基于心跳技术的3阶段提交协议

赵高峰　胡运发
(复旦大学计算机信息与技术系,上海 200433)
E-mail:gaofengz@sina.com

摘　要　在数据库技术的发展过程中,事务机制一直是个热点和关键技术,在分布式数据库尤其如此。该文概述了分布式数据库中事务提交的主要算法,并提出了基于"心跳技术"的另一种三阶段提交协议。采用该技术使得对于大多数站点来说,事务的提交机制更为简单灵活有效。

关键词　心跳技术　分布式数据库　三阶段提交协议(3PC)

文章编号 1002-8331-(2004)11-0177-03　**文献标识码** A　**中图分类号** TP311.13

3-Phase Commitment Protocol Based on Heart-Beat Technology

Zhao Gaofeng　Hu Yunfa
(Department of Computer Information and Technology, Fudan University, Shanghai 200433)

Abstract: In the development of database technology, transaction mechanism is a Hotpoint and a key technology, especially in Distributed Database Management System(DDBMS). In this article, the main algorithm about the transaction commitment DDBMS is stated. And another 3-Phase Commit protocol, which is based on "Heart-Beat" technology, is given. This protocol, using redundant technology, makes the mechanism of transaction commitment in most sites of the system simpler.

Keywords: Heart-Beat Technology, Distributed database, 3-Phase Commitment(3PC)

1　引言

事务处理一直都在数据库管理系统中占据重要地位,尤其随着技术发展和需求增长而出现分布式数据库之后,分布式事务处理更是保证数据库各场所的数据之间一致性的关键技术。

分布式事务是传统事务的扩充。它继承了传统事务的定义,具有同样的ACID特性,即:原子性(Atomicity),一致性(Consistency),隔离性(Isolation),持久性(Durability)。但是,由于分布式数据库系统的分布特性,分布式事务更带有分布执行时的新特性。例如,在分布式数据库系统中,为了保证事务的原子性,组成这个分布式事务的各个子事务,要么全都提交(成功结束),要么全都撤销(不成功结束)。这就需要对各子事务进行协调和控制。因此分布式事务的提交机制更为复杂。

传统的分布式数据库环境下的提交协议主要有:2阶段提交(2PC),3阶段提交(3PC)。该文提出了一种基于心跳技术的3阶段事务提交协议。并对比了该协议和传统的3PC之间的差别。

该文内容组织如下:第2部分概述了已经提出的分布式事务提交协议,并指出了各种协议的优缺点;在第3部分中,提出了一种新的3阶段提交协议,给出了该协议的算法描述,以及该协议与其他协议的对比;第4部分是总结。

2　分布式事务提交协议

在分布式事务机制中,对提交协议的主要需求是因为它们保持了分布式事务的原子性。这意味着即使分布式事务的执行包含许多站点,其中一些也可能在执行过程中失败,该事务对分布式数据库的影响仍然是要么全部或要么都没有,即"原子提交"。另外,如果一个协议允许事务在非失效站点终结而不必等待失效站点的恢复,它就被称为非阻断的。笔者希望协议是非阻断的,这将极大地改进事务的响应时间。

对于分布式事务,主要存在两种协议:2阶段提交协议,3阶段提交协议。

2.1　2阶段提交协议(2-Phase Commitment,2PC)

在2PC中,把分布式事务的某一个代理指定为协调者,所有其他代理称为参与者。只有协调者才有掌握提交或者撤销事务的决定权,而其他参与者各自负责在本地数据库中执行写操作,并向协调者提出撤销或提交子事务的意向。

2PC协议分为两个阶段。

在第一阶段进行投票,从而形成一个共同的决定。协调者给所有参与者发送"准备提交"消息,并进入等待状态。当参与者收到"准备提交"消息后,它检查是否能提交本地事务,并发送相应的投票给协调者。协调者收到所有参与者的投票后,做出是否提交事务的决定。只要有一个参与者投了"终止"票,或者在规定的时间未对协调者做出响应,则协调者发出"全局撤销"消息给所有参与者;否则发送"全局提交"。

第二阶段是执行阶段。根据协调者的指令,参与者或者提交事务,或者撤销事务,并给协调者发送确认消息。此时,协调者在日志中写入一条事务结束记录并中止事务。

2PC既简单又精巧,它把本地原子性提交行为的效果扩展到分布式事务,保证了分布式事务提交的原子性,可以实现快速故障恢复,提高了分布式数据库系统的可靠性。但是,2PC存

作者简介:赵高峰,男,在读硕士研究生,研究方向:分布式数据库技术,数据挖掘等。胡运发,男,教授,博士生导师,研究方向:数据库技术,知识工程,人工智能等。

在一个站点等待其他站点信息的可能，也就是说可能引发阻塞。

2.2 3PC 协议

3PC 协议包括：参与者投票表决阶段、预提交阶段、协调者决策阶段。其基本思想为：在 2PC 的参与者投票和协调者决策之间增加了“预提交”阶段。协调者在接受到所有的参与者的提交票后发送一个全局预提交命令，当参与者接收到全局预提交命令之后，它就得知其他的参与者都投了提交票，从而确定它自己在稍后肯定会执行提交操作，除非它失败了。每个参与者都对全局预提交发确认，协调者一旦接收到了所有参与者的确认，它再发出“全局性提交”。

3PC 协议在站点失败，甚至是所有的站点都失败的情况下也不会带来阻塞。

2.3 3PC 终止协议

在 3PC 进行过程中，参与者有两个等待阶段：一是第二阶段等待协调者发来的“全局预提交”或者“全局终止”命令；二是在第三阶段等待“全局提交”。无论哪种情况，如果参与者已经投了“提交”，他就必须与其他进程联系而不能单独行动。如果超时，则它断定协调者已经失败了，执行终止协议且唤起选举协议选举一个新的协调者。

新的协调者首先确定每个参与者在旧的协调者失败时所处的状态。他向每个参与者发送一个“询问状态”命令，之后检查参与者回送的信息，做出一个与所有参与者已经做出的决定一致的全局决策。

3 基于心跳技术的 3PC 协议

基于心跳技术的 3PC 协议(Heart Beat Based 3-Phase Commitment，HBB-3PC)协议基于以下观察和启发。

观察 1：2PC 可能产生阻塞的原因是，参与者在投了“提交”票后等待协调者决策这个时段，其状态是不确定的。3PC 消除了这个阶段。

启发 1：新的协议应该采用传统 3PC 的做法来消除阻塞。

观察 2：在 3PC 中，参与者不仅需要关注和协调者之间通讯以及本地事务的执行，也要负责在协调者超时的情况下，发起新的协调者选举。而在分布式事务提交机制中参与者占主要成分。

启发 2：如果可以使得占多数节点的参与者不关心协调者是否存活，也就是说协调者的死亡与新协调者的产生对于参与者是透明的，那么整个系统的体系结构将更为清晰，效率也更高。

观察 3：在分布式系统中经常采用的冗余技术，对于比较关键的节点，采取冗余的方法，在该节点死亡后，后备节点接管工作。

启发 3：协调者在 3PC 中地位非常特殊而且重要，通过为其增加冗余来实现新的 3PC 协议。然而，这里引入了心跳技术，并不完全等同于传统意义上的冗余。

3.1 HBB-3PC 协议描述

HBB-3PC 以传统 3PC 协议为基础，同时增加了冗余的协调者，但是对于参与者来说，冗余的协调者并不可见。该协议描述如下。

正常情况下：当 HBB-3PC 开始时，存在一个主协调者，该协调者负责协调参与者，其功能和传统的 3PC 协议中的协调者相同；同时，系统中存在一个从协调者，该协调者作为主协调者的后备，监听发送给主协调者的信息(因此从协调者也可以叫作监听协调者)，并执行与主协调者同样的动作，但是，该协调者并不向参与者发送各种命令，亦即，从协调者除了不发命令到参与者，其他动作同主协调者相同。因此，在任意时刻，从协调者和主协调者的状态完全一致。对于参与者来说，它们并不能感知主、从协调者，对它们来说，和传统的 3PC 一样，只知道有一个协调者存在，但是他们永远相信协调者不会死亡。

从协调者通过“心跳(Heart Beat)”来感知主协调者是否存活。所谓“心跳”技术，是指主协调者每间隔一定的时间发送消息给此协调者，这个消息称为心跳。两个心跳消息之间的间隔称为心跳间隔，其应该远远小于传统 3PC 协议中的超时。从协调者从主协调者处得到心跳，从而知道主协调者的存活。如果从协调者间隔一定时间(比如 3 个心跳间隔)没有收到主协调者的心跳，则认为主协调者失败。

当从协调者发现主协调者失败时，提升自己为主协调者，并接管已经失败的主协调者的工作，向参与者发送命令。该接管过程是无缝的。

对于参与者来说，它并不需要关心当前的协调者是原来的主协调者，还是主协调者失败后由从协调者提升而来的。也就是说，主协调者的更替对其来说是透明的，它可以把所有的协调者看成一个整体，参与者只需永远信任协调者不会失败。

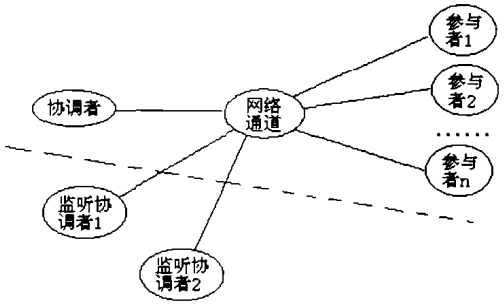

图 1

图 1 说明了引入从协调者后，3PC 协议实现时的系统结构。其中的虚线表示监听协调者对于参与者来说是透明的。在图中，使用了两个监听协调者，其中监听协调者 1 作为主协调者的冗余，监听协调者 2 作为{主协调者和监听协调者 1}簇的冗余；当监听协调者 2 发现主协调者和监听协调者 1 都死亡时，提升自己为主协调者。

3.2 算法描述

```
Step 1 投票
  在日志中写入“开始事务”
  如果本协调者当前为主协调者，则发送“投票”指令给所有参与者
  等待；如果超时，转 Step 2B
Step 2A 预提交
  如果收到所有参与者的“提交”回应
  { 写“预提交”到日志
    如果本协调者当前为主协调者，则发送“预提交”命令给所有参与者}
Step 2B 全局取消
  如果收到至少一个参与者的“放弃”回应，或者协调者超时
  { 写“全局放弃”到日志中
    如果本协调者当前为主协调者，则发送“全局放弃”命令给参与者
  转 Step4}
Step 3 全局提交
  等待，直到收到所有参与者的“预提交确认”回应
  写“全局提交”到日志中
  如果本协调者当前为主协调者，则发送“全局提交”到参与者
Step 4 终结
  等待，直到收到所有参与者的确认
  写“结束事务”到日志中
```

图 2

HBB-3PC 的算法实现是基于传统 3PC 的。和传统 3PC 相比，HBB-3PC 无需“终结协议”。

HBB-3PC 算法的协调者算法描述如图 2，参与者算法描述如图 3。

```
Step 0 等待投票指令
Step 1 投票
  如果参与者准备提交，则发送“提交”回应给协调者
  否则发送“放弃”回应
  等待协调者的指令
Step 2A 预提交
  如果收到“预提交”指令，则转 Step3 等待全局提交
Step 2B 取消
  执行本地事务放弃处理
Step 3 提交
  等待，直到收到“全局提交”命令
  执行本地提交处理
Step 4 终结
  给协调者发送确认
```

图 3

3.3 与 3PC 的比较

与基本的 3PC 协议相比，HBB-3PC 的算法具有如下优点。

(1)简单高效。从 3.2 的算法描述可以看出，HBB-3PC 的实现相对来说简单许多，而且 HBB-3PC 无需参与者选举新协调者，新主协调者产生后也无需和参与者进行额外的通信，就避免了额外的通信负担和选举延迟，提高了效率。

(2)透明性。多个监听协调者的存在对于参与者是完全透明的，参与者无需关心主协调者的死亡，他们甚至根本不知道有多个监听协调者的存在。也就是说，通过协调者的冗余，使得在参与者看来，协调者簇(包括主、从协调者)作为一个整体是不会失败的，他们只需要永远相信协调者即可。

(3)可扩充性。为了提高协调者整体的可靠性，可以扩充冗余协调者的数量。给每个协调者分配一个优先级，每个协调者的优先级不同。在某一时刻，优先级最高的协调者作为主协调者，其他作为从协调者。每个协调者获取比他优先级高的协调者的心跳信息，如果所有优先级比自己高的协调者都失败，则该协调者充当主协调者。

(4)HBB-3PC 中采用的冗余不同于传统意义上的冗余。监听协调者提升自己为主协调者时的工作接管是无缝的，不会导致状态的不一致。

当然，HBB-3PC 也存在着不足。比如，即使有协调者冗余，也不能从理论上 100%地保证协调者簇不会失败，但是它可以在很多场合达到应用中要求的可靠性。

4 结论

相对于传统 3PC，引入心跳和冗余技术的 HBB-3PC 建立在新的系统架构之上，带来了相当大的优越性，使得系统的设计更加模块化，具有更强的可扩充性和可维护性。

(收稿日期：2003 年 6 月)

参考文献

1.邵佩英.分布式数据库系统及其应用[M].北京：科学出版社，2000
2.廖国琼，李陶深.分布式工程数据库系统中事务提交机制的研究[J].计算机辅助设计与图形学学报，2001；13(4)
3.(美)Jie Wu 著.分布式系统设计[M].机械工业出版社，2000
4.Elmasri Ramez，Navathe Shamkant B 著.Fundamentals of Database Systems[M].2nd Ed，Benjamin，1994

(上接 164 页)

代理与普通代理一样，都是遵从代理服务用户管理系统的，没有授权的用户是不能访问的。

(5)Web 服务器

系统提供简单的 Web 服务器功能，虽然还不能支持 Script 和 Web 应用程序，但已实现了最基本的 Web 服务。因为该系统的内核结构是兼容 Web 服务的，所以提供这一功能并不复杂，有需要的管理员可以启用这一功能。

(6)Web 远程管理

系统在基于 Web 服务器的基础上构建了 Web 方式的远程管理系统。通过浏览器就可以轻松实现在异地管理代理服务器。

4 结束语

该文分析了代理服务器工作的基本原理，并对其实现做了较深入的研究。实际开发了一个代理服务器系统，从代理服务器的实现过程来看，Proxy 是 Browser 访问 Server 的中介，它可以监控 Browser 发出的所有的 HTTP 数据包，对这些数据包进行阻塞、转发和流量记录。同时，它也可以监控 Server 发出的所有的 HTTP 数据包，对这些数据包进行阻塞、转发和流量记录。对代理服务器进行简单扩展，就可以完成许多其它功能，如对数据包进行过滤，就构成了一个 Firewall；如果对流量进行记录，就构成了一个计费系统。该代理服务器用 Borland Delphi7 在 Microsoft Windows XP Professional (Build 2600.1 English Version)With SP1 环境下开发，在实验室进行一系列功能测试后(测试服务器：Google 搜索引擎 http://www.google.com；中国程序员开发网 http://www.csdn.net；北京大学图书馆 ftp://ftp.lib.pku.edu.cn；本地 LAN 服务器：IIS Web/FTP/Serv-U；Borland FTP：ftp://ftp.borland.com；ASUS 华硕德国站点：ftp://ftp.asuscom.de 等)，现已在某企业网上正式运行。运行结果显示，系统能稳定地代理各类 Internet 访问服务；企业利用系统基于用户的管理机制，加强了访问管理和控制。可以说系统很好地适应了企业的需求。(收稿日期：2003 年 6 月)

参考文献

1.Jones A，OhlundJ.Network programming for microsoft windows[M].Washington：Microsoft Press，2000：89~171
2.曾明，李建军.Internet 访问管理与代理服务器[M].北京：人民邮电出版社，1999-12：3~6
3.Anthony Northrup.NT Network Plumbing：Routers，Proxies，and Web Services.IDG Books Worldwide，Inc，1998：293~303
4.ICS 组件的 FAQ 和相关 Demo.http://www.overbyte.be
5.WolfeD.郭文健译.MicrosoftProxyServer 开发指南[M].北京：电子工业出版社，1998：23~180
6.张宝社.Windows 下的网络编程[M].合肥：中国科学技术大学出版社，1997
7.潘志松.网络防火墙中的代理技术[N].计算机世界报，1999-3-8-10