

减少SQLServer数据库死锁的方法计算机等级考试 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/555/2021\\_2022\\_\\_E5\\_87\\_8F\\_E5\\_B0\\_91SQLS\\_c98\\_555779.htm](https://www.100test.com/kao_ti2020/555/2021_2022__E5_87_8F_E5_B0_91SQLS_c98_555779.htm)

如果两个用户进程分别锁定了不同的资源，接着又试图锁定对方所锁定的资源，就会产生死锁。此时，SQL Server将自动地选择并中止其中一个进程以解除死锁，使得另外一个进程能够继续处理。系统将回退被中止的事务，并向被回退事务的用户发送错误信息。大多数设计良好的应用都会在接收到这个错误信息之后重新提交该事务，此时提交成功的可能性是很大的。但是，如果服务器上经常出现这种情况，就会显著地降低服务器性能。为避免死锁，设计应用应当遵循一定的原则，包括：让应用每次都以相同的次序访问服务器资源。在事务期间禁止任何用户输入。应当在事务开始之前收集用户输入。尽量保持事务的短小和简单。如合适的话，为运行事务的用户连接指定尽可能低的隔离级别。[适用于6.5，7.0，2000]此外，对于SQL Server的死锁问题，下面是几则实践中很有用的小技巧。

使用SQL Server Profiler的Create Trace Wizard运行“Identify The Cause of a Deadlock”跟踪来辅助识别死锁问题，它将提供帮助查找数据库产生死锁原因的原始数据。[适用于7.0，2000] 如果无法消除应用中的所有死锁，请确保提供了这样一种程序逻辑：它能够在死锁出现并中止用户事务之后，以随机的时间间隔自动重新提交事务。这里等待时间的随机性非常重要，这是因为另一个竞争的事务也可能在等待，我们不应该让两个竞争的事务等待同样的时间，然后再在同一时间执行它们，这样的话将导致新的死锁。[适用于6.5

[适用于6.5, 7.0, 2000] 尽可能地简化所有T-SQL事务。此举将减少各种类型的锁的数量,有助于提高SQL Server应用的整体性能。如果可能的话,应将较复杂的事务分割成多个较简单的事务。

[适用于6.5, 7.0, 2000] 所有条件逻辑、变量赋值以及其他相关的预备设置操作应当在事务之外完成,而不应该放到事务之内。永远不要为了接受用户输入而暂停某个事务,用户输入应当总是在事务之外完成。[适用于6.5, 7.0, 2000]

在存储过程内封装所有事务,包括BEGIN TRANSACTION和COMMIT TRANSACTION语句。此举从两个方面帮助减少阻塞的锁。首先,它限制了事务运行时客户程序和SQL Server之间的通信,从而使得两者之间的任何消息只能出现于非事务运行时间(减少了事务运行的时间)。其次,由于存储过程强制它所启动的事务或者完成、或者中止,从而防止了用户留下未完成的事务(留下未撤销的锁)。

[适用于6.5, 7.0, 2000] 如果客户程序需要先用一定的时间检查数据,然后可能更新数据,也可能不更新数据,那么最好不要在整个记录检查期间都锁定记录。假设大部分时间都是检查数据而不是更新数据,那么处理这种特殊情况的一种方法就是:先选择出记录(不加UPDATE子句。UPDATE子句将在记录上加上共享锁),然后把它发送给客户。如果用户只查看记录但从来不会更新它,程序可以什么也不做。反过来,如果用户决定更新某个记录,那么他可以通过一个WHERE子句检查当前的数据是否和以前提取的数据相同,然后执行UPDATE。类似地,我们还可以检查记录中的时间标识列(如果它存在的话)。如果数据相同,则执行UPDATE操作。如果记录已经改变,则应用应该提示用户以便用户决定如何处理。虽然这种方法需要编

写更多的代码，但它能够减少加锁时间和次数，提高应用的整体性能。[适用于6.5，7.0，2000] 尽可能地为用户连接指定具有最少限制的事务隔离级别，而不是总是使用默认的READ COMMITTED。为了避免由此产生任何其他问题，应当参考不同隔离级别将产生的效果，仔细地分析事务的特性。[适用于6.5，7.0，2000] 使用游标会降低并发性。为避免这一点，如果可以使用只读的游标则应该使用READ\_ONLY游标选项，否则如果需要更新，尝试使用OPTIMISTIC游标选项以减少加锁。设法避免使用SCROLL\_LOCKS游标选项，该选项会增加由于记录锁定引起的问题。[适用于6.5，7.0，2000] 如果用户抱怨说他们不得不等待系统完成事务，则应当检查服务器上的资源锁定是否是导致该问题的原因。进行此类检查时可以使用SQL Server Locks Object: Average Wait Time (ms)，用该计数器来度量各种锁的平均等待时间。如果可以确定一种或几种类型的锁导致了事务延迟，就可以进一步探究是否可以确定具体是哪个事务产生了这种锁。Profiler是进行这类具体分析的最好工具。[适用于7.0，2000] 使用sp\_who和sp\_who2(SQL Server Books Online没有关于sp\_who2的说明，但sp\_who2提供了比sp\_who更详细的信息)来确定可能是哪些用户阻塞了其他用户。[适用于6.5，7.0，2000] 试试下面的一个或多个有助于避免阻塞锁的建议:1)对于频繁使用的表使用集簇化的索引.2)设法避免一次性影响大量记录的T-SQL语句，特别是INSERT和UPDATE语句.3)设法让UPDATE和DELETE语句使用索引.4)使用嵌套事务时，避免提交和回退冲突。[适用于6.5，7.0，2000] 2009年最新愚人节整人绝招！2009年上半年全国计算

机等级考试参考答案请进入计算机考试论坛 2009年上半年全国计算机等级考试报名信息汇总 2009年NCRE考试有新变化 2009年全国计算机等级考试大纲 2009年上半年全国计算机二级考试试题及答案 2009年上半年全国计算机等级考试试题答案汇总 100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)