

省时又省力用Oracle扩展SQL跟踪数据 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/143/2021_2022__E7_9C_81_E6_97_B6_E5_8F_88_E7_c102_143000.htm 使用扩展SQL跟踪数据来了解是什么在耗费这么长的时间。假如有一天你开车去上班，但最后还是没能及时参加一个重要会议。你无法将你的革命性的想法呈现给客户，所以他们也不会采用。你的拖拖拉拉使你感到沮丧，你发誓决不再犯同样的错误。那么，为了不再发生类似情况，你怎么判断问题的原因呢？检查汽车外表是否有缺陷，因为外表有缺陷会使汽车的最高速度降低1%或更多。检查车轮定位，因为外倾角、后倾角或前束角不合适都会导致汽车的操纵不灵活并且耗费时间。检查发动机，以确保达到额定马力的99%或更高。如果不是这样，则要考虑重装或更换发动机。不，你可能不会采用这种检查方法；那样太可笑了。你可能会以完全不同的方式来判断问题之所在，可能只是问你自己一个简单的问题：什么事情让我花了这么长时间？从这个角度出发，问题就迎刃而解了。如果开车需要40分钟，而你在会议开始前20分钟才动身，那么下次就要提前30分钟动身。如果因为交通拥堵浪费了20分钟，那么，下次要么再早一些动身，换条路线，要么更仔细地查看早7点的路况报告。如果是你迷了路，结果浪费了20分钟去兜圈子，那么下次你大概就要事先看看地图。如此等等。我感到奇怪的是，那些擅长解决日常性能优化问题的数据库专业人员在工作中却使用完全不同的方法来解决数据库性能问题。许多数据库"调优人员"从来不问，"是什么让这个程序运行了这么长时间？"相反，他们会参考检查内容清单，并试

图阻止错误发生：检查所有Oracle块请求是否都由数据库缓存提供服务. 检查是否有全表扫描. 检查所有排序是否都在内存中进行. 检查重做日志是否与其他所有数据库文件进行了适当的隔离等等。对于某些工作来说，使用检查内容清单也许很好。但是对于判断性能问题这样的工作，试图确定理论上可能会出错的每一件事，从而对这个问题进行处理的做法的效率会很低。更有效的方法就是找到这个简单问题的答案：是什么花了这么长时间？用于优化Oracle程序的好的策略就如同日常生活中用到的策略。就像这样：1. 使用专门的仪器来测定程序的性能，从而监视运行速度慢的程序。2. 为运行慢的程序创建资源描述，把程序的响应时间细分为几种有用的类型。3. 通过首先处理响应时间最长的部分来缩短程序的响应时间。当你了解了若干技术细节之后，这个方法就非常简单了。如果你真的这样做，那么每次你都能获得一个有用的方法，久而久之，你将能在进行性能改进之前预知其结果。

跟踪 如果你有用于收集程序中每个执行步骤的时间统计信息的高级工具，那就用吧。但只收集汇总数据（如通过对系统全局区[SGA]或其基础共享存储段采样获得的数据）的工具对于某些类型的问题就不适合。使用昂贵的监控工具时最常见的汇总错误是它们会跨整个Oracle数据库实例来汇总某一给定时间间隔内资源的使用情况。但是，运行速度慢的程序实际上可能不受资源争用问题的影响，而这个问题却完全控制着系统中一些不太重要的程序的性能。即便是那些在Oracle数据库会话级上汇总信息的工具在诊断一些重要的问题类型时也有着缺陷。例如，假设一个程序运行10分钟，调用了10000次Oracle SQL*Net message from client 这一"等待事件"，

会话等待该事件的总用时为8.3分钟。这意味着会话对SQL*Net message from client事件的等待时间平均为3秒。但是单从汇总数据看，你无法知道这10000次调用是否每次都用了3秒，还是这些调用中也许有一个用了5分钟，而其余9999次调用每次只用0.02秒。这两种情况需要进行完全不同的处理。在这种情况下最能为你提供帮助的诊断数据是Oracle的扩展SQL跟踪数据。扩展SQL跟踪文件按时间顺序显示了Oracle数据库内核在指定时间内所完成工作的逐条记录。收集扩展SQL跟踪数据几乎是免费的。最大的花销是存储每一个需要引起注意的跟踪文件所需磁盘空间（很少超过几兆字节）的费用。跟踪自己的代码。如果能访问程序的源代码，则打开其扩展SQL跟踪就非常容易。首先必须确保会话的TIMED_STATISTICS和MAX_DUMP_FILE_SIZE参数设置正确：Code: [Copy to clipboard] alter session set timed_statistics=true. alter session set max_dump_file_size=unlimited. 如果没有设置TIMED_STATISTICS=TRUE，则数据库内核将把0值而不是真正的持续时间发送到跟踪文件中。如果对MAX_DUMP_FILE_SIZE严加限制，则会在跟踪文件中生成下面这样的消息，而不是你想要的时间数据：***DUMP FILE SIZE IS LIMITED TO 1048576 BYTES***接下来是激活跟踪。有几种方法可以采用。过去的方法是使用ALTER SESSION命令，如下所示：Code: [Copy to clipboard] alter session set events 10046 trace name context forever, level 12 /* code to be traced goes here */ alter session set events 10046 trace name context off 更好的方法是使用DBMS_SUPPORT包来激活扩展SQL跟踪：Code: [Copy to

```
clipboard] dbms_support.start_trace(waits=>true, binds=>true) /*  
code to be traced goes here */ dbms_support.stop_trace() 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