**Updateserver启动流程和配置文件说明**

文件分为两个部分，第一个是Updateserver启动流程，第二块是配置文件说明。

* 1. 一、UpdateServer 主备启动

无论Master和Slave启动的命令行都需要指定"-f config\_file"，并且只需要这一个参数就可以启动。**需要注意，如果没有vip时，[update\_server]域内的vip参数必须设置为本机地址。**

另外，目前Updateserver Slave从Master同步旧日志文件是使用rsync工具，所以需要Master机器和Slave 机器互有信任关系。

其他参数参考后面一部分配置文件说明。

**【Master启动流程】**

1. 读取配置文件；

2. 初始化组件；

3. 监听服务端口；

4. 向Rootserver 获取schema；

5. 回放日志；

6. 启动读服务线程和写服务线程；

7. 开始正常服务。

**【Slave启动流程】**

1. 读取配置文件；

2. 初始化组件；

3. 监听服务端口；

4. 向master注册；

5. 向rootserver获取schema；

6. 启动获取日志线程（从回放点开始，获取旧日志）；

7. 启动日志回放线程；

8. 开始正常服务。

**【Slave切换成Master流程】**

1. 检查线程发现vip漂移到本机；

2. 等待一段时间（即用于检查vip的周期）后，停止日志同步线程；

3. 等待日志回放线程回放完毕（如果旧日志没有同步完全，则日志回放线程不会结束）；

4. 启动读服务线程；

5. 启动写服务线程；

6. 开始作为Master正常服务；

**【Master失去vip流程】**

1. 检查线程发现vip漂移到本机；

2. UPS退出；

* 1. 二、Updateserver配置文件说明

正常启动Updateserver，并且保证主备正常工作，需要注意的参数如下：

* [update\_server]域内vip项指定了updateserver使用的vip地址，在测试时如果没有配置vip，则将该参数设置为master所在机器地址；
* 因为updateserver的master和slave都会向RootServer请求schema，所以[root\_server]域的vip, port两个参数要设置正确的RootServer的地址和服务端口号；
* 强烈建议程序普通日志和Updateserver的操作日志存放在不同的磁盘上，当程序压力较大时，两个日志量都会很大，分开写不同磁盘可以避免互相干扰；

|  |  |  |
| --- | --- | --- |
| 配置项 | 默认值 | 说明 |
| **[public]** |  |  |
| pid\_file | server.pid | 该文件存放Updateserver守护进程的PID |
| log\_file | server.log | Updateserver日志文件 |
| log\_level | info | Updateserver日志级别 |
|  |  |  |
| **[update\_server]** |  |  |
| vip = 10.232.36.29 | 无 | Updateserver提供服务的vip地址 |
| port = 8888 | 无 | UpdateServer提供服务的端口号 |
| log\_dir\_path = ./commitlog | 无 | 操作日志存放目录 |
| log\_size\_mb = 64 | 64 | 单个操作日志文件大小 |
| dev\_name = bond0 | 无 | 网络通讯设备名 |
| app\_name = collect | 无 | 应用名 |
| fetch\_schema\_times = 10 | 10 | Master向Root Server获取schema的重试次数 |
| fetch\_schema\_timeout\_us = 3000000 | 3000000 | Master每次获取schema的超时时间 |
| resp\_root\_times = 10 | 10 | Master向Root Server发送任务结束包的重试次数 |
| resp\_root\_timeout\_us = 1000000 | 1000000 | Master向Root Server发送任务结束包的超时时间 |
| log\_sync\_type = 0 | 0 | 是否在写日志时开启fsync，0表示不开启，1表示开启 |
| read\_thread\_count = 14 | 14 | 读线程个数, 建议配置为core个数减2 |
| read\_task\_queue\_size = 1000 | 1000 | 读任务队列大小 |
| write\_task\_queue\_size = 1000 | 1000 | 写任务队列大小，写线程个数永远为1 |
| write\_thread\_batch\_num = 1024 | 1024 | Group Commit上限 |
| log\_task\_queue\_size = 1000 | 1000 | 同步日志队列大小 |
| register\_times = 10 | 10 | Slave向Master注册的重试次数 |
| register\_timeout\_us = 2000000 | 3000000 | Slave向Master每次注册的超时时间 |
| log\_sync\_limit\_kb = 40960 | 40960 | Slave同步旧日志限速 |
| replay\_wait\_time\_us = 100000 | 100000 | Slave回放线程在遇到日志末尾时，重试等待的时间 |
| state\_check\_period\_us = 500000 | 500000 | 检查VIP周期 |
| total\_memory\_limit = 25 | 25 | Update Server总内存使用限制，单位是G， 建议配置为总内存大小的80% |
| table\_memory\_limit = 10 | 10 | Update Server中一个内存表的内存使用限制，单位是G， 建议配置为total\_memory\_limit值的40% |
| lease\_interval\_us = 8000000 | 8000000 | lease超时时间 |
| lease\_reserved\_time\_us = 5000000 | 5000000 | lease续订时间 |
| lease\_on = 1 | 1 | 是否开启lease功能，0不开启，1开启 |
| log\_sync\_timeout\_us = 500000 | 500000 | 主同步日志给备时的超时时间 |
| max\_thread\_memblock\_num = 10 | 10 | 每个线程私有的2MB内存块个数 |
|  |  |  |
| **[root\_server]** |  |  |
| vip = 10.232.36.29 | 无 | Root Server的VIP |
| port = 6666 | 无 | Root Server的端口 |
|  |  |  |