



链滴

记一次对达梦数据库的优化过程

作者: [jianzh5](#)

原文链接: <https://ld246.com/article/1574159839794>

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>某年某月某日的一个下午，接收到监控服务器的一条告警短信：
尊敬的运维工程师 XX，你好：

“192.168.136.200”数据库服务器 CPU 异常，CPU 使用率 98.7%，请尽快处理。
看到这个消息浑身一紧，赶紧掐灭手中的烟，跑回办公室。</p>

<p>以上段子纯属捏造，如有雷同，我反正是不会改。</p>

<p>言归正传，本文是记录一次对达梦数据库的优化过程。</p>

<p>处理问题的第一步，是需要了解当前服务器的状况，我们通过以下两种手段确认服务器瓶颈。</p>

<h3 id="系统状况">系统状况</h3>

<p>通过服务器性能监控大盘观察当前系统性能

</p>

<p>通过上图我们看出 CPU 基本耗尽，IO 飙升。</p>

<p>通过 sar 命令观察服务器实时状态

<code>sar 10 3</code>

</p>

<p>确认 CPU 被耗满，没有空闲。</p>

<p>通过我的细致观察，发现服务器 CPU 被耗满。接下来需要查看数据库服务器的配置参数是否合理，是否有慢查询脚本。</p>

<h3 id="参数优化">参数优化</h3>

<p>查看 dm 配置文件

<code>cd /dm7/dmdbms/devdb cat dm.ini | grep -E "MEMORY_POOL|MEMORY_TARGET|BUFFER"</code>

</p>

<p>发现数据库参数配置为安装时候的默认配置，参数不合理，需要优化参数配置。</p>

<p>备份原配置文件

<code>cp dm.ini dm.ini.bak</code></p>

<p>修改配置

修改如下几个关键参数，根据之前文章<a href="https://ld246.com/forward?goto=https%3A%2Fmp.weixin.qq.com%2Fs%3F_biz%3DMzg3NjE0ODM2NA%3D%3D%26mid%3D224748376%26idx%3D1%26sn%3Dfe5fbf12670ffd06149b26edc06026fd%26chksm%3Dcf37e277f8406b162eec10e94d51a662eea571627f3dc45bcc60ce9c589eec15512dc017fdb%26mpshare%3D1%26scene%3D1%26srcid%3D1030jwdKZfrkBFGsqcYEwGR1%26sharer_sharetime%3D157244352467%26sharer_shareid%3Df4a0969410a6220c2924a1da2e3895da%26key%3D96490f73691e5d5cd21411e09fa82e605939bf4459edee0183ccfb038dabc3714729a23138af597bc82b87379966884e55b24c8f5c2d79a2517b9b4890ffdbdd84d74c16f3c224c2e6a9e28623dfd5%26scene%3D1%26uin%3DMTQ4ODgxOTEyOA%253D%253D%26devicetype%3DWindows%2B10%26version%3D62070152%26lang%3Dzh_CN%26pass_ticket%3DTfixyow%252FySFwy6RobtF7Vg2i%2

2BnJstXTK1v3TkWFL%252BkjrfoRyhYfXVo32uzEcrWf%252B" target="_blank" rel="nofollow u c">数据库优化-实例优化中的表格进行优化 (ps: 当前数据库内存 2G) </p>

<table>
<thead>
<tr>
<th>参数</th>
<th>优化建议</th>
<th>优化后的值, 单位 M</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEMORY_POOL</td>
<td>建议为内存的 90%</td>
<td>1800</td>
</tr>
<tr>
<td>MEMORY_TARGET</td>
<td>建议为内存的 90%</td>
<td>1800</td>
</tr>
<tr>
<td>BUFFER</td>
<td>建议为内存的 60%</td>
<td>1200</td>
</tr>
<tr>
<td>MAX_BUFFER</td>
<td>建议为内存的 70%</td>
<td>1400</td>
</tr>
<tr>
<td>MAX_SESSIONS</td>
<td></td>
<td>1000</td>
</tr>
</tbody>
</table>

重启数据库服务

<code>service DmServerdm restart</code>

<p>参数优化后我们尝试找出当前数据库存在的慢查询 SQL, 看看是否可以优化。 </p>
<h3 id="慢-SQL-优化">慢 SQL 优化</h3>
<p>达梦数据库不像 MySQL 可以直接将慢查询存放在指定位置, 达梦需要通过 AWR 报告中找出慢查询。(AWR 报告大家自行百度)

启用 DM 快照需要调用 DBMS_WORKLOAD_REPOSITORY 包。 </p>

<p>使用 DBA 账户登录数据库

<code>disql SYSDBA/password</code> </p>

-
<p>创建 DBMS_WORKLOAD_REPOSITORY 系统包, 开启 AWR 快照功能。

<code>SP_INIT_AWR_SYS(1);</code> </p>

<p>启用状态检测。

<code>SELECT SF_CHECK_AWR_SYS;</code> </p>

<p>设置 AWR 快照间隔时间 (30 分钟)

<code>CALL DBMS_WORKLOAD_REPOSITORY.AWR_SET_INTERVAL(30);</code> </p>

<p>手动创建快照:

<code>DBMS_WORKLOAD_REPOSITORY.CREATE_SNAPSHOT();</code>

这里我们可以间隔几分钟多执行几遍创建几个不同的快照。 </p>

<p>查看创建的快照信息, 包括快照 id:

<code>SELECT * FROM SYS.WRM\$_SNAPSHOT;</code>

 </p>

<p>查看 AWR 报告内容

<code>SELECT * FROM TABLE (DBMS_WORKLOAD_REPOSITORY.AWR_REPORT_HTML(1,2));</code>

查看 snapshot 的 id 在 1~2 范围内的 AWR 分析报告的带 html 格式的内容。

 </p>
<p>这个内容格式基本没办法看, 我们需要将其转化成 html 页面查看。 </p>

<p>生成 HTML 文件(需要先对 awr 文件夹授权) </p>
<pre> <code class="highlight-chroma"> chmod 777 /awr
 SYS.AWR_REPORT
HTML(1,2,'/awr','awr1.html');
 </code> </pre>

<p>通过 AWR 报告找出慢 SQL

 </p>
<p>SQL Ordered by Elapsed Time 的内容就是慢查询语句。

在拿到慢查询语句后我们需要联系开发人员修改查询语句, 这次优化过程中我通过给相关字段添加索引, 改写一部分 SQL 完成。 </p>
<p>但是数据表本身设计不合理这个没有优化, 由于设计不合理导致查询没办法走索引; 而有些查询则需要从业务角度进行优化, 比如是否有必要对大表进行全表查询然后再排序? 等等。。。(至于数据库 SQL 优化的具体策略我们下期再聊) </p>

<p>在完成优化后重启应用, 再次通过 <code>sar 10 3</code> 观察 CPU 性能, 较优化前还是有少的提升的, 又可以抽空去抽根烟了。

 </p>