[**mysql的pt-heartbeat监控备库延迟记录**](http://blog.csdn.net/z1547840014/article/details/10174667)

**工作原理：**

1，在主上创建一张heartbeat表，按照一定的时间频率更新该表的字段（把时间更新进去）。

2，连接到从上检查复制的时间记录，和从库的当前系统时间进行比较，得出时间的差异。

**常用参数：**

|  |
| --- |
| 注意：需要指定的参数至少有 --stop，--update，--monitor，--check。其中--update，--monitor和--check是互斥的，--daemonize和--check也是互斥。  --ask-pass  隐式输入MySQL密码  --charset  字符集设置  --check  检查从的延迟，检查一次就退出，除非指定了--recurse会递归的检查所有的从服务器。  --check-read-only  如果从服务器开启了只读模式，该工具会跳过任何插入。  --create-table  在主上创建心跳监控的表，如果该表不存在。可以自己建立，建议存储引擎改成memory。通过更新该表知道主从延迟的差距。  CREATE TABLE heartbeat (  ts varchar(26) NOT NULL,  server\_id int unsigned NOT NULL PRIMARY KEY,  file varchar(255) DEFAULT NULL, -- SHOW MASTER STATUS  position bigint unsigned DEFAULT NULL, -- SHOW MASTER STATUS  relay\_master\_log\_file varchar(255) DEFAULT NULL, -- SHOW SLAVE STATUS  exec\_master\_log\_pos bigint unsigned DEFAULT NULL -- SHOW SLAVE STATUS  );  heratbeat表一直在更改ts和position,而ts是我们检查复制延迟的关键。  --daemonize  执行时，放入到后台执行  --user  -u，连接数据库的帐号  --database  -D，连接数据库的名称  --host  -h，连接的数据库地址  --password  -p，连接数据库的密码  --port  -P，连接数据库的端口  --socket  -S，连接数据库的套接字文件  --file 【--file=output.txt】  打印--monitor最新的记录到指定的文件，很好的防止满屏幕都是数据的烦恼。  --frames 【--frames=1m,2m,3m】  在--monitor里输出的[]里的记录段，默认是1m,5m,15m。可以指定1个，如：--frames=1s，多个用逗号隔开。可用单位有秒（s）、分钟（m）、小时（h）、天（d）。  --interval  检查、更新的间隔时间。默认是见是1s。最小的单位是0.01s，最大精度为小数点后两位，因此0.015将调整至0.02。  --log  开启daemonized模式的所有日志将会被打印到制定的文件中。  --monitor  持续监控从的延迟情况。通过--interval指定的间隔时间，打印出从的延迟信息，通过--file则可以把这些信息打印到指定的文件。  --master-server-id  指定主的server\_id，若没有指定则该工具会连到主上查找其server\_id。  --print-master-server-id  在--monitor和--check 模式下，指定该参数则打印出主的server\_id。  --recurse  多级复制的检查深度。模式M-S-S...不是最后的一个从都需要开启log\_slave\_updates，这样才能检查到。  --recursion-method  指定复制检查的方式,默认为processlist,hosts。  --update  更新主上的心跳表。  --replace  使用--replace代替--update模式更新心跳表里的时间字段，这样的好处是不用管表里是否有行。  --stop  停止运行该工具（--daemonize），在/tmp/目录下创建一个“pt-heartbeat-sentinel” 文件。后面想重新开启则需要把该临时文件删除，才能开启（--daemonize）。  --table  指定心跳表名，默认heartbeat。 |

1、在主服务器下创建一个定期更新heartbeat表

|  |
| --- |
| [root@node1 ~]# pt-heartbeat --user=root --ask-pass --host=192.168.70.71 --create-table -D zhoujh --interval=1 --update --replace --daemonize |



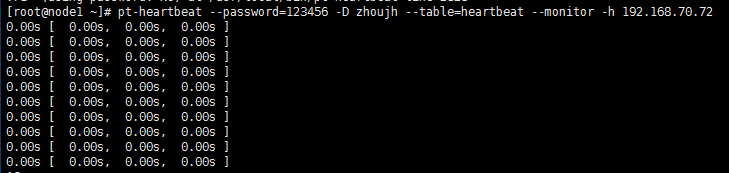
|  |
| --- |
| [root@node1 ~]# pgrep -fl pt-heartbeat |



参数说明：-D 需要监控的数据库，--update会在指定的时间更新一个时间戳,-u 用户，-p 密码 -P端口 -h 主机 --create-table自动生成heartbeat表,--daemonize在后台运行。

2、在主上运行监测复制延迟

|  |
| --- |
| [root@node1 ~]# pt-heartbeat --password=123456 -D zhoujh --table=heartbeat --monitor -h 192.168.70.72 |



0表示从没有延迟。 [ 0.00s, 0.00s, 0.00s ] 表示1m,5m,15m的平均值。可以通过--frames去设置。

复制没有延迟，可以使用压力工具测试一下。（或者停掉slave，观察）在主库创建测试数据库sbtest；

|  |
| --- |
| mysql> create database sbtest;  Query OK, 1 row affected (0.00 sec) |

开始生成100w记录的数据

|  |
| --- |
| [root@yayun\_192.168.0.10 ~]# sysbench --test=oltp --oltp-table-size=1000000 --oltp-read-only=off --init-rng=on --num-threads=16 --max-requests=0 --oltp-dist-type=uniform --max-time=1800 --mysql-user=root --mysql-socket=/tmp/mysql.sock --mysql-password=123456 --db-driver=mysql --mysql-table-engine=innodb --oltp-test-mode=complex prepare  sysbench 0.4.10: multi-threaded system evaluation benchmark |

3、再次检测复制延时：

|  |
| --- |
| [root@node1 ~]# pt-heartbeat --password=123456 -D zhoujh --table=heartbeat --monitor -h 192.168.70.72 |

我们还可以使用--check监测一次就退出:

|  |
| --- |
| [root@node1 ~]# pt-heartbeat --password=123456 -D zhoujh --table=heartbeat --check -h 192.168.70.72  0.00 |

如何关闭主上面执行的后台进程。可以用参数--stop 去关闭：

|  |
| --- |
| [root@node1 ~]# pt-heartbeat --stop  Successfully created file /tmp/pt-heartbeat-sentinel |

这样就把在主上开启的进程杀掉了，后续要继续开启后台进行的话，需要把/tmp/pt-heartbeat-sentinel 文件删除，否则启动不了。

**总结：**

      通过pt-heartbeart工具可以很好的弥补默认主从延迟的[问题](http://www.woqutech.com/?p=1116)，但需要搞清楚该工具的原理。而默认的Seconds\_Behind\_Master值是通过将服务器当前的时间戳与二进制日志中的事件时间戳相对比得到的，所以**只有在执行事件时才能报告延时**。备库**复制线程没有运行**，也会报延迟null。还有一种情况：**大事务**，一个事务更新数据长达一个小时，最后提交。这条更新将比它实际发生时间要晚一个小时才记录到二进制日志中。当备库执行这条语句时，会临时地报告备库延迟为一个小时，执行完后又很快变成0。

|  |
| --- |
| 参考文章： <http://www.cnblogs.com/gomysql/p/3687329.html>  http://www.cnblogs.com/zhoujinyi/p/3494258.html |