链滴

Redis 高级应用 -- 主从复制

作者：GaoWentian
原文链接：https://ld246.com/article/1576595846983
来源网站：链滴

<h3 id="简介">简介</h3>
<h4 id="高并发">高并发</h4>
互联网架构追求高并发，高性能和高可用。其中高并发指允许大量用户同时访问，高可用指服务宕机时间少。业界高可用目标为 5 个 9，即可用性达到 99.999%，也就是说服务器年宕机时长低于 35 秒，计算公式为：（31536000-宕机时间）/ 31536000 * 100%　（31536000 = 365 * 24* 60 * 60 =1 年）
<h4 id="主从复制作用">主从复制作用</h4>
1、读写分离：master 写，slave 读，提高服务器的读写负载能力
2、负载均衡：基于主从结构，配合读写分离，由 slave 分担 master 负载，并根据需求的变化，改变 lave 的数量，通过多个从节点分担数据读取负载，由 slave 提供服务，实现快速的故障恢复
3、故障恢复：当 master 出现问题时，由 slave 提供服务，实现快速的故障恢复
4、数据冗余：实现数据热备份，是持久化之外的一种数据冗余方式
5、高可用基石：基于主从复制，构建哨兵模式与集群，实现 Redis 的高可用方案
<h3 id="主从复制工作流程">主从复制工作流程</h3>
主从复制可分为三个阶段：

建立连接阶段
数据同步阶段
命令传播阶段

<h3 id="建立连接阶段工作流程">建立连接阶段工作流程</h3>
<table>
<thead>
<tr>
<th>master</th>
<th>slave</th>
</tr>
</thead>
<tbody>
<tr>
<td>/</td>
<td>1、发送指令：slaveof ip port</td>
</tr>
<tr>
<td>2、接收指令，响应 slave</td>
<td>/</td>
</tr>
<tr>
<td>/</td>
<td>3、保存 master 的 ip 和端口</td>
</tr>
<tr>
<td>/</td>
<td>4、根据保存的信息创建连接 master 的 socket</td>
</tr>
<tr>
<td>/</td>
<td>5、周期性发送命令：ping</td>
</tr>
<tr>
<td>6、响应 pong</td>
<td>/</td>
</tr>
<tr>

<td>/</td>
<td>7、发送指令：auth password</td>
</tr>
<tr>
<td>8、验证授权</td>
<td>/</td>
</tr>
<tr>
<td>/</td>
<td>9、发送指令：replconflistening-port </td>
</tr>
<tr>
<td>10、保存 slave 的端口号</td>
<td>/</td>
</tr>
</tbody>
</table>
<h4 id="主从连接指令-slave使用-">主从连接指令（slave 使用）</h4>
方式一：客户端发送命令
<pre><code class="highlight-chroma">slaveof <masterip> <masterport>
</code></pre>
方式二：启动 slave 服务器时加参数
<pre><code class="highlight-chroma">redis-server -slaveof <masterip> <masterport>
</code></pre>
方式三：服务器配置(保存在 conf 文件中)
<pre><code class="highlight-chroma">slaveof <masterip> <masterport>
</code></pre>
<h4 id="主从断开连接-slave使用-">主从断开连接（slave 使用）</h4>
slave 客户端发送命令：
<pre><code class="highlight-chroma">slaveof no one
</code></pre>
<h3 id="数据同步阶段工作流程">数据同步阶段工作流程</h3>
<table>
<thead>
<tr>
<th>master</th>
<th>slave</th>
</tr>
</thead>
<tbody>
<tr>
<td>/</td>
<td>1、发送指令：psync2</td>
</tr>
<tr>
<td>2、执行 bgsave</td>
<td>/</td>
</tr>
<tr>
<td>3、第一个 slave 连接时，创建复制缓冲区（复制缓存区中保存生成 RDB 文件时 master 服务

执行的命令）</td>
<td>/</td>
</tr>
<tr>
<td>4、生成 RDB 文件，通过 socket 发送给 slave</td>
<td>/</td>
</tr>
<tr>
<td>/</td>
<td>5、接收 RDB 文件，清空数据，执行 RDB 文件恢复过程</td>
</tr>
<tr>
<td>/</td>
<td>6、发送命令告知 RDB 恢复已经完成</td>
</tr>
<tr>
<td>7、发送复制缓冲区数据</td>
<td>/</td>
</tr>
<tr>
<td>/</td>
<td>8、接收信息，执行 bgrewriteaof,恢复数据</td>
</tr>
</tbody>
</table>
数据同步阶段，1-5 称为全量复制，使用 RDB 方式同步；6-8 称为部分复制，使用 AOF 方式同。这是因为全量复制时，还会有 master 服务器还会执行指令，因此需要保存这些指令，待全量复制，同步这一部分数据。
<h4 id="部分复制阶段">部分复制阶段</h4>
短时间断网，可以使用部分复制来实现同步，而不必全量复制。使用部分数据需要三个要素：</>

服务器运行 id（run id）；
主服务器的复制积压缓冲区
主从服务器的复制偏移量

<h5 id="服务器运行id">服务器运行 id</h5>

服务器运行 id 是每台服务器运行时的身份识别码，一台服务器运行多次可以生成多个运行 id

运行 id 由 40 位字符组成，是随机十六进制字符
用于服务器之间传输，识别身份

<h5 id="复制缓冲区">复制缓冲区</h5>

复制缓冲区又叫复制积压缓冲区，是一个先进先出的队列，用于存储服务器执行过的命令，每次播命令，master 将命令记下来，存储在复制缓冲区。例如下面命令：

<pre><code class="highlight-chroma">set name gavin
</code></pre>
保存为 AOF 格式为：
<pre><code class="highlight-chroma">$3 \r\n #3表示指令的大小

set \r\n
$4 \r\n
name \r\n
$5 \r\n
gavin \r\n
</code></pre>
然后将上面的数据保存在复制缓冲区中，复制缓冲区由偏移量和字节值组成,字节值表示上面指令偏移量表示每个字节值代表的递增编号：
<table>
<thead>
<tr>
<th>偏移量</th>
<th>9041</th>
<th>9042</th>
<th>9043</th>
<th>9044</th>
<th>9045</th>
<th>9046</th>
<th>9047</th>
<th>9048</th>
<th>9049</th>
<th>9050</th>
<th>9051</th>
<th>9052</th>
<th>9053</th>
<th>9054</th>
</tr>
</thead>
<tbody>
<tr>
<td>字节值</td>
<td>...</td>
<td>$</td>
<td>3</td>
<td>\r</td>
<td>\n</td>
<td>s</td>
<td>e</td>
<td>t</td>
<td>\r</td>
<td>\n</td>
<td>$</td>
<td>4</td>
<td>\r</td>
<td>...</td>
</tr>
</tbody>
</table>
<h5 id="偏移量">偏移量</h5>

偏移量是一个数字，描述缓冲区中指令字节位置
master 端每发送一次指令记录一次，slave 端接收一次指令记录一次
slave 断线后对比 master 和 slave 的差异，然后恢复数据

<table>
<thead>
<tr>
<th>master</th>
<th>slave</th>
</tr>
</thead>
<tbody>
<tr>
<td>/</td>
<td>1、发送指令： psync2 runid offset</td>
</tr>
<tr>
<td>2、执行 bgsave 生成 RDB 文件，记录当前复制偏移量 offset</td>
<td>/</td>
</tr>
<tr>
<td>3、发送 +FULLRESYNC runid offset 发送 RDB 文件给 slave</td>
<td>/</td>
</tr>
<tr>
<td>/</td>
<td>4、收到 +FULLRESYNC 保存 master 的 runid 和 offset，清空全部数据，接收 RDB 文件，恢
RDB 数据</td>
</tr>
<tr>
<td>/</td>
<td>5、发送命令：psync2 runid offset</td>
</tr>
<tr>
<td>6、接收命令，判断 runid 是否匹配，判断 offset 是否在复制缓冲区中</td>
<td>/</td>
</tr>
<tr>
<td>7、如果 runid 或 offset 有一个不满足，执行全量复制</td>
<td>/</td>
</tr>
<tr>
<td>7、如果 runid 或 offset 校验通过，offset 与 offset 相同，忽略，不用执行同步</td>
<td>/</td>
</tr>
<tr>
<td>7、如果 runid 或 offset 校验通过，offset 与 offset 不同，发送 + CONTINUE offset ，发送
制缓冲区中两个 offset 之间的数据</td>
<td>/</td>
</tr>
<tr>
<td>/</td>
<td>8、收到 +CONTINUE 保存 master 的 offset 接收 信息后，执行 bgrewriteaof，恢复数据</t
>
</tr>
</tbody>
</table>
<h3 id="命令传播阶段工作流程">命令传播阶段工作流程</h3>

<h4 id="心跳机制">心跳机制</h4>

master 心跳：

指令：PING
周期：由 repl-ping-slave-period 决定，默认 10 秒
作用：判断 slave 是否在线
查询：INFO replication (获取 slave 最后一次连接时间间隔，lag 维持在 0 或者 1 视为正常)</li

slave 心跳

指令：REPLCONF ACK {offset}
周期：1 秒
作用：汇报 slave 自己的复制偏移量，获取追星数据变更指令，判断 master 是否在线

<pre><code class="highlight-chroma"><span class="highlight
cl">min-slaves-to-write 2
min-slaves-max-l
g 8
</code></pre>
slave 数量少于 2 个，或者所有 slave 的延迟都大于等于 8 时，此时 master 只能读，不能写，
时关闭数据同步功能。
<h4 id="命令传播">命令传播</h4>
<table>
<thead>
<tr>
<th>master</th>
<th>slave</th>
</tr>
</thead>
<tbody>
<tr>
<td>1、发送命令：ping</td>
<td>1、发送命令：replconf ack offset</td>
</tr>
<tr>
<td>2、接收命令，判断 offset 是否在复制缓冲区中</td>
<td>/</td>
</tr>
<tr>
<td>3、offset 不在缓冲区，执行全量复制</td>
<td>/</td>
</tr>
<tr>
<td>3、如果 offset 在缓冲区，master 的 offset 与 slave 的 offset 相同，忽略，不用执行同步</t
>
<td>/</td>
</tr>
<tr>
<td>3、如果 offset 在缓冲区，master 的 offset 与 slave 的 offset 不同，发送 + CONTINUE offse

，发送复制缓冲区中两个 offset 之间的数据</td>
<td>/</td>
</tr>
<tr>
<td>/</td>
<td>4、收到 +CONTINUE 保存 master 的 offset 接收 信息后，执行 bgrewriteaof，恢复数据</t
>
</tr>
</tbody>
</table>
<h3 id="数据同步说明">数据同步说明</h3>
1、复制缓冲区大小设定不合理，会导致数据溢出。如进行全量复制周期太长，进行部分复制时现数据已经存在丢失情况，必须进行第二次全量复制，导致 slave 陷入死循环，因此可以设置复制缓区大小：
<pre><code class="highlight-chroma">repl-backlog-size 1mb
</code></pre>
2、master 单机内存占用主机内存比例不应过大，建议使用 50%~70% 的内存，留下 30%~50% 的内存用于执行 bgsave 命令和创建复制缓冲区。
3、为避免 slave 进行全量复制、部分复制时服务器响应阻塞或数据不同步，建议关闭此期间的对外务
<pre><code class="highlight-chroma">slave-serve-stale-data yes|no
</code></pre>