一种适应性复制协议的研究与设计

赵 东[1],姚绍文[1,2],周明天[1]

(1. 电子科技大学计算机科学与工程学院,四川成都 610054;2. 云南大学信息学院,云南昆明 650091)

摘 要: 传统的复制协议往往对如何适应遗留系统和引入复制技术对系统性能造成的影响方面考虑不够深入.本文在分析传统复制协议的基础上,提出了一种新协议.其设计采用复制实例数目可动态伸缩的逻辑令牌环结构,因而在满足可用性需求的前提下,可支持不同分布系统对适应性的需求,同时保证引入复制技术后系统的性能.文中详细论述了相应的系统模型和服务请求的处理,并通过与几种典型的传统复制协议的比较,证明了该协议的可行性.

关键词: 复制; 适应性; 可用性; 性能

中图分类号: TP302 **文献标识码:** A **文章编号:** 0372-2112 (2002) 12A-1991-04

Research and Design of an Adaptable Replication Protocol

ZHAO Dong[1], YAO Shao-wen[1,2], ZHOU Ming-tian[1]

(1. *College of Computer Sci. and Eng., University of Elec. Sci. and Tech. of China, Chengdu, Sichuan 610054, China*;
2. *School of Information, Yunnan University, Kunming, Yunnan 650091, China*)

Abstract: Traditional replication protocols don't take those issues into consideration such as adaptation to legacy system as well as affection to the system performance after introducing replication. Based on discussion about these protocols, a new replication protocol is addressed in this paper. Its design adopts logical token ring architecture with dynamically increasing/decreasing replicas. This paper discusses the system model and the request processing procedure of the protocol. With precondition to meet availability requirement, it not only enables adaptability to various distributed systems but also guarantees high performance of these systems. The comparison between this protocol and other typical replication protocols shows that it is feasible.

Key words: replication; adaptability; availability; performance

1 引言

随着分布系统在许多关键业务领域的使用变得越来越广泛,相应地,对于高可用服务的需求日益增长.目前,复制技术已广泛用于分布系统中以支持高可用服务.支持复制服务的典型协议包括 Active Replication(也称为状态机方法[1]),Primary Backup[2],ROWAA[3](Read Once/Write All Available),Majority[4]和 Grid[5]等.

不幸的是,传统的复制协议在支持可用性的同时,往往也损害了分布系统的性能并对系统的设计施加了许多限制,从而导致它们大多只能适应某一类型的分布应用.针对这一问题,我们设计了一种可支持不同类型分布应用的高性能复制协议.本文给出了该协议的系统模型和协议描述,详细说明了该协议如何提供高可用性,适应各种不同类型应用,并进行了相应的性能分析,论证了该协议能够保证系统的整体性能.

2 协议设计

我们的复制协议称为 ROWC(Read Once/Write Circularly),其设计思想如下:(1)通过在客户端和复制服务之间引入复制服务适配器来适应遗留系统需求,保证采用不同通信协议的各种客户端都能透明地访问高可用的复制服务;(2) 在保障系统必须满足的可用性前提下提高系统的性能;(3) 对复制服务实例数目基本无限制,且在配置同样数目复制服务实例的前提下提供最大的可用性,从而可充分利用现有资源.

为方便讨论,本文以下部分假设所有的复制服务运行在同一个支持广播技术的局域网中,因而不考虑网络分区故障,客户则可通过广域网或局域网请求服务.考虑到本节的重点在于 ROWC 协议的系统模型和服务请求的正常处理过程,因而略去了故障处理和协议的正确性证明.

2.1 基本定义

本文中,接收定义为一个服务 s 接收请求信息 m 的动作,交付则定义为 s 将请求传递给应用层进行处理的动作,分别记为 $s.receive(m)$ 和 $s.deliver(m)$.配置定义为某一时间段 T 内活动的服务集合 C,即 $C=\{s \mid s \text{ is running in } T\}$.

许多复杂的分布系统中往往有各种不同类型的应用,这些应用对于复制服务有各不相同的一致性和正确性需求,相应地出现了不同的复制服务类型[6],按其复杂程度从低到高依次为:基本级(basic),先进先出级(fifo),因果顺序级(causal),一致级(agreed)和安全级(safe).由于提供可靠的基本级服务和先进先出级服务相对比较简单,因此本文不再赘述.而 ROWC 的设计通过对一致级服务的优化大大减小了其开销,使得单独提供因果顺序级服务也没有必要.因此,本文以下部分着重说明 ROWC 如何支持高可用的一致级服务与安全级服务.

收稿日期:2002-05-30;修回日期:2002-09-08
基金项目:国家高技术研究发展计划项目(No.863-301-7-9);四川省重点科技攻关项目(No.SG95.17.1)

文中,一致级服务定义为不同请求间的全序关系,即任意两个消息间都具有全局的先后顺序.在某个配置 C 中,当且仅当满足如下条件时,s 按一致级顺序交付消息 m:(1)$s\in C$,$s.receive(m)$已发生,且 m 是由 C 中或 C 的前一配置中的某个成员生成的;(2)s 不会在 C 中的同一时刻按一致级顺序交付两条不同的消息;(3)s 已交付了先于 m 且发生在 T 内的所有消息 m';(4)对所有 $r\in\{r\mid r\in C$ 且 $r\neq s\}$,若在 C 中 $s.deliver(m)$ 先于 $s.deliver(n)$发生,则 $r.deliver(n)$不会先于 $r.deliver(m)$发生.

安全级服务则定义为:当且仅当满足如下条件时,配置 C 中的服务 s 按安全级顺序交付消息 m:(1)$s.deliver(m)$满足一致级顺序;(2)对于所有 $r\in\{r\mid r\in C$ 且 $r\neq s\}$,s 知道 $r.receive(m)$已发生,将进行 $r.deliver(m)$.

2.2 系统模型

如图 1 所示,ROWC 协议的系统模型采用分层设计,基于分布对象观点建立.每种服务对象可部署多个复制实例(图中 SR_1,SR_2 和 SR_3)到不同的服务器(图中 S_1,S_2 和 S_3)中,这些实例组成一个复制服务组.服务器负责为复制实例提供运行时环境,通常位于不同的主机中.复制服务有自己的内部状态和服务方法,客户端对服务的请求体现为相应的方法调用.对于服务状态的影响而言,方法可分为"读"方法和"写"方法.读方法不改变服务的状态,而写方法可能会改变服务的状态.

根据我们先前的研究[7],ROWC 中引入了 SA(Service Adapter,即服务适配器),用于分隔客户端与复制服务.SA 支持各种类型的现有客户端透明地访问服务端,即感觉不到引入复制服务后的影响,从而解决了支持复制服务与兼容遗留系统之间的矛盾,,即 ROWC 可满足适应性需求.

采用图 1 的系统模型,客户端与 SA 之间的通信协议 CAP(Client-Adapter Protocol)独立于 SA 与后端复制服务之间通信所采用的协议 ARP(Adapter-Replica Protocol).因此 ROWC 可支持现有的各种协议,如 RMI,IIOP,DCOM,HTTP 或 TCP 协议等等,只需要实现支持这些 CAP 协议的 SA 即可.每个 SA 配置一个 CRQ(Client Request Queue,即客户请求队列)和一个 SRQ(Service Request Queue,即服务应答队列),可支持多个客户的并发访问.同一个客户在任一时刻可连接到一个 SA 上并请求服务,SA 则负责调度复制实例完成请求,并将应答返回给客户.同时,SA 还将应答保存在 SRQ 中,直到客户确认已收到该应答.设置 CRQ 的作用主要是为了检测客户端的重复请求,使之对后端的复制服务透明.此外,在重载情况下,CRQ 还可提供服务端的第一层缓冲,以免使复制服务过载.设置 SRQ 的目的则用于防止应答的丢失.

属于同一复制服务组的各个实例构成一个逻辑令牌环.在任意时刻,复制服务组中有且仅有一个实例持有令牌.SA 可以通过 ARP 协议向复制组发送消息或接收复制组成员的消息.然而,对于复制组成员而言,只有持有令牌的成员才能向环中发送消息,而其它成员只能接收消息.组成员之间的通信协议称为 TRP(Token Ring Protocol),在该协议中,令牌持有者负责对它接收到但尚未交付的消息进行全局排序,以提供一致级服务和安全级服务所需的全局顺序.TRP 协议采用捎带技术,全局排序信息在令牌中传递.每个成员持有令牌的时间是受限的.当超过限制时间时,若持有令牌的成员仍未收到 SA 多播的请求消息,则它多播一个空令牌给整个复制组,通知令牌已转移给环中的后继实例.为支持排序,每个复制实例都有自己的 ARQ(Adapter Request Queue,即适配器请求队列),用于缓存接收到的 SA 消息.此外,每个实例还配置了自己的 RSRQ(Replicated Service Reply Queue,即复制服务应答队列),用于保存回送给 SA 的应答消息.

2.3 一致级服务写请求的正常处理过程

图 2 示出当复制实例数为 3 时,ROWC 协议对一条一致级服务写请求进行处理的正常过程.客户端 C 通过某种 CAP 协议向 SA 发送服务请求消息 m_{creq},SA 将接收到的 m_{creq} 添加到 CRQ 中并赋予一个序号.SA 接着使用 ARP 协议将赋予 SA 序号的服务请求 m_{mreq} 多播给整个复制组.对于不持有令牌的成员,它们只是简单地添加该请求到 ARQ 中.当前正持有令牌的复制实例 SR_2 接收到该请求后,则要负责为 m_{mreq} 赋予一个全局序号.全局序号与 SA 序号区别在于它对于来自所有 SA 的每条请求消息是唯一的,而 SA 序号只对生成该序号的 SA 唯一.随后,SR_2 将 SA 序号、新生成的全局序号和 SR_2 的实例标识符添加到令牌消息 m_{mt} 中,并多播 m_{mt} 给整个复制组,表示将令牌传递给其后继 SR_3.按照 ROWC 的应答策略,复制组中仅有一个实例,即生成对应请求全局序号的 SR_2,负责回送应答给 SA,以免产生 Primary Backup 和 Active Replication 中易导致的应答风暴.若 SR_2 已交付小于新全局序号的所有请求,则它可交付新请求,并通过 ARP 协议向 SA 单播发送处理结果 m_{mrly}.最后,SA 向客户发送应答 m_{srly},完成该次服务请求.当 SR_1 和 SR_3 接收到 SR_2 多播的令牌后,获知 m_{mreq} 的全局序号和 SA 序号,并可在 ARQ 中找到先前缓存的 m_{mreq}.若已交付先于 m_{mreq} 的所有请求,则可交付 m_{mreq};否则将 m_{mreq} 赋予令牌中指定的全局序号并重新插入到 ARQ 相应的位置中.

2.4 一致级服务读请求的正常处理过程

为防止与写请求产生冲突,ROWC 同样需要对读请求进行全局排序,以保证数据完整性.由于读请求不改变复制服务的状态,因此,不同的复制服务实例可并行执行多个连续的读请求(即其中不含有写请求).与写请求处理过程的区别是,每个读请求只会在选定的实例中执行一次.如图 3 所示,SA_1 和 SA_2 通过 ARP 协议分别发送来自于 C_1 的读请求 mc_{req2} 和 C_2 的读请求 m_{creq1} 给复制组.当持有令牌的复制实例 SR_2 接收到多个连续的读请求(图中 m_{mreq1} 和 m_{mreq2})时,它为这些读请求赋予与写请求类似的递增的全局序号.接下来的处理则与写请求不同.该复制实例按照某种负载均衡策略(例如简单轮转法)分别选定执行不同读请求的复制实例,随后在多播的令牌消息中捎带这一信息,通知选定的这些实例该执行哪些读请求.每个选定的实例接收到该令牌时,首先确认是否已交付了全局序号小于指定该实例执行的读请求的所有写请求.若已交付,则可立即执行读请求并向 SA 返回应答;否则需要将读请求在 ARQ 中排队,等待实例交付所有这些写请求后再处理.

2.5 安全级服务请求的处理过程

在 ROWC 中,我们对安全级服务进行了增强,将其从

络通信级上移到了应用级，因此现在安全级服务的定义变为，当且仅当满足如下条件时，配置 C 中的服务 s 已按安全级顺序交付消息 m：(1) $s.deliver(m)$ 满足一致级顺序且已执行. (2) 对于所有 $r \in \{r \mid r \in C$ 且 $r \neq s\}$，s 知道 $r.deliver(m)$ 已发生. 显然，这一定义更符合要求提供安全级服务的客户需求.

安全级服务的读请求处理过程和一致级服务相同，而其写请求过程大部分与一致级服务类似，区别是在该请求之后的其它服务请求（即全局序号 > 安全写请求序号的所有请求）必须等到安全写请求已交付后才能执行. 当令牌轮转一圈，回到生成安全请求消息 m 的全局序号的复制实例那里时，若令牌显示环中的其它实例都已交付 m，该实例才可回送 RSRQ 缓存的应答给 SA 并接着执行 m 之后的其它请求.

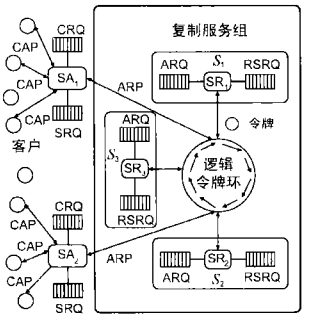

图 1　ROWC 的系统模型

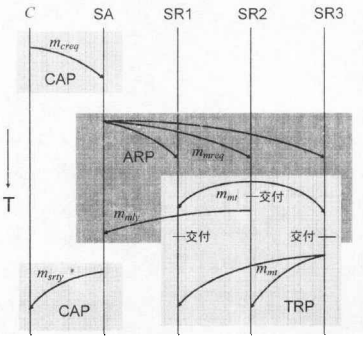

图 2　一致级服务的写请求消息正常处理时序图

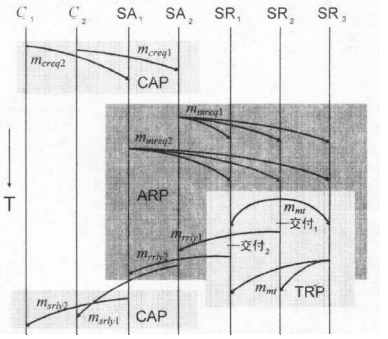

图 3　一致级服务的读请求消息正常处理时序图

3　性能分析与测试

3.1　性能分析

以下假设系统中有 n 台主机和多种复制服务. 为简化分析过程，不访假设每种复制服务均有 n 个复制实例，分别运行在 n 台主机中，服务请求均属于一致级服务，可分为读/写两种类型，写请求类型的比率为 w，每种协议用于处理读请求的实例数为 rq，处理写请求的实例数为 wq，每台主机每秒可处理 L 个请求.

在可用性方面，ROWC 与 Active Replication，Primary Backup 和 ROWAA 相同，只要 n 台主机还有一台可用，则客户仍然能访问复制服务，即可用性 $av = 1 - (1 - p)^n$. 文献[8]说明，在实例数目 n 相同的前提下，ROWAA 能提供比 Majority[4] 和 Grid[5] 更高的可用性，因此 ROWC 也比这些复制协议具有更好的可用性.

实际应用于分布系统中的服务主机的可靠性通常很高，因此在分析性能时可忽略故障的影响，则服务请求在每个实例中的平均执行时间 $\bar{t} = \dfrac{rq(1-w) + wq * w}{nL}$. 为保证公平性，对每种协议取其最优情况时的 rq 值和 wq 值[1~5]，可计算出它们的平均执行时间（表 1）.

表 1　六种协议的平均执行时间

复制协议	rq	wq	平均执行时间
Active Replication	n	n	$1/L$
Primary Backup	n	n	$1/L$
ROWAA	1	n	$(1-w+wn)/(nL)$
Majority	$(n+1)/2$	$(n+1)/2$	$(n+1)/(2nL)$
Grid	$\sqrt{n}$	$2\sqrt{n}-1$	$(\sqrt{n}+w\sqrt{n}-w)/(nL)$
ROWC($1 \leqslant x \leqslant n$)	1	x	$(1-w+wx)/(nL)$

由表 1 可见，当读/写密度发生变化时，前五种传统复制协议的执行时间变化趋势不同. 因此，ROWC 在协议设计时考虑到了对读/写密度不同的分布应用的适应性，并权衡了不同方案的代价. 当请求频度为读密集型时，ROWC 处理服务请求采用与 ROWAA 类似的方式，对于占多数的读请求只选择一个实例执行，因此保证了系统性能；而当请求频度为写密集型时，ROWC 在满足分布系统要求的复制服务最低可用性的前提下，采用重构令牌环，动态减少对占多数的写请求进行处理的复制实例数 x 的方式，加快请求的响应，其代价是此时复制服务的可用性比 ROWAA 有所下降.

3.2　性能测试

我们基于先前研究[7]的原型实现了 ROWC 协议，并将它和 Active Replication，Primary Backup，ROWAA，Majority，Grid 进行了性能对比测试. 测试环境为 4 台 Sun Ultra-5 工作站，通过 1 个 100M 以太网交换机连接，每台工作站中运行一个实现为 Java 对象的复制实例，即复制实例数为 4. ARP 协议和 TRP 协议采用支持多播的 UDP 实现. 图 4 和图 5 分别给出在 w 为 0.2 和 w 为 0.8，且系统中无失效实例时，客户端采用改进的 RMI 访问服务端的平均请求响应时间. 该图反映了在不同的压力下，不同复制协议的响应时间变化.

由图中可见，与其它复制协议相比，ROWC 协议的响应时间较低，而且在重载下的差别尤其明显. 这是由于 TRP 协议采用了逻辑令牌环结构，从而在轻载时能充分利用系统资源，在重载时则能通过流控技术防止复制服务过载，减缓响应时

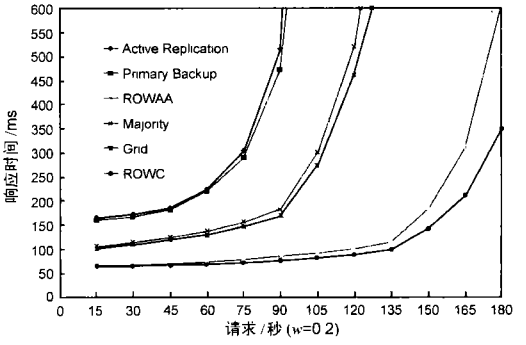

图 4　不同复制协议的响应时间曲线（$w = 0.2$）

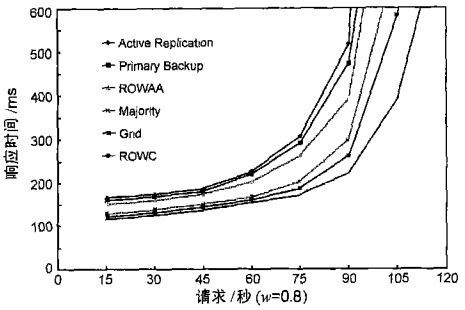

图 5 不同复制协议的响应时间曲线($w=0.8$)

间急剧上升的趋势.对于读密集型应用,由于 *ROWC* 只须在一个复制实例中执行读请求,从而具有与 *ROWAA* 相似的响应速度,而明显比其他协议优越.对于写密集型应用,由于 *ROWC* 在写请求频度上升时相应地减小处理写请求的复制实例数,因此仍然比其他协议的响应更快.

4 相关工作及其比较

各种传统复制协议[1~5]的研究,往往侧重于如何提高系统的可用性,而忽视了引入复制技术后对系统性能的影响,也缺乏对于客户和复制服务之间如何通信的考虑.而很少的关于客户与复制服务之间通信协议的研究[9],往往不是要求改造现有客户端,以便采用智能存根支持复制技术,就是要求客户和服务器之间可以支持多播通信.由于这些要求均违反了适应性需求(例如,由于安全限制,防火墙外的客户往往无法通过多播方式与防火墙内的服务通信),因而限制了这类研究在遗留系统中的应用.

ROWC 中 TRP 协议采用的令牌环结构与 Totem[6]类似.两者之间关键的区别之一是 ROWC 将服务请求按读/写频度的差别进行分类,并采取不同的处理策略,从而在不同类型的应用中仍能保持较好的性能.与 Totem 的另一区别是,ROWC 中的一致级服务请求可由令牌持有者排序后立即交付并返回应答,而无须等待令牌绕环半周,从而提高了响应速度;对于安全级服务,ROWC 同样只需要等待令牌绕环一周即可返回应答给客户,而无须等待令牌绕环两周.此外,Totem 假设客户和复制服务实例属于同一个多播组,因而也缺乏适应性.

RAWA[10]的设计思想在动态伸缩处理请求的复制实例数方面与 ROWC 类似,但它同样未考虑不同读/写频度的服务请求对性能造成的不同影响.

5 结束语

针对现有复制协议很难适应遗留系统的适应性缺陷和导致响应速度不够理想的性能缺陷,本文提出了 ROWC 协议.通过引入 SA,ROWC 的设计可满足适应性需求.而性能分析与测试结果表明,复制实例数目动态伸缩的逻辑令牌环结构可对不同读/写频度的服务请求进行分类处理,从而使 ROWC 提供比其它传统复制协议更好的性能.

目前,ROWC 协议设计为单令牌环结构,适用的环境为局域网.下一步,我们一方面考虑引入多令牌环结构以扩展 ROWC,支持广域网中服务分散的大型分布系统,在保证高性能的前提下支持高可用性;另一方面,我们考虑将 ROWC 与移动计算相结合,进一步提高 ROWC 的适应性,以适应应用范围更广的分布系统.

参考文献:

[1] F B Schneider. Replication Management Using the State-Machine Approach[M]. New York: ACM Press/Addison-Wesley Publishing Corporation. By Sape Mullender, editor, Distributed systems, Second Edition, ACM Press Books, Chapter 7, 1993. 169 - 198.

[2] N Budhiraja, K Marzullo, F B Schneider, S Toueg. The Primary-Backup Approach[M]. New York: ACM Press/Addison-Wesley Publishing Corporation. By Sape Mullender, editor, Distributed Systems, Second Edition, ACM Press Books, Chapter 8, 1993. 199 - 216.

[3] P A Bernstein, V Hadzilacos, N Goodman. Concurrency Control and Recovery in Database Systems[M]. Boston, MA: Addison Wesley Longman Publishing Co, Inc, Chapter 8, 1987. 265 - 294.

[4] R H Thomos. A majority consensus approach to concurrency control for multiple copy databases[J]. ACM Transactions on Database Systems, 1979, 4(9): 180 - 209.

[5] S Y Cheung, M Ahamad, M H Ammar. The grid protocol: a high performance scheme for maintaining replicated Data[J]. IEEE Transactions on Knowledge and Data Engineering, 1992, 4(6): 582 - 592.

[6] Y Amir, L E Moser, P M Melliar-Smith, D A Agarwal, P Ciarfella. Fast message ordering and membership using a logical token-passing ring [A]. Proceedings of the 13th International Conference on Distributed Computing Systems [C]. Pittsburgh: PA, 1993. 551 - 560.

[7] D Zhao, S Yao, M Zhou. Research and design of a middleware for supporting wide-area distributed applications[DB/OL]. http://www.computer.org/proceedings/ipdps/1573/workshops/15730217abs.htm. 2002: 217.

[8] R Jimenez-Peris, M Patino-Martinez, G Alonso, B Kemme. How to select a replication protocol according to scalability, availability and communication overhead[DB/OL]. http://citeseer.nj.nec.com/cache/papers/cs/22836/http:zSzzSzwww.inf.ethz.chzSzpersonalzSzalonsozSzPAPERSzSzSRDS01-Proceedings.pdf/jimenez-peris01how.pdf. 2001: 24 - 35.

[9] C Karamanolis, J Magee. Client-access protocols for replicated services [J]. IEEE Transactions on Software Engineering, 1999, 25(1): 3 - 21.

[10] 钱方,贾焰,等.提高冗余服务性能的动态容错算法[J].软件学报,2001,12(6):928 - 935.

作者简介:

赵 东 男,1972 年 4 月生于四川省成都市,1993 年毕业于电子科技大学计算机系获计算机及应用专业学士学位.1996 年毕业于电子科技大学计算机学院获计算机软件专业硕士学位,1998 年入电子科技大学计算机学院攻读计算机应用专业博士学位,主要研究方向为分布对象技术和信息安全技术.

姚绍文 男,博士,1966 年 8 月生,湖南省永顺县人,云南大学信息学院副教授、网络智能计算研究室主任、中国计算机学会西南网络与 MIS 专委会副主任委员,1998 年至 1999 年为澳大利亚南澳大利亚大学电信研究所访问学者,主要研究方向为语义 Web 技术、网络协议工程、网络分布式计算、着色 Petri 网(CPN)建模、知识工程技术等.近年来,先后在国内外发表学术论文 30 余篇.