



平民软件

oneproxy-monitor 使用手册

编 写	黄辉	编写 时间	2016-12-29
审 批	审批者姓名（及其职务）	审批 时间	
版本	V1.0		

平民软件有限公司版权所有

目录

1. 引言

- 1.1 编写目的
- 1.2 项目背景
- 1.3 定义
- 1.4 使用范围

2. 软件概述

- 2.1 目标
- 2.2 功能
 - 2.2.1 监控功能
 - 2.2.2 多端口功能
 - 2.2.3 reuseport 功能
 - 2.2.4 前后端密码分离
 - 2.2.5 连接池
 - 2.2.6 读写分离
 - 2.2.7 连接清理
 - 2.2.8 负载均衡
 - 2.2.9 数据库混合搭配
 - 2.2.10 显示交互数据
 - 2.2.11 显示前端操作的 sql 语句
 - 2.2.12 在线更新配置
 - 2.2.13 keepalive 功能
 - 2.2.14 虚拟 ip(vip)功能
 - 2.2.15 审计功能

3. 安装

4. 获取帮助

5 配置文件说明

- 5.1 [oneproxy]节的配置
- 5.2 [database_xxxx]节的配置
- 5.3 数据库组节的配置

6. 附录



- 6.1 混合数据库的配置样例
- 6.2 sql server 数据库配置样例
- 6.3 postgresql 数据库配置样例
- 6.4 直接转发时的配置样例

1. 引言

1.1 编写目的

编写本文档是为了帮助用户了解本软件的安装与使用、了解软件的功能、性能等特性。

读者对象：使用本软件的所有用户。

1.2 项目背景

在当前大数据量的情况下，很多场景中需要把数据进行实时复制，采用一对多的复制方式。其中一份数据用于查询以及备份，另外一份数据用于修改等，在这种场景中需要用到读写分离的功能特性。

很多场景中前端应用操作数据库都是采用短连接的方式，而在短连接中建立连接的过程占用了很大比例的时间。如果能够加快连接的建立过程，则可以大大的提供数据库的性能。为了解决这个问题，目前的解决方案是使用连接池的功能，在连接池中针对建立的连接进行维护和管理。本项目中提供了一种稍微有点差异的连接池，既可以有效的维护连接，又可以动态调整连接池的大小。

现在市面上的数据库的种类繁多，并且每个数据库都有自己的优点和缺点。同一家公司中可能使用了多种数据库。随着数据库种类的增加，管理也特别的困难。前端应用程序需要在不同的应用程序中写入不同数据库的个性化数据。本项目可以同时管理多种数据库，方便前端应用程序对数据库的使用。

1.3 定义

oneproxy-monitor: 平民软件提供的数据库中间件

1.4 使用范围

由于 onepoxy-for-postgresql, onepoxy-for-sqlserver 都是在 onepoxy-monitor 框架下开发的，故本配置也适合于上述两个程序的配置。

oneproxy-monitor 使用的范围为：

- 1) 如果只是转发，不进行协议解析。则不受限制
- 2) 进行协议解析：
 - i) sqlserver 支持 2008 及以后的版本。
 - ii) postgresql 支持 9.4 及以上的版本。

iii)长连接不能有效的使用连接池。

2. 软件概述

2.1 目标

- 1) 提供监控页面，在监控页面中提供登录信息、sql 语句信息、事务信息、数据库使用量信息等。
- 2) 提供多个端口功能，并且提供从不同端口接收到的数据转发到不同数据库，或者同一个数据库的功能。
- 3) 提供负载均衡的功能，可以在同一个数据库组中的不同数据库间进行负载均衡操作。
- 4) 提供连接池功能，使用连接池来管理连接
- 5) 提供前后端密码分离功能，加强安全性管理
- 6) 提供读写分离功能。从 slave 数据库读取数据，写数据到 master 数据库。

2.2 功能

2.2.1 监控功能

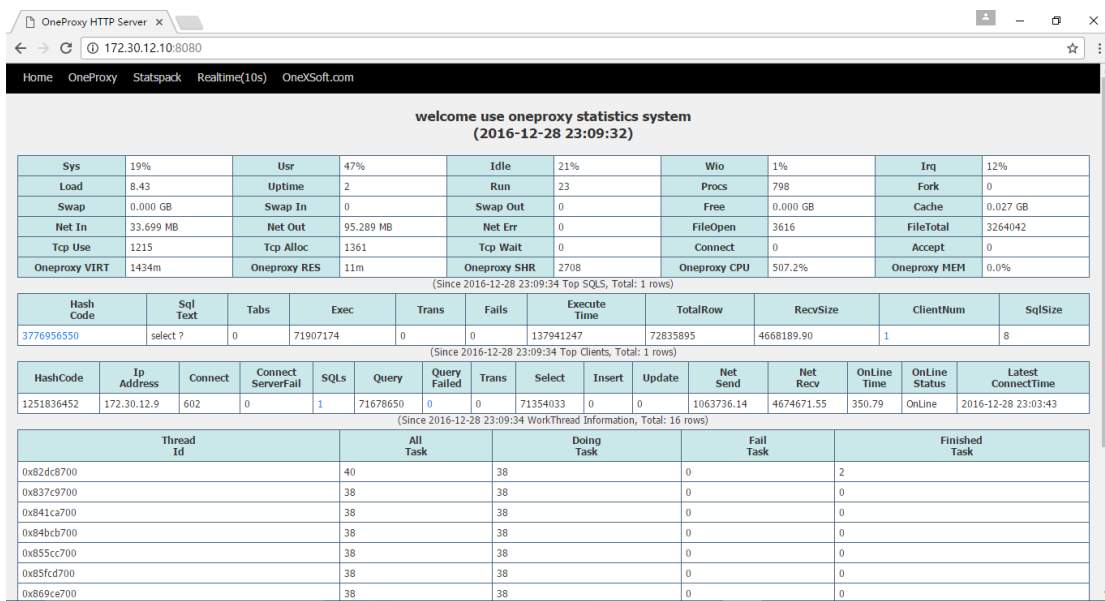
oneproxy-monitor 提供了部分监控的功能，通过监控功能可以查看到应用程序的登录和查询数据的情况，以及操作了哪些表和操作方式。可以在配置文件中通过 `httpserver_addr` 和 `httpserver_port` 两个参数来指定监控页面的地址和端口信息。下面是默认的配置：

```
httpserver_addr = 0.0.0.0
httpserver_port = 8080
```

下面详细介绍所有的监控功能：

2.2.1.1 监控首页

在监控首页上面提供了当前主机的系统信息，部分查询的 sql 语句，部分登录的应用程序信息，以及 oneproxy-monitor 处理任务的负载信息（见图 2.2.1.1.1）。



welcome use oneproxy statistics system
(2016-12-28 23:09:32)

Sys	19%	Usr	47%	Idle	21%	Wio	1%	Irq	12%
Load	8.43	Uptime	2	Run	23	Procs	798	Fork	0
Swap	0.000 GB	Swap In	0	Swap Out	0	Free	0.000 GB	Cache	0.027 GB
Net In	33.699 MB	Net Out	95.289 MB	Net Err	0	FileOpen	3616	FileTotal	3264042
Tcp Use	1215	Tcp Alloc	1361	Tcp Wait	0	Connect	0	Accept	0
Oneproxy VIRT	1434m	Oneproxy RES	11m	Oneproxy SHR	2708	Oneproxy CPU	507.2%	Oneproxy MEM	0.0%

(Since 2016-12-28 23:09:34 Top SQLs, Total: 1 rows)

Hash Code	Sql Text	Tab	Exec	Trans	Fails	Execute Time	TotalRow	RecvSize	ClientNum	SqlSize
3776956550	select ?	0	71907174	0	0	137941247	72835895	4668189.90	1	8

(Since 2016-12-28 23:09:34 Top Clients, Total: 1 rows)

Hash Code	Ip Address	Connect	Connect ServerFail	SQLs	Query	Query Failed	Trans	Select	Insert	Update	Net Send	Net Recv	OnLine Time	OnLine Status	Latest ConnectTime
1251836452	172.30.12.9	602	0	1	71678650	0	0	71354033	0	0	1063736.14	4674671.55	350.79	OnLine	2016-12-28 23:03:43

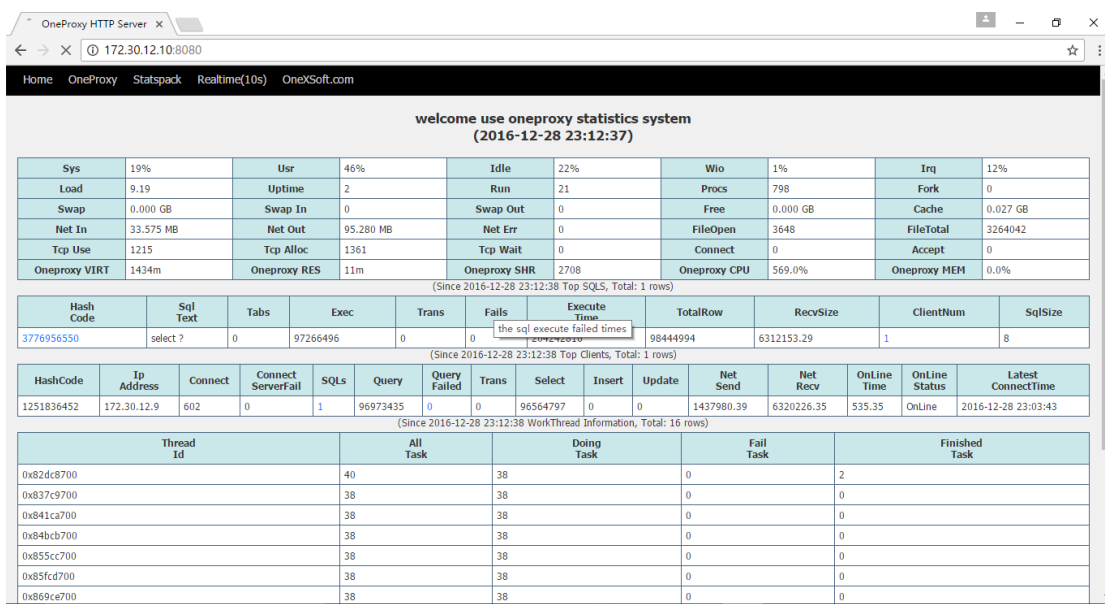
(Since 2016-12-28 23:09:34 WorkThread Information, Total: 16 rows)

Thread Id	All Task	Doing Task	Fail Task	Finished Task
0x82dc8700	40	38	0	2
0x837c9700	38	38	0	0
0x841ca700	38	38	0	0
0x84bcb700	38	38	0	0
0x855cc700	38	38	0	0
0x85fcd700	38	38	0	0
0x869ce700	38	38	0	0

图 2.2.1.1.1

从上图可以看到，整个首页分成了 4 部分，从上往下依次是：主机信息、sql 语句信息、客户端操作信息、oneproxy-monitor 任务负载信息。

针对表格中 title 的解释可以直接把鼠标停留在 title 上面，即可查看（见图 2.2.1.1.2）。



welcome use oneproxy statistics system
(2016-12-28 23:12:37)

Sys	19%	Usr	46%	Idle	22%	Wio	1%	Irq	12%
Load	9.19	Uptime	2	Run	21	Procs	798	Fork	0
Swap	0.000 GB	Swap In	0	Swap Out	0	Free	0.000 GB	Cache	0.027 GB
Net In	33.575 MB	Net Out	95.280 MB	Net Err	0	FileOpen	3648	FileTotal	3264042
Tcp Use	1215	Tcp Alloc	1361	Tcp Wait	0	Connect	0	Accept	0
Oneproxy VIRT	1434m	Oneproxy RES	11m	Oneproxy SHR	2708	Oneproxy CPU	569.0%	Oneproxy MEM	0.0%

(Since 2016-12-28 23:12:38 Top SQLs, Total: 1 rows)

Hash Code	Sql Text	Tab	Exec	Trans	Fails	Execute Time	TotalRow	RecvSize	ClientNum	SqlSize
3776956550	select ?	0	97264496	0	0	the sql execute failed times	98444994	6312153.29	1	8

(Since 2016-12-28 23:12:38 Top Clients, Total: 1 rows)

Hash Code	Ip Address	Connect	Connect ServerFail	SQLs	Query	Query Failed	Trans	Select	Insert	Update	Net Send	Net Recv	OnLine Time	OnLine Status	Latest ConnectTime
1251836452	172.30.12.9	602	0	1	96973435	0	0	96564797	0	0	1437980.39	6320226.35	535.35	OnLine	2016-12-28 23:03:43

(Since 2016-12-28 23:12:38 WorkThread Information, Total: 16 rows)

Thread Id	All Task	Doing Task	Fail Task	Finished Task
0x82dc8700	40	38	0	2
0x837c9700	38	38	0	0
0x841ca700	38	38	0	0
0x84bcb700	38	38	0	0
0x855cc700	38	38	0	0
0x85fcd700	38	38	0	0
0x869ce700	38	38	0	0

图 2.2.1.1.2

上图中，把鼠标停留在了第二部分的 Fails 表格上面，则就弹出了提示信息。这个提示信息就是针对 Fails 的解释。

首页的显示情况也可以进行个性化配置的，目前支持选择是否显示 sql 语句、是否显示客户端操作信息、是否显示 oneproxy-monitor 任务负载情况。这个配置可以通过 Home->Setting 来配置（见图 2.2.1.1.3）

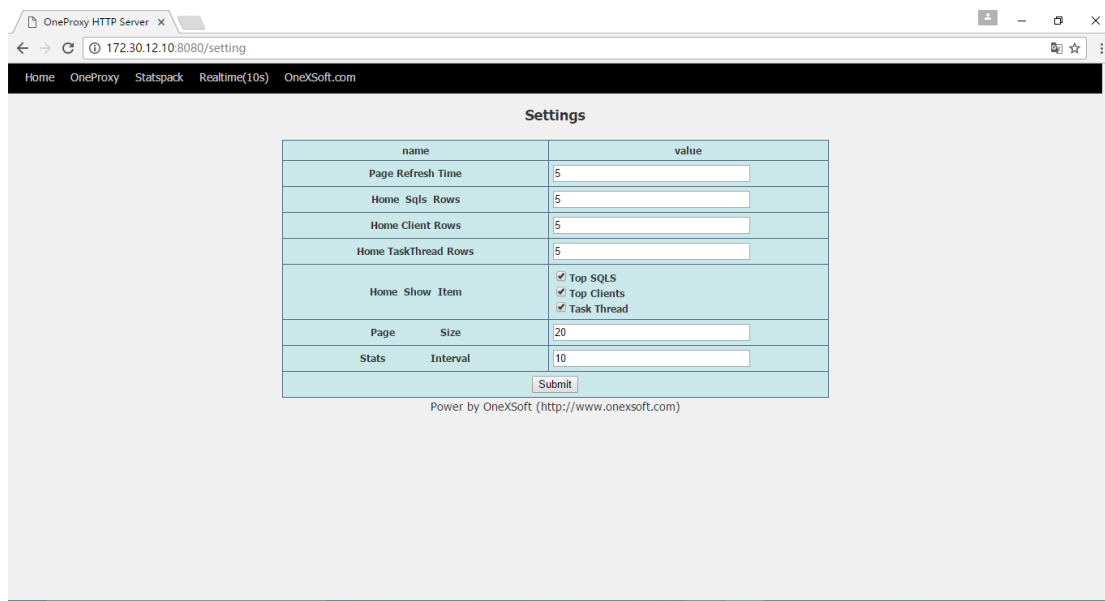


图 2.2.1.1.3

上图为 setting 页面的情况，可以通过选择 Home Show Item 中的不同选项来决定首页的显示情况。

2.2.1.2 数据库信息

在 OneProxy->DataBase 页面提供了数据信息情况（见图 2.2.1.2.1）。在这个页面中显示了每个数据库的地址、端口、权值、连接数情况、以及是否为 active。

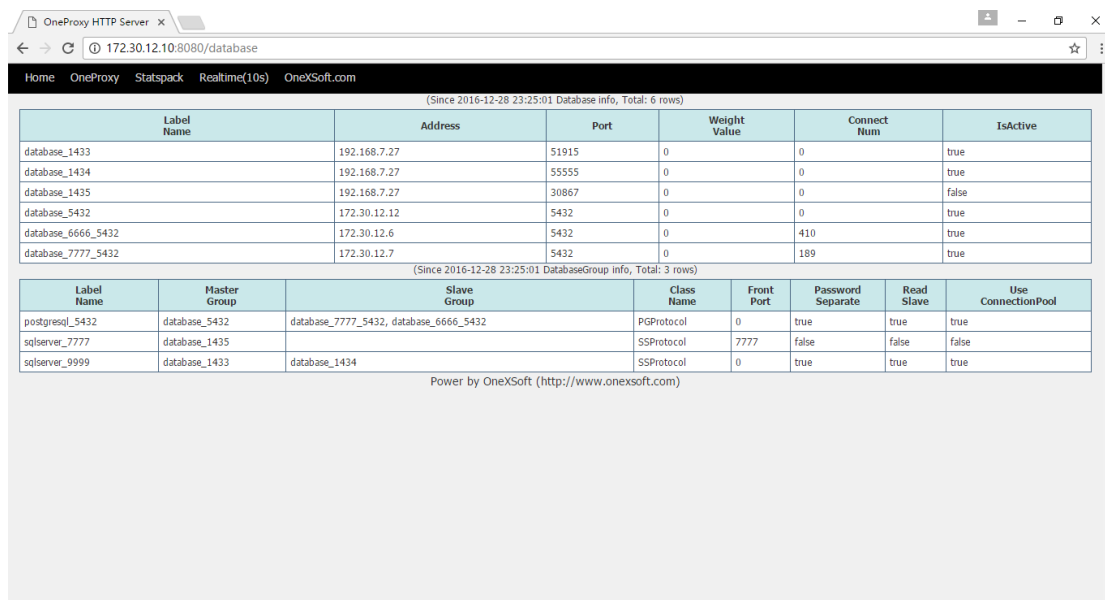


图 2.2.1.2.1

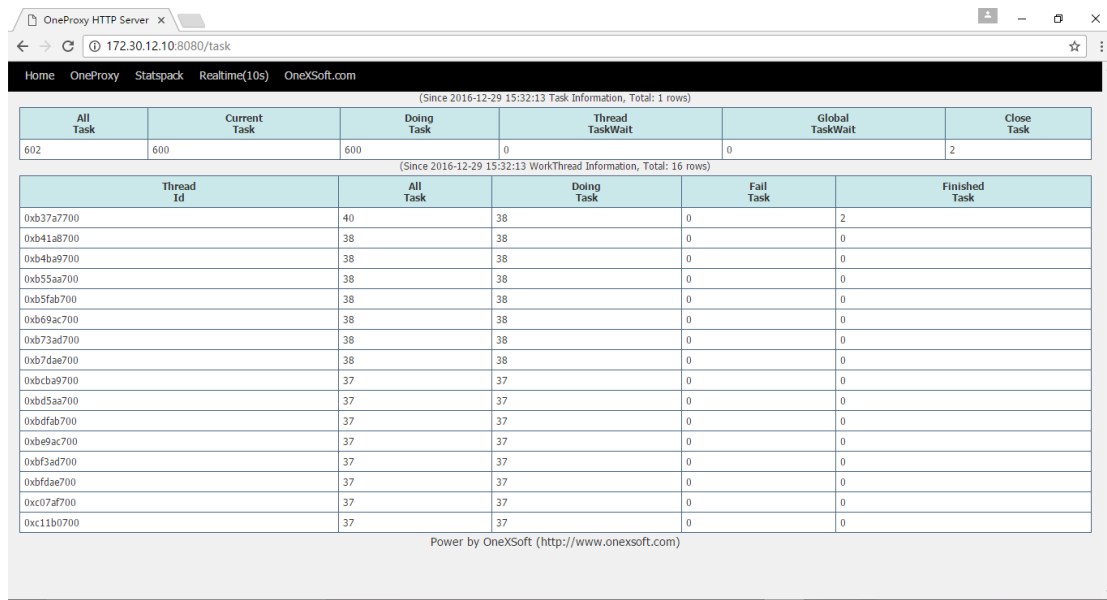
上图页面被分成了两个部分，第一部分是单个数据库的情况。在单个数据库中 LabelName 是单个数据库的别名，是在配置文件中指定的。这个别名不能够重复，并且必须以 database_ 开头。



第二部分是数据库组情况，前端只能够连接到对应的数据库组上面，其中数据库组的 LabelName 可以是任意的，但是不能是 database_ 开头。数据库组需要指定 master 数据库，slave 数据库的 labelName。如果有多个 master 数据库或者多个 slave 数据库则以逗号进行分隔，如上图所示。同时需要指定使用的处理类，目前只提供了三个处理类，sqlserver 使用 SSProtocol. postgresql 数据库使用 PGProtocol。其他的使用 FakeProtocol。数据库组中的 FrontPort 字段指定处理前端哪个端口来的数据，如果为 0 表示处理所有前端端口的数据。

2.2.1.3 任务信息

Oneproxy->Task 页面中显示的是 oneproxy-monitor 当前的任务情况（见图 2.2.1.3.1）。



All Task	Current Task	Doing Task	Thread TaskWait	Global TaskWait	Close Task
602	600	600	0	0	2

Thread Id	All Task	Doing Task	Fail Task	Finished Task
0xb37a7700	40	38	0	2
0xb41a8700	38	38	0	0
0xb4ba9700	38	38	0	0
0xb55aa700	38	38	0	0
0xb5fab700	38	38	0	0
0xb69ac700	38	38	0	0
0xb73ad700	38	38	0	0
0xb7dae700	38	38	0	0
0xbcb9a700	37	37	0	0
0xb55aa700	37	37	0	0
0xbdfab700	37	37	0	0
0xb69ac700	37	37	0	0
0xbf3ad700	37	37	0	0
0xbfd9e700	37	37	0	0
0xc07af700	37	37	0	0
0xc11b0700	37	37	0	0

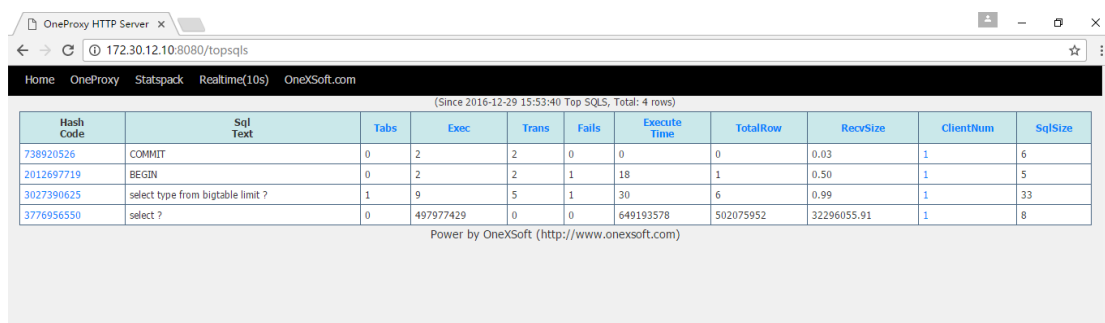
图 2.2.1.3.1

Task 页面也分成了两部分：第一部分是从 oneproxy-monitor 启动以来执行任务的情况。第二部分是每个工作线程执行任务的情况。每个部分都有总的处理任务的数量（AllTask），正在处理的任务数量(DoingTask)。在第一部分中还有 Thread TaskWait 表示任务已经分配给线程，但是线程在处理的过程中前端还没有获取到后端数据库连接的任务数量。GlobalTaskWait 表示还没有分配给工作线程的任务的数量。如果连接过多，已经超过配置中 maxconnectnum 指定的任务时，则会进去到 GlobalTaskWait 中进行排队处理。如果前端连接没有超过 maxconnectnum,则不会进去全局排队队列，直接分配给了工作线程。

第一部分的 closeTask 是第二部分的所有 FinishedTask 之和。第二部分的 FailTask 是当前工作线程处理失败的 task 的数量。

2.2.1.4 sql 统计

Statspack->Top SQLS 页面展示从启动或者重置开始到现在的 sql 执行情况（见图 2.2.1.4.1）。



Hash Code	Sql Text	Tabs	Exec	Trans	Fails	Execute Time	TotalRow	RecvSize	ClientNum	SqlSize
738920526	COMMIT	0	2	2	0	0	0	0.03	1	6
2012697719	BEGIN	0	2	2	1	18	1	0.50	1	5
3027390625	select type from bigtable limit ?	1	9	5	1	30	6	0.99	1	33
3776956550	select ?	0	497977429	0	0	649193578	502075952	32296055.91	1	8

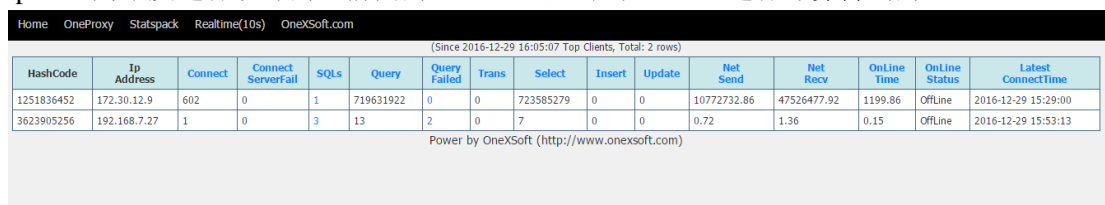
Power by OneXSoft (http://www.onexsoft.com)

图 2.2.1.4.1

可以通过点击每条 sql 语句前的 HashCode 查看完整的信息。clientNum 字段显示了有多少个客户端执行了此条 sql 语句，也可以通过点击表格的 title 来按照相应的字段进行排序。每个字段的含义可以通过把鼠标停留在 title 上面进行查看，在此就不过多进行解释了。

2.2.1.5 client 统计

Statspack->Top Clients 页面展示的是客户端的信息（见图 2.2.1.5.1），客户端的信息是以 ip 地址为维度进行统计的，前面的 hashcode 是针对 IP 地址进行计算得到的。



HashCode	Ip Address	Connect	Connect ServerFail	SQLs	Query	Query Failed	Trans	Select	Insert	Update	Net Send	Net Recv	OnLine Time	OnLine Status	Latest ConnectTime
1251836452	172.30.12.9	602	0	1	719631922	0	0	723585279	0	0	10772732.86	47526477.92	1199.86	OffLine	2016-12-29 15:29:00
3623905256	192.168.7.27	1	0	3	13	2	0	7	0	0	0.72	1.36	0.15	OffLine	2016-12-29 15:53:13

Power by OneXSoft (http://www.onexsoft.com)

图 2.2.1.5.1

下面表格是部分字段的解释：

Connect	对应的客户端建立连接的次数或者登录的次数
SQLs	当前客户端执行 sql 语句的条数，通过点击对应的数字 可以查看具体的 sql 语句
QueryFailed	是在 sql 语句执行的过程中得到 ERROR 包的次数
Trans	客户端执行事务的数量
OnLineTime	在线的累计时间
OnLineStatus	当前是否在线
LatestConnectTime	最后一次登录的时间

2.2.1.6 table 统计

Statspack->Top Tables 页面是针对数据库表的使用统计（图 2.2.1.6.1）。



Home

OneProxy

Statspack

Realtime(10s)

OneXSoft.com

(Since 2016-12-29 16:35:24 Top Tables, Total: 30 rows)Next

TableName	Sqls	Select			Insert			Update			Delete		
		Exec	Rows	Time	Exec	Rows	Time	Exec	Rows	Time	Exec	Rows	Time
ONLY	1	1	0	0	0	0	0	0	0	0	0	0	0
bigtable	2	2	101	32	0	0	0	0	0	0	0	0	0
pg_am	2	16	0	107	0	0	0	0	0	0	0	0	0
pg_attrdef	2	5	0	74	0	0	0	0	0	0	0	0	0
pg_attribute	6	12	1	115	0	0	0	0	0	0	0	0	0
pg_class	10	40	0	229	0	0	0	0	0	0	0	0	0
pg_collation	3	6	0	126	0	0	0	0	0	0	0	0	0
pg_constraint	6	29	1	205	0	0	0	0	0	0	0	0	0
pg_database	1	4	16	40	0	0	0	0	0	0	0	0	0
pg_default_acl	1	20	0	26	0	0	0	0	0	0	0	0	0
pg_depend	3	17	1	122	0	0	0	0	0	0	0	0	0
pg_description	19	48	4	458	0	0	0	0	0	0	0	0	0
pg_event_trigger	1	1	0	54	0	0	0	0	0	0	0	0	0
pg_extension	1	1	1	63	0	0	0	0	0	0	0	0	0
pg_index	3	20	0	174	0	0	0	0	0	0	0	0	0
pg_inherits	2	6	0	39	0	0	0	0	0	0	0	0	0
pg_language	4	7	0	89	0	0	0	0	0	0	0	0	0
pg_namespace	19	49	15	524	0	0	0	0	0	0	0	0	0
pg_proc	7	10	6	155	0	0	0	0	0	0	0	0	0
pg_rewrite	1	4	0	11	0	0	0	0	0	0	0	0	0

Power by OneXSoft (http://www.onexsoft.com)

Power by OneXSoft (http://www.onexsoft.com)

图 2.2.1.6.1

在表的统计中，是按照 select, insert, update, delete 维度来进行统计的。可以通过 sqls 来查看有多少个 sql 语句包含了此表。

2.2.1.7 表和 SQL 的映射关系统计

Statspack->Top TablesMap 页面展示的是表与 sql 语句的关系（图 2.2.1.7.1）。

Home OneProxy Statspack Realtime(10s) OneXSoft.com		
(Since 2016-12-29 16:41:05 Top TablesMap, Total: 32 rows)Next		
Tables	Exec	SqlHashCode
qry:ONLY	1	3792735339
qry:bigtable	2	60132373,2000981005
qry:pg_am,pg_class,pg_constraint,pg_depend,pg_description,pg_index,pg_namespace,pg_tablespace	16	2951271640,3267027008
qry:pg_attrdef,pg_attribute,pg_class,pg_collation,pg_constraint,pg_description,pg_index,pg_namespace,pg_seclabels,pg_type	4	330567416
qry:pg_attrdef,pg_attribute,pg_class,pg_namespace,pg_type	1	2268492359
qry:pg_attribute	3	1482402270,3408683773,3556422247
qry:pg_attribute,pg_inherits	4	22517547
qry:pg_class,pg_constraint,pg_description,pg_namespace	4	252495316
qry:pg_class,pg_constraint,pg_description,pg_namespace,pg_type	4	3925891711
qry:pg_class,pg_description,pg_language,pg_namespace,pg_proc,pg_trigger	4	2635176213
qry:pg_class,pg_description,pg_namespace,pg_rewrite	4	3772743732
qry:pg_class,pg_description,pg_seclabels	1	1585578489
qry:pg_class,pg_inherits,pg_namespace	2	1717712648
qry:pg_collation,pg_description,pg_namespace	1	1751783144
qry:pg_collation,pg_description,pg_namespace,pg_seclabels,pg_type	1	2232011352
qry:pg_constraint,pg_depend,pg_trigger	1	624113126
qry:pg_database,pg_shdescription,pg_shseclabel,pg_tablespace	4	3627035216
qry:pg_default_acl	20	3328925387
qry:pg_description,pg_language,pg_namespace,pg_proc,pg_seclabels,pg_type	2	1069018668,2715385687
qry:pg_description,pg_namespace	1	1991334079

图 2.2.1.7.1

从图中可以看到一组表可能在多个 sql 语句中被使用。通过这个统计可以查看出 sql 语句的复杂度。也可以看到具有强复杂度的 sql 语句的数量。

2.2.2 多端口功能

oneproxy-monitor 在读取客户端连接的时候是系统调用，如果有大量的并发请求，则可能导致 listen 得队列满而导致部分客户端无法连接的问题。而从 listen 队列中读取客户端连接是 accept 函数进行系统调用，每次只能读取一个连接。当有大量并发请求时，此处有瓶

颈问题。oneproxy-monitor 提供了注册多个监听端口的解决方案，这样可以把不同的应用程序连接到不同的端口，或者不同的数据库使用不同的端口。具体的配置方法如下：

设置 config 标签下的 listen_port 的值，如果是多个端口，则以逗号分隔，例如：

```
listen_port = 9999,7777,6666
```

这样就开启多个监听端口了。如果需要把不同端口得到的数据转发到不同的数据库组，则还需要配置数据库组下的 frontport 的值为对应的监控端口号，例如：

```
[sqlserver_7777]
dbmastergroup = database_1435
classname = SSProtocol
frontport = 7777
```

sqlserver_7777 数据库组就只会处理从 7777 监听端口读取到的数据。如果想让 sqlserver_7777 处理所有前端监听端口的数据，则可以配置 frontport 为 0 即可。

2.2.3 reuseport 功能

Linux 在 2.6.32-573 及以上版本中提供了 reuseport 功能。同一个进程中可以使用多个线程针对同一个端口进行监听。Linux 自动进行负载均衡，把进行了三次握手后的连接分配到不同的 accept 调用线程中。Linux 通过这种方式来解决大量并发的问題，oneproxy-monitor 目前也已经支持这个特性，可以通过配置 config 标签下的 acceptthreadnum 来决定使用多少个线程进行监听。默认配置为 2，同时受 cpu core 的数量的限制。如果只有一个 cpu core，oneproxy-monitor 把 acceptthreadnum 设置为 1。

2.2.4 前后端密码分离

oneproxy-monitor 提供了密码分离功能，这样前端应用程序中可以不用根据不同的数据库使用不同的密码了。如果需要使用此功能，则需要进行如下配置：

- 1) 在 config 标签下的 clientusername 设置客户端使用的用户名
- 2) 在 config 标签下的 clientpassword 设置客户端使用的密码
- 3) 在数据库组的标签下设置 passwordseparate = true（默认配置为 true）。如果需要设置数据库组不使用读写分离，则可以设置 passwordseparate = false。例如：

```
[sqlserver_7777]
dbmastergroup = database_1435
classname = SSProtocol
frontport = 7777
passwordseparate = false
```

- 4) 在数据库标签下设置对应的数据库的密码，即登录到数据库需要的用户名和密码。例如：

```
[database_1435]
host = 192.168.7.27
port = 30867
```

```
username = sa
password = 0000
```

- 5) 如果是 postgresql 数据库,则需要在数据库组下增加 clientauthmethod 为 md5(默认为 md5)。指定前端的验证方式为 md5 的方式。

通过上面五步的配置后,则数据库组就开启了读写分离功能,前端需要使用 config 标签下指定的用户名和密码进行登录。oneproxy-monitor 则采用数据库标签下对应的用户名和密码登录到数据库。

2.2.5 连接池

为了加快客户端建立连接的速度,oneproxy-monitor 提供了连接池功能。oneproxy-monitor 提供的连接池功能不主动创建连接,只对连接进行管理。oneproxy-monitor 提供的连接池的原理如下:首先客户端发起连接,oneproxy-monitor 检查连接池中是否有连接,如果没有,则向对应的数据库发起连接请求,建立连接。并且把建立后的连接分配给客户端。当客户端完毕后,把连接到数据库的连接保存到连接池中。连接池会隔一段时间就检查连接是否已经空闲超时,如果超时了还没有被使用,那么就是否此连接。

从 oneproxymonitor 连接池的原理来看,如果客户端向 oneproxymonitor 发起请求的频率是恒定的,则连接池中的连接能够有效的被复用。如果客户端暴涨暴跌的发起连接,oneproxy-monitor 也能够不受限制的发起到数据库的连接。连接池也能够及时的释放空闲的连接,从而降低数据库的连接数。

使用连接池功能,则需要进行如下的配置:

- 1) 配置支持前后端密码分离。
- 2) 配置 config 标签下的 poolconncheckactivetime 的值。此配置是指定在多长时间对连接池中的连接进行检查是否还是 active,如果不是 active 就释放连接。单位为秒。默认为 5 秒。
- 3) 配置 config 标签下的 poolconntimeoutreleasetime 的值。如果 poolconntimeoutreleasetime 时间内连接没有被使用,则连接会被连接池释放掉。单位是秒。默认为 60 秒。
- 4) 配置数据库组下的 useconnectionpool 的值为 TRUE。默认为 TRUE。

例如:

```
[postgresql_5432]
dbmastergroup = database_5432
dbslavegroup = database_7777_5432, database_6666_5432
classname = PGProtocol
frontport = 0
passwordseparate=true
useconnectionpool=true
clearbackendconnection = true
```

- 5) 配置 clearbackendconnection 的值为 true,当连接在放回连接池时是否进行清理操作。postgresql 必须配置为 true。

经过上面 4 个步骤的配置,就可以使用连接池功能了。

切记：一定配置支持前后端密码分离。

注：为了支持 prepared 方式，目前只支持短连接使用连接池。如果是长连接就没有连接池的效果。

2.2.6 读写分离

为了从 slave 数据上面读取数据，写数据到 master 上面。oneproxy-monitor 也提供了读写分离的功能。

配置 onepoxy-monitor 支持读写分离的方法为：

- 1) 配置 onepoxy-monitor 支持前后端密码分离
- 2) 在数据库组下增加 readslave = true 的配置即可

例如：

```
[sqlserver_9999]
dbmastergroup = database_1433
dbslavegroup = database_1434
classname = SSProtocol
readslave = true
passwordseparate = true
```

切记：配置支持前后端密码分离。

2.2.7 连接清理

如果需要把连接放到连接池之前进行连接的清理工作，则需要配置支持连接清理的选项。可以通过在数据库组下增加 clearbackendconnection=true。默认情况下已经配置为 TRUE。建议所有数据库都设置为 TRUE。

2.2.8 负载均衡

oneproxy-monitor 支持同一个数据库组配置多个 slave 的方式。当前端连接到来时，则可以通过均衡的方式把连接分配到不同的数据库上面去。为了支持负载均衡，则需要进行下列的配置：

- 1) 配置每个数据库的权值 (weightvalue)，默认为 0。weightvalue 的取值范围为 0-100。如果都是 0，则会大概均衡的把连接分配到不同的数据库上面。
- 2) 配置多个需要均衡处理连接的数据库到数据库组中。例如：

```
[database_5432]
host = 172.30.12.12
port = 5432
username = db_user
password = 0000
```

```
[database_7777_5432]
host = 172.30.12.7
port = 5432
username = db_user
password = 0000

[database_6666_5432]
host = 172.30.12.6
port = 5432
username = db_user
password = 0000

[postgresql_5432]
dbmastergroup = database_5432
dbslavegroup = database_7777_5432, database_6666_5432
classname = PGProtocol
frontport = 0
```

则上面的配置会把读进行负载均衡到两个 slave 数据库上面。

默认情况下是根据前端连接分配得到的数据库的情况进行负载均衡的，oneproxy-monitor 中还提供了根据真正到数据库的连接数的情况进行负载均衡。可以通过在数据库组中增加：usedatabaseconnecttobalace 来指定不同的方式。
usedatabaseconnecttobalace = true 使用正在建立到数据库的连接的情况来进行负载均衡。针对短连接比较多的情况下比较有效。
usedatabaseconnecttobalace = false 使用预分配给前端连接的数据库情况来进行负载均衡，这种情况下可能某些客户端连接并没有建立到数据库的连接。这种方式针对长连接比较有效。

2.2.8.1 master 提供查询服务

如果需要把 master 数据库也当做 slave 数据库作为负载均衡部分的数据库，则可以在数据库组的标签下增加：usemasterasslave=true。

例如：

```
[postgresql_5432]
dbmastergroup = database_5432
dbslavegroup = database_7777_5432, database_8888_5432
classname = PGProtocol
frontport = 0
passwordseparate=true
useconnectionpool=true
```

```
readslave = true  
clearbackendconnection = true  
usemasterasslave = true
```

2.2.9 数据库混合搭配

oneproxy-monitor 已经支持多种数据库进行混搭的模式。目前支持 sqlserver, postgresql 进行统计, 读写分离等功能, 其他的数据库可以直接进行转发。使用的方法是:

- 1) 开启多端口功能
- 2) 不同的端口使用不同的处理类进行处理, 目前支持的处理类有 SSProtocol, PGProtocol, FakeProtocol。其中 SSProtocol 是针对 sql server 的处理类, PGProtocol 是针对 postgresql 的处理类。其他的数据库则可以使用 FakeProtocol 进行直接转发处理。下面的配置是一个支持混合数据库处理的配置:

```
[oneproxy]  
logfile = oneproxy_log.log  
pidfile = oneproxy_pid.pid  
listen_addr = 0.0.0.0  
listen_port = 9999,7777,6666  
clientusername = sa  
clientpassword = 0000  
  
[database_1433]  
host = 192.168.7.27  
port = 51915  
username = sa  
password = 0000  
  
[database_1434]  
host = 192.168.7.27  
port = 55555  
username = sa  
password = abcdefuz  
  
[database_1435]  
host = 192.168.7.27  
port = 30867  
username = sa  
password = 0000
```



```
[database_5432]
host = 172.30.12.12
port = 5432
username = db_user
password = 0000

[database_7777_5432]
host = 172.30.12.7
port = 5432
username = db_user
password = 0000

[database_6666_5432]
host = 172.30.12.6
port = 5432
username = db_user
password = 0000

[database_1521]
host = 172.30.12.10
port = 1521

[sqlserver_9999]
dbmastergroup = database_1433
dbslavegroup = database_1434
classname = SSProtocol
useconnectionpool = true
readslave = true
passwordseparate = true
frontport=9999

[postgresql_5432]
dbmastergroup = database_5432
dbslavegroup = database_7777_5432, database_6666_5432
classname = PGProtocol
frontport = 7777
passwordseparate=true
useconnectionpool=true
readslave = true
clearbackendconnection = true

[other_database]
```

```
dbmastergroup = database_1521
classname = FakeProtocol
frontport = 6666
```

上面的配置指出，客户端可以通过 9999 端口来处理 sqlserver 的请求，通过连接 7777 端口来处理 postgresql 的请求，通过连接 6666 连处理其他数据库的请求。

2.2.10 显示交互数据

oneproxy-monitor 也可以使用作为调试工具，通过开启 config 下的 data_dump 功能来打印前端和后端交互的数据。输出格式如下所示：

```
[2016-12-29 20:25:53][dump][0x97fff700]front =====> backend:77
00 00 00 4D 00 03 00 00      |...M....|
75 73 65 72 00 73 61 00      |user.sa.|
64 61 74 61 62 61 73 65      |database|
00 70 67 62 65 6E 63 68      |.pgbench|
00 61 70 70 6C 69 63 61      |.applica|
74 69 6F 6E 5F 6E 61 6D      |tion_nam|
65 00 70 73 71 6C 00 63      |e.psql.c|
6C 69 65 6E 74 5F 65 6E      |lient_en|
63 6F 64 69 6E 67 00 55      |coding.U|
54 46 38 00 00              |TF8..|
```

上面的数据是从前端发送到后端的数据，以及大小（77）。

2.2.11 显示前端操作的 sql 语句

有时为了监控客户端的操作情况，需要知道客户端进行了哪些查询。则需要输出进行操作的 sql 语句。目前只支持 sqlserver 和 postgresql。则可以通过设置 config 标签下的 log_sql 为 true 即可。

2.2.12 在线更新配置

oneproxy-monitor 支持在线更新配置，从而保证在不中断前端连接的情况下实现 oneproxymonitor 的在线更新。操作的方法为：

- 1) 更新配置文件 config.ini 或者可执行程序。
- 2) 像第一次启动程序一样启动可执行程序(例如：./oneproxy-monitor)
经过上面的两步，就可以完成 oneproxymonitor 的在线更新配置。

注意点：

- 1) 需要配置配置文件中 oneproxymonitor 节下的 pidfile。例如：

```
[oneproxy]
logfile = oneproxymonitor_log.log
pidfile = oneproxymonitor_pid.pid
```

- 2) 在线更新配置使用了 pid 文件，如果没有 pid 文件，将无法让原来的 oneproxymonitor

退出。故启动后生成的 pid 文件不能人为删除，程序退出时会自动删除自身生成的 pid 文件。

- 3) 原来的 oneproxy-monitor 在新的 oneproxy-monitor 启动后就停止接收客户端的连接。但是不会马上退出，会等待已经连接的程序执行完毕后才退出。故如果存在长连接的话，会导致 oneproxy-monitor 很难退出。如果长连接一直在工作（有数据交换），原来的 oneproxy-monitor 不会退出，如果长连接没有工作（没有数据交换），则在指定的时间后被强制关闭。通过在 oneproxy 节下的 connecttimeout 来指定超时时间，单位为 second。这个超时时间是指没有数据交换后的超时时间，如果一直有数据交换，此时间不影响长连接的操作。默认为：86400 秒。

2.2.13 keepalive 功能

oneproxy-monitor 提供了 keepalive 功能，这样当程序在意外退出情况下，可以被父进程快速启动，从而不影响向前端提供服务。需要注意的是当 oneproxy-monitor 运行在前台时，可以通过 ctrl+c 的方式来退出程序。

配置方法：

oneproxy 节下面增加 keepalive=1。例如：(见黑体)

```
[oneproxy]
logfile = oneproxy_log.log
pidfile = oneproxy_pid.pid
listen_addr = 0.0.0.0
listen_port = 9999,7777,6666
httpserver_addr = 0.0.0.0
httpserver_port = 8080
log_level = error
data_dump = false
log_sql = false
clientusername = sa
clientpassword = 0000
keepalive = 1
acceptthreadnum = 1
maxconnectnum = 40000
```

在启动 oneproxy-monitor，通过 ps 可以看到有两个 oneproxy-monitor 的进程存在，一个是父进程，一个是子进程。

2.2.14 虚拟 ip(vip)功能

oneproxy-monitor 提供了 vip 功能，可以通过在配置文件中的 oneproxy 节下面增加 vip_ifname 和 vip_address 配置来使用。例如：(见黑色部分)

```
[oneproxy]
```

```
logfile = oneproxy_log.log
pidfile = oneproxy_pid.pid
listen_addr = 0.0.0.0
listen_port = 9999,7777,6666
httpserver_addr = 0.0.0.0
httpserver_port = 8080
log_level = error
data_dump = false
log_sql = false
clientusername = sa
clientpassword = 0000
keepalive = 1
acceptthreadnum = 1
maxconnectnum = 40000
vip_ifname = em1:0
vip_address = 172.30.12.22
```

上面的配置中 em1 是本主机的网卡名称。em1:0 的含义是在已经存在的物理网卡 em1 上面建立虚拟网卡 em1:0。虚拟 IP 地址为 172.30.12.22。前端应用程序则可以通过 172.30.12.22 来访问 oneproxy-monitor 了。

oneproxy-monitor 会在后台不停的检测当前的 vip 地址是否可以 ping 通，如果不通时，则会把 vip 设置在自身的物理网卡上面。从而达到局域网内多个 oneproxy-monitor 的协同工作。

2.2.15 审计功能

为了查看客户端进行了哪些查询，以及在查询时使用的条件等。oneproxy-monitor 提供了把 prepared 中的动态变量回填显示的功能。

可以通过在配置文件中[config]标签下增加 logclientsql=true 来开启此功能，默认情况下是关闭的。

输出的 sql 语句是回填值后的 sql 语句，输出到日志文件中。下面是输出的示例：

```
[2017-01-12 20:14:08][0x2fc1700][192.168.7.58:43564] select dbtype from dbo.bigtable
where id = 10
[2017-01-12 20:14:08][0x2fc1700][192.168.7.58:43564] set implicit_transactions on
[2017-01-12 20:14:08][0x2fc1700][192.168.7.58:43564] select dbtype from dbo.bigtable
where id = 10
[2017-01-12 20:14:08][0x2fc1700][192.168.7.58:43564] IF @@TRANCOUNT > 0
COMMIT TRAN
[2017-01-12 20:14:08][0x2fc1700][192.168.7.58:43564] IF @@TRANCOUNT > 0
COMMIT TRAN set implicit_transactions off
[2017-01-12 20:14:08][0x2fc1700][192.168.7.58:43564] select dbtype from dbo.bigtable
```

```
where id = 10
[2017-01-12 20:14:08][0x2fc1700][192.168.7.58:43564] begin tran
[2017-01-12 20:14:08][0x2fc1700][192.168.7.58:43564] select dbtype from dbo.bigtable
where id = 10
[2017-01-12 20:14:08][0x2fc1700][192.168.7.58:43564] commit
[2017-01-12 20:14:08][0x2fc1700][192.168.7.58:43564] select dbtype from dbo.bigtable
where id = 10
[2017-01-12 20:14:08][0x2fc1700][192.168.7.58:43564] set implicit_transactions on
[2017-01-12 20:14:08][0x2fc1700][192.168.7.58:43564] select id, name, age, dbtype from
dbo.bigtable where id = 10
[2017-01-12 20:14:08][0x2fc1700][192.168.7.58:43564] IF @@TRANCOUNT > 0
COMMIT TRAN
[2017-01-12 20:14:08][0x2fc1700][192.168.7.58:43564] select id, name, age, dbtype from
dbo.bigtable where id < 10
```

3. 安装

- 1) 下载可执行程序 and 配置文件

<https://github.com/onexsoft/oneproxy-monitor/tree/master/bin>

配置文件叫做 config.ini。配置文件只要是以.ini 结尾即可。

- 3) 把配置文件和可执行程序放在同一个目录下
- 4) 直接执行可执行程序即可。或者使用 -f 参数指定配置文件来启动可执行程序。

4. 获取帮助

- 1) 可以从平民软件官网获取联系方式
- 2) 加入群《数据库监控 521095285》
- 3) 关注 <https://github.com/onexsoft/oneproxy-monitor>

5 配置文件说明

配置文件的格式见附录。本节主要针对配置文件的组织关系进行说明，方便读者的使用与配置。

oneproxy-monitor 的配置文件被分成了 3 个大的部分：

第一部分：[oneproxy]节，这个节下面的配置信息是 oneproxymonitor 的全局配置信息。

第二部分：[database_xxxx]节，这个是配置后端数据库信息的节，节的名称必须为 database_ 开头。

第三部分：其他的节（除了[oneproxy]节,[database_xxxx]节），是数据库组的信息，这些信息与每个连接相关。

下面介绍每个节完整的配置：

5.1 [oneproxy]节的配置

配置名	默认值	取值范围	功能说明
logfile	oneproxy_log.log	有效文件名	
pidfile	oneproxy_pid.pid	有效文件名	见 2.2.12
listen_addr	0.0.0.0		
listen_port	9999		见 2.2.2
httpserver_addr	0.0.0.0		见 2.2.1
httpserver_port	8080		见 2.2.1
log_level	error	fatal, error, warning, info, debug	
data_dump	false	false, true	见 2.2.10
log_sql	false	false, true	见 2.2.11
tryconnservertimes	3	如果小于 1，自动设置为 1	连接数据库的重试次数
maxconnectnum	2000	如果小于 1，自动设置为 1	数据库的连接总数
keepalive	false	true, false	见 2.2.13
vip_ifname			见 2.2.14
vip_address			见 2.2.14
threadnum	0	如果小于等于 0，自动设置为 cpu core 数	工作线程数量
clientusername	admin		见 2.2.4
clientpassword	0000		见 2.2.4
poolconncheckactivetime	5	大于 0，单位为秒	见 2.2.5
poolconntimeoutreleasetime	60	大于 0，单位为秒	见 2.2.5
connecttimeout	86400	大于 0，单位为秒	见 2.2.12
acceptthreadnum	1	大于 0	见 2.2.3
listenbacklog	1000	大于等于 128	socket 的监听队列

5.2 [database_xxxx]节的配置

配置名	默认值	取值范围	功能说明
labelname	database_fakedb		配置中没有配置取节名称
host	127.0.0.1		数据库地址
port	1433		数据库端口
weightvalue	0		权值
username	sa		用户名
password	0000		密码
connectnumlimit	1000		连接数

5.3 数据库组节的配置

配置名	默认值	取值范围	功能说明
labelname	group_fake		配置中没有配置取节名
dbmastergroup	database_fakedb		master 数据库的节名（多个以逗号分隔）
dbslavegroup			slave 数据库的节名(多个以逗号分隔)
classname	FakeProtocol	SSProtocol, PGProtocol, FakeProtocol	处理类的名称。 SSProtocol 是 sql server 数据库的处理类 PGProtocol 是 postgresql 数据库的处理类 FakeProtocol 是直接转发的类
frontport	0		见 2.2.2
passwordseparate	true		见 2.2.4
readslave	true		见 2.2.6
useconnectionpool	true		见 2.2.5
clientauthmethod	md5		见 2.2.4
clearbackendconnection	true		见 2.2.5

6. 附录

6.1 混合数据库的配置样例

```
[oneproxy]
logfile = oneproxy_log.log
pidfile = oneproxy_pid.pid
listen_addr = 0.0.0.0
listen_port = 9999,7777,6666
clientusername = sa
clientpassword = 0000
```

```
[database_1433]
host = 192.168.7.27
port = 51915
username = sa
password = 0000
```

```
[database_1434]
host = 192.168.7.27
port = 55555
username = sa
password = abcdefuz
```

```
[database_1435]
host = 192.168.7.27
port = 30867
username = sa
password = 0000
```

```
[database_5432]
host = 172.30.12.12
port = 5432
username = db_user
password = 0000
```

```
[database_7777_5432]
host = 172.30.12.7
port = 5432
username = db_user
```



```
password = 0000

[database_6666_5432]
host = 172.30.12.6
port = 5432
username = db_user
password = 0000

[database_1521]
host = 172.30.12.10
port = 1521

[sqlserver_9999]
dbmastergroup = database_1433
dbslavegroup = database_1434
classname = SSProtocol
useconnectionpool = true
readslave = true
passwordseparate = true
frontport=9999

[postgresql_5432]
dbmastergroup = database_5432
dbslavegroup = database_7777_5432, database_6666_5432
classname = PGProtocol
frontport = 7777
passwordseparate=true
useconnectionpool=true
readslave = true
clearbackendconnection = true

[other_database]
dbmastergroup = database_1521
classname = FakeProtocol
frontport = 6666
```

6.2 sql server 数据库配置样例

```
[oneproxy]
logfile = oneproxy_log.log
```

```
pidfile = oneproxy_pid.pid
listen_addr = 0.0.0.0
listen_port = 9999,7777,6666
httpserver_addr = 0.0.0.0
httpserver_port = 8080
log_level = error
data_dump = false
log_sql = false
clientusername = sa
clientpassword = 0000
keepalive = 0
acceptthreadnum = 1
maxconnectnum = 40000
#threadnum = 1

[database_1433]
host = 192.168.7.27
port = 51915
username = sa
password = 0000

[database_1434]
host = 192.168.7.27
port = 55555
username = sa
password = abcdefuz

[database_1435]
host = 192.168.7.27
port = 30867
username = sa
password = 0000

[sql server_5432]
dbmastergroup = database_1433
dbslavegroup = database_1434, database_1435
classname = SSProtocol
frontport = 0
```

6.3 postgresql 数据库配置样例

```
[oneproxy]
logfile = oneproxy_log.log
pidfile = oneproxy_pid.pid
listen_addr = 0.0.0.0
listen_port = 9999,7777,6666
httpserver_addr = 0.0.0.0
httpserver_port = 8080
log_level = error
data_dump = false
log_sql = false
clientusername = sa
clientpassword = 0000
keepalive = 0
acceptthreadnum = 1
maxconnectnum = 40000
#threadnum = 1

[database_5432]
host = 172.30.12.12
port = 5432
username = db_user
password = 0000

[database_7777_5432]
host = 172.30.12.7
port = 5432
username = db_user
password = 0000

[database_6666_1435]
host = 172.30.12.6
port = 5432
username = db_user
password = 0000

[sql server_5432]
dbmastergroup = database_5432
dbslavegroup = database_7777_5432, database_6666_1435
classname = PGProtocol
```

```
frontport = 0
```

6.4 直接转发时的配置样例

```
[oneproxy]
logfile = oneproxy_log.log
pidfile = oneproxy_pid.pid
listen_addr = 0.0.0.0
listen_port = 9999,7777,6666
httpserver_addr = 0.0.0.0
httpserver_port = 8080
log_level = error
data_dump = false
log_sql = false
clientusername = sa
clientpassword = 0000
keepalive = 0
acceptthreadnum = 1
maxconnectnum = 40000
#threadnum = 1

[database_5432]
host = 172.30.12.12
port = 5432
username = db_user
password = 0000

[database_7777_5432]
host = 172.30.12.7
port = 5432
username = db_user
password = 0000

[sql server_5432]
dbmastergroup = database_5432
dbslavegroup = database_7777_5432
classname = FakeProtocol
frontport = 0
```



平民软件