链滴

redis【主从复制 & 哨兵】

作者：haxLook

原文链接：https://ld246.com/article/1617268134683

来源网站：链滴

<h3 id="1-简介">1、简介</h3>
<h4 id="多台服务器连接方案">多台服务器连接方案</h4>

提供数据方：master

主服务器，主节点，主库
主客户端

接收数据的方：slave

从服务器，从节点，从库
从客户端

需要解决的问题

数据同步

核心工作

master 的数据复制到 slave 中

<h4 id="主从复制">主从复制</h4>
主从复制即将 master 中的数据即时、有效的复制到 slave 中
特征：一个 master 可以拥有多个 slave，一个 slave 只对应一个 master
职责：

master:

写数据
执行写操作时，将出现变化的数据自动同步到 slave
读数据（可忽略）

slave:

读数据
写数据（禁止）

<h3 id="2-作用">2、作用</h3>

读写分离：master 写、slave 读，提高服务器的读写负载能力
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 slave 的数量，通过多个从节点分担数据读取负载，大大提高 Redis 服务器并发量与数据吞吐量
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

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
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="3-工作流程">3、工作流程</h3>
<h4 id="总述">总述</h4>

主从复制过程大体可以分为 3 个阶段

建立连接阶段（即准备阶段）
数据同步阶段
命令传播阶段

<h4 id="阶段一-建立连接">阶段一：建立连接</h4>

建立 slave 到 master 的连接，使 master 能够识别 slave，并保存 slave 端口号

**主从连接（slave 连接 master） **

方式一：客户端发送命令
<pre><code class="highlight-chroma"><span class="highlight
cl">slaveof <masterip> <masterport>Copy
</code></pre>

方式二：启动服务器参数
<pre><code class="highlight-chroma"><span class="highlight
cl">redis-server -slaveof <masterip> <masterport>Copy
</code></pre>

方式三：服务器配置 （常用）
<pre><code class="highlight-chroma"><span class="highlight
cl">slaveof <masterip> <masterport>
</code></pre>

授权访问

master 客户端发送命令设置密码
<pre><code class="highlight-chroma"><span class="highlight
cl">requirepass <password>Copy
</code></pre>

master 配置文件设置密码
<pre><code class="highlight-chroma"><span class="highlight

cl">config set requirepass <password>
config get requireassCopy
</code></pre>

slave 客户端发送命令设置密码
<pre><code class="highlight-chroma">auth <password>Copy
</code></pre>

slave 配置文件设置密码
<pre><code class="highlight-chroma">masterauth <password>Copy
</code></pre>

slave 启动服务器设置密码
<pre><code class="highlight-chroma">redis-server –a <password>
</code></pre>

<h4 id="阶段二-数据同步阶段">阶段二：数据同步阶段</h4>
<p>

全量复制

将 master 执行 bgsave 之前，master 中所有的数据同步到 slave 中

部分复制（增量复制）

将 master 执行 bgsave 操作中，新加入的数据（复制缓冲区中的数据）传给 slave，slave 通过 grewriteaof 指令来恢复数据

<h5 id="数据同步阶段master说明">数据同步阶段 master 说明</h5>

如果 master 数据量巨大，数据同步阶段应避开流量高峰期，避造成 master 阻塞，影响业务正常执行
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，进行部分复制时发现据已经存在丢失的情况，必须进行第二次全量复制，致使 slave 陷入死循环状。

<pre><code class="highlight-chroma">repl-backlog-size 1mbCopy
</code></pre>
<ol start="3">
master 单机内存占用主机内存的比例不应过大，建议使用 50%-70% 的内存，留下 30%-50% 内存用于执 行 bgsave 命令和创建复制缓冲区

<h5 id="数据同步阶段slave说明">数据同步阶段 slave 说明</h5>

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服务

<pre><code class="highlight-chroma">slave-serve-stale-data yes|noCopy
</code></pre>
<ol start="2">
数据同步阶段，master 发送给 slave 信息可以理解 master 是 slave 的一个客户端，主动向 slave 发送命令
多个 slave 同时对 master 请求数据同步，master 发送的 RDB 文件增多，会对带宽造成巨大冲，如果 master 带宽不足，因此数据同步需要根据业务需求，适量错峰
slave 过多时，建议调整拓扑结构，由一主多从结构变为树状结构，中间的节点既是 master，也 slave。注意使用树状结构时，由于层级深度，导致深度越高的 slave 与最顶层 master 间数据同步迟较大，数据一致性变差，应谨慎选择

<h4 id="阶段三-命令传播阶段">阶段三：命令传播阶段</h4>

当 master 数据库状态被修改后，导致主从服务器数据库状态不一致，此时需要让主从数据同步一致的状态，同步的动作称为命令传播
master 将接收到的数据变更命令发送给 slave，slave 接收命令后执行命令
主从复制过程大体可以分为 3 个阶段

建立连接阶段（即准备阶段）
数据同步阶段
命令传播阶段

<h5 id="命令传播阶段的部分复制">命令传播阶段的部分复制</h5>

命令传播阶段出现了断网现象

网络闪断闪连
短时间网络中断
长时间网络中断

部分复制的三个核心要素

服务器的运行 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-runid-">服务器运行 ID（runid）</h5>

概念：服务器运行 ID 是每一台服务器每次运行的身份识别码，一台服务器多次运行可以生成多运行 id
组成：运行 id 由 40 位字符组成，是一个随机的十六进制字符 例如- -

fdc9ff13b9bbaab28db42b3d50f852bb5e3fcdce

作用：运行 id 被用于在服务器间进行传输，识别身份

如果想两次操作均对同一台服务器进行，必须每次操作携带对应的运行 id，用于对方识别

实现方式：运行 id 在每台服务器启动时自动生成的，master 在首次连接 slave 时，会将自己的行 ID 发送给 slave，slave 保存此 ID，通过 info Server 命令，可以查看节点的 unid

<h5 id="复制缓冲区">复制缓冲区</h5>

概念：复制缓冲区，又名复制积压缓冲区，是一个先进先出（FIFO）的队列，用于存储服务器执行过的命 令，每次传播命令，master 都会将传播的命令记录下来，并存储在复缓冲区
由来：每台服务器启动时，如果开启有 AOF 或被连接成为 master 节点，即创建复制缓冲区
作用：用于保存 master 收到的所有指令（仅影响数据变更的指令，例如 set，select）
数据来源：当 master 接收到主客户端的指令时，除了将指令执行，会将该指令存储到缓冲区中

<h5 id="复制缓冲区内部工作原理">复制缓冲区内部工作原理</h5>

组成

偏移量
字节值

工作原理

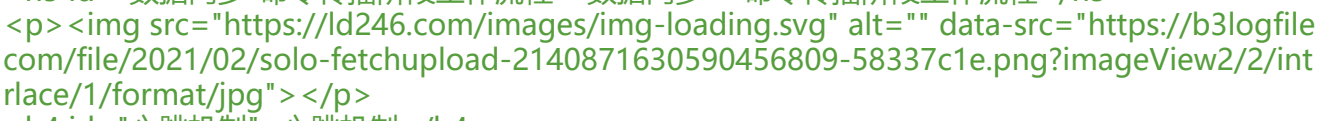
通过 offset 区分不同的 slave 当前数据传播的差异
master 记录已发送的信息对应的 offset
slave 记录已接收的信息对应的 offset

<h5 id="主从服务器复制偏移量-offset-">主从服务器复制偏移量（offset）</h5>

概念：一个数字，描述复制缓冲区中的指令字节位置
分类：

master 复制偏移量：记录发送给所有 slave 的指令字节对应的位置（多个）
slave 复制偏移量：记录 slave 接收 master 发送过来的指令字节对应的位置（一个）

数据来源： master 端：发送一次记录一次 slave 端：接收一次记录一次
作用：同步信息，比对 master 与 slave 的差异，当 slave 断线后，恢复据使用

<h5 id="数据同步-命令传播阶段工作流程">数据同步 + 命令传播阶段工作流程</h5>

<h4 id="心跳机制">心跳机制</h4>

进入命令传播阶段候，master 与 slave 间需要进行信息交换，使用心跳机进行维护，实现双方连接保持在线
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 项维持在 0 或 1 视为正常

slave 心跳任务

指令：REPLCONF ACK {offset}
周期：1 秒
作用 1：汇报 slave 自己的复制偏移量，获取最新的数据变更指令
作用 2：判断 master 是否在线

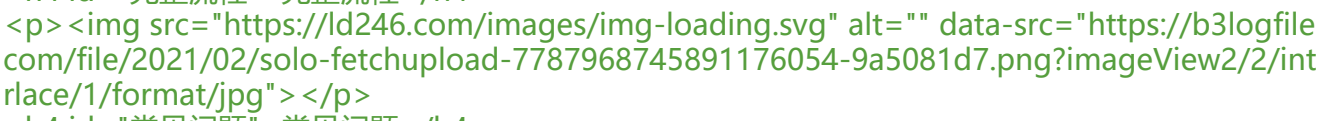
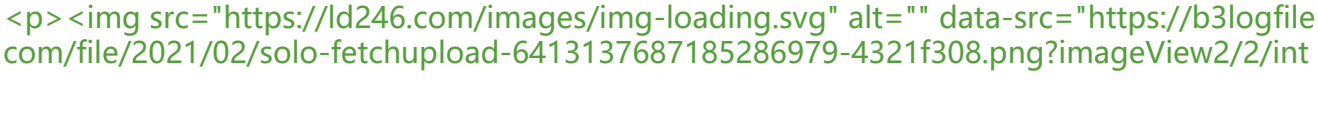
<h5 id="心跳阶段注意事项">心跳阶段注意事项</h5>

当 slave 多数掉线，或延迟过高时，master 为保障数据稳定性，将拒绝所有信息同步操作
<pre><code class="highlight-chroma">min-slaves-to-write 2
min-slaves-max-lag 8Copy
</code></pre>

slave 数量少于 2 个，或者所有 slave 的延迟都大于等于 10 秒时，强制关闭 master 写功能，止数据同步

slave 数量由 slave 发送 REPLCONF ACK 命令做确认

slave 延迟由 slave 发送 REPLCONF ACK 命令做确认

<h4 id="完整流程">完整流程</h4>

<h4 id="常见问题">常见问题</h4>


rlace/1/format/jpg">
<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfile
com/file/2021/02/solo-fetchupload-8718154284112138521-899bac18.png?imageView2/2/int
rlace/1/format/jpg">
<h4 id="频繁的网络中断">频繁的网络中断</h4>
<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfile
com/file/2021/02/solo-fetchupload-2257674675303304015-d7fa8ed1.png?imageView2/2/int
rlace/1/format/jpg">
<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfile
com/file/2021/02/solo-fetchupload-2975987306211380567-709e7410.png?imageView2/2/int
rlace/1/format/jpg">
<h4 id="数据不一致">数据不一致</h4>
<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfile
com/file/2021/02/solo-fetchupload-1134452756522449296-15b5cf8c.png?imageView2/2/inte
lace/1/format/jpg">
<h2 id="哨兵">哨兵</h2>
<h3 id="简介">简介</h3>
哨兵(sentinel) 是一个分布式系统，用于对主从结构中的每台服务器进行<s
rong>监控，当出现故障时通过投票机制选择新的 master 并将所有
lave 连接到新的 master。
<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfile
com/file/2021/02/solo-fetchupload-6962502688184549821-e3beca0e.png?imageView2/2/int
rlace/1/format/jpg">
<h3 id="2-作用-">2、作用</h3>

监控

不断的检查 master 和 slave 是否正常运行。 master 存活检测、master 与 slave 运行情况检测
/li>

通知（提醒）

当被监控的服务器出现问题时，向其他（哨兵间，客户端）发送通知。

自动故障转移

断开 master 与 slave 连接，选取一个 slave 作为 master，将其他 slave 连接到新的 master，
告知客户端新的服务器地址

注意：
哨兵也是一台 redis 服务器，只是不提供数据服务 通常哨兵配置数量为<stron
>单数
<h3 id="3-配置哨兵">3、配置哨兵</h3>

配置一拖二的主从结构
配置三个哨兵（配置相同，端口不同）

参看 sentinel.conf

启动哨兵
<pre><code class="highlight-chroma"><span class="highlight
cl">redis-sentinel sentinel端口号 .conf
</code></pre>

<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfile
com/file/2021/02/solo-fetchupload-6477996835666298089-dff24203.png?imageView2/2/inte
lace/1/format/jpg">
<h3 id="4-工作原理">4、工作原理</h3>
<h4 id="监控阶段">监控阶段</h4>

用于同步各个节点的状态信息

获取各个 sentinel 的状态（是否在线）

获取 master 的状态

master 属性

runid
role：master
各个 slave 的详细信息

获取所有 slave 的状态（根据 master 中的 slave 信息）

slave 属性

runid
role：slave
master_host、master_port
offset
...

<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfile
com/file/2021/02/solo-fetchupload-2074677286196273116-52008a1e.png?imageView2/2/int
rlace/1/format/jpg">
<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfile
com/file/2021/02/solo-fetchupload-5164207109190913403-aee9af4a.png?imageView2/2/inte
lace/1/format/jpg">
<h4 id="通知阶段">通知阶段</h4>

各个哨兵将得到的信息相互同步（信息对称）

<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfile
com/file/2021/02/solo-fetchupload-6358088504638452982-11f6a937.png?imageView2/2/int

rlace/1/format/jpg">
<h4 id="故障转移">故障转移</h4>
<h5 id="确认master下线">确认 master 下线</h5>

当某个哨兵发现主服务器挂掉了，会将 master 中的 SentinelRedistance 中的 master 改为 <strng>SRI_S_DOWN（主观下线），并通知其他哨兵，告诉他们发现 master 挂掉了。
其他哨兵在接收到该哨兵发送的信息后，也会尝试去连接 master，如果超过半数（配置文件中置的）确认 master 挂掉后，会将 master 中的 SentinelRedistance 中的 master 改为 SRIO_DOWN（客观下线）

<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfilecom/file/2021/02/solo-fetchupload-8510307774929586900-47bf14f2.png?imageView2/2/intelace/1/format/jpg">
<h5 id="推选哨兵进行处理">推选哨兵进行处理</h5>

在确认 master 挂掉以后，会推选出一个哨兵来进行故障转移工作（由该哨兵来指定哪个 slave 做新的 master）。
筛选方式是哨兵互相发送消息，并且参与投票，票多者当选。

<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfilecom/file/2021/02/solo-fetchupload-5770214268910276611-a615a5e4.png?imageView2/2/interlace/1/format/jpg">
<h5 id="具体处理">具体处理</h5>

由推选出来的哨兵对当前的 slave 进行筛选，筛选条件有：

服务器列表中挑选备选 master
在线的
响应慢的
与原 master 断开时间久的
优先原则

优先级
offset
runid

发送指令（ sentinel ）

向新的 master 发送 slaveof no one(断开与原 master 的连接)
向其他 slave 发送 slaveof 新 masterIP 端口（让其他 slave 与新的 master 相连）

很多资源来自网络，转载，如有侵权请联系删除！
转载地址,谢谢博主