Priority-Based Test Point Incremental Backup Strategy of Real-Time Database

Liangzhu Xu,Lin Lei,Jianhua Wang,Xunrong Liu,Lijuan Ma
KaiLi PowerSupply,Kaili,Guizhou 556000, *China*

Abstract: Checkpoint technology is real-time memory database recovery one of the key technologies, while real-time database by reference to the data test points can be defined by real-time database backup point. This paper analyzes the characteristics of real-time memory database data based on comprehensive consideration given time constraints of data and transaction data checkpoint priority calculation method, then, combined with memory database-stage storage structure, discussed the inspection point based on priority incremental backup strategy. The simulation tests show that the proposed test point priority incremental backup strategy is more effective to find real-time database to carry out incremental backups incremental point.

Key words:real-time database; Incremental backup;Priority checkpoints

基于检验点优先级的实时数据库增量备份策略研究

许良柱，雷霖，王建华，刘顺荣，马丽娟
贵州凯里供电局，凯里 贵州 556000

摘　要：检验点技术是实时内存数据库恢复的关键技术之一，同时通过参照实时数据库的数据检验点可以定义出实时数据库的备份点。本文在分析实时内存数据库数据特征基础上，给出了综合考虑数据和事务定时约束的数据检验点优先级计算方法，然后，结合内存数据库段式存储结构，讨论了一种基于检验点优先级的增量备份策略。通过仿真测试，表明所提出的检验点优先级增量备份策略能比较有效的找到实时数据库的增量点从而进行增量备份。

关键词：实时数据库；增量备份；检验点优先级

1. 引言

实时数据库是数据库系统发展的一个分支，是实时系统技术和数据库技术相结合的产物。它具有高实时性、高数据吞吐量等特点，在电力系统得到了广泛地应用。但随着电力系统实时监控的应用趋于大型化、普遍化，系统内的信号数量成倍增加同时系统长期运行的需要，要保存的历史数据量非常大。因此，实时数据库中数据库的增量备份技术就显得尤为重要。将增量备份的策略引入实时数据库是非常必要的，可以达到节省存储空间、增加数据库容量、节省系统资源、提高备份效率等效果。

与传统基于磁盘的数据库系统相同，实时数据库(real-time database system，RTDBS)检验点的目的是在永久存储设备(如磁盘)中维持数据库最新版本、确定恢复起始点及减少故障恢复时间。同时由于检验点(check pointing，CKP)是 RTDBS 进行磁盘数据 I/O 的惟一机制，其效率高低直接影响系统性能好坏。因此通过找到实时数据库的检验点，可以设定增量备份的备份点从而进行增量备份。

本文将给出一种综合考虑数据和事务定时约束、基于于数据段检验点优先级的分区模糊检验点策略，能较好地满足 RTDBS 的增量备份要求。

2. 检验点优先级研究

2.1 检验点模式

RTDBS 中的数据表现为多种特征，如有效期、存取频率、关键性及存取事务的优先级等,其检验点策略应能考虑这些特征。其次,RTDBS 的检验点操作不应阻塞正常事务的执行，否则会延长事务执行时间而影响其定时限制的满足。显然，传统数据库检验点策略不能满足 RTMMDBS 这些要求。目前内存数据库检验点策略可分 3 类：非模糊检验点(non-fuzzy check pointing)策略、模糊检验点(fuzzy check pointing)策略和日志驱动检验点(log-driven check pointing)策略等[1-3]，但这些策略只考虑了内存的易失性,而未考虑事务和数据的定时约束。迄今为止，有关 RTDBS

检验点的研究并不多见。文献[4,5]分别讨论了两种分区检验点策略 UFPC (update frequency partition check pointing)和 UFVIPC (update Frequency Valid interval partition check pointing)。但它们都只考虑了数据的时间特性，而未考虑事务定时约束。

因此，在本文中设计了一种模糊检验点模式，在检验点触发时机上，不是采用周期性触发策略，而是根据日志存储区空间使用率（简记为 RLU）是否到达设定的阈值 α 来决定是否触发检验点进程。当 RLU 大于 α 时，检验点被触发执行。α 的初始值可根据应用的要求预先设定，在系统运行过程中可通过修改 α 值来动态调整检验点进程的执行频率。

在检验点执行时，检验点日志记录被写入检验点日志存储区。检验点日志记录包括四个部分：检验点位域、恢复时标(RTS)、检验点时标(CTS)和更新页位域。检验点位域包括 S-bit 和 F-bit。S-bit 表示系统需要备份时，本次检验点过程是否成功完成。S-bit 为 1，表示成功完成；为 0，表示在执行检验点过程中，系统发生故障。F-bit 表示本次检验点执行过程中，LMDB 中全部更新是否已被写入到 LSDB。F-bit 为 1，表示成功备份；为 0，表示备份过程中发生系统故障。RTS 表示系统进行备份时，必须执行 Redo 操作的最早记录时系统的逻辑时标。更新页位域为 LMDB 中每一数据页设置一位，用来表示检验点时刻对应数据页状态（已更新和未被更新两种状态）。已更新对应位置 1，未被更新，对应位置 0。同样，LMD 中每一数据页也设置一更新位，更新位为 1，表示对应数据页已被更新，但结果并未被写入到 LSDB；为 0，表示自上次检验点后该块未被更新，内外存处在一致状态。

2.2 检验点优先级

通常，RTDBS 中的数据按特征可做不同类。按有效期长短，可分为时序数据和非时序数据。特别地，一类时序数据其值在从磁盘读入内存之就已变为无效，即其有效期小于 AT（AT 为系统成一次磁盘读写所需的平均时间），称之为短时限序数据。按更新频率高低，可分为高频和低频数据。按数据对事务处理的重要性影响程度（即关键度不同），可分为关键数据和一般数据。按存取数据事的优先级高低，可分为高优先级顶数据和低优先顶数据[6]。数据的优先级是指存取该数据的全部务的最高优先级。文献[7]详细分析了这些特征 RTMMDBS 恢复的影响。具体对检验点而言，我们有如下原则：

1)较短有效期数据应及早地刷新到磁盘，以少数据失效率；

2)短时限时序数据无须刷新到磁盘，以减少必要的恢复开销；

3)高频数据应经常刷新到磁盘，以减少日志理时间；

4)关键数据应优先刷新到磁盘，以保证更多要事务满足定时约束；

5)优先级顶高的数据应优先刷新到磁盘，以证更多高优先级事务满足定时约束

设 x 为数据库中任一数据对象，$evi(x)=[evi_b(x),evi_e(x)]$ 为 x 的有效期，$evi_b(x)$ 为有效期起始时刻，$evi_e(x)$ 为有效期终止时刻；$UF(x)$、$k(x)$ 和 $PC(x)$ 分别为 x 的更新频率、关键度和最大优先值；$CP(x)$ 为 x 做检验点的优先级。于是可根据式(1)计算 $CP(x)$。其中 W_i （i=1,2,3,4）是加权值。$CP(x)$ 值越大，意味着 X 越应优先完检验点操作。

$$CP(x)=\begin{cases}0\\ \dfrac{W_1}{evi(x)}+UF(x)\times W_2+k(x)\times W_3+PC(x)\times W_4\end{cases}$$

$$evi(x)>AT \text{ 且 } UF(x)>0$$

设实时内存数据库 D 由 m 个数据段（一连续物理存储区域）组成，而每个数据段又包含 k 个数据对象，即 $D=\{S_i\mid 1\le i\le m\}$，$S_i=\{x_{ij}\mid 1\le j\le k\}$。

我们定义以下四个变量：

S_i 的有效期：S_i 所包含的全部数据对象的有效期的最小值

$$evi(x_i)=\min(evi(x_{ij})),1\le j\le k$$

S_i 的更新频率：S_i 在单位时间 Δt 内被更新的次数 N_i

$$UF(x_i)=N_i/\Delta t$$

S_i 的关键度：S_i 所包含全部数据对象关键度的最大值

$$k(x_i)=\max(k(x_{ij})),1\le j\le k$$

S_i 的最大优先级值：S_i 所包含全部数据对象的优先级值最大值的最大值

$$PC(x_i)=\max(PC(x_{ij})),1\le j\le k$$

于是，根据式(7)可以计算 S_i 的检验点优先级 $CP(x_i)$：

978-1-935068-17-4 © 2010 SciRes.

$$CP(x_i)=\begin{cases}0\\ \dfrac{W_1}{evi(x_i)}+UF(x_i)\times W_2+k(x_i)\times W_3+PC(x_i)\times W_4\end{cases}$$

$evi(x_i)>AT$ 且 $UF(x_i)>0$

3. 增量数据备份策略

实时内存数据库由两个不可分割的部分组成内存数据库（main memory databases，MMDB）和磁盘数据库（secondary databases，SDB），其中 MMDB 存放数据库的工作版本，而 SDB 则作为 MMDB 的备份。检验点操作就是定期刷新 MMDB 的最新变化到 SDB，以保证 SDB 与 MMDB 状态保持最近一致。基于操作点的增量备份过程主要分为两个阶段：分析阶段和备份阶段。

3.1 分析阶段

分析阶段通过分析日志来确定所有需要备份（Back）的事务并确定 Back 起始点。如图 1 所示，LS_Checkpoint 表示在备份操作之前最后一次成功完成的检验点；T1 表示在该检验点开始记录写入日志文件以前活动日志区已刷新到日志文件的事务；T2 表示部分日志记录在检验点开始时间前写入，部分日志在检验点开始时间后写入的事务；T3 表示日志记录在检验点开始时间后，但在检验点结束之前写入的事务；T4 表示部分日志记录在检验点结束时间前写入，部分日志在检验点结束时间后写入的事务；T5 表示日志记录在检验点结束时间后，但在备份操作前写入的日志文件的事务。对于 T1 类事务，在基于先来先服务调度策略的传统数据库系统中，由于所有日志记录在 LS_Checkpoint 开始之前已写入日志文件，按照先来先服务调度策略，它们对数据库的更新也应该在 LS_Checkpoint 执行前被写入到 SDB，因此无需再进行备份。其具体的操作步骤如下：

（1） 置 T2 类事务的集合 $S_{T2}=\varnothing$

（2） 从日志文件尾开始反向扫描，直到 LS_Checkpoint 的开始时间记录为止

（3） 根据 LS_Checkpoint 的开始时间日志记录的事务信息表确定集合 S_{T2}

（4） 从 LS_Checkpoint 的开始时间日志记录处反向扫描日志文件，直到 $S=\varphi$

（5） 将当前日志记录的位置记录下来作为 Back 起始点

3.2 Back 阶段

Back 阶段依据分析阶段确定的备份起始点，执行备份操作，将 SDB 中的部分数据进行备份。其具体实现步骤描述如下：

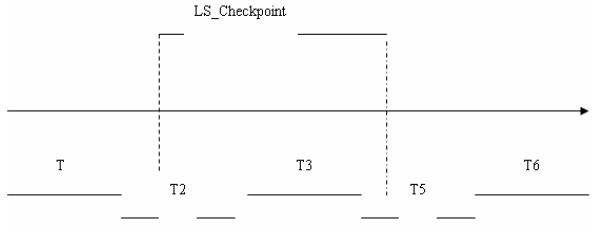

图 1 日志记录事务图

（1） 从起始点开始，正向扫描日志文件，直到日志文件尾

（2） 对于每一 Back 日志记录所描述事务 S，覆盖 MMDB 中对应数据页的相应数据对象，并将修改后的数据页刷新到 SDB

5. 结束语

检验点技术是实时内存数据库恢复的关键技术之一，同时通过参照实时数据库的数据检验点可以定义出实时数据库的备份点。我们在分析实时内存数据库数据特征基础上，给出了综合考虑数据和事务定时约束的数据检验点优先级计算方法，设计了一种基于检验点优先级的增量备份策略。通过仿真测试，表明所提出的检验点优先级增量备份策略能比较有效的找到实时数据库的增量点从而进行增量备份。

References (参考文献)

[1] H·V·Jagadish, A·Silberschatz, S·Sudarshan·Recovering frommain memory lapses[C]·The 19th Conf·Very Large Databases,Dublin, Ireland, 1993·391~404.

[2] S·K·Woo, M·H·Kim, Y·J·Lee·An effective recovery under fuzzy checkpointing in main memory databases [J]·Informa-tion and Software Technology, 2000, (42): 185~196.

[3] Dongho Lee, Haengrae Cho·Checkpointing schemes for fast restart in main memory database systems [C]·In: IEEE PacificRim Conf·Communications, Computers and Signal Processing,vol·2·Piscataway, NJ: IEEE Press, 1997·663~668.

[4] Jing Huang, Le Gruenwald·Crash recovery for real-time main memory database systems [C]·In: Proc·ACM Sym-posium on Applied Computing·New York: ACM Press, 1996·145~149.

[5] Jing Huang, Le Gruenwald·An up-date-frequency-valid-interval partition checkpoint tech-nique for real-time main memory databases[C]·The Workshop on Real-Time Databases, Newport Beach,CA, 1996·130~137.

[6] Liu Yunsheng·Advanced Database Technology [M]·Beijing: Defense Industry Press, 2001 (in Chinese).

[7] Liao Guo-Qiong, LiuYun-Sheng, Xiao Yin-Yuan. CPU scheduling in an embedded active real-time database sys-tem,Proceedings of the 11th ISPE International Confer-ence onConcurrent Engineering. Beijing, 2004: 903-908.