

适合分布式实时内存数据库的全局一致性模糊备份策略^{*})

肖迎元 刘云生 邓华锋 刘小峰

(华中科技大学计算机科学与技术学院 武汉 430074)

摘要 当站点介质故障导致本地磁盘数据库和日志文件都遭受破坏的情况下,为了确保能迅速而有效地将全局数据库恢复到某个一致性状态,全局备份操作策略变得非常关键。本文提出了一个适合分布式实时内存数据库的全局一致性模糊备份策略,在此基础上给出了相应的故障恢复方法。

关键词 分布式实时内存数据库,介质故障,全局一致性备份操作,模糊备份策略

Global Consistency Fuzzy Backup Strategy for Distributed Real-time Main Memory Database

XIAO Ying-Yuan LIU Yun-Sheng DENG Hua-Feng LIU Xiao-Feng

(College of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074)

Abstract When both local disk database and log file in a certain site are destructed by medium failure, global backup strategy is very vital to recover global database to certain consistent state rapidly and effectively. This paper presents a global consistency fuzzy backup strategy suitable for distributed real-time main memory database. On the basis of the proposed strategy, the corresponding recovery method of failure is presented.

Keywords Distributed real-time main memory database, Medium failure, Global consistency backup, Fuzzy backup strategy

1 引言

分布式实时数据库系统是分布式数据库系统和实时数据库系统相结合的产物,是事务和数据都可以具有定时特性或显式的定时限制的分布式数据库系统。为了更好地满足事务和数据的定时限制,分布式实时数据库系统通常需要内存数据库技术支持。内存数据库(MMDB)要求数据库工作版本常驻内存,外存版本作为内存版本的备份。通过一定的内、外存数据交换策略,MMDB能确保事务执行过程中无数据 I/O^[1]。我们将集成了 MMDB 技术的分布式实时数据库系统称为分布式实时内存数据库系统(DRTMMDBS; Distributed Real-Time Main Memory Database System),它集成了分布式数据库、实时数据库和内存数据库的能力,但并非三者概念、技术、机制上的简单组合,而有一系列问题需要研究和解决,如事务调度、并发控制、故障恢复、安全性等。

目前,对于 DRTMMDBS 故障恢复技术的相关研究不多,已有的工作主要是针对单一的分布式数据库、实时数据库、内存数据库的恢复策略和技术的研究。文[2,3]研究了适合实时数据库的系统故障恢复方法。文[4,5]研究了适合内存数据库基于日志的恢复策略。文[6,7]研究了分布式数据库系统全局一致性检验点策略。

本文研究在站点介质故障导致本地磁盘数据库和日志文件都遭受破坏的情况下,如何实现 DRTMMDBS 迅速而有效的恢复,提出了一个适合 DRTMMDBS 的全局一致性模糊备份策略,在此基础上给出了相应的故障恢复方法。

2 系统模型和假定

一个 DRTMMDBS 由 N 个站点通过固定网络连接而成,其中每个站点都有一个数据库服务器,所有的数据库服务器共同构成一个分布式实时数据库系统,支持全局实时事务处理。每个数据库服务器包含一个全局实时数据库管理系统、一个局部实时数据库管理系统(LDBMS)、一个通信管理器、一个全局数据字典、一个局部数据字典、一个局部内存数据库(LMDB)和一个局部外存数据库(LSDB)。每个数据库服务器均有场地自治性,支持局部实时事务处理,所有的 LMDB 共同构成逻辑的全局内存数据库(GMDB)。所有的 LSDB 共同构成逻辑的全局外存数据库(GSDB)。

按照实时事务执行时所需访问站点的个数,我们将 DRTMMDBS 中的实时事务分为分布式实时事务(DRTT)和局部实时事务(LRTT)两大类。DRTT 涉及到对多个(两个及以上)站点的 LMDB 存取操作,LRTT 仅涉及到对单个站点的 LMDB 存取操作。DRTT 对全局数据库的存取经转换、分解、优化后,产生一个涉及用通信原语联系的对多个站点的 LMDB 进行存取的实时子事务的集合。同一站点的各实时子事务构成该站点的实时子事务集合,在该站点的局部实时数据库管理系统 LDBMS 控制下并发执行和提交。

当一个实时子事务(或一个本地事务)开始执行,LDBMS 为其分配一定数量的内存空间作为事务私有数据缓冲区,读操作 $read(X)$ 表示把数据对象 X 的值从 LMDB 读到相应事务私有数据缓冲区。写操作 $write(X)$ 表示数据对象 X 的新值被写入到对应事务私有数据缓冲区。由一个事务写入或更

^{*})国家自然科学基金(60203017)、“十·五”国防预研重点项目(413150403)。肖迎元 博士研究生,研究兴趣为移动、分布式、实时、内存数据库理论及实现,数据库和信息系统开发,计算机图形学和 CAD 技术;刘云生 教授,博士生导师,主要研究方向为现代(非传统)数据库理论与技术、软件方法与支撑环境;邓华锋 博士研究生,研究兴趣为数据集成、实时仿真技术;刘小峰 博士研究生,研究兴趣为时空数据库技术。

新的私有数据缓冲区,在该事务提交时才被刷新到 LMDB,即采用了“延迟写”技术。当执行检验点时,LMDB 中更新被写出到 LSDB。

3 两级备份恢复模型

由于内存的易失性,在 DRTMMDBS 中我们采用两级备份机制来分别针对系统故障和介质故障。图 1 给出了基于日志技术的两级备份恢复模型。

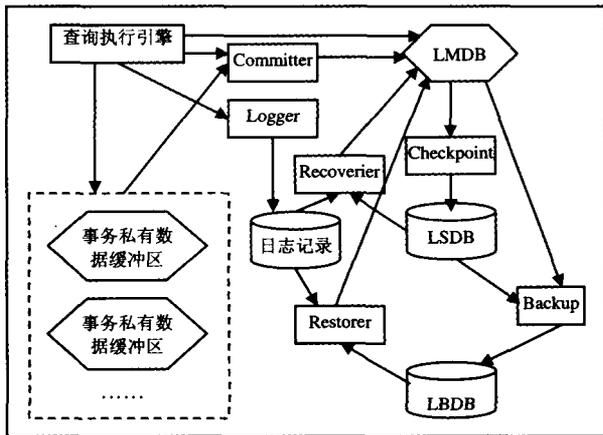


图 1 单个站点的两级备份恢复模型

对每一个读、写操作,查询执行引擎调用 Logger(日志管理器)记录相应日志记录。当事务提交时,Committer(提交管理器)负责将事务的更新写到 LMDB(采用了延迟更新策略)。一级备份是将 LMDB 中的更新数据保存到 LSDB 中,已备站点系统故障后 LMDB 的恢复,该级备份由本地检验点(Checkpoint)操作完成。二级备份是将本地数据库(LMDB 或者 LSDB)当前状态保存到独立的备份存储设备中,已备站点介质故障后进行 LSDB 和 LMDB 的恢复,该级备份由 Backup 操作完成。Recoverier(恢复管理器)负责站点系统故障后的恢复处理,Restorer(修复管理器)负责站点介质故障后的恢复处理。LBDB 表示本地备份数据库。本文主要研究 Backup 策略和介质故障后的恢复处理技术。

4 全局一致性模糊备份

对于站点系统故障,LMDB 数据丢失,但 LSDB 未遭破坏,因此可依据 LSDB 和日志重建 LMDB 并将 LMDB 恢复到故障前的一致性状态。而对于站点介质故障,由于 LSDB 遭到破坏,仅仅依靠日志已无法将数据库恢复到故障前的一致性状态。对于介质故障后的恢复处理通常基于备份技术,备份技术通过建立数据库的后备副本(存储在独立的备份存储设备中也称为备份数据库)来确保介质故障发生后数据库的可恢复性,即通过备份数据库和日志将数据库从备份时刻的状态恢复到故障前的一致性状态。

若介质故障同时造成日志被破坏,由于无日志信息可用,故无法将数据库从备份时刻的状态恢复到故障前的一致性状态,仅能依据备份数据库将数据库恢复到备份时刻的状态。对于集中式数据库系统而言,若能确保备份过程中事务的一致性(即对任何一个事务,备份数据库要么包含该事务的所有更新操作的效果,要么不包含该事务的任何更新操作的效果),则通过备份数据库可将数据库恢复到备份时刻的一致性状态。而对于 DRTMMDBS,由于全局数据库被分片、分布到

各站点,故当站点发生介质故障后,依据故障站点的一致性备份数据库并不能确保全局数据库的一致性(在备份之后介质故障发生前成功提交的且介质故障站点作为参与者的 DRTT 其效果部分反映到数据库中,部分未反映到数据库中)。因此,对于 DRTMMDBS,当介质故障同时造成日志信息不可用时,除了根据本地备份数进行恢复以外,而涉及到重构全局数据库一致性的问题。重构全局一致性数据库涉及到各个站点对大量的已提交的事务的 UNDO 操作,这些 UNDO 操作又可能导致新的 UNDO 操作(串联 UNDO 操作),从而可能形成“多米诺”效果(Domino effect),最坏的情况将导致全局数据库回滚到初始的状态。重构全局一致性数据库高昂的代价,显然不能满足 DRTMMDBS 高性能的需求。因此,对于 DRTMMDBS 必须研究介质故障后无需执行全局数据库一致性重构的高效的全局一致性备份技术。

4.1 相关概念

定义 1 本地备份操作(LB):在某个站点执行的、将本地数据库当前状态复制到独立备份存储设备中去的操作。

由于 DRTMMDBS 采用了内存数据库技术,即数据库工作版本 LMDB 常驻内存,若 LMDBLSDB,即 LMDB 包含了 LSDB 的全部数据对象,本地备份操作负责将 LMDB 的当前状态复制到独立备份存储设备中;若 LMDBLSDB 不成立,即 LSDB 中有部分数据对象不在 LMDB 中,本地备份操作负责将 $LMDB \cup (LSDB - (LSDB \cap LMDB))$ 中数据对象的当前状态复制到独立备份存储设备中。

各个站点间的本地备份操作可以相互协作,也可各自独立进行。

在下面的定义中,我们用 LBDB 表示由本地备份操作 LB 产生的存储在独立备份存储设备中本地备份数据库;符号 $T \text{ In } LBDB$ 表示事务 T 的执行结果被包含在 LBDB 中,即 T 的所有更新操作的效果都包含在 LBDB 中;符号 $T \text{ Out } LBDB$ 表示事务 T 的执行结果都不包含在 LBDB 中,即 T 的任一更新操作的效果都不包含在 LBDB 中。

定义 2 本地一致性备份操作(LCB):一个本地备份操作 LB 被称为本地一致性备份操作,如果满足下面的条件:

- (1)对每一个本地事务或子事务 T,都有 $(T \text{ In } LBDB) \vee (T \text{ Out } LBDB)$;
- (2)假定 T 存取了由本地事务或子事务 T(更新了的数据对象,若 $T \text{ In } LBDB$,则 $T' \text{ In } LBDB$ 。

定义 3 本地模糊备份操作(LFB):若在执行本地备份操作时允许执行本地事务或子事务,则称该本地备份操作为本地模糊备份操作。

定义 4 本地一致性模糊备份操作(LCFB):一个本地备份操作 LB 被称为本地一致性模糊备份操作,如果满足下列条件:

- (1)LB 为本地一致性备份操作;
- (2)LB 为本地模糊备份操作。

定义 5 全局备份操作(GB):一个全局备份操作定义为各站点本地备份操作的集合,即 $GB = \{LB_1, LB_2, \dots, LB_n\}$,这里 $LB_i (i=1, 2, \dots, n)$ 表示在站点 i 上执行的本地备份操作。

我们用符号 GBDB 表示由全局备份操作 GB 产生的全局备份数据库,即 $GBDB = \{LBDB_1, LBDB_2, \dots, LBDB_n\}$,这里 $LBDB_i (i=1, 2, \dots, n)$ 表示在站点 i 上执行本地备份操作 LB_i 产生的本地备份数据库。介质故障发生后,全局备份数据库

被用来实现数据库的全局重构。

定义 6 全局一致性备份操作(GCB):一个全局备份操作 $GB = \{LB_1, LB_2, \dots, LB_n\}$ 被称为全局一致性备份操作,如果满足下面的条件:

(1)构成 GB 的每一个本地备份操作都是本地一致性备份操作;

(2)对每一个分布式事务(全局事务)T,都有 $(T \text{ In } GBDB) \vee (T \text{ Out } GBDB)$ 。

这里,符号 $T \text{ In } GBDB$ 表示分布式事务 T 的所有更新操作的效果都包含在 GBDB 中; $T \text{ Out } GBDB$ 表示分布式事务 T 的任一更新操作的效果都不包含在 GBDB 中。

容易证明:

定理 1 对于分布式事务 $T = \{ST_1, ST_2, \dots, ST_m\}$,若对任意的子事务 $ST_i \in T$ 都有 $ST_i \text{ In } LBDB_i$, 则可得 $T \text{ In } GBDB$;若对任意的子事务 $ST_i \in T$ 都有 $ST_i \text{ Out } LBDB_i$, 则可得 $T \text{ Out } GBDB$ 。

定义 7 全局模糊备份操作(GFB):若构成 GB 的每一个本地备份操作都是本地模糊备份操作,则称该 GB 为全局模糊备份操作。

与模糊备份操作相对应的为静态备份操作,静态备份操作在备份过程中不允许存在活跃的事务。由于备份过程中存在大量的磁盘 I/O 操作,显然静态备份操作不能够满足 DRTMMDBS 高性能的需求。

定义 8 全局一致性模糊备份操作(GCFB):一个全局备份操作 GB 被称为全局一致性模糊备份操作,如果满足下列条件:

(1)GB 为全局一致性备份操作;

(2)GB 为全局模糊备份操作。

定义 9 全局备份协调者(GBC):全局备份操作 GB 的发起和协调站点称为全局备份协调者。

在上一个全局备份操作未结束前,GBC 不会发起下一个全局备份操作,因此在任何时候,系统中至多存在一个全局备份操作。

定义 10 全局备份参与者(GBP):除 GBC 以外,构成 DRTMMDBS 的其它站点称为全局备份参与者。

4.2 全局一致性模糊备份策略

实现全局一致性备份操作需要构成全局备份操作各本地备份操作相互协作,通常都包含下面两个基本步骤:

(1)各站点对哪些事务的结果应包含在 GBDB 中、哪些事务的结果应不被包含在 GBDB 中达成一致性决定;

(2)基于上述决定,各站点实现本地一致性备份操作。

为了实现步骤(1),我们将系统中所有事务按照与当前全局备份操作的时序关系分为两大类:全局备份操作前事务(TBGB)和全局备份操作后事务(TAGB)。TBGB 表示结果必须包含在 GBDB 中的事务;TAGB 表示结果全不包含在 GBDB 中的事务。由于只读事务不涉及到对数据库的更改,因此上面的事务分类并不考虑只读事务,都是针对更新事务(包含有更新操作的事务)。由于在 DRTMMDBS 中,各站点维持自己独立的逻辑时标,系统无统一的时标。因此,为了实现 TBGB 和 TAGB 划分,我们预先给出如下定义:

定义 11 事务提交时标(TCTS):对本地事务,TCTS 定义为该本地事务记 Commit 日志记录时刻站点的逻辑时标值;对分布式事务,TCTS 定义为该分布式事务所有子事务记 Commit 日志记录的逻辑时标值最大值;对子事务而言,

TCTS 定义为该子事务所对应分布式事务的 TCTS,即子事务与其对应的分布式事务有相同的 TCTS。

定义 12 全局备份协调者时标(GBCTS):GBCTS 定义为全局备份操作发起时刻全局备份协调者站点的逻辑时标值。

在我们的全局一致性模糊备份策略(GCFBS)中,全局备份操作由全局备份协调者发起,全局备份协调者通过广播包含 GBCTS 的消息,发起全局备份操作请求。

定义 13 全局备份参与者时标(GBPTS): $GBPTS = \text{Max}(GBCTS, CST)$,这里 GBCTS 为接收到的全局备份协调者时标;CST 为全局备份参与者收到全局备份协调者的全局备份操作请求时,全局备份参与者站点的逻辑时标值。

定义 14 全局一致性备份时标(GCBTS):GCBTS 定义为所有全局备份参与者时标的最大值,即 $GCBTS = \text{Max}(GBPTS_i), i=1, 2, \dots, n-1$ 。这里,GBPTS_i 表示第 i 个站点的全局备份参与者时标。

在后面的全局一致性模糊备份操作的实现策略中,我们以全局一致性备份时标 GCBTS 作为划分 TBGB 和 TAGB 的基准时标。凡是事务提交时标 TCTS 不超过(小于或等于)全局一致性备份时标 GCBTS 的事务被划分为 TBGB,而事务提交时标 TCTS 大于 GCBTS 的事务视为 TAGB。

显然有下面的定理:

定理 2 当前全局备份参与者时标不小于当前全局备份协调者时标,当前全局一致性备份时标不小于当前全局备份参与者时标,即 $GCBTS \geq GBPTS \geq GBCTS$ 。

我们的全局一致性模糊备份策略 GCFBS 的实现描述如下:

(1)全局备份协调者设置全局备份协调者时标 GBCTS 的值,广播包含 GBCTS 的消息,发起全局备份操作请求,启动全局备份操作;同时,在全局备份协调者站点,所有 TCTS 不超过 GBCTS 的事务可确定为 TBGB,这些事务的更新在它们提交时被直接刷新到 LMDB 中。而对于 TCTS 大于 GBCTS 的事务,尚不能确定它们是否一定属于 TBGB(可能有部分属于 TBAB,可能有部分属于 TBGB),下面称这些事务为临时未确定事务(TNDT: Temporary Non-decision Transactions)。这些临时未确定事务的更新在它们提交时并不直接刷新到 LMDB 中,而是刷新到内存临时提交区(TCA: Temporary Committed Area)中。若有多个 TNDT 先后对同一个数据对象进行了更新,则在 TCA 中依次维护该数据对象的多个版本。

(2)全局备份参与者收到来自全局备份协调者的全局备份操作请求后,设置全局备份参与者时标 GBPTS 的值,即 $GBPTS = \text{Max}(GBCTS, CST)$,并将 GBPTS 发送给全局备份协调者;同时,在全局备份参与者站点,所有 TCTS 不超过 GBPTS 的事务可确定为 TBGB,这些事务的更新在它们提交时被直接刷新到 LMDB 中。同样,对于 TCTS 大于 GBPTS 的事务(即 TNDT),它们的更新在提交时并不直接刷新到 LMDB 中,而是刷新到内存 TCA 中。

(3)全局备份协调者在收到所有全局备份参与者发送的 GBPTS 后,设置全局一致性备份时标 GCBTS 的值,即 $GCBTS = \text{Max}(GBPTS_i), i=1, 2, \dots, n-1$ 。这里,GBPTS_i 表示第 i 个站点的全局备份参与者时标,并将 GCBTS 向各全局备份参与者广播。同时,全局备份协调者将本站点临时未确定事务中所有 TCTS 不大于 GCBTS 的事务定为 TBGB。

若这些 TBGB 尚未提交,则提交时它们的更新效果被直接刷新到 LMDB 中;若这些 TBGB 已提交,则将 TCA 中相应的更新写到 LMDB 中,并释放相应的空间。未确定事务中余下的事务被确定为 TBAB,它们的更新在提交时被写入到 TCA 中。

(4) 全局备份参与者收到来自全局备份协调者的 GCBTS 消息后,将本站点临时未确定事务中所有 TCTS 不大于 GCBTS 的事务定为 TBGB,如同全局备份协调者站点类似处理策略,它们的更新效果被反映到 LMDB 中。未确定事务中余下的事务被确定为 TBAB,它们的更新在提交时被写入到内存 TCA 中。

(5) 当某个站点上的所有 TBGB 完成了提交活动,在该站点上开始执行本地备份操作,本地备份操作负责将 LMDB 复制到 LBDB 中。为了缩短复制时间,对于过期或即将过期的实时数据,无需从 LMDB 复制到 LBDB 中。进一步为提高效率,在复制时可采用增量复制技术,即只复制上一次备份以来更新了的数据。

(6) 当站点完成了本地备份操作后,TCA 中数据被刷新到 LMDB,同时释放 TCA 的相应的内存空间。当所有站点都释放了各自 TCA 的相应的内存空间,全局备份操作结束。

定理 3 GCFBS 能确保实现全局一致性模糊备份操作。

证明: 设 $GB = \{LB_1, LB_2, \dots, LB_n\}$ 是采用上述 GCFBS 策略的任一全局备份操作。

(1) 首先证明: $\forall LB_i \in GB, LB_i$ 为本地一致性备份操作。

设 T 为在站点 i 上执行的任一本地事务或子事务,则根据 GCFBS 中事务分类策略, T 要么为 TBGB,要么为 TBAB。若确定 T 为 TBGB,根据 GCFBS,仅当站点 i 上所有 TBGB 都完成提交后且更新刷新到 LMDB 后,才开始执行 LB_i ,因此可以确保:若 T 为 TBGB,则 $T \text{ In } LBDB$ 。若确定 T 为 TBAB,根据 GCFBS, T 的更新在提交时仅仅写入到站点 i 的 TCA 中,并且仅当 LB_i 完成后才将 TCA 刷新到 LMDB。因此可以肯定:若 T 为 TBAB,则 $T \text{ Out } LBDB$ 。综上所述,对站点 i 上的任一事务都有: $(T \text{ In } LBDB) \vee (T \text{ Out } LBDB)$;假定 $T \text{ In } LBDB$ 且 T 存取了由本地事务或子事务 T' 更新了的数据对象,由并发控制机制可以确保 T' 一定在 T 之前完成提交,即有 $TCTS(T') < TCTS(T)$,这里 $TCTS(T')$ 和 $TCTS(T)$ 分别表示事务 T' 和 T 的提交时标。由 $T \text{ In } LBDB$ 可得 $TCTS(T) \leq GCBTS$,又由 $TCTS(T') < TCTS(T)$ 可推出 $TCTS(T') < GCBTS$,由此可得 $T' \text{ In } LBDB$ 。综上所述证明了: $\forall LB_i \in GB, LB_i$ 为本地一致性备份操作。

(2) 其次证明:对任意分布式事务 T ,都有 $(T \text{ In } GBDB) \vee (T \text{ Out } GBDB)$ 。

假设分布式事务 $T = \{ST_1, ST_2, \dots, ST_m\}$,根据事务提交时标的定义, $TCTS(ST_1) = TCTS(ST_2) = \dots = TCTS(ST_m) = TCTS(T)$ 。若 $TCTS(T) \leq GCBTS$,则 $TCTS(ST_1) = TCTS(ST_2) = \dots = TCTS(ST_m) = TCTS(T) \leq GCBTS$,因而可知 $ST_i \text{ In } LBDB, i=1, 2, \dots, m$ 。由定理 1 可得 $T \text{ In } GBDB$;若 $TCTS(T) > GCBTS$,则 $TCTS(ST_1) = TCTS(ST_2) = \dots = TCTS(ST_m) = TCTS(T) > GCBTS$,因而可知 $ST_i \text{ Out } LBDB, i=1, 2, \dots, m$ 。由定理 1 可得 $T \text{ Out } GBDB$,即对任意分布式事务 T ,都有 $(T \text{ In } GBDB) \vee (T \text{ Out } GBDB)$ 。

(3) 最后证明 GB 为全局模糊备份操作。

因为 $\forall LB_i \in GB$,在站点 i 上,事务和 LB_i 可并发存在,

即 LB_i 为本地模糊备份操作,故可得 GB 为全局模糊备份操作。

(4) 综上所述,对采用 GCFBS 策略的任意全局备份操作 GB 都可推出 GB 为全局一致性模糊备份操作,即定理得证。

5 故障恢复处理

全局一致性模糊备份操作的目的是确保 DRTMMDBS 在介质故障发生且日志不可用的情况下能迅速而有效地将全局数据库恢复到一致性状态,以便 DRTMMDBS 能尽快地恢复服务。

基于上述 GCFBS 的介质故障恢复处理的算法描述如下:

(1) 故障站点重启系统后,向构成 DRTMMDBS 的各站点广播全局数据库重构请求,进入介质故障恢复处理状态;

(2) 各站点收到全局数据库重构请求后,停止系统服务,夭折所有活跃事务,进入介质故障恢复处理状态;

(3) 处在介质故障恢复处理状态各站点,从 LBDB 中装入数据到内存中,重建 LMDB。在站点完成了全部数据装入后,实现检验点磁盘和备份磁盘间的切换,即将原来的检验点磁盘置为备份磁盘,将原来的备份磁盘置为检验点磁盘,从而完成恢复处理,恢复站点服务。

在上述算法描述中,检验点磁盘指的是存储 LSDB 的磁盘存储器;备份磁盘指的是存储 LBDB 的磁盘存储器。

结束语 当站点介质故障导致本地磁盘数据库和日志文件都遭受破坏的情况下,为了确保能迅速而有效地将全局数据库恢复到某个一致性状态,全局备份操作策略变得非常关键。本文首先针对 DRTMMDBS 给出了一个两级备份的恢复模型,然后给出了本地一致性备份操作、本地模糊备份操作、全局一致性备份操作、全局一致性模糊备份操作等相关概念。在此基础上,提出了一个确保全局一致性模糊备份操作的全局备份策略 GCFBS,并证明了其正确性。最后,基于提出的全局备份策略 GCFBS,给出了一个有效的站点介质故障恢复方法。GCFBS 结合上述故障恢复方法能避免重构全局一致性数据库所需的高昂代价,有利于确保 DRTMMDBS 的实时性能。

参考文献

- 1 刘云生. 现代数据库技术. 北京: 国防工业出版社, 2001
- 2 Huang J, Gruenwald L. Impact of timing constraints on real-time database recovery. In: Proceedings of Workshop on Databases: active and real-time table of contents, Maryland, United States, Nov. 1996. 54~58
- 3 Sivasankaran R M, Stankovic K, Stankovic J, et al. Data placement, logging and recovery in real-time active database. International Workshop on Active and Real-time Database Systems, Jun. 1995
- 4 Xiao Yingyuan, Liu Yunsheng, Liao Guoqiong, et al. An Efficient Crash Recovery Technique for Real-Time Main Memory Database. Wuhan University Journal of Natural Sciences, Jan. 2005
- 5 Garcia-Molina H, Salem K. Main memory database systems: an overview. IEEE Transactions on Knowledge and Data Engineering, 1992, 4(6): 509~516
- 6 Kim J L, Park T. An efficient recovery scheme for locking-based distributed database systems. In: Proceedings of the 13th Symposium on Reliable Distributed Systems, Oct. 1994. 116~125
- 7 Lin Jun-Lin, Dunham M H. A Survey of Distributed Database Checkpointing. Distributed and Parallel Databases, 1997. 289~319