agentserver 配置说明

Nginx 工程配置

agentserver是通过反向代理的方式来实现的。

反向代理方式实际上就是一台负责转发的代理服务器，貌似充当了真正服务器的功能，但实际上并不是，代理服务器只是充当了转发的作用，并且从真正的服务器那里取得返回的数据。nginx完成的就是这样的工作。

下面看看nginx需要做哪些配置：

- 设置**user**

  `user rxm108934 users;`（这个设置非常重要，很多时候就是因为user没有设置好导致运行不正确）

  `worker_processes 8;`

- 设置**root,location,proxy_pass**

  root 是文件存放的主目录

  location是用来定义url路径的

  proxy_pass 是url链接的agentserver服务

```
server {
    listen       80;
        server_name  localhost;
        root   /home/rxm108934;
        index  index.html index.htm index.php;

        location ~ \.php$ {
        fastcgi_pass   127.0.0.1:9000;
        fastcgi_index  index.php;
        fastcgi_param  SCRIPT_FILENAME  $document_root$fastcgi_script_name;
        include        fastcgi_params;
        }

        location /nginx_status {
        #stub_status on;
        access_log   off;
        }

        #agentserver
        location /agentserver/busservice{
        proxy_pass http://0.0.0.0:7143;

        add_header 'Access-Control-Allow-Origin' *;
        add_header 'Access-Control-Allow-Credentials' 'true';
        add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS';
        add_header 'Access-Control-Allow-Headers' 'DNT,X-Mx-ReqToken,Keep-Alive
anbo-Header,Client-SequenceId';

        }
        location /busservice{
        proxy_pass http://10.19.1.121:22248/;

        add_header 'Access-Control-Allow-Origin' *;
        add_header 'Access-Control-Allow-Credentials' 'true';
        add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS';
        add_header 'Access-Control-Allow-Headers' 'DNT,X-Mx-ReqToken,Keep-Alive
anbo-Header,Client-SequenceId';
```

需要链接到的服务的地址和端口号

- 添加需要的**service** 到**nginx**

到nginx.conf文件中copy 相应的service代码(譬如下图中driveservice)，修改location即可完成添加。

```
        #agentserver
        location /agentserver/busservice{
        proxy_pass http://0.0.0.0:7143;

        add_header 'Access-Control-Allow-Origin' *;
        add_header 'Access-Control-Allow-Credentials' 'true';
        add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS';
        add_header 'Access-Control-Allow-Headers' 'DNT,X-Mx-ReqToken,Keep-Alive,User-Agent,X-Reques
Wuranbo-Header,Client-SequenceId';

        }
        location /busservice{
        proxy_pass http://10.19.1.121:22248/;

        add_header 'Access-Control-Allow-Origin' *;
        add_header 'Access-Control-Allow-Credentials' 'true';
        add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS';
        add_header 'Access-Control-Allow-Headers' 'DNT,X-Mx-ReqToken,Keep-Alive,User-Agent,X-Reques
Wuranbo-Header,Client-SequenceId';

        }
        location /agentserver/driveservice{
                proxy_pass http://0.0.0.0:7143;

                add_header 'Access-Control-Allow-Origin' *;
                add_header 'Access-Control-Allow-Credentials' 'true';
                add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS';
                add_header 'Access-Control-Allow-Headers' 'DNT,X-Mx-ReqToken,Keep-Alive,User-Agent,
nt-Type,Wuranbo-Header,Client-SequenceId';

        }
        location /agentserver/driveservicedebug{
        proxy_pass http://0.0.0.0:1234;
```

- 启动**nginx**

启动nginx的时候一定要以当前用户启动，启动之后运行下面命令查看：

```
netstat -tulnp
```

如果可以看到有两个nginx（一个80，一个443），那么用户设置成功，启动正确。

```
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      7119/nginx
tcp        0      0 0.0.0.0:8080            0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:17776         0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:17777         0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:17779         0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:8182          0.0.0.0:*               LISTEN      -
tcp        0      0 100.81.11.237:8182      0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:442             0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:443             0.0.0.0:*               LISTEN      7119/nginx
tcp        0      0 127.0.0.1:13772         0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:9151            0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:15776         0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:15777         0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:19777         0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:15778         0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:15779         0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:41701         0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:7143            0.0.0.0:*               LISTEN      8473/./agentserver
tcp        0      0 127.0.0.1:199           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:3306            0.0.0.0:*               LISTEN      -
udp        0      0 0.0.0.0:111             0.0.0.0:*                           -
udp        0      0 100.81.11.237:123       0.0.0.0:*                           -
udp        0      0 127.0.0.1:123           0.0.0.0:*                           -
udp        0      0 0.0.0.0:123             0.0.0.0:*                           -
udp        0      0 0.0.0.0:161             0.0.0.0:*                           -
```

到此，nginx设置完成。接下来设置agentserver

agentserver 设置

- 设置agentserver的端口号

这个设置和nginx中的服务建立连接

```
#include "agentserver/trainservice.h"
#include "agentserver/drivedecoder.h"

DEFINE_string(server_address, "0.0.0.0:7143", "Default server address");

DEFINE_int32(worker_num, 4, "Default worker thread num");

// namespace agent{
// class AgentServer:public HttpServer {
```

- 在agentserver中添加service类

```
agent::CyclingService* cycling_service = new agent::CyclingService();
cycling_service->Initialize();
server->RegisterHttpService("/agentserver/cyclingservice", cycling_service);

agent::DriveDecoder::InitStatusMap();
agent::DriveService* drive_service = new agent::DriveService();
drive_service->Initialize();
server->RegisterHttpService("/agentserver/driveservice",drive_service);

// snowman ETA
agent::ETAService* eta_service = new agent::ETAService();
eta_service->Initialize();
server->RegisterHttpService("/agentserver/etaservice", eta_service);

// seamless service
agent::SeamlessService* seamless_service = new agent::SeamlessService();
seamless_service->Initialize();
server->RegisterHttpService("/agentserver/seamlessservice", seamless_service);
```

- blade build 编译 agentserver 为blade build添加配置

```
proto_library(
  name='route_proto',
  srcs=[
    'route.proto',
  ]
)
proto_library(
  name='walk_responce',
  srcs=[
    'walk.responce.proto',
  ]
)
cc_binary(
  name='agentserver',
  srcs=[
    'walkservice.cc',
    'nwalkservice.cc',
    'walkdecoder.cc',
    'cyclingdecoder.cc',
    'cyclingservice.cc',
    'loadhistorytrafficservice.cc',
    'util.cc',
    'driveservice.cc',
    'etaservice.cc',
    'seamlessservice.cc',
    'pathfeaturesservice.cc',
    "navidataservice.cc",
    'drivedecoder.cc',
    'drivedecoder21.cc',
    'drivedecoder22.cc',
    'drivedecoder24.cc',
    'drivedecoder25.cc',
    'drivedecoder252.cc',
    'drivedecoder253.cc',
    'drivedecoder30.cc',
    'busservice.cc',
    'trainservice.cc',
    'agentserver.cc',
    'constructpath_service.cc',
    'pathrestoration_service.cc',
```

配置添加完成之后，使用命令：

```
blade build
```

开始编译

agentserver使用说明(俊惠整理)

升级解码器时：

- 后台
- 开发： 将decoder文件拷贝到/home/www/TestTools/agentserver/onlinenavi/src/Decoder25下，运行命令：**blade build**，确认编译通过。 如果需要下发给前端新的JSON参数，请更改/home/www/TestTools/agentserver里的route.proto文件。执行blade build，生成的proto文件在 `./build64_release/agentserver/` 下。 请将route.pb.h或route.pb.cc拷贝到/home/www/TestTools/agentserver里。 需要下发给前端的参数，请更改drivedecoder25.h和drivedecoder25.cc文件。
- 编译： 运行命令：blade build。生成的文件agentserver在./agentserver/build64_release/agentserver/agentserver。
- 启动服务： kill掉现在执行的agentserver程序； 将可执行程序拷贝到：/home/www/TestTools/agentserver_online，替换路径下同名可执行文件agentserver； 启动程序： `nohup ./agentserver &`
- 前端

程序位置：/home/nginx_www

- **crontab**配置(定时执行任务)

```
/home/www/TestTools/agentserver_online/agent_server_monitor.sh
```