



链滴

数据库优化 - 实例优化

作者: [jianzh5](#)

原文链接: <https://ld246.com/article/1574040337731>

来源网站: [链滴](#)

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

<p></p>

<p>从网上去搜数据库优化基本都是从 SQL 层次进行优化的，很少有提及到数据库本身的实例优化就算有也都是基于某个特定数据库的实例优化，本文涵盖目前市面上所有主流数据库的实例优化（Oracle、MySQL、POSTGRES、达梦），按照文章的配置能够将你数据库性能用到 80% 或以上。</p>

<p>这部分为理论知识，不感兴趣的同学可以直接跳到后面参数配置部分。</p> <p>根据角色的不同，数据库优化分为以下几个目标：</p> 业务角度（关键用户）：
 减少用户页面响应时间 数据库角度（开发）：
 减少数据库 SQL 响应时间 数据库服务器角度（运维）：
 充分使用数据库服务器物理资源
 减少数据库服务器 CPU 使用率
 减少数据库服务器 IO 使用率
 减少数据库服务器内存使用率 SQL 平均响应时间变短 优化前：数据库平均响应时间 500ms 优化目标：数据库平均响应时间 200ms 数据库服务器 CPU 占用率变少 优化前：数据库高峰期 CPU 使用率 70% 优化目标：数据库高峰期 CPU 使用率 50% 数据库服务器 IO 使用率变低 优化前：数据库 IO WAIT 为 30% 优化目标：数据库 IO WAIT 低于 10% <p>在进行数据库优化的时候可能会有以下几个误区：</p> 优化之前一定要深入了解数据库内部原理
 优化是有“套路”的，照着这些“套路”你也可以很好的完成数据库优化 不断调整数据库参数就可以最终实现优化
 有时候设计不合理怎么调整参数都不行 不断调整操作系统参数就可以最终实现优化
 同上 数据库性能由应用、数据库架构决定，与应用开发关系不大
 恰恰相反，应用开发的关系很大 必须要做读写分离，必须要弄分库分表
 原文链接：[数据库优化 - 实例优化](#)

数据量级只有达到一定的比例才有必要做读写分离，分表分库，否则徒增复杂度。一般来说 Oracle 单表量级可以达到 1 亿，MySQL 到 1000 万~2000 万

数据库优化流程

完整的数据库优化流程如下：

 <https://b3ogfile.com/file/2019/11/image-05fdb9c5.png?imageView2/2/interlace/1/format/jpg>

首先需要尽可能的了解优化问题，收集问题期间系统信息并做好存档。根据当前系统问题表现制定优化目标并与客户沟通目标达成一致；通过一系列工具分析系统问题，制定优化方案，方案评审完成由各负责人员进行实施。若达到优化目标则编写优化报告，否则需要重新制定优化方案。

数据库实例优化

数据库实例优化遵循三句口诀：**日志不能小、缓存足够大、连接要够用。**

数据库事务提交后需要将事务对数据页的修改刷（fsync）到磁盘上，才能保证数据的持久性。个刷盘，是一个随机写，性能较低，如果每次事务提交都要刷盘，会极大影响数据库的性能。数据库架构设计中都会采用如下两个优化手法：

- 先将事务写到日志文件 RedoLog（WAL），将随机写优化成顺序写

- 加一层缓存结构 Buffer，将每次写优化成顺序写

所以日志跟缓存对数据库实例尤其重要。而连接如果不够用，数据库会直接抛出异常，系统无法问。

数据库参数优化

主流数据库架构都有如下的共同点：

- 数据缓存

- SQL 解析区

- 排序内存

- REDO 及 UNDO

- 锁、LATCH、MUTEX

- 监听及连接

- 文件读写性能

接下来我们根据不同的数据库调整参数以使数据库达到最佳性能。

ORACLE

||
||
||

参数分类

参数名

参数值

备注

||
||
||

--

数据缓存

SGA_TAGET、MEMORY_TARGET

物理内存 70-80%

越大越好

--

数据缓存

DB_CACHE_SIZE

<td>物理内存 70-80%</td>
<td>越大越好</td>
</tr>
<td>SQL 解析</td>
<td>SHARED_POOL_SIZE</td>
<td>4-16G</td>
<td>不建议设置过大</td>
</tr>
<td>监听及连接</td>
<td>PROCESSES、SESSIONS、OPEN_CURSORS</td>
<td>根据业务需求设置</td>
<td>一般为业务预估连接数的 120%</td>
</tr>
<td>其他</td>
<td>SESSION_CACHED_CURSORS</td>
<td>大于 200</td>
<td>软软解析</td>
</tr>

</tbody>
</table>
<h4 id="MYSQL-INNODB-">MYSQL (INNODB) </h4>
<table>
<thead>
<tr>
<th>参数分类</th>
<th>参数名</th>
<th>参数值</th>
<th>备注</th>
</tr>
</thead>
<tbody>
<tr>
<td>数据缓存</td>
<td>INNODB_BUFFER_POOL_SIZE</td>
<td>物理内存 50-80%</td>
<td>一般来说越大性能越好</td>
</tr>
<tr>
<td>日志相关</td>
<td>Innodb_log_buffer_size</td>
<td>16-32M</td>
<td>根据运行情况调整</td>
</tr>
<tr>
<td>日志相关</td>
<td>sync_binlog</td>
<td>1、100、0</td>
<td>1 安全性最好</td>
</tr>
<tr>
<td>监听及连接</td>

max_connections	根据业务情况调整	可以预留一部分值
-----------------	----------	----------

innodb_flush_log_at_trx_commit	2	安全和性能的折中考虑
--------------------------------	---	------------

其他	wait_timeout, interactive_timeout	28800	避免应用连接定时中断
----	-----------------------------------	-------	------------

POSTGRES

参数分类	参数名	参数值	备注
数据缓存	SHARED_BUFFERS	物理内存 10-25%	
数据缓存	CACHE_BUFFER_SIZE	物理内存 50-60%	
日志相关	wal_buffer	8-64M	不建议设置过大过小
监听及连接	max_connections	根据业务情况调整	一般为业务预估连接数的 120%

其他	maintenance_work_mem	512M 或更大	
其他	work_mem	8-16M	原始配置 1M 过小
其他	checkpoint_segments	32 或者更大	

达梦数据库

参数分类	参数名	参数值	备注
数据缓存	MEMROY_TARGET、MEMROY_POOL	物理内存 90%	
数据缓存	BUFFER	物理内存 60%	数据缓存
数据缓存	MAX_BUFFER	物理内存 70%	最大数据缓存
监听及连接	max_sessions	根据业务需求设置	一般为业务预估连接数的 120%

</tbody>

</table>

<h2 id="总结">总结</h2>

<p>数据库的优化手法太多太多，有换磁盘阵列升级硬件，有改写 SQL 脚本添加索引，还有数据库数调整优化性能，甚至还可以调整数据库架构。本文从数据库本身参数进行调优，大家根据上面几张中的参数进行调整基本能达到数据库最佳性能的 80%。</p>