

# MySQL 慢查询日志

作者: [zouchanglin](#)

原文链接: <https://ld246.com/article/1571233408416>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

 <https://b3logfile.com/bing/20181115.jpg?imageView2/1/w/960/h/540/interlace/1/q/100>

## 慢查询日志

## 什么是慢查询日志

MySQL 的慢查询日志是 MySQL 提供的一种日志记录，它用来记录在 MySQL 中响应时间超过值的语句，具体指运行时间超过 long\_query\_time 值的 SQL，则会被记录到慢查询日志中

具体指运行时间超过 long\_query\_time 值的 SQL，则会被记录到慢查询日志中。long\_query\_time 的默认值为 10，意思是运行 10 秒以上的语句


由它来查看哪些 SQL 超出了我们的最大忍耐时间值，比如一条 SQL 执行超过 5 秒钟，我们就算 SQL，希望能收集超过 5 秒的 SQL，结合之前 explain 进行全面分析

默认情况下，MySQL 数据库没有开启慢查询日志，需要我们手动来设置这个参数。当然，如果是调优需要的话，一般不建议启动该参数，因为开启慢查询日志会或多或少带来一定的性能影响。慢查询日志支持将日志记录写入文件！

## 如何开启慢查询

查看开启状态


```
SHOW VARIABLES LIKE '%slow_query_log%'
```

 <https://b3logfile.com/file/2019/10/mysql01-ee66e8fb.png?imageView2/2/interlace/1/format/jpg>

开启慢查询

```
set global slow_query_log = 1
```

使用 set global slow\_query\_log = 1 开启了慢查询日志只对当前数据库生，如果 MYSQL 重启则会失效。

 <https://b3logfile.com/file/2019/10/mysql02-30d35e55.png?imageView2/2/interlace/1/format/jpg>

如果要永久生效，就必须修改配置文件 my.cnf（其它系统变量也是如此）修改 my.cnf 文件，[mysqld]下增加或修改参数

slow\_query\_log 和 slow\_query\_log\_file 后，然后重启 MySQL 服务器。也即将如下两行配置进 my.cnf 文件

```
slow_query_log = 1
slow_query_log_file = /var/lib/mysql/tim-slow.log
```

关于慢查询的参数 slow\_query\_log\_file，它指定慢查询日志文件的存放路径，系统默认会给一个省的文件 host\_name-slow.log（如果没有指定参数 slow\_query\_log\_file 的话）

那么开启慢查询日志后，什么样的 SQL 参会记录到慢查询里面？

通过 show variables like 'long\_query\_time%' 来查看默认时间长度，单位是秒：

 <https://b3logfile.com/file/2019/10/mysql03.png?imageView2/2/interlace/1/format/jpg>

s://b3logfile.com/file/2019/10/mysql03-5006b3f6.png?imageView2/2/interlace/1/format/jpg  
> </p>

<p>同样的，可以使用命令修改，也可以在 my.cnf 里面配置。假如运行时间正好等于 long\_query\_time 的情况，并不会被记录下来。也就是说，在 MySQL 源码里是判断大于 long\_query\_time，而非于等于!</p>

<p>设置记录的阈值： </p>

```
<pre> <code class="language-sql highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl"> <span class="highlight-k">set</span> <span class="highlight-w"> </span> <span class="highlight-k">global</span> <span class="highlight-w"> </span> <span class="highlight-n">long_query_time</span> <span class="highlight-o">=</span> <span class="highlight-mi">3</span> <span class="highlight-p">;</span> <span class="highlight-w"> </span> </span> </pre>
```

<p> </p>

<p>设置了但是还是没有发生更改？为什么？此时需要重新开启一个会话才可以： </p>

<p> </p>

<p>接下来实行一个较慢的查询，如下图，但是记得要在配置文件中做如下配置： </p>

<p> </p>

<p>接着去日志文件中查看存在哪些超过阈值的 SQL 就好了： </p>

<p> </p>

<p>查询当前系统中有多少条慢查询记录： </p>

<p> </p>

<p>记载一下我的配置文件</p>

```
<pre> <code class="language-ini highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl"> <span class="highlight-na">slow_query_log</span> <span class="highlight-o">=</span> <span class="highlight-s">1</span> </span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="highlight-na">slow_query_log_file</span> <span class="highlight-o">=</span> <span class="highlight-s">/var/lib/mysql/tim-slow.log</span> </span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="highlight-na">long_query_time</span> <span class="highlight-o">=</span> <span class="highlight-s">3</span> </span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="highlight-na">log_output</span> <span class="highlight-o">=</span> <span class="highlight-s">FILE</span> </span> </span> </code> </pre>
```

<h2 id="日志分析工具mysqldumpshow">日志分析工具 mysqldumpshow</h2>

<p> </p>

<p>在生产环境中，如果要手工分析日志，查找、分析 SQL，显然是个体力活，MYSQL 提供了日志析工具 mysqldumpshow</p>

<ul>

<li>s:是表示按何种方式排序</li>

<li>c:访问次数</li>

- <li>l:锁定时间</li>
- <li>r:返回记录</li>
- <li>t:查询时间</li>
- <li>al:平均锁定时间</li>
- <li>ar:平均返回记录数</li>
- <li>at:平均查询时间</li>
- <li>t:即为返回前面多少条的数据</li>
- <li>g:后边搭配一个正则匹配模式，大小写不敏感的</li>

<p>下面是使用示例: </p>

<p>得到返回记录集最多的 10 个 SQL</p>

```
<pre> <code class="language-bash highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl">mysqldumpslow -s r -t <span class="highlight-m">10</span> /var/lib/mysql/tim-slowlog</span> </span> </code> </pre>
```

<p>得到访问次数最多的 10 个 SQL</p>

```
<pre> <code class="language-bash highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl">mysqldumpslow -s c-t <span class="highlight-m">10</span> /var/lib/mysql/tim-slow log</span> </span> </code> </pre>
```

<p>得到按照时间排序的前 10 条里面含有左连接的查询语句</p>

```
<pre> <code class="language-bash highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl">mysqldumpslow -s t -t <span class="highlight-m">10</span> -g <span class="highlight-s2">"left join"</span> /var/lib/mysql/tim-slowlog</span> </span> </code> </pre>
```

<p>另外建议在使用这些命令时结合和 more 使用，否则有可能出现爆屏情况</p>

```
<pre> <code class="language-bash highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl">mysqldumpslow -s r -t <span class="highlight-m">10</span> /var/lib/mysql/tim-slow.log <span class="highlight-p">|</span> more</span> </span> </code> </pre>
```