

第 21 讲：数据库崩溃恢复

数据库系统（2021 年秋季）

<https://15445.courses.cs.cmu.edu/fall2021/>

卡内基·梅隆大学

林马

1 碰撞恢复

DBMS 依赖于其恢复算法来确保数据库的一致性、事务的原子性和尽管失败的持久性。每个恢复算法都由以下两部分组成：

- 在正常事务处理期间执行的操作，以确保 DBMS 可以从故障中恢复
- 将数据库恢复到确保事务的原子性、一致性和持久性的状态失败后的操作。

语义利用算法(aries)是 IBM 研究在 20 世纪 90 年代早期为 DB2 系统开发的一种恢复算法。

aries 恢复协议中有三个关键概念：

- 日志记录：在将数据库更改写入磁盘（窃取+存储 noforce）。
- 在重做过程中重复历史记录：在重新启动时，重新跟踪操作并将数据库恢复到崩溃前的精确状态。
- 记录撤消期间的更改：记录撤消操作以确保在重复失败时不重复操作。

2 WAL 记录

预写日志记录扩展了 DBMS 的日志记录格式，以包括一个全局唯一的日志序列号(LSN)。图 1 显示了如何编写使用 LSN 的日志记录的高级图。

所有的日志记录都有一个 LSN。每次事务修改页面中的记录时，都会更新 pageLSN。每次 DBMS 将 WAL 缓冲区写入磁盘时，内存中的 LSN 将更新。

系统中的各种组件都在跟踪与它们相关的 lsn。这些 lsn 的一个表如图 2 所示。

每个数据页面都包含一个 pageLSN，它是对该页面的最新更新的 LSN。DBMS 还跟踪到目前为止刷新的最大 LSN(流量 LSN)。在 DBMS 可以将页面 i 写入磁盘之前，它必须至少将日志刷新到页面 LSN “<刷新 LSN” 的位置

3 正常执行

每个事务都调用一系列读写，然后是提交或中止。恢复算法必须具备的就是这一系列事件。

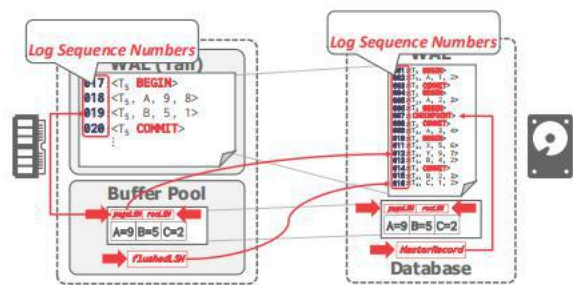


图 1：写入日志记录—每个 WAL 都有一个 LSNs 计数器，每一步都递增。该页面还保留一个 pageLSN 和一个 recLSN，它存储使页面变脏的第一个日志记录。流量 LSN 是指向写入磁盘的最后一个 LSN 的指针。主记录器指向最后一个成功通过的检查点。

名称	其中	定义
flushedLSN	记忆力	登录到磁盘上的最后一个 LSN
pageLSN	页码，	最新更新到页面。
recLSN	页码，	最老的更新到页面，因为它是最后一次被冲洗过的
lastLSN	T)	txnT 的最新记录)
主记录	磁盘	最新检查点的 LSN

图 2：LSN 类型——系统的不同部分还维护存储相关信息的不同类型的 LSN。

交易提交

当事务进入提交状态时，DBMS 首先将提交记录写入内存中的日志缓冲区。然后 DBMS 将所有日志记录刷新到并包括事务的提交记录到磁盘。请注意，这些日志刷新是对磁盘的顺序同步写入。每个日志页可以有多个日志记录。一个事务提交的图表如图 3 所示。

一旦提交记录安全地存储在磁盘上，DBMS 将向事务已提交的应用程序返回一个确认信息。稍后，DBMS 将写入一个特殊的 TXN-END 记录。这表明事务在系统中已完全完成，并且将不再有日志记录。这些 TXN-END 记录用于内部记账，不需要立即刷新。

交易中止

中止事务是仅应用于一个事务的请求请求文件撤销操作的特殊情况。

在日志记录中添加一个称为 prevLSN 的附加字段。这对应于该事务的前一个 LSN。DBMS 使用这些 preveLSN 值来为每个事务维护一个链表，以便更容易地浏览日志以找到其记录。

还引入了一种称为补偿日志记录 (CLR) 的新型记录。CLR 描述了

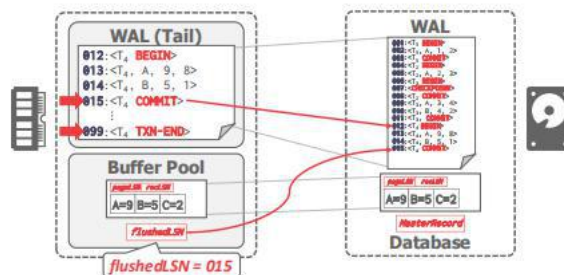


图 3: 事务提交一在事务提交 (015) 后, 日志被刷新, f 润滑 LSN 被修改以指向生成的最后一个日志记录。稍后, 写入事务结束消息在日志中表示该事务不会再次出现。

为撤消以前的更新记录的操作所采取的操作。它包含一个更新日志记录的所有字段，并加上 undoNext 指针（即，下一个要撤销的 LSN）。DBMS 像其他记录一样将 c1r 添加到日志中，但它们永远不需要撤消。

要中止事务，DBMS 首先将中止记录附加到内存中的日志缓冲区。然后，它会反向撤消事务的更新，以便从数据库中删除它们的影响。对于每个未完成的更新，DBMS 将在日志中创建 CLR 条目并恢复旧值。在逆转所有中止事务的更新后，DBMS 将写入 TXN-END 日志记录。该图如图 4 所示。

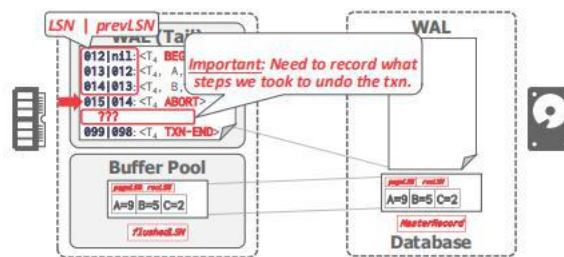


图 4: 事务中止-DBMS 为事务创建的每个日志记录维护一个 LSN 和 preveLSN。当事务中止时, 之前的所有更改都会被逆转。反向更改的日志条目到达磁盘后, DBMS 会将 TXN-END 记录附加到中止事务的日志中。

4 检查点

DBMS 定期使用检查点，并将其缓冲区池中的脏页写入磁盘。这用于最小化在恢复时需要重播的日志的数量。

下面讨论的前两种阻塞检查点方法是在检查点支持期间暂停事务

过程。为了确保 DBMS 在检查点期间不会错过对页面的更新。然后，提供了一种更好的方法，允许在检查点期间继续执行事务，但要求 DBMS 记录其他信息，以确定它可能错过的更新。

阻止检查点

DBMS 在需要检查点时停止事务和查询的执行，以确保它将数据库的一致快照写入磁盘。与上一节课中讨论的方法相同：

- 停止任何新事务的开始。
- 等待，直到所有活动的事务完成执行。
- 请将脏页刷新到磁盘上。

阻止检查点

与以前的检查点方案一样，除了 DBMS 不需要等待活动的事务完成执行。DBMS 现在记录到检查点开始时的内部系统状态。

- 停止任何新事务的开始。
- 在 DBMS 接受检查点时暂停事务。

活动事务表(ATT)：ATT 表示在 DBMS 中主动运行的事务的状态。在 DBMS 完成该事务的提交/中止过程后，该事务的条目将被删除。对于每个事务处理条目，ATT 包含以下信息：

- 事务处理范围：唯一的事务处理标识符
- 状态：事务的当前“模式”（运行、提交、撤销候选项）。
- 最近的 LSN

请注意，ATT 包含了所有没有 TXN-END 日志记录的交易。这包括正在提交或正在中止的两个事务。

“脏页表” (DPT)：DPT 包含有关缓冲池中被未提交的事务修改的页面的信息。每个脏页都有一个包含 recLSN 的条目(即，第一次导致页面变脏的日志记录的 LSN)。

DPT 包含缓冲区池中已脏的所有页面。这些更改是由正在运行、提交或中止的事务引起的并不重要。

总的来说，ATT 和 DPT 可以帮助 DBMS 通过文件恢复协议恢复崩溃前数据库的状态。

模糊检查点

模糊检查点是 DBMS 允许其他事务继续运行的检查点。这是 aris 在协议中使用的。

DBMS 使用其他日志记录来跟踪检查点边界：

- <检查点-开始>：指示检查点的开始。此时，DBMS 获取当前 ATT 和 DPT 的快照，它们在<检查点-END>记录中被引用。
- <检查点-END>：当检查点已经完成时。它包含 ATT+DPT，就在写入<检查点开始>日志记录时捕获。

5 个日记恢复

该协议由三个阶段组成。在崩溃后启动时，DBMS 将执行以下阶段，如图 5 所示：

1. 分析：读取 WAL，以识别缓冲池中的脏页和崩溃时的活动事务。在分析阶段结束时，ATT 告诉 DBMS 在崩溃时哪些事务是活动的。DPT 告诉 DBMS 哪些脏页可能没有到达磁盘。
2. 重做：从日志中的点开始重复所有操作。
3. 撤销：撤销在崩溃前未提交的事务的操作。

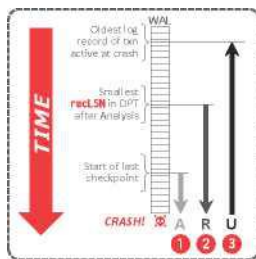


图 5: a 羊座恢复: DBMS 通过检查通过 MasterRecord 找到的最后一个开始检查点开始的日志来开始恢复过程。然后开始分析阶段, 通过向前扫描时间, 构建 ATT 和 DPT。在 Redo 阶段, 算法会跳转到最小的 recLSN, 这是最古老的日志记录, 可能已经修改了未写入磁盘的页面。然后, DBMS 将应用最小的 recLSN 中的所有更改。撤消阶段从崩溃时活动的事务的最古老的日志记录开始, 并逆转迄今为止的所有更改。

分析阶段

从通过数据库的主记录 LSN 找到的最后一个检查点开始。

1. 从检查点向前扫描日志。
2. 如果 DBMS 找到了一个 TXN-END 记录，请从 ATT 中删除其事务。
3. 所有其他记录，将事务添加到状态为 UNDO 的 ATT，并在提交时，更改事务状态以提交。
4. 对于更新日志记录，如果 P 页不在 DPT 中，则将 P 添加到 DPT 中，并将 P 的 recLSN 设置为日志记录的 LSN。

重做阶段

这个阶段的目标是让 DBMS 重复历史，重建其状态，直到崩溃的时刻。它将重新应用所有更新（甚至是中止的事务）和重做 CLR。

DBMS 从 DPT 中包含最小 recLSN 的日志记录中向前扫描。对于每个使用给定 LSN 记录的更新日志记录或 CLR，DBMS 将重新应用该更新，除非：

- 受影响的页面不在 DPT 中，或
- 受影响的页面在 DPT 中，但该记录的 LSN 小于 DPT 中该页面的 recLSN，或
- 受影响的 pageLSN（在磁盘上）>LSN。

要重做操作，DBMS 将重新应用日志记录中的更改，然后将受影响页面的 pageLSN 设置为该日志记录的 LSN。

在重做阶段结束时，为所有状态为提交的事务写入 TXN-END 日志记录，并将其从 ATT 中删除。

撤消阶段

在最后一个阶段，DBMS 将反转在崩溃时激活的所有事务。这些都是在分析阶段之后的 ATT 中处于 UNDO 状态的事务。

DBMS 使用 lastLSN 以反向 LSN 顺序处理事务，以加快遍历速度。当它反转事务的更新时，DBMS 会为每次修改向日志写入一个 CLR 条目。

一旦成功中止了最后一个事务，DBMS 将刷新日志，然后准备开始处理新事务。