

收稿日期:2013-02-16;修回日期:2013-05-11。

作者简介:陈雅琳(1987-),女,四川成都人,硕士研究生,主要研究方向:信号与信息处理;黄宏光(1963-),男,四川成都人,副教授,主要研究方向:宽带通信网、网络管理与监测;李燕斌(1970-),男,四川成都人,高级工程师,主要研究方向:通信信号与信息处理、软件无线电。

特定对象的存根)或动态地建立请求。用动态和静态接口建立请求使用的是相同的请求语法,信息的接收方分辨不出请求是用哪种方式建立的。在执行一些功能时客户端也可直接与 ORB 交互。

ORB 将对象执行 (Object Implementation) 相应的执行代码、传输参数和传输控制信息放入 IDL 骨架或动态骨架。骨架是与特定的接口和对象适配器相对应的。对象执行可通过静态 IDL 骨架或动态骨架来接收请求。在处理对象请求或其他情况下,对象执行可能会包含从对象适配器传来的 ORB 服务。当请求执行完成后,控制和执行结果将会返回给客户端。

客户端和对象执行是如何获悉接口和执行信息的呢,接口定义在 OMG IDL 或接口库中,这些定义将用于产生客户端存根和对象执行骨架;对象执行信息则是存储在执行库中。

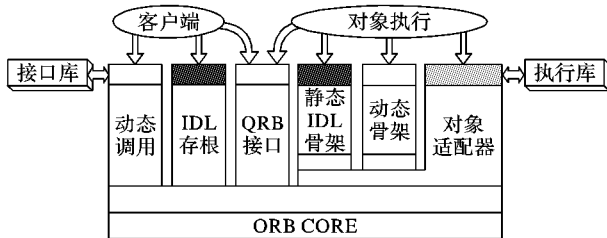


图1 CORBA 体系结构

1.2 基于 FT-CORBA 的分布式系统

FT-CORBA 通过对象冗余来实现容错,一个操作交由多个复制对象来执行,将相同的大多数复制对象执行的结果作为这个操作的最终结果,这就需要有一个对象管理器来管理这些复制对象。现在许多应用需要在一个非常复杂和庞大的分布式系统中来完成,由一个复制管理器来管理庞大的分布式系统中所有的应用变得不切实际。如图 2,将分布式系统分为若干个容错域,一个容错域包含一个复制管理器、若干个主机和多个对象组;一个主机可包含多个复制对象,也可以属于多个容错域。

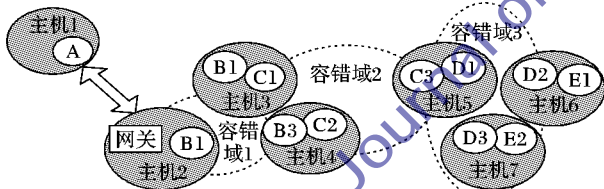


图2 基于 FT-CORBA 的分布式系统

2 FT-CORBA

2.1 FT-CORBA 模型

FT-CORBA 标准致力于为应用提供完整的可靠性,它要求整个过程中不能出现任何错误。容错机制^[7]通过组件冗余、故障监测和恢复来实现。组件冗余在 FT-CORBA 中体现为对象复制,与服务器复制相比,在复制个数和主机分配的配置管理方面,对象复制显得更为灵活。

复制管理器将一个操作分配给多个复制对象执行,并选出一个作为复制主对象,当复制对象出现故障时根据故障报告采取相应措施,如重新选择复制对象,或者从复制对象中重新选择主对象等。复制对象服务端完成用户请求的操作,并将复制对象状态及操作结果发送给复制主对象。复制主对象服务端除了完成复制对象需要完成的任务以外,还要接收所有其他复制对象的操作结果及复制对象状态,并判定接收到的复制对象状态是否为最新,接收到的复制对象数是否满足判决要求,如果满足要求,则将大部分相同复制对象返回的结果作为该操作的结果返回给用户对象;如果不满足,则重新执

行该操作。故障监测器通过定时地向对象服务端发送请求来监测该对象是否出现故障或者是否被分割到其他子网中,如果复制对象出现故障,对象服务端的故障监测器向故障通知器发送故障通知,故障通知器将故障报告发送给复制管理器。恢复机制模块存储对象最新状态,当复制对象出现故障需要复时,就调用恢复机制模块里的信息恢复。日志机制模块存储对象服务端的状态信息。

FT-CORBA 标准支持一系列容错机制,包括重复请求、重定向到其他服务器、主动复制和被动复制,并且允许用户为单个复制对象定义容错参数。标准 FT-CORBA 主要包括图 3 中的复制管理器、故障通知和故障检测器三部分。

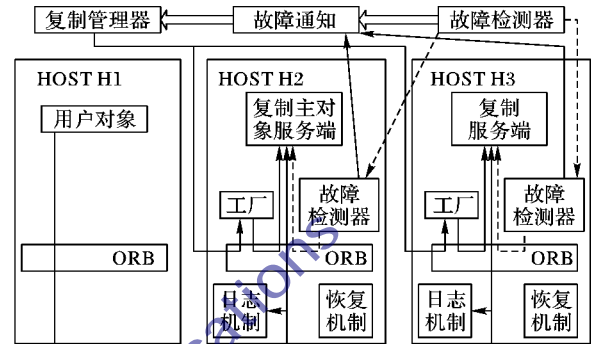


图3 标准 FT-CORBA 框图

1) 创建对象组,实现对象冗余。

a) 用户端应用调用 `resolve_initial_references()` 函数获取复制管理器引用。

b) 用户端应用通过复制管理器里 `GenericFactory` 接口,调用 `create_object()` 函数提供 `type_id` 和 `the_criteria`,然后返回对象组引用和 `factory_creation_id` 给复制管理器,其中 `factory_creation_id` 用于之后复制管理器调用 `GenericFactory` 接口函数 `delete_object()` 删除对象组。

c) 参数管理器接口为对象组提供设置参数的操作,例如复制类型、成员类型、一致性类型、最小复制数、初始化复制数等。

d) 复制管理器调用本地对象库,在合适的位置创建对象组,并且满足 `InitialNumberReplicas` 和 `MinimumNumberReplicas` 参数值,即对象组的初始对象数和执行过程中对象组包含的最小对象数。

e) 复制管理器决定出对象组中的主对象,创建包含容错域对象代表的 `TAG_FT_GROUP` 组件,并允许其他组件访问主对象和对象组中各对象的状态版本;然后复制管理器创建对象组引用。

f) 对每一个网关,复制管理器建立一个 `TAG_INTERNET_IOP` 文件,它包含网关内主机、端口和允许被访问的 `TAG_FT_GROUP` 组件;然后复制管理器在 `TAG_INTERNET_IOP` 文件中增加对象组地址。

g) 复制管理器记录对象组引用。

h) 对每一个成员,复制管理器将复制对象成员添加到对象组;根据复制类型激活成员;核对复制类型、故障监测类型和故障监测间隔,然后决定是否初始化该成员的故障监测功能;最后,复制管理器将自己或由它创建的对象登记到故障通知器,用来接收该成员的故障通知。

i) 对于 `COLD_PASSIVE` 和 `WARM_PASSIVE` 复制类型,复制管理器决定对象组的主对象,将 `TAG_FT_PRIMARY` 组件包含到该对象的文件中,只有主对象才能执行调用到对象组上的方法。

j) 复制管理器将对象组引用返回给应用。

2) 故障监测和恢复。

a) 复制管理器告知故障监测器开始监测对象。

b) 故障监测器定期调用 `is_alive()` 函数作用于对象组内的主对象, 主对象回复该请求。

c) 若主对象出现故障, 没有回复 `is_alive()`, 监测器等待超时, 则向故障通知器报告主对象出现故障, 然后故障通知器告知复制管理器主对象出现故障。

d) 复制管理器确认对象组包含这个主对象和该对象组的复制类型, 并从对象组中移除该主对象, 并调用容错组件为对象组指定新的主对象。如果当前对象组的成员少于最小成员数, 那复制管理器开始建立新的成员。

e) 如果候补对象还没有进入工作状态, 那么复制管理器激活该对象。

f) 恢复机制调用 `set_state()`, 返回信息给新主对象。

3) 容错机制。

a) 设一致性类型为 `CONS_INF_CTRL`, 容错组件通过检查、日志、激活和恢复操作维持复制对象一致性。

b) 日志机制定期调用 `get_state()` 获取主对象的状态和行为, 并将其录入日志, 当主对象出现故障时, 可取出这些状态信息来恢复一个后补对象或建立一个新对象。

c) 通常情况下采用大数表决法^[5]来实现容错, 表决器从参数管理器处获取对象组成员个数 N , 当主对象接收到 $N/2$ 或 $N/3$ 个相同消息, 则判决该消息正确, 无需等待其他消息到达, 并丢弃掉与正确消息不同的对象。若全部消息都已到达, 但仍没有接收到 $N/2$ 或 $N/3$ 个相同消息, 则重启应用。

d) 状态传输之后, 一个对象组内的成员状态必须与主对象保持一致。

2.2 FT-CORBA 缺点

由 2.1 节可以知道, 系统中出现的错误, 可通过对象冗余的方法解决, 但上述 FT-CORBA 有一个主要缺点, 在网络分割的情况下, FT-CORBA 的可执行性大大降低甚至为零。网络分割^[3]即当网络不稳定时, 网络中的部分节点与其他节点间的所有网络连接同时发生故障, 造成节点与节点间不能互通, 形成两个或多个相互隔绝的子网, 如图 2, 容错域 3 可能与容错域 1 和容错域 2 完全隔绝。这样就导致容错域 3 中的复制对象无法与容错域 1 或 2 中的对象组中的主对象保持一致, 不同子网中的对象状态版本可能不同, 如 C3 与 C2 和 C1 的状态因为网络故障无法保持一致。由于这些差异只有当网络故障被维护、系统恢复后才能监测得到, 那么在网络分割的情况下就无法保持所有对象状态一致性, 如果依照 FT-CORBA 标准则会大大限制操作, 降低系统的可行性。例如在 2.1 节中, 出现网络分割时, 2) 中的 b)、c)、d) 和 3) 中的 c) 操作会反复执行, 操作达不到对象组所要求的最少成员数, 或使应用频繁地替换主对象, 甚至重启, 这样大大增加了网络负担, 且操作不能继续往下执行。

3 改进后的 FT-CORBA

本文在已有的 CORBA 架构添加协议组件, 在出现网络分割时, 根据协议执行额外的一系列操作来提高系统的执行能力, 这种方法的重大优势在于不用改变 ORB 源码, 且可移植性、可扩展性强。根据 OMG 颁布的 FT-CORBA, 网络中各对象状态严格一致, 当出现网络分割时, 由于子网中各节点之间不能通信, 因而无法满足标准 FT-CORBA 中对象状态的完整一致性要求。本文将增加的针对网络分割的操作规范封装

到协议组件^[8]中。该协议允许对象状态在网络出现故障的情况下有差异; 当系统恢复正常时, 各对象状态又重新保持严格一致; 在网络分割环境下系统已经做出的操作, 根据应用的一致性等级采取不同的恢复措施。

出现网络分割的节点数是随机的, 且节点恢复的顺序与产生分割的顺序是相互独立的。只有当网络连接恢复时, 才能区分节点是出现故障还是网络分割, 因此, 将节点故障按照出现网络分割的情况处理。

下面将讨论改进后的容错机制, 由于容错机制的复制和状态一致性机制的实现是通过在 CORBA 上添加中间件附件, 协议内容由专门的协议组件来完成, 所以无需改变已有的 ORB 源代码^[8-9]。

3.1 改进后的容错机制

本文提出的容错机制放宽了对象状态在网络分割时对对象状态一致性的要求, 但当网络恢复正常时, 对象状态又重新保持严格一致。

对象状态需满足所有约束条件操作才能继续往下执行, 而约束条件则与具体的对象相关。标准 FT-CORBA 要求所有对象状态都是最新的, 即严格与主对象保持同步; 同时, 在操作执行前后对象状态要满足先决条件或者后决条件。例如网上购票系统, 所有对象状态必须一致, 如图 2, 不能出现在 B1、B2 对象上同一班次火车票为可购买, 在 B3 对象上该次列车火车票为已售完; 其次, 在购票操作前, 对象要满足先决条件, 即一个用户最多购五张票; 同时满足后决条件, 即同一座位火车票只能被一人购买。

将对象一致性要求分为 C1 和 C2 级: C1 级为严格的一致性要求, 即要求各对象状态必须保持严格一致; C2 级则允许对象间的状态存在差异。改进后的容错机制, 在 2.1 节的 3) 中加入了以下操作:

1) 如果无法连接上主对象, 新的对象会被选出执行主对象的任务, 但当该操作有 C1 级先决条件或后决条件时, 就不能执行该操作。

2) 如果在评估后决条件时, 对象状态还没有更新, 那么该对象回滚, 即恢复到先前状态, 然后重新执行操作。

3) 如果对象状态没有及时更新, 而对象一致性要求为 C1 级, 操作将直接被丢弃; 若一致性要求为 C2, 操作将保留。

4) 网络恢复时, 各对象状态恢复一致后, 对操作的后决条件重新评估。

5) 当操作包含需回滚的对象时, 那么该操作不能继续执行。

仍以网上购票系统为例, 先决条件可以设为 C1 级, 即出现网络分割, 由于主对象所在子网没能更新用户对象状态, 用户实已购到五张票而显示不足五张, 用户在此状态下购第六张票仍可继续执行且有效; 后决条件同一座位只能被一人购买设为 C1 级, 即网络分割条件下, 主对象不能获取该座位最新状态是空闲还是已售出, 则此操作将回滚。

3.2 仿真实验

假设分布式系统包含 100 个节点, 500 个对象, 每个对象有 10 个复制服务对象, 所有对象的执行都没有约束条件。

图 4 为标准 FT-CORBA 的分布式系统的可行度, 纵坐标 av 表示系统可行度, 横坐标 pm 为包含主对象的分割子网的节点数占全网节点数的百分比, 设其余子网包含相同的节点数。由于产生其不可行的原因为分割子网的节点数达不到一个操作所需的最小复制数的要求及其在进行容错判决时对复制对象个数的要求, 这里设子网节点数为系统总节点数的

80% 以上为满足以上两个要求。此外,操作直接调用对象在分割子网外也会造成操作失败。当子网节点数不满足最小复制对象个数和容错判决对象个数的要求时,系统的可行度为 0。当满足临界条件时,系统可行度可表示为子网包含所有直接调用的对象的概率:

$$P = pm/N \quad (1)$$

其中: pm 为最大子网包含节点数占分布式系统中总节点数的百分比, N 为分布式系统的总节点数。

基于改进后的 FT-CORBA 的分布式系统,造成操作不能继续执行的原因就只有在分割网络中无法连接到复制对象。则系统可行度可用式(2)^[10]表示:

$$P = 1 - \sum_{i=1}^q \frac{P_i}{N} \left[1 - \left(1 - \left(1 - \frac{r}{N} \right)^{P_i} \right)^{oD} \right] \quad (2)$$

其中: q 为分割子网的个数, P_i 为第 i 个子网包含的节点数, N 为总的节点数, r 为每个对象的复制个数, oD 为一个操作直接调用的对象总数。设一个操作直接调用的对象数为 3, 一个操作有 2 层调用, 即调用的对象再调用一次对象。这里对比子网数 $q = 3$ 和 $q = 10$ 的情况下, 标准 FT-CORBA 和改进后的 FT-CORBA 的可行性, 其余参数模拟实际情况设为定值(如表 1), 利用 Matlab 进行仿真。

表 1 仿真参数

仿真参数	参数值
系统节点数	100
对象数	500
每个对象的复制服务对象数	10
分割子网数	$q = 3$ 和 $q = 10$
一个操作平均直接调用对象数	3
一个操作平均调用层数	2

图 4 为标准 FT-CORBA 系统可行度, 因为只有当子网包含节点数超过分布式系统总节点数 80% 时主对象复制服务端才达到判决条件, 所以当没有子网数超过总节点数 80% 时, 系统重复执行删除对象组、建立新对象组、选择主对象等操作, 无法进入到判决阶段, 系统可行度为 0。当最大子网节点数超过 80% 时, 系统可行度为子网包含所有直接调用对象的概率。

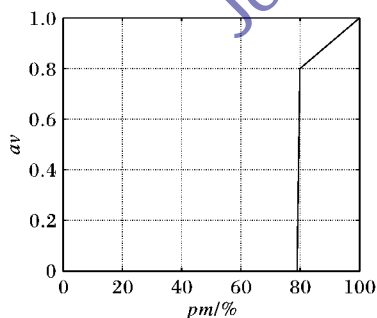


图 4 标准 FT-CORBA 系统可行度

图 5 为改进后 FT-CORBA 系统在 $q = 10$ 和 $q = 3$ 时的可行度比较。当 $q = 10$ 时, 系统可行度与包含最多节点子网子节点数近似成线性关系, 当所有子网包含的节点数小于系统总节点数的 15% 时, 这种情况下网络几乎处于瘫痪状态, 整个分布式系统因为网络故障被分割成 10 个零碎的子网, 系统可行度趋于 0。随着最大子网包含的节点数增加, 系统的可行度呈线性上升趋势, 在最大子网数低于总节点数 80% 时, 系统可行度远优于标准 FT-CORBA; 当最大子网数高于总节点数 80% 时, 系统可行度与标准 FT-CORBA 相同。当 $q = 3$

时, 不论子网以怎样的形式被分割, 系统的可执行度均在 0.65 以上, 可执行度远优于标准 FT-CORBA。

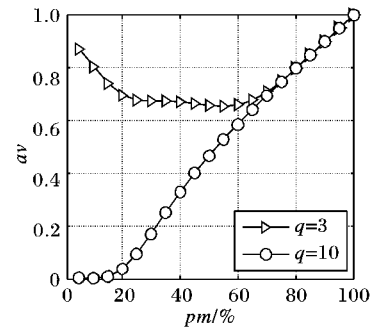


图 5 改进后的 FT-CORBA 系统可行度

由于网络分割成子网的大小都是随机的, 所以可行度会因为各子网大小的不同会有一些浮动, 但由于本例是以将其中一个子网所占节点由 1% 遍历至 100%, 其余子网包含相同的节点数, 基本涵盖了所有子网分割的特性, 所以系统可行度总体趋势与图 5 相同。可见改进后的 FT-CORBA 对网络分割有很好的适应性。

综上所述, 当最大子网包含节点超过 80% 时, 改进后的 FT-CORBA 与标准 FT-CORBA 的可执行度近乎完全相同, 只有当最大子网包含节点数低于 80% 时, 改进后的 FT-CORBA 优势才突显出来。当网络分割情况不严重, 即分割子网数比较少, 或者最大子网包含节点数较多时, 改进的 FT-CORBA 在很大程度上能提高系统的可行度。当网络故障严重, 网络被分割成的子网规模越小, 数量越多, 系统可行性也随着降低, 此时应该采取措施恢复网络, 但总体系统的可行度大大高于基于标准 FT-CORBA 的分布式系统的可行度。

对于对象执行有约束条件的, 由于标准 FT-CORBA 没有考虑这方面情况, 主对象无法获得复制对象的最新状态, 不能对约束条件进行评估, 导致操作不能往下执行, 系统可行性几乎为 0。而改进后的 FT-CORBA 的可行度在有约束条件的情况下有一定幅度的下降, 当此类对象比例很高时, 系统可行度也会很低, 但仍然比基于标准 FT-CORBA 系统可行度要高。

4 结语

本文详细分析介绍了 FT-CORBA 的工作原理和流程, 指出其存在的主要缺点及产生的原因。提出了解决网络分割问题的方法, 改进了原 FT-CORBA 中对象复制和恢复机制, 通过在 CORBA 上添加中间件附件的方法实现改进后的复制和恢复机制, 并在一定程度上验证对比了该方法的有效性。本文仿真验证只是讨论了对象执行没有约束条件的情况, 可以针对一个对象操作具有多个约束条件, 且可能处于不同一致性等级的实际情况作进一步研究。

参考文献:

- [1] LI B, BISWAS S, ORTIZ A, *et al.* Survivability analysis of reconfigurable systems [C]// Proceedings of 2007 IEEE International Conference on Industrial Engineering and Engineering Management. Piscataway: IEEE, 2007: 663-667.
- [2] Object Management Group. Common object request broker architecture: core specification, Version 3.0.3-2004 [S]. [S. l.]: OMG, 2004.
- [3] 杨茂江, 孙星明, 朱建秋, 等. 基于 CORBA-WEB 的分布式应用系统开发策略 [J]. 计算机工程与应用, 2000, 36(2): 21-24.

(下转第 2135 页)

高,在相同时间内识别的标签就越多,所以吞吐率也会越高。总体来说,多标签时,基于标签分组的算法性能优于另两种常用算法。

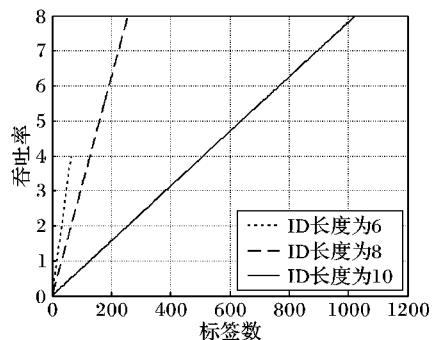


图2 标签分组算法在不同ID长度下的吞吐率

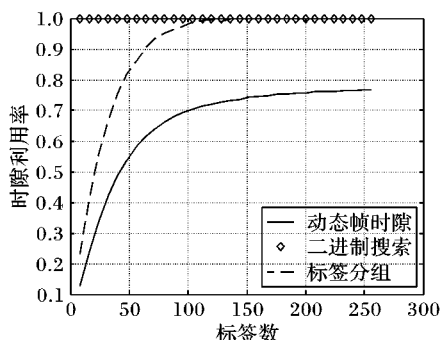


图3 各算法的时隙利用率比较

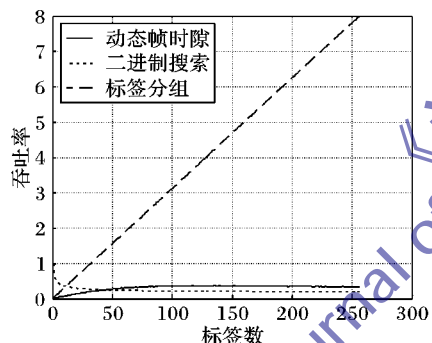


图4 各算法的吞吐率比较

4 结语

本文在介绍常用的 ALOHA 和树型防碰撞算法后设计了结合两者优点的基于标签分组的帧时隙 ALOHA 算法,并对三种算法的吞吐率进行了比较。由于新算法将标签分组,并利用特征位对同组内的标签进行区别,既避免了不同组标签的碰撞,又使得一个时隙内可以鉴别一组标签,在标签数较多

时具有明显的优势。文中对标签分组的基本思想和实现步骤分别作了介绍,并用软件进行了验证。当然该算法也存在不足,当标签在码表中比较分散或标签较少时将不可避免产生大量的空闲时隙,性能将明显下降,因此算法还有待改善与提高。

参考文献:

- [1] 彭力. 无线射频识别(RFID)技术基础[M]. 北京:北京航空航天大学出版社,2012:130-152.
- [2] 赵军辉. 射频识别技术与应用[M]. 北京:机械工业出版社,2008:198-207.
- [3] 胡玲敏. RFID系统的防碰撞算法研究[D]. 杭州:杭州电子科技大学,2011.
- [4] CHOI J S, LEE H, ENGELS D W, *et al.* Passive UHF RFID-based localization using detection of tag interference on smart shelf[J]. *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, 2012, 42(2): 268-275.
- [5] 魏欣. RFID标签及阅读器防冲突算法研究[D]. 成都:电子科技大学,2009.
- [6] 王建伟,赵玉萍, KORHONEN T. RFID系统防碰撞协议研究——设计与优化[J]. *电子与信息学报*, 2009, 31(1): 1-4.
- [7] 陈中平. 基于RFID的混合型标签防碰撞算法研究[D]. 南昌:南昌大学,2012.
- [8] LI J, JIN M S, LIU J. An enhanced slotted ALOHA algorithm for anti-collision in RFID system[C]// *Advances in Future Computer and Control Systems, Advances Intelligent and Soft Computing* 159. Berlin: Springer, 2012, 1: 119-123.
- [9] CHEN Y H, HORNG S J, RUN R S, *et al.* A novel anti-collision algorithm in RFID systems for identifying passive tags[J]. *IEEE Transactions on Industrial Informatics*, 2010, 6(1): 105-121.
- [10] AGRAWAL T, BISWAS P K, RAOOT A D. An optimized query tree algorithm in RFID inventory tracking a case study evidence[J]. *International Journal of Computer Science Issues*, 2012, 9(4): 85-93.
- [11] CHOI S, CHOI J, YOO J. An efficient anti-collision protocol for tag identification in RFID systems with capture effect[C]// *Proceedings of 2012 Fourth International Conference on Ubiquitous and Future Networks*. Piscataway: IEEE, 2012: 482-483.
- [12] 刘迟时,王春华,伏奎. 新的基于分组处理的射频识别标签防碰撞算法[J]. *计算机应用*, 2013, 33(2): 592-599.
- [13] 张学军,王娟. 基于标签识别码分组的连续识别防碰撞算法研究[J]. *电子与信息学报*, 2011, 33(5): 1159-1165.
- [14] 陆宝春,丁日春. 基于自动分组排列的电子标签防碰撞算法[J]. *南京理工大学学报*, 2012, 36(1): 122-126.
- [15] 孙丽华. 信息论与编码[M]. 北京:电子工业出版社,2009:260-275.
- [16] ZHEN B, KOBAYASHI M, SHIMIZU M. Framed ALOHA for multiple RFID objects identification[J]. *IEICE Transactions on Communications*, 2005, 88(B): 991-999.

(上接第2127页)

- [4] 陈源,王元钦,刘莹. 基于CORBA的扩展型事件服务模型设计[J]. *计算机应用*, 2011, 31(S1): 138-140, 143.
- [5] 郑尚书,沈立炜,彭鑫,等. 基于CORBA的自适应系统实现[J]. *计算机工程*, 2011, 37(19): 239-242, 257.
- [6] 赵瑜,刘勇,孔捷. 基于CORBA组件的分布式网管软件设计[J]. *无线电工程*, 2012, 42(7): 4-6, 54.
- [7] 刘宏月,马建峰,王超. 基于容错CORBA的可生存网络应用模型[J]. *华中科技大学学报:自然科学版*, 2010, 38(10): 26-30.
- [8] BEYER S, BAÑULS M-C, GALDÁMEZ P, *et al.* Increasing availability in a replicated partitionable distributed object system[C]// *ISPA 2006: Proceedings of the 4th International Conference on Par-*

allel and Distributed Processing and Applications, LNCS 4330. Berlin: Springer-Verlag, 2006: 682-695.

- [9] BEYER S, MUNOS-ESCOI F D, GALDÁMEZ P. Implementing network partition-aware fault-tolerant CORBA systems[C]// *ARES07: Proceedings of the Second International Conference on Availability, Reliability and Security*. Washington, DC: IEEE Computer Society, 2007: 69-76.
- [10] OSRAEL J, FROIHOFFER L, GOESCHKA K M, *et al.* A system architecture for enhanced availability of tightly coupled distributed systems[C]// *ARES 2006: Proceedings of the First International Conference on Availability, Reliability and Security*. Washington, DC: IEEE Computer Society, 2006: 400-407.