



▲ 关系型数据库是如何运作的(上)

12

▼

数据库 (<http://www.csdn.net/tag/数据库/news>)

运算 (<http://www.csdn.net/tag/运算/news>)

合并 (<http://www.csdn.net/tag/合并/news>)



一说到关系型数据库，我总感觉缺了点什么。如果你尝试透过“关系型数据库是如何运作的”的关键词句来进行搜索，其搜索结果是少量的而且内容是简短的。难道说是由于它已经太老旧而已经不再流行吗？

作为一名开发者，我讨厌使用我不明白的技术。此外，关系型数据库已经使用超40年，肯定有它过人的原因。因此，我花了大量时间来想真正弄懂它里面如同黑盒子那样的奥秘。关系型数据库实际上是非常有趣的，因为它是基于实用和复用的概念。但是限于篇幅，以下我将把重点放在数据库如何处理SQL查询的问题上。本文内容大致划分为以下三部分：

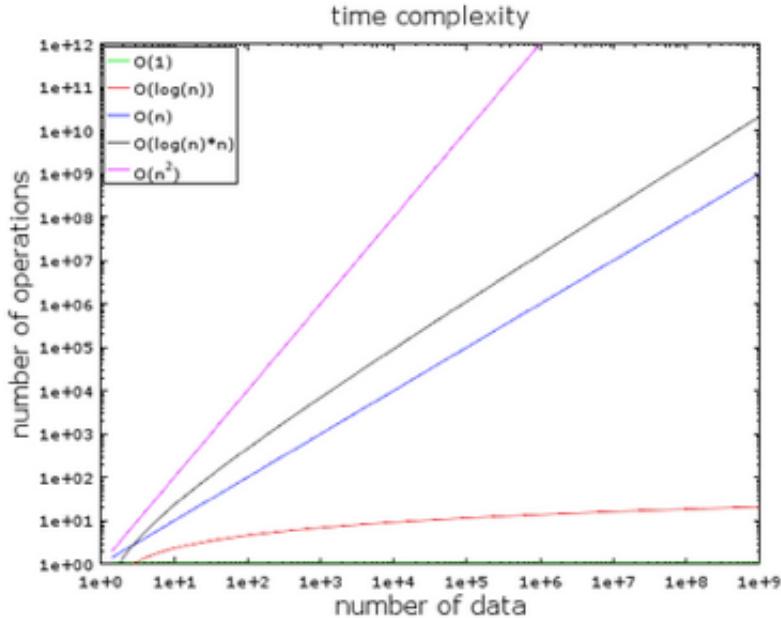
- 1.低阶数据库和高级数据库组成概述
- 2.查询优化流程的处理概述
- 3.事务和缓冲池管理概述

## 基本概念回顾

在编程年代早期，开发者是必须要理解清楚自己所进行操作的原理的。他们对于所使用的算法和数据结果是了然于胸的，因为他们很注重在计算机配置较低时于CPU和内存上的开销。在这一节，我首先要介绍的是数据库索引。

O(1) vs O(n2)

时间复杂度用于计算算法处理数据的用时。科学家使用大O表示法来进行时间复杂度描述，其定义是对于输入的数据算法需要进行多少步运算。这里要强调的是，它的核心是数据量增加对运算增加的影响而不是数据量的多少。时间复杂度不会直接给出精确的运算步数，而是以趋势的方式展示。



在上图中，你可以看到不同复杂度的发展趋势，我使用的方法是对数法。换言之，数据量将会从1快速地增加到10亿。我们可以得出以下结论：

- $O(1)$ 或常数复杂度是维持不变的
- $O(\log(n))$ 在处理10亿数据量时也维持与一个较低复杂度水平
- $O(n^2)$ 复杂度增长最快
- 其余两种复杂度位于中游

## 举例说明

如果是处理少量数据， $O(1)$ 和 $O(n^2)$ 的差别是不明显的。例如是2000个运算元素：

- $O(1)$  的运算量是1
- $O(\log(n))$  的运算量是7
- $O(n)$  的运算量是2000
- $O(n*\log(n))$  的运算量是14000
- $O(n^2)$  的运算量是4 000 000

尽管 $O(1)$  和  $O(n^2)$ 的运算量的差距是4百万，但是这仅需2ms，也就是眨眼的功夫。此外，如果使用的是多核处理器，其运算速度会更快。所以性能和优化问题在现在的重视程度无法跟以往相比。

如果处理的数据量是1 000 000，其结果又会如何呢？

- $O(1)$  的运算量是1
- $O(\log(n))$  的运算量是14

- $O(n)$  的运算量是1 000 000
- $O(n \log(n))$  的运算量是14 000 000
- $O(n^2)$  的运算量是1 000 000 000 000

这样一来，你可以先喝杯咖啡休息下再回来看结果了！如果再加个0，你可以先进行午休了！

## 进一步说明

这里有几点提示：

在一个完整hash表中进行一次搜索会提交一个元素给 $O(1)$

在一个全平衡树种进行一次搜索会提交一个结果给 $O(\log(n))$

在一个数组中进行一次搜索会提交一个结果给 $O(n)$

最优排序算法的时间复杂度与 $O(n \log(n))$ 相当

低效排序算法的时间复杂度与 $O(n^2)$ 相当

注意：具体算法和数据结果会在本文稍后列示

时间复杂度的类型有：

- 平均事件场合
- 最佳时间场合
- 最差时间场合

时间复杂度通常是最差时间场合。除了时间复杂度，复杂度还可以用来表示内存使用和磁碟I/O占用情况等。诚然，比 $n^2$ 更复杂的计算有 $n^4$ ,  $3n$ ,  $nn$ 。

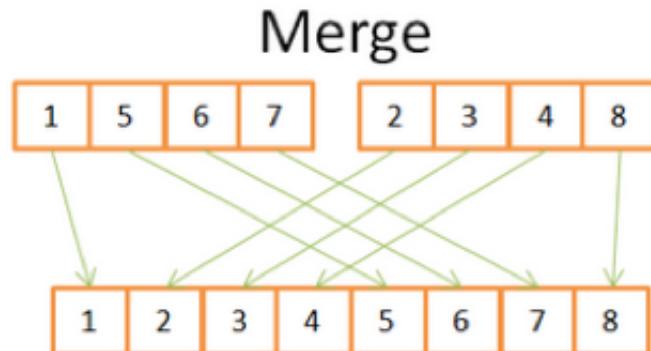
## 合并排序

如果你要对一个集合进行排序该如何做呢？什么？使用sort()？听起来是个好的答案。

但如果排序对象是一个数据库，你就务必知道sort()的工作原理。这里我介绍排序算法中最重要的一种：合并排序。对合并排序理解透彻，一方面可以掌握如何进行查询优化，二来可以更好地理解本文稍后说到的合并join运算。

## 合并(Merge)

合并排序的运算过程是：合并两个已排序的 $N/2$ 数组到一个已排序 $N$ 个元素数组，例如下图所示：



以上是本系列文的上篇，更多内容请关注后续文章，后续内容简述：全局概念，客户管理器，查询管理器，数据管理器。

英 文 链 接 : <http://coding-geek.com/how-databases-work/> (<http://coding-geek.com/how-databases-work/>)

(译者/伍昆 责编/夏梦竹)



(<http://geek.csdn.net/user/publishlist/rnifeeasy>)

**rnifeeasy** (<http://geek.csdn.net/user/publishlist/rnifeeasy>)

发布于 数据科学 (<http://geek.csdn.net/forum/125>) 2016-01-22 15:36