

35-38

面向对象数据库中的故障恢复^{*}

Recovery in Object Oriented Database System

王意洁 胡守仁

TP311.13

(国防科技大学计算机系 长沙 410073)

Abstract In this paper, the recovery of the system fault in object oriented database is discussed, a practical and efficient recovery strategy is proposed, which is implemented in the object-oriented database system KDOODB.

Keywords Object oriented database, System fault, Recovery, Page

1 引言

系统故障的恢复是面向对象数据库(OODB)中的关键技术之一,尤其是在支持嵌套事务模型的OODB系统中,系统故障的恢复尤为重要。页服务器已经在OODB的Client/Server体系结构中占主导地位,目前既采用页服务器结构又支持嵌套事务模型的OODB系统大多没有充分考虑系统故障恢复的有效实现。顺应OODB的发展趋势,我们针对页服务器结构和嵌套事务模型的特点,提出了一种基于页的恢复策略WAL-P,它能够全面有效地处理系统故障,并实际应用于自行研制的KDOODB系统中。

2 日志文件

WAL-P的日志文件包括一系列日志记录、事务表和脏页表。

日志记录包括以下几个域:LSN(日志序号)、Type(类型)、Subtype(子类型)、Tid(事务标识)、Pid(页标识)、Last-LSN、RedoNext-LSN、UndoNext-LSN、Length、Data。实际上,每个页也有一个LSN域,它随着页面内容的改变而改变,它与日志记录的LSN是对应的。

事务表是事务管理的基础,在事务执行过程中,事务表被动态更新,在设置检查点时,事务表将被写入日志文件,等到系统重启时用于确定圆满事务和

夭折事务。事务表包括以下几项:Tid、Last-LSN、State。

脏页表管理已经在缓存中被更新但尚未存入磁盘数据库的页(即:脏页)。每当某页被存入磁盘数据库时,就从脏页表中删除与其对应的项。在设置检查点时,脏页表将被写入日志文件,等到系统重启时用于确定恢复的终点。脏页表包括以下两项:Pid和Dirty-LSN。

3 事务的正常执行

3.1 更新操作

更新操作的执行将引起一个日志记录的产生以及页中LSN域的更新。一个事务的日志记录是通过后向指针链接的,后向链接则通过记录的Last-LSN域实现,它的值是根据事务表的Last-LSN域来设置的。

3.2 事务的卷回

事务夭折将引起事务的全部卷回,即:根据日志自后向前卷回所有的更新操作,并且每卷回一个更新操作就产生一个补偿日志记录,直到遇到该事务的Start类型日志记录,产生一个Abort类型日志记录,然后结束卷回过程。

3.3 事务的提交和夭折

我们采用两段锁协议2PL进行并发控制,用以控制事务提交。两段提交协议的基本思想是:在第一阶段,首先,保证有关的日志记录全部写入日志文

^{*} 本文得到九五国防科技预研项目的资助。王意洁 博士生,研究领域为面向对象数据库和并行分布处理技术;胡守仁 博士生导师,研究领域为面向对象数据库、虚拟现实、高性能机体系结构及并行分布处理技术。

件,该事务的更新页是否写入磁盘数据库取决于缓存管理器的具体策略:强制策略和非强制策略。然后,向日志文件写入一个 Prepare 记录,第一阶段到此结束,在第二阶段,向日志文件写入一个 Commit 记录,释放该事务拥有的锁,事务提交到此结束。

根据嵌套事务的行为规则,事务提交的处理过程是:

- 若该事务有父事务,则:①利用文献[1]中提出的事务标识分配策略可以很容易地找到其父事务;②产生一个 Prepare 类型的日志记录,并在事务表中设置它的状态为 pre-commit,更新其 Last-LSN 域为 Prepare 记录的 LSN;③更新父事务日志——将当前事务的日志链接到父事务的日志上,具体而言,就是将它的 Start 记录的 Lst-LSN 域指向父事务的当前最后一个记录,这可通过事务表中父事务的 Last-LSN 域得到;④更新父事务状态——在事务表中,根据当前事务的 Last-LSN 域更新父事务的 Last-LSN 域值;⑤将当前事务拥有的锁转交给父事务。

- 若该事务没有父事务,则:①确保它的日志记录(包括其所有 pre-commit 状态的子事务的日志记录)已全部写入日志文件;②将一个 Prepare 类型的日志记录写入日志文件,并在事务表中设置它的状态为 pre-commit,更新其 Last-LSN 域为 Prepare 记录的 LSN;③将一个 Commit 类型的日志记录写入日志文件,释放它拥有的锁,并从事务表中删除它和它所有 pre-commit 状态的子事务。

事务夭折的处理过程是:1)事务(包括其所有 pre-commit 状态的子事务)全部卷回;2)释放它拥有的锁,并从事务表中删除它和它所有 pre-commit 状态的子事务。

3.4 检查点

KDOODB 系统选择所谓的模糊检查点,设置模糊检查点分两个阶段进行:首先,产生一个 begin-checkpoint 日志记录,并记住它的 LSN;然后,产生一个 end-checkpoint 日志记录,它包括当前最新的脏页表和事务表。当 end-checkpoint 日志记录被写入日志文件时,将 begin-checkpoint 日志记录的 LSN 存入磁盘上的指定位置,它是系统重启的开始点。

4 系统重启

当发生系统故障时,需要进行系统重启以使数据库恢复到一致状态,从而保证事务的原子性和持

久性。基于 WAL-P 的系统重启可以分为三个阶段(如图 1):分析阶段、undo 阶段和 redo 阶段,其目的是撤销夭折事务的更新操作(保证原子性)和重做(redo)圆满事务的更新操作(保证持久性)。许多恢复策略在处理系统故障时都或多或少地进行了不必要的 undo 和 redo 操作,这严重影响了恢复的效率, WAL-P 充分考虑了恢复的效率问题,在系统重启时进行了有选择的必要的 undo 和 redo 操作。

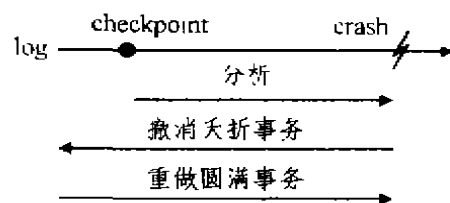


图 1 基于 WAL-P 的系统重启

4.1 分析阶段

分析阶段是为下面两个阶段作准备的,它从磁盘的指定位置取出最后一个检查点的 LSN,从该检查点取出信息来初始化事务表和脏页表;然后,从该检查点开始依次读取每个日志记录并进行处理,直到日志文件的结尾。对日志记录的处理是根据其类型进行的:

- Start:在事务表中产生一个新项,其状态置为 active,其 Last-LSN 域置为 Start 记录的 LSN;
- Update or CLR:更新事务表中相应事务的 LSN 域为该记录的 LSN。如果脏页表中没有与被更新页对应的项,则为其创建一个新项,新项的 Dirty-LSN 域置为该记录的 LSN,标志着这是对它的第一次更新;
- Prepare:在事务表中,更新相应事务的状态为 pre-commit,且 Last-LSN 域更新为该记录的 LSN;
- Savepoint:不作任何处理;
- Commit or Abort:从事务表中删除相应的事务及其所有 pre-commit 状态的子事务。

到达日志文件尾时,分析阶段宣告结束,此时已能确定所有圆满事务和夭折事务。它为下面两个阶段提供的信息有:事务表、脏页表和最后一个写入日志文件的记录 Last-record。

4.2 undo 阶段

undo 阶段的任务主要是:从 Last-record 开始后向前依次读取每个记录,并进行判断和相应的处理:如果该记录对应于夭折事务的更新操作,且更

新结果已存入磁盘数据库,则进行 undo 操作,并产生一个 CLR 记录;如果该记录对应于圆满事务的更新操作,且更新结果未存入磁盘数据库,则将其加入前向链,前向链是通过日志记录的 RedoNext-LSN 域实现的,为 redo 阶段作准备。另外,若是 Start 记录,则从事务表中删除与该事务对应的项。

对系统故障恢复的较高要求是:即使在系统重启时反复发生故障,数据库仍然能恢复到相同的一

致状态。为了达到这个要求,需要恢复策略具有幂等性。恢复策略的幂等性是指多次恢复后的数据库状态与一次恢复后的数据库状态相同。一般来说,基于值日志的恢复是幂等的,而单纯依靠操作日志的恢复是不幂等的。WAL-P 既支持基于值日志的恢复又支持基于操作日志的恢复,所以,借助于日志记录序号和补偿日志记录来实现基于操作日志的恢复的幂等性,从而使 WAL-P 真正具有幂等性。

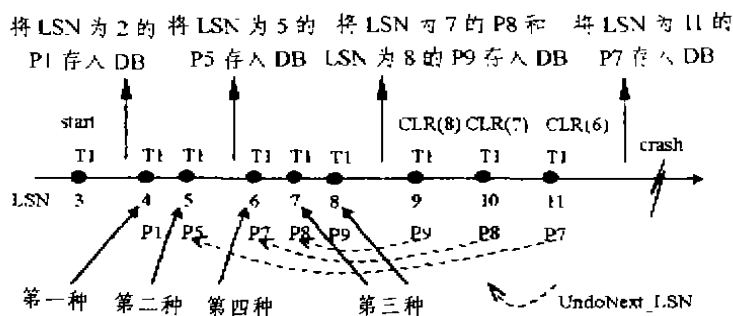


图2 四种更新操作

为了保证 undo 阶段的幂等性,需要严格区分夭折事务的各种更新操作并采取不同的处理方式(如图2):

1)更新操作的结果未存入磁盘数据库——不作任何处理,因为更新操作未对数据库产生影响;

2)更新操作的结果已存入磁盘数据库,且没有相应的补偿日志记录——撤销更新操作,并产生一个相应的补偿日志记录;

3)更新操作的结果已存入磁盘数据库,且有相应的补偿日志记录,但其对应的 undo 操作的结果为存入磁盘数据库(即:更新操作并未真正卷回)——撤销更新操作,并将页的 LSN 域设置为已有的补偿日志记录的 LSN;

4)更新操作及其 undo 操作的结果均已存入磁盘数据库(即:更新操作真正卷回)——不作任何处理,因为更新操作对数据库的影响已被消除。

undo 阶段的工作比较复杂,经常会出现一些异常情况。下面,我们讨论如何处理这些异常情况。

在系统故障发生时就已经进行了或正在进行卷回处理的夭折事务,系统重启时的 undo 阶段和事务正常执行时的卷回处理过程是不一样的。事务正常执行时的卷回处理过程无需考虑与补偿日志记录对应的 undo 操作的结果是否存入磁盘数据库,并可以利用补偿日志记录的 UndoNext-LSN 域跳过已经“撤销”的记录。系统重启时的 undo 阶段不能利用

补偿日志记录的 UndoNext-LSN 域跳过已经“撤销”的记录而是要遍历日志文件,判断更新操作的类型,考虑与补偿日志记录对应的 undo 操作的结果是否存入磁盘数据库。如果对那些在系统故障发生时就已经进行了或正在进行卷回处理的夭折事务处理不当将会引起异常情况。如果恢复是基于操作日志的而且颠倒次序的记录是针对同一页的,那么 undo 阶段的次序改变将导致恢复的失败。

为了避免异常情况的出现,undo 阶段需要满足以下条件:

1)undo 操作必须按照正确的次序进行;

2)不能为补偿日志记录产生它的补偿日志记录,否则在反复发生系统故障的情况下将导致日志记录数目的指数级增长。

为了满足上述条件,WAL-P 在 undo 阶段为每个夭折事务维护两个 LIFO(后进先出)链:已撤销记录链和伪撤销记录链。在 undo 阶段,每当遇到某事务的第一个补偿日志记录时,就为其创建这两个链,并对遇到的每一个补偿日志记录进行如下判定和处理:

△若与补偿日志记录对应的 undo 操作的结果已存入磁盘(即:页的 LSN 大于等于记录的 LSN),则在已撤销记录链中建立一个新项,它包含被该记录补偿的记录的 LSN(可以通过补偿日志记录的 RedoNext-LSN 域得到);

△ 若与补偿日志记录对应的 undo 操作的结果未存入磁盘(即:页的 LSN 小于记录的 LSN),则在伪撤销记录链中建立一个新项,它包含该记录的 LSN 和被其补偿的记录的 LSN(即:补偿日志记录的 LSN 域和 RedoNext_LSN 域)。

当遇到对应于更新操作的记录时,根据当前的已撤销记录链和伪撤销记录链来确定是否执行 undo 操作以及是否产生补偿日志记录:

△ 如果在已撤销记录链中存在相应的项,那么无需任何处理;

△ 如果在伪撤销记录链中存在相应的项(说明存在相应的补偿日志记录),那么比较页的 LSN 和该记录的 LSN,若前者大于等于后者(页已被更新),则执行 undo 操作,并更新页的 LSN 为已有补偿日志记录的 LSN;否则,页未被更新,所以不作任何处理;

△ 如果在已撤销记录链和伪撤销记录链中均不存在相应的项(说明不存在相应的补偿日志记录),那么比较页的 LSN 和该记录的 LSN,若前者大于等于后者(页已被更新),则执行 undo 操作,产生相应的补偿日志记录,并更新页的 LSN 为补偿日志记录的 LSN;否则,页未被更新,不作任何处理。

• 伪更新操作。对于某个对应于更新操作的记录而言,如果在已撤销记录链和伪撤销记录链中均不存在相应的项,并且更新操作的结果未存入磁盘,那么 WAL-P 将不作任何处理,我们称这样的更新操作为伪更新操作。WAL-P 对伪更新操作不作任何处理是不妥当的,这会引起异常情况的出现。

为了避免异常情况的发生,WAL-P 对伪更新操作也进行适当的处理:为其产生相应的补偿日志记录。也就是说,WAL-P 通过保证每一个更新操作都有相应的补偿日志记录来有效地避免异常情况的出现。

4.3 redo 阶段

redo 阶段依据前两个阶段的结果选择性地地进行某些 redo 操作,从而保证事务的持久性。如前所述,它的准备工作已由 undo 阶段完成(建立前向链),它只需获取前向链的头,然后依次重做前向链中的每一个操作。由于前向链是存于日志缓冲区中的,所以 redo 阶段不必访问日志文件,从而节约了大量的时间开销。

结论 面向对象数据库系统 KDOODB 采用页服务器结构,符合 ODMG-93 国际标准,支持嵌套事务模型。KDOODB 系统中的恢复策略 WAL-P 具有如下特点:

• 采用先写日志协议; • 采用基于页的日志; • 有力支持嵌套事务模型; • 有力支持页面级的并发控制; • 既支持操作日志又支持值日志; • 能够有效地处理事务故障和系统故障; • 提供对检查点的支持,为恢复的有效实现奠定基础; • 发生系统故障后,WAL-P 选择性地地进行 redo 和 undo 操作,从而使系统故障的恢复更加高效; • 具有一定的可扩展性,稍加修改,即可为基于对象的事务管理提供支持。

参考文献

- 1 王意洁,胡守仁. 面向对象数据库中的事务标识分配策略. 计算机工程,1998,6
- 2 Harder T. Principles of Transaction-Oriented Database Recovery. ACM Computing Surveys,1983,15(4),287~317
- 3 Gray J N, Reuter A. Transaction Processing Concepts and Techniques. Morgan Kaufmann Publishers, USA, 1993
- 4 王意洁,王勇军,胡守仁. 面向对象数据库管理系统中的事务管理. 计算机科学,1996,23(6),59~62

《计算机科学》在国外发行

国外总发行:中国国际图书贸易公司(北京 399 信箱)
国外发行代号:6210M