

## 一种改进的事务提交机制\*

## An Improved Transaction Committing Mechanism

李卫玲,李陶深

Li Weiling, Li Taoshen

(广西大学计算机与电子信息工程学院,广西南宁 530004)

(School of Comp., Elec. &amp; Info., Guangxi Univ., Nanning, Guangxi, 530004, China)

**摘要:**结合文献[3,4]以及已有的一阶段提交协议,提出一种改进的Early no Prepare(ENP)协议。该协议消除了两阶段提交协议(2PC)中的处理消息阶段,参与者无需向协调者发送提交消息,子事务可自行决定提交或是终止,不必等待协调者的最终命令,从而降低消息复杂度和日志复杂度,保证了状态的一致性。

**关键词:**事务 提交协议 局部事务 全局事务

中图分类号:TP311.13 文献标识码:A 文章编号:1002-7378(2005)04-0239-02

**Abstract:** Referred the literatures[3,4] and one-phase commit protocol, an improved commit protocol, called Early no Prepare (ENP), is presented. The protocol eliminates the voting phase of 2PC. In this protocol, the participants don't need to send back "commit (ready)" message, the sub-transaction can decide by itself to commit or abort, and it is unnecessary to wait for coordinator's decision. In this way, message and log complexities are reduced. At the same time, it promises the consistency.

**Key words:** transaction, commit protocol, local transaction, global transaction

事务是访问并可能更新各种数据项的程序执行单元<sup>[1]</sup>。事务分为全局事务和局部事务,需访问和更新多个局部数据库中数据的事务称为全局事务,仅访问和更新一个局部数据库中数据的事务称为局部事务<sup>[1]</sup>。事务的基本特性有:原子性、一致性、隔离性和持久性<sup>[2]</sup>。

在分布式数据库系统中,原子提交协议是用来保证系统的一致性状态的。传统的提交协议是两阶段提交协议(2PC)和三阶段提交协议(3PC)<sup>[2]</sup>。3PC协议虽说在一定程度上避免了阻塞,但由于其需要较多的通信,实现起来比较复杂,所以很少被采用。目前应用较多的是2PC的各种变体,如O2PC, Presumed Abort (PrA), Presumed Commit (PrC), 及Early Prepare协议等。

2PC分为投票阶段和决策阶段<sup>[2]</sup>。2PC协议在执行过程中,若协调者在发出请求处理消息后失效,

此时若参与者未做出回答,则将发生阻塞情况。文献[3]提出一种先提交协议,它允许局部事务在全局决定作出之前提交,局部事务独立于全局事务管理提交或撤消。该协议的优点在于有较高的并发性,因为子事务提交后可释放持有的锁;不足在于当局部的子事务终止时,也需要终止全局事务,因此所有已提交的站点都要进行取消操作,取消日志和并发控制机制由全局系统实现,影响了局部事务的自治性。文献[4]提出一种改进的提交协议,省去了处理消息阶段,子事务完成时,能立即返回提交消息,等待最终决定(提交/终止),且在有某站点事务终止时,可向协调者发送通知,协调者再向所有站点子事务发送终止命令,整个事务终止。这种协议有较高的消息复杂度。文献[5]提出一阶段提交协议(1PC)。该协议假定协议发起之前事务操作都已成功执行完毕。1PC的典型协议有IYV(Implicit Yes-Vote)<sup>[5]</sup>和CL(Coordination Log)两种。1PC比2PC获得了较好的性能,但是1PC引入了太多的假定条件。本文结合文献[3,4]及已有的一阶段提交协议,提出一种改进的Early no Prepare(ENP)协议。

收稿日期:2005-06-24

修回日期:2005-08-10

作者简介:李卫玲(1980-),女,河南周口人,硕士研究生,主要从事分布式工程数据库系统研究。

\* 广西自然科学基金(桂科自9912009)项目资助。

## 1 ENP 协议的思路

取消处理阶段,参与者无需向协调者发送提交消息,而是可以自行决定提交,不必等待协调者的提交命令;若参与者操作失败,则在向协调者发送终止消息后可自行终止,不必等待协调者发来的终止命令。协调者在收到某一参与者发来的终止消息后,向各参与者站点发送终止命令。对于子事务已提交的站点,需要根据取消日志记录(取消日志存储逆事务)进行撤消操作。

## 2 ENP 协议

### 2.1 ENP 协议的执行过程

(1)参与者完成后自行决定提交/终止,若是提交状态,则释放所占有资源,并写逆操作日志,不必向协调者发送任何消息;若决定终止,则向协调者发送终止消息,不写日志记录。

(2)协调者若没有收到任何消息,则可认为全局事务已完成提交,写提交日志。

(3)协调者若收到终止消息,则向各站点发送终止命令,并写终止日志记录。

(4)对于未完成的参与者站点,收到终止命令后,决定终止,不必再继续执行。

(5)对于已提交的站点,根据取消日志记录进行撤消操作。

### 2.2 ENP 协议的性能分析

假设有  $n + 1$  个事务,一个为协调者,其余  $n$  个为参与者,在不考虑故障的情况下,ENP 协议与其它 4 种协议进行性能比较的结果见表 1 和表 2。

表 1 事务提交时的性能比较

协议类型	消息复杂度 <sup>[6]</sup>	日志复杂度 <sup>[6]</sup>
2PC	$4n$	$2n + 1$
IYV	$2n$	$n + 1$
先提交协议	$2n$	$n + 1$
dEDBMS 的加工协议	$3n$	$2n + 1$
ENP	0	$n + 1$

表 2 事务夭折时的性能比较

协议类型	消息复杂度 <sup>[6]</sup>	日志复杂度 <sup>[6]</sup>
2PC	$4n$	$2n + 1$
IYV	$n$	0
先提交协议	$3n$	$n + 1$
dEDBMS 的加工协议	$3n$	$2n + 1$
ENP	$n + m$	$n - m + 1$

表 1 显示,在事务提交时,ENP 协议的日志复杂度为  $n + 1$ ;消息复杂度为 0,比其他 4 种协议的消息复杂度都要小得多。表 2 显示,ENP 协议的消息复杂度为  $n + m (1 \leq m \leq n)$ ,仅次于 IYV 协议的  $n$ ;日志复杂度为  $n - m + 1 (1 \leq m \leq n)$ ,当  $m = n$  时日志复杂度为 1,略大于 IYV 协议而优于其他 3 种协议,当  $m = 1$  时(即 1 个站点事务终止其余全提交的情况),日志复杂度为  $n$ ,也是略大于 IYV 协议而好于其他 3 种协议。考虑到一阶段协议的假设条件过于理想化(协调者确信所有参与者都不会投否定票),因此,可以认为 ENP 协议相对比较优越。

在网络环境中,ENP 协议能较好地保证站点的自治性,并且不影响全局的一致性。

**定理 1** 在 ENP 协议中,所有的参与者要么全部提交,要么最后不产生任何影响,保证了状态的一致性。

**证明** 若协调者没有收到任何消息,则由 ENP 协议(2)可知所有站点都已自行提交,相当于全局事务已经完成,已到达一个新的状态。

若协调者收到终止消息,则向各站点发终止命令,由 ENP 协议(4)可知,未完成的参与者自行终止;而对于已提交的站点,通过进行取消操作,也回到最初未执行前的开始状态。因此,最后的结果是回到了事务未执行时的状态,未发生任何改变。

由上述证明可知,ENP 协议保证了所有的参与者站点要么全部提交,最后到达一个新的状态,要么通过终止和撤消,仍处于事务执行前的状态,这保证了状态的一致性。

## 3 结束语

ENP 协议保证了站点的自治性,同时由于消除了 2PC 中的处理消息阶段,不再发送提交消息,降低了消息复杂度和日志复杂度,并且不影响全局的一致性。本文从理论上证明了 ENP 协议的正确性。ENP 协议的这些特点,将使它在对自治性要求很高的网格环境中有良好的应用前景。

参考文献:

- [1] Araham Silberschatz, Henry Fkorth, S Sudarshan. 数据库系统概念[M]. 杨冬青,唐世渭译. 北京:机械工业出版社,2003.
- [2] 贾 焰,王志英,韩伟红,等. 分布式数据库技术[M]. 北京:国防工业出版社,2000.

head 节点所对应的状态(head, treeNode)出发的状态迁移以及迁移至的目标状态,分别填充 head.t 和 head.childTree,即把迁移至的状态作为节点 head 的子节点;同时置好各个子节点的属性值,其中 nodeLevel 为 head.nodeLevel+1,从 root 节点开始层次遍历测试树(从第 0 层至 head.nodeLevel 层),如果在 head 的子节点中存在某个节点 s,其所对应的状态已在第 0 层至 head.nodeLevel 层中出现过,则该节点 s 不再扩展,即为叶子节点。把其它没有出现过的子节点加入到队列尾部。

步骤 4:head 指向队列中的下一个节点,重复第二步,直至队列为空。

在步骤 3 中,如果某个迁移对应的目标状态已经在测试树中出现过,就不再考虑这个状态,所有并不加入到队列尾部。这样就可以有效地避免了重复构造节点,同时又不降低测试的覆盖率。通过上述步骤就可以构造出 UML 状态图对应的测试树。

测试树构造算法能很好地支持多个并发的子状态的情况,只是节点标识为并发子状态的合集;如果某个事件触发其它事件而引起一系列的状态迁移时,只要把最终的状态作为节点加入到测试树中。另外,本算法同样适用于除 UML 状态图之外的其它状态图,如扁平状态机或 D. Kung 的 COSD。

通过本算法构造出的 CoinBox 类的测试树如图 3 所示。图 3 测试树中的非黑体符号节点表示该节

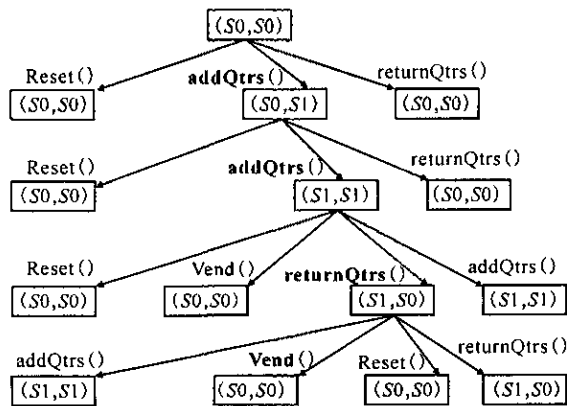


图 3 类 CoinBox 的测试树

点代表的状态已经在之前出现过,所以不再扩展从而成为叶子节点。通过测试树可以很容易的构造出测试用例。从根节点开始沿着各个分支往下直到叶子节点,每条这样从根节点开始到某个叶子节点结束的路径上的事件按顺序组合在一起,就成为基于对象状态测试的一个测试用例。那个造成错误的事件序列很容易地被检测出来,如图 3 中黑体事件就是被检测出来的错误事件。

### 3 结束语

UML 的状态图支持嵌套和并发,把状态的复杂度控制在和状态属性相关的线性级别;其次 UML 状态参数图是在面向对象软件开发生命周期中的早期设计阶段确定的,是对对象状态的完整的描述,并不依赖于源代码,既保证了状态描述的完整性,又可以在开发早期进行测试,尽早发现与状态相关的错误,避免将错误带入到后面的开发阶段。因此可以用 UML 的状态图来产生有效的测试用例,这大大提高了测试的灵活性和有效性,是一种实际可行的方案。

参考文献:

- [1] 夏耘,林华.面向对象测试技术的研究与应用[J]. 计算机应用与软件,2002,(2):17-20.
- [2] John D. McGregor David A, Sykes. 面向对象的软件测试[M]. 杨文宏,李新辉,杨洁,等译.北京:机械工业出版社,中信出版社,2003. 235.
- [3] Binder R V. Testing of Object-oriented System[M]. 北京:人民邮电出版社,2001. 157.
- [4] Dirk Seiferk, Steffen Helke, Thomas Santen. Conformance testing for statecharts[R]. Technical report 03-01; Technical university of berlin, 2003.
- [5] OMG. Unified modeling language specification[Z]. version 1.4. 2001.

(责任编辑:韦廷宗 邓大玉)

(上接第 240 页)

- [3] 卢正鼎,陈光.一种基于两级事务模型的先提交协议[J]. 华中理工大学学报,1999,27(9):101-104.
- [4] 王韬.一种新的分布事务处理模型和算法[J]. 机械工程学院学报,1998,10(4):51-56.
- [5] Abdallah M, Guerraoui R, Pucheral P. One-Phase Commit: Does It Make Sense[A]. Proceeding of the

- International conference on Parallel and Distributed Systems, 1998, 182-192.
- [6] 陈国宁,李陶深,廖国琼.一个带有时限的工程设计事务提交协议[J]. 计算机工程与应用,2004,40(14): 178-180, 195.

(责任编辑:邓大玉 韦廷宗)