

数据库的使用你可能忽略了这些

原创：IT米粉 IT米粉 2017-11-01

数据库的管理是一个非常专业的事情，对数据库的调优、监控一般是由数据库工程师完成，但是开发人员也经常与数据库打交道，即使是简单的增删改查也是有很多窍门，这里，一起来聊聊数据库中很容易忽略的问题。

字段长度省着用

先说说我们常用的类型的存储长度：

列类型	存储长度
tinyint	1字节
smallint	2字节
int	4字节
bigint	8字节
float	4字节
decimal(m,d)	0-4字节
datetime	8字节
timestamp	4字节
char(m)	m个字节

列类型	存储长度
varchar(m)	可变长度
text	可变长度

很明显，不同的类型存储的长度有很大区别的，对查询的效率有影响，字段长度对索引的影响是很大的。

- 字符串字段长度都差不多的，可以预估长度的，用char
- 字符串长度差异大，用varchar,限制长度，不要浪费空间
- 整型根据大小，选择合适的类型
- 时间建议用timestamp
- 建议使用decimal，不建议使用float,如果是价格，可以考虑用int或bigint，如1元，存储的就是100

放弃uuid(guid)的使用

不管是uuid，还是guid，使用的时候都是为了避免同时生成重复的ID，但是建议考虑其他方案，原因如下：

- uuid没有顺序
- uuid太长
- uuid规则完全不可控

推荐的方案用bigint（首选），或者char来存储，生成方式参考snowflake的算法，有顺序、长度固定、比uuid更短，当然，也几乎不会重复。

大表减少联表，最好是单表查询

单表查询的优势很多，查询效率极高，便于分表分库扩展，但是很多时候大家都觉得真正实现起来不太现实，完全失去了关系数据库的意义，但是单表的性能优势太明显，一般总会有办法解决的：

- 合理的冗余字段
- 配合内存数据库（redis\mongodb）使用
- 联表变多次查询（下文会有说明）

如果考虑都后期数据量大，需要分表分库，就应该尽早实时单表查询，现在的数据库分表分库的中间件基本都无法支持联表查询。即使如mycat最多支持两个表的联表查询，但是也有很明显的性能损耗。

索引的正确处理方式

索引的优势这里就不多说了，索引使用不当会有反效果：

- 数据量很小的表，不需要索引
- 一个表的索引不宜过多，建议最多就5个，索引不可能满足所有的场景，但是了个满足绝大部分的场景
- mysql 和 sqlserver的索引差别还挺大的，需要注意。例如：mysql索引字段的顺序对性能有很大影响，sqlserver优化过，影响很小

多查几次比联表可能要好

提出这个方案相信会得到很多人的反对，但是我相信这个结论还是非常适合数据量大的场景。多查几次数据库有这么几个弊端：

- 增加了网络消耗
- 增加了数据库的连接数

其实，这两个问题在现在基本都可以忽略的，数据库和应用的连接基本都是内网，这个网络连接的效率还是很高的。数据库对连接池的优化已经比较成熟了，连接数只要不是太多，影响也不会太严重，但是多查几次的优势却很多：

- 单表效率更高
- 便于后期扩展分表分库库
- 有效利用数据库本身的结果缓存

- 减少锁表，联表会锁多个表

当然，多查几次这个度一定要把握。千万不要在一个循环里面查询数据库。我们也应该尽量减少查询数据库的次数。我们可以接受1次查询变2次查询，如果你变成10次查询，那就要放弃了。

举个例子：

查询商品的时候，需要显示分类表的分类名

```
select category.name,product.name from product inner join category on p.categoryid=category.id
```

建议的方式：

```
select categoryid,name from product  
select categoryname from category where categoryid in ('','''','''')
```

当然，你可以再优化一下，查询分类名之前，对product的categoryid排序一下，这样速度更快。因为我们前面已经用snowflake生成了有顺序的主键了。补充一下，in的效率并不是你想象的那么慢，如果保持在100个节点（很多书籍介绍1000个节点，我们保守一点），性能还是很高的。

尽量使用简单的数据库脚本

很多用过 .net Entity Framework 的人都说这个框架太慢，其实慢主要是两点：错误的使用延迟加载（外键关联）、生成SQL编译太慢。Entity Framework生成的SQL脚本有太多没用的东西，导致编译太慢。

数据库脚本尽量使用简单的，不要用太长的一个SQL脚本，会导致初次执行的时候，编译SQL脚本花费太多的时间。

尽量去避免聚合操作

聚合操作如count,group等，是数据库性能的大杀手，经常会出现大面积的表扫描和索表的情况，所以大家能看到很多平台都把数量的计算给隐藏了，商品查询不去实时显示count的结果。如淘宝，就不显示查询结果的数量，只是显示前100页。

避免聚合操作的方法就是将实时的count计算结果用字段去存储，去累加这个结果。当然，也可以考虑用spark等实时计算框架去处理，这种高深的技术，不在此次讨论范围内。（PS:主要是我也不懂）

总结

程序的优化很多时候都是一些细节的问题，更应该注意平时的积累，阿里SQL的规范有很多可以吸取的地方，以上也是自己工作中的一些总结。

（完）

欢迎大家关注我的公众号交流、学习、第一时间获取最新的文章。

微信号：itmifen



