

2.1 优化论

这个标题有点投机取巧，没有深入研究理论知识之前，其实也谈不上“论”。所以，这里只是给出一些常规的优化步骤和手段，希望通过这些流程性的展示，让读者能够自己动手处理大部分常见的性能问题，从而达到本书的学习目的。

首先，在面对性能问题的时候，要足够冷静，不要马上去网上搜，“期望”快速解决问题。只有冷静下来分析问题，才能做出最好的分析和判断，从而快速解决问题。笔者见过有的DBA，在面对性能问题和其他服务器问题的时候，手忙脚乱，一下子蒙了，方向也没有了，局外人一看就知道的问题，他们竟然忽略了。所以，第一步是让自己冷静下来。

在你冷静下来之后，就开始进入分析阶段。通常情况下，我们会遇到两类性能问题。

- 1) 某个功能很慢或者突然变慢，比如某个存储过程、查询等。
- 2) 整个系统都很慢。

面对第一种情况，由于对象比较明确，所以处理起来相对轻松。大部分情况下，你只需要研究执行计划（将在第5章和第8章详细介绍）就可以解决绝大部分问题。这种情况下通过改写查询、改变表结构等操作，通常可以得到明显的效果。对于一些极端情况，就需要考虑更多的方面了，而不能仅从执行计划去分析。

如果面对的是第二种情况，由于对象不明确，因此让人无从下手，此时需要通过多种手段来寻找瓶颈，然后再进行处理。这个过程的难点就是定位瓶颈，由于性能是一个系统级别的事情，所以很多现象都是有关联的，有时甚至会造成误导，比如CPU使用率高，可能是由内存不足、语句不够优化、缺失索引等导致的，如果单纯定位到CPU存在压力，就要求升级CPU，是不合理的做法。面对这类问题，可以通过如性能计数器（第11章）、SQL Profiler/SQL Trace，甚至是SQL Server 2008出现的Extended Events等工具来收集性能方面的数据，或分析等待状态（第7章），然后进行全面、系统的汇总分析，最后定位瓶颈。根据经验来看，大部分这种类型的性能问题其瓶颈集中于低效的语句、不合理的设计、资源配置不合理及没有进行适当的维护等方面。

除了上述两种情况以外，还存在一些特例，本书也会适时对此进行一些介绍。

下面来看看图2-1。这个图展示了常规的性能优化步骤，适合上面的两种情况及部分极端情况。如果通过图中展示的步骤处理之后，还不能优化甚至还不能定位瓶颈，那么就只有两种可能了：要么只有微软才知道（毕竟外人不能知道太多内幕），要么就是无解的问题（经过微软的统计，在性能优化过程中，存在4%的可能性是运气因素，希望我们大家都不要遇到这4%）。

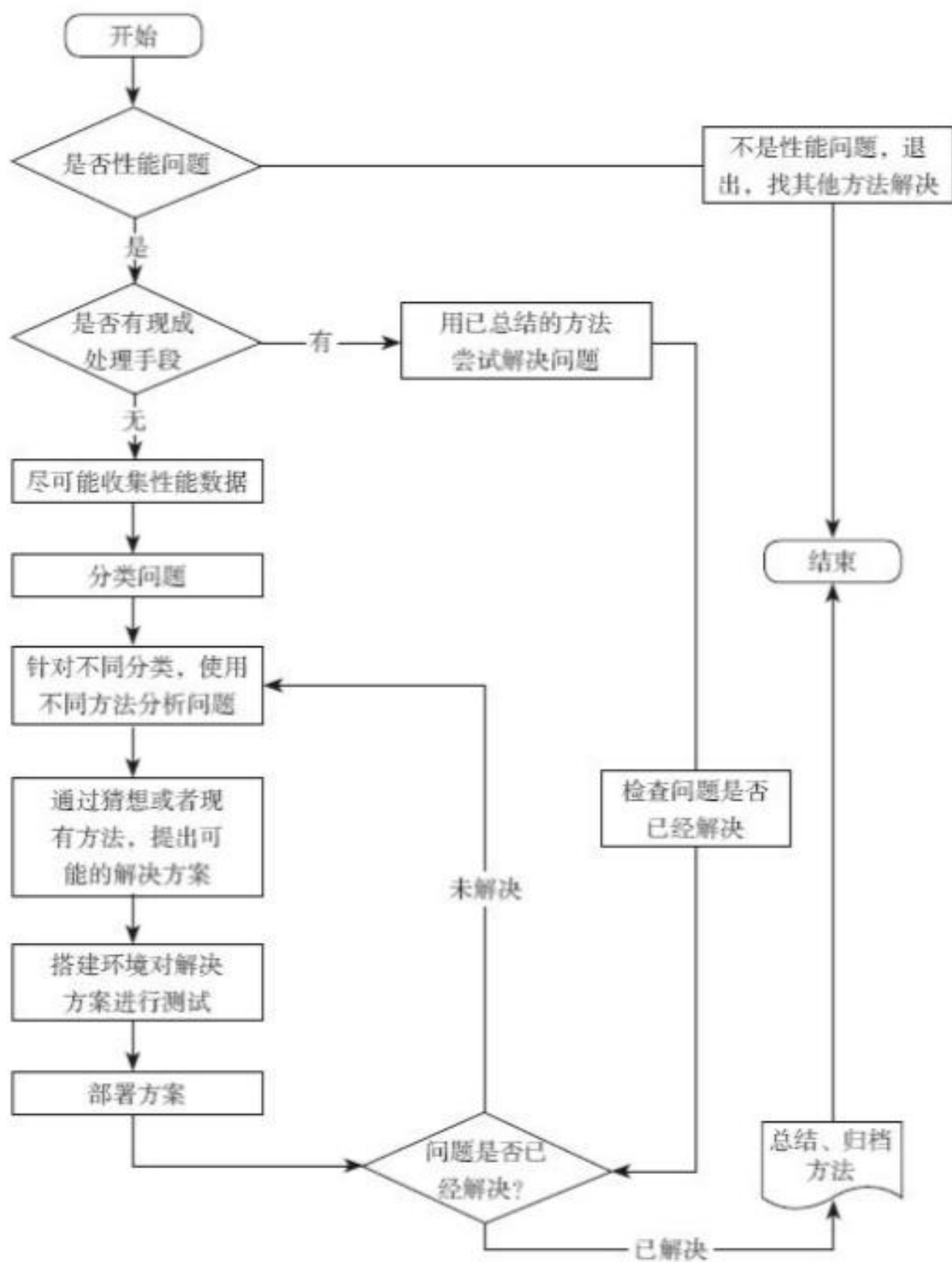


图2-1 性能优化步骤