计算机研究与发展
Journal of Computer Research and Development
ISSN 1000-1239/CN 11-1777/TP
43(7):1291~1296, 2006

实时内存数据库分区模糊检验点策略

廖国琼[1,2] 刘云生[1] 肖迎元[1]
[1](华中科技大学计算机科学与技术学院 武汉 430074)
[2](西门子(中国)有限公司西门子中国研究院 北京 100102)
(liaoguoqiong@163.com)

A Partition Fuzzy Checkpointing Strategy for Real-Time Main Memory Databases

Liao Guoqiong[1,2], Liu Yunsheng[1], and Xiao Yingyuan[1]
[1](*School of Computer Science & Technology, Huazhong University of Science & Technology, Wuhan* 430074)
[2](*Corporation Technology, Siemens Limited China, Beijing* 100102)

Abstract Checkpointing is one of the important recovery techniques of real-time main memory database systems (RTMMDBS). Through analyzing the data characteristics in RTMMDBS, a method to calculate data checkpointing priorities is presented, which takes the timing constraints of both data and transactions into consideration. A partition fuzzy checkpionting strategy based on data segment checkpointing priority—PFCS-SCP is suggested, and the correctness of PFCE-SCP is also discussed. It is shown through performance testing that PFCS-SCP strategy can decrease the missing ratio of transactions in RTMMDBS.

Key words real-time database; main memory databases; database recovery; fuzzy checkpointing

摘 要 检验点技术是实时内存数据库恢复的关键技术之一. 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,讨论了一种基于数据段检验点优先级的分区模糊检验点策略 PFCS-SCP. 通过性能测试,表明所提出的检验点策略能减低超截止期事务比率.

关键词 实时数据库;内存数据库;数据库恢复;模糊检验点

中图法分类号 TP311.13

1 引 言

与传统基于磁盘的数据库系统相同,实时内存数据库(real-time main memory database system, RTMMDBS)检验点的目的也是在永久存储设备(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. 但由于检验点(checkpointing, CKP)是 RTMMDBS 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.

首先,RTMMDBS 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,RTMMDBS 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 checkpointing)策略、模糊检验点(fuzzy checkpointing)策略和日志驱动检验点(log-driven checkpointing)策略等[1~3],

收稿日期:2005-01-25;修回日期:2006-01-24
基金项目:中国博士后科学基金项目(2003034482)

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 RTMMDBS 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 checkpointing)和 UFVIPC (update-frequency-valid-interval partition checkpointing).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.

该文将给出一种综合考虑数据和事务定时约束、基于数据段检验点优先级的分区模糊检验点策略 PFCS-SCP,能较好地满足 RTMMDBS 恢复要求.

2 数据段检验点优先级

通常,RTMMDBS 中的数据按特征可做不同分类.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前就已变为无效,即其有效期小于 AT(AT 为系统完成一次磁盘读写所需的平均时间),称之为短时限时序数据.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.按存取数据事务的优先级高低,可分为高优先级顶数据和低优先级顶数据.数据的优先级顶是指存取该数据的全部事务的最高优先级.文献[6]详细分析了这些特征对 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,以减少数据失效率;

2) 短时限时序数据无须刷新到磁盘,以减少不必要的恢复开销;

3) 高频数据应经常刷新到磁盘,以减少日志处理时间;

4) 关键数据应优先刷新到磁盘,以保证更多重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,以保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、关键度和优先级顶;$CP(X)$ 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 越应优先完成检验点操作.

$$CP(X)=\begin{cases}0, & evi(X)\leqslant AT \text{ 或 } UF(X)=0,\\ \left\lceil \dfrac{W_1}{evi(X)}+UF(X)\times W_2+\right. & \\ \left. K(X)\times W_3+PC(X)\times W_4\right\rceil, & \\ evi(X)>AT \text{ 且 } UF(X)>0. & \end{cases}\tag{1}$$

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 $D=\{S_i\mid 1\leqslant i\leqslant m\}$,$S_i=\{X_{ij}\mid 1\leqslant j\leqslant k\}$.

定义 1. S_i 所包含全部数据对象的有效期的最小值称为 S_i 的有效期,记为 $evi(S_i)$,即 $evi(S_i)=\min(evi(X_{ij}))$,$1\leqslant j\leqslant k$.

定义 2. S_i 在单位时间 Δt 内被更新的次数 N_i 称为 S_i 的更新频率,记为 $UF(S_i)$,即 $UF(S_i)=N_i/\Delta t$.

定义 3. S_i 所包含全部数据对象关键度的最大值称为 S_i 的关键度,记为 $K(S_i)$,即 $K(S_i)=\max(K(X_{ij}))$,$1\leqslant j\leqslant k$.

定义 4. S_i 所包含全部数据对象的优先级顶的最大值称为 S_i 的优先级顶,记为 $PC(S_i)$,即 $PC(S_i)=\max(PC(X_{ij}))$,$1\leqslant j\leqslant k$.

于是,根据式(2)可计算 S_i 的检验点优先级 $CP(S_i)$:

$$CP(S_i)=\begin{cases}0, & evi(S_i)\leqslant AT \text{ 或 } UF(S_i)=0,\\ \left\lceil \dfrac{W_1}{evi(S_i)}+UF(S_i)\times W_2+\right. & \\ \left. K(S_i)\times W_3+PC(S_i)\times W_4\right\rceil, & \\ evi(S_i)>AT \text{ 且 } UF(S_i)>0. & \end{cases}\tag{2}$$

3 分区模糊检验点策略 PFCS-SCP

计算出数据段的检验点优先级后,则可根据该优先级对数据段进行逻辑分区.设有 n 个分区 P_1,P_2,…,P_n,除分区 P_n 外,其余区间长度都设为 L(正整数),则第 1 到第 n 个分区的检验点优先级范围依次为 $[1,L]$,$(L,2L]$,…,$((n-2)L,(n-1)L]$,$((n-1)L,\infty)$.基于检验点优先级的数据段分区过程可描述为

1) 当数据段 S_i 发生更新时,根据式(2)计算 $CP(S_i)$.

2) 若 $CP(S_i)=0$,则不予以考虑;否则,根据 $CP(S_i)$计算 S_i 应在的分区号 $v(1\leqslant v\leqslant n)$:

if $\left\lceil \frac{CP(S_i)}{L} \right\rceil \geqslant n$ then

$v:=n$;

else

$v:=\left\lceil \frac{CP(S_i)}{L} \right\rceil$.

3) 在进行检验点期间,对于由新执行事务(因为是模糊检验点)更新的数据段,若之前不属于任何分区(即新更新的),则将其加入分区 v;否则,根据数据段分区调整原则(第 4 节详树)确定该段要调入的新分区号.

于是可对不同分区安排不同的检验点频率,以满足不同特征数据的检验点要求. 设计分区检验点频率的总原则是:具有较高优先级的分区安排较高的检验点频率.

定义 5. 设 ACP_i 为 P_i 的平均检验点优先级(average checkpointing priority),则

$$ACP_i=\begin{cases}\dfrac{(2i-1)L+1}{2}, & 1\leqslant i<n,\\ (n-1)L, & i=n.\end{cases} \tag{3}$$

定义 6. 设 $RACP_i$ 为 P_i 的相对平均检验点优先级(relative average checkpointing priority),则

$$RACP_i=\frac{ACP_i}{\sum_{i=1}^{n}ACP_i}. \tag{4}$$

定义 7. 设 CF_i 为 P_i 的检验点频率(checkpointing frequency),则

$$CF_i=\left\lceil RACP_i\times n \right\rceil. \tag{5}$$

式(5)体现了若 $ACP_i<ACP_j$,则 $CF_i<CF_j$ 的原则. 基于数据段检验点优先级的分区模糊检验点策略 PFCS-SCP 算法描述如下:

Proc PFCS-SCP(/* n 为分区数,L 为分区优先级长度 */

Step1. for $i=1$ to n

根据式(3)计算各分区的 ACP_i;

Step2. for $i=1$ to n

根据式(4)计算各分区的 $RACP_i$;

Step3. for $i=1$ to n

根据式(5)计算各分区的初始检验点频率 CF_i;

Step4. 在全局检验点中记录数据库检验点开始日志;

Step5. for (all $CF_i>0, i=1,2,\cdots,n$)

Step5.1 查找具有最高检验点频率的分区 P_i:$CF_i>CF_j, j=1,2,\cdots,n, i\neq j$

Step5.2 将属于 P_i 的所有数据段依次刷新到外存;

Step5.3 降低 P_i 的检验点频率:$CF_i:=CF_i-\alpha$;/* α 为可调节不同检验点频率差别的常数 */

Step6. 在全局检验点日志记录数据库检验点结束日志,

Return./* 所有分区已完成检验点 */

4 PFCS-SCP 的正确性

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.

定义 8. 在时刻 t,S_i 在 MMDB 中的状态(或值)称为 S_i 在 t 的当前映像(current image),记为 $CI_t(S_i)$.

定义 9. 在时刻 t,S_i 在 SDB 中的状态(或值)称为 S_i 在 t 的备份映像(backup image),记为 $BI_t(S_i)$.

定义 10. 称 S_i 为"不稳定(unstable)"数据段,当且仅当在时刻 t,$BI_t(S_i)\neq CI_t(S_i)$.

检验点操作实际上就是将所有"不稳定"数据段的当前映像去更新其备份映像,使得在时刻 t,$BI_t(S_i)=CI_t(S_i)$.

在 PFCS-SCP 策略中,一个完整的数据库检验点由各分区的检验点操作组成. 令每个数据库检验点开始时记录 B-CKP 日志(记录时刻为 bt),结束时记录 E-CKP 日志(记录时刻为 et),分区 P_i 检验点开始时记录 $B\text{-}CKP_i$ 日志(记录时刻为 bt_i),结束时记录 $E\text{-}CKP_i$ 日志(记录时刻为 et_i),则称 $DCI=[bt,et]$为一个数据库检验点间隔,$PCI_i=[bt_i,et_i]$为分区 P_i 的检验点间隔.

定义 11. 在一个 DCI 内,若 $BI_{bt}(S_i)\neq CI_{bt}(S_i)$,则称 S_i 相对于该 DCI 是"旧不稳定"数据段(old unstable segment);若 $BI_{bt}(S_i)=CI_{bt}(S_i)$ 但 $BI_t(S_i)\neq CI_t(S_i)$,$bt<t\leqslant et$,则称 S_i 相对于该 DCI 是"新不稳定"数据段(new unstable segment).

定义 12. 称一个数据库的检验点是完整的,当且仅当在 DCI 内数据库中所有“旧不稳定”数据段至少刷新到磁盘一次.

定义 13. 称一个分区 P_i 的检验点是完整的,当且仅当在 PCI_i 内该分区内所有“旧不稳定”数据段至少刷新到磁盘一次.

因此,一个数据库检验点操作完成后,其所有“旧不稳定”数据段都已刷新到磁盘.而在完成检验点的过程中产生的“新不稳定”数据段可能刷新也可能未刷新到磁盘.但未刷新的“新不稳定”数据段在下一个检验点开始时将变为“旧不稳定”数据段而被刷新到磁盘.

引理 1. 在一个数据库检验点间隔 DCI 内,当所有“旧不稳定”数据段的分区不变时,若所有分区的检验点是完整的,则相应的数据库检验点也是完整的.

证明. 由 PFCS-SCP 算法可知,在一个 DCI 内,每个分区至少做一次检验点操作.当所有“旧不稳定”数据段所属分区在一个 DCI 其间不发生改变时,若分区 P_i 的检验点是完整的,则由定义 13,P_i 中的所有“旧不稳定”数据段至少刷新到磁盘一次.因此,若所有分区的检验点是完整的,即数据库中的全部“旧不稳定”数据段必定至少刷新到磁盘一次,故相应的数据库检验点是完整的. 证毕.

然而,由于 PFCS-SCP 采用模糊检验点策略,即允许事务处理与检验点操作并发执行.在检验点过程内,当有更新操作发生时,一数据段的所属分区可能发生变化.由式(2)可知,一数据段的检验点优先级在一个 DCI 内可能增加,即一个数据段可能由优先级较低的分区向较高分区调整.而数据库的检验点操作则是从最高检验点优先级的分区向较低检验点优先级分区进行的.这样,当一个“旧不稳定”数据段由一个优先级较低的分区调整到一个在 DCI 内已完成检验点的分区时,该数据段在该 DCI 内就得不到刷新的机会,从而导致了数据库检验点的不完整(图 1).

从图 1 可看出,在一个 DCI 内,若能保证“旧不稳定”数据段不调整到已完成检验点的分区时,仍能保持数据库检验点的完整性.

设“旧不稳定”数据段原始分区号为 u,新计算的分区号为 v,当前正在做检验点的分区号为 j,若当 $v \neq u$ 时,则有“旧不稳定”数据段分区调整原则:若 $v \geqslant j$,将“旧不稳定”数据段调整到当前最高检验点频率的分区(分区 j 或 $j-1$) 否则,调整到分区 v.

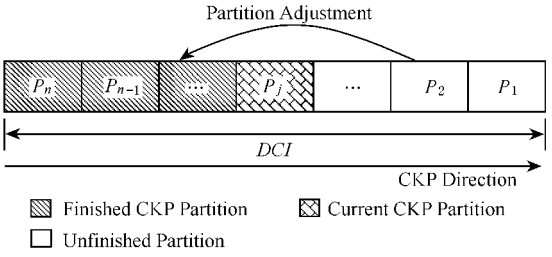

Fig. 1 Incomplete checkpointing.

图 1 不完整检验点

定理 1. 在一个数据库检验点间隔 DCI 内,若所有“旧不稳定”数据段都按上述分区调整原则进行分区调整时,则在 DCI 内数据库检验点是完整的.

定理 1 显然.于是,我们可设计一种基于 PFCS-SCP 的 REDO-only 恢复策略.限于篇幅,在此不再赘述,详细内容请见文献[7].

5 性能测试及评价

我们在自行研制的嵌入式实时内存数据库系统 ARTs-EDB[8] 上完成了对 PFCS-SCP 的测试.实验的性能指标为实时数据库通常采用的事务超截止期比率 MR(missing ratio):$MR = NumMiss/NumTotal \times 100\%$,其中 $NumMiss$ 表示超截止期事务数,$NumTotal$ 表示事务总数.表 1 为主要实验参数.

Table 1 Experiment Parameters

表 1 实验参数

Parameter	Meaning	Default	Range
seg_number	Segment number	1000	500~2000
per_temporal	Percent of temporal data	40%	30%~80%
per_persistent	Percent of non-temporal data	60%	20%~70%
per_short	Percent of short time period data	10%	5%~50%
part_number	Partition number	10	1~12
part_length	Priority length of partitions	60	10~100
arrive_rate	Arrival ratio of transactions	100transactions/s	50~300
pupdate	Probability of update operations	0.4	
slack	Slack factor	U[2.0,4.0]	
op_number	Operation number in each transaction	4	

在表 1 中,U[2.0,4.0] 表示在区间[2.0,4.0]上满足均匀分布的随机变量.实验中的事务主要为软实时,优先级分派采用最早截止期优先(EDF)策略.一个实时事务 T 的截止期按如下公式计算:

$Deadline(T) = AT(T) + Slack \times ET(T)$. 其中,$AT(T)$表示实时事务 T 到达系统的时间,即事务被接纳的时间;$Slack$ 表示松弛因子,为一个满足均匀分布的随机变量;$ET(T)$表示 T 的估计执行时间,$ET(T)$按如下公式估算:$ET(T) = op\_number \times op_t$,这里,$op\_number$ 表示事务包含的操作数,op_t 表示单个操作的平均执行时间.

分区数($part\_number$)是影响 PFCS-SCP 性能的因素之一. 由于随着分区数的增加,系统需记录各分区检验点日志的开销也相应增加. 从图 2 可看出,当分区数增加时,MR 减低. 但当分区数达到 8 个时,增加分区数对 MR 的改善程度不大.

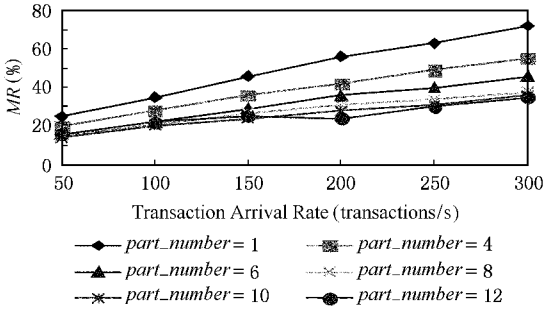

Fig. 2 Impact of partition number on MR.

图 2 分区数对 MR 的影响

分区长度($part\_length$)是影响 PFCS-SCP 性能的另一关键因素. 由于 PFCS-SCP 采用分区动态调整策略,若分区长度较短,则数据段进行分区调整较为频繁而增加系统开销. 图 3 是在 $part\_number$ 为 10 时不同优先级区间长度的性能测试结果. 结果表明,当 $part\_length$ 超过 60 时,MR 反而增加.

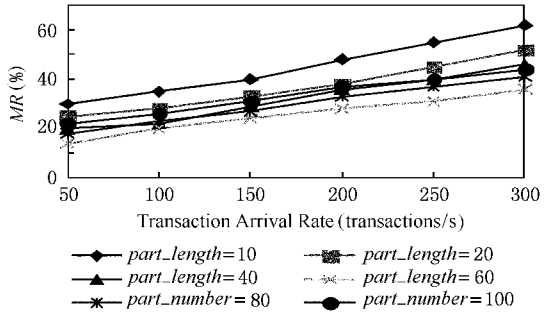

Fig. 3 Impact of partition number on MR.

图 3 分区优先级区间长度对 MR 的影响

与 UFVIPC 相比,PFCS-SCP 不仅考虑了数据本身的定时约束,而且考虑了与之相关事务的定时约束. 而且,PFCS-SCP 策略更加灵活和符合实际应用. UFVIPC 策略中的数据页(page)式分区是静态的,一旦确定不再改变. 而实际情况是,数据库中的数据对象除外部有效期长度可事先说明外,而数据的更新频率、优先级顶及关键度都与做检验点前存取该数据的事务相关,因此对数据段的检验点分区进行动态调整更能反映数据的'当前'定时特征.

图 4 和图 5 是 PFCS-SCP 在 $part\_number = 10$ 和 $part\_length = 60$ 情形下,与 UFVIPC(检验点分区数为 10)及传统数据库模糊检验点策略(只有 1 个检验点分区)在不同事务到达率与故障率情形下的性能测试情况. 结果表明 PFCS-SCP 均具有较好性能,这与 PFCS-SCP 同时考虑了事务的定时约束是相一致的.

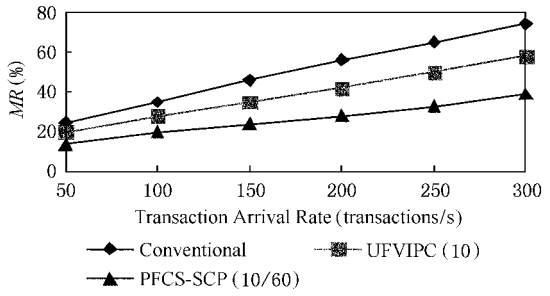

Fig. 4 MR comparison of checkpointing strategies on different transaction arrive rates.

图 4 检验点策略在不同事务达到率下的 MR 比较

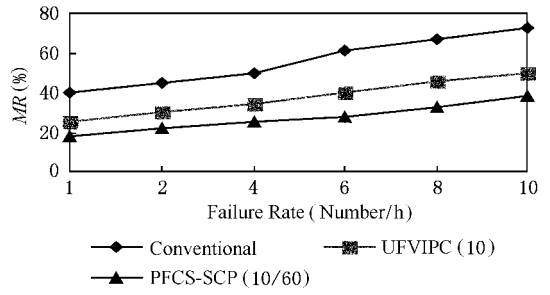

Fig. 5 MR comparison of checkpointing strategies on different failure rates.

图 5 检验点策略在不同故障发生率下的 MR 比较

6 总结与展望

为减低实时内存数据库系统超截止期事务比率,设计了一种灵活高效的分区模糊检验点策略 PFCS-SCP. 全文工作归纳如下:

1) 根据 RTMMDBS 的数据特征给出了考虑数据和事务定时约束的数据段检验点优先级计算方法.

2) 讨论了一种基于数据段检验点优先级的分区模糊检验点策略 PFCS-SCP,并分析了其正确性.

3) 完成了对 PFCS-SCP 的性能测试. 实验表明

PFCS-SCP 具有较好的性能.

随着嵌入式计算、移动通信技术的迅速发展,RTMMDBS 正在广泛应用于嵌入式移动计算环境.因此,在该文基础上继续研究在嵌入式移动计算环境中的 RTMMDBS 的检验点策略将是我们下一步方向.

参 考 文 献

1 H. V. Jagadish, A. Silberschatz, S. Sudarshan. Recovering from main 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]. Information 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 Pacific Rim 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 Symposium on Applied Computing. New York: ACM Press, 1996. 145~149

5 Jing Huang, Le Gruenwald. An update-frequency-valid-interval partition checkpoint technique 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)
(刘云生. 现代数据库技术[M]. 北京:国防工业出版社, 2001)

7 Liao Guoqiong. Research on recovery processing of embedded real-time databases: [Postdoctoral Research Report] [D]. Wuhan: Huazhong University of Science and Technology, 2005 (in Chinese)
(廖国琼. 嵌入式实时数据库恢复处理研究:[博士后研究报告][D]. 武汉:华中科技大学, 2005)

8 Liao Guoqiong, Liu Yunsheng, Xiao Yingyuan. CPU scheduling in an embedded active real-time database system [C]. The 11th ISPE Int'l Conf. Concurrent Engineering, Beijing, 2004

Liao Guoqiong, born in 1969. Ph. D. His main research interests include advanced databases theory and technology, including real-time database, active database, main memory database, mobile database and engineering database, etc.

廖国琼,1969 年生,博士,主要研究方向为现代(实时、主动、内存、移动、工程等非传统)数据库理论与技术.

Liu Yunsheng, born in 1940. Professor and Ph. D. supervisor. His main research interests include advanced databases, including real-time database, active database, main memory database, and mobile database, etc, and their integration; database and information system development; real-time data engineering; and software methodology and engineering technology.

刘云生,1940 年生,教授,博士生导师,主要研究方向为现代(实时、主动、内存、移动等非传统)数据库理论与技术及其集成实现、数据库与信息系统开发、实时数据工程、软件方法学与工程技术.

Xiao Yingyuan, born in 1969. Ph. D., senior member of CCF. His main research interests include advanced database technology, real-time information processing, and computer graphics & CAD.

肖迎元,1969 年生,博士,中国计算机学会高级会员,主要研究方向为现代数据库技术、实时信息处理、计算机图形学与 CAD.

Research Background

Checkpointing technique is the only one disk I/O mechanism in real-time main memory databases (RTMMDBS). Whether it is better or not will influence the performances of RTMMDBS.

Differing from disk-based databases, checkpointing techniques in RTMMDBS are required to consider data characteristics and can't interrupt the execution of normal transactions. PFCS-SCP, a partition fuzzy checkpionting scheme based on segment checkpionting priority, has taken into account the timing constraints of both data and transactions. As an update event occurs on a data segment, a checkpionting priority of the segment should be calculated by its effective period, access frequency, criticality and priority ceiling. Then, the segment will enter a logical partition which is determined in advance by its priority. A partition with higher priority has higher checkpionting frequency than a partition with lower priority.

This research is supported by the China National Postdoctoral Foundation and Defense "Ten-Five Plan" Key Project.