

Nginx反向代理

原创

GeorgeKai

2018-02-10 14:02:07

评论(0)

731人阅读

作者：Georgekai
归档：学习笔记
2018/2/7

Nginx反向代理

1.1 集群介绍

集群概念：一群干相同事情的服务器，称为集群

1.1.1 集群作用

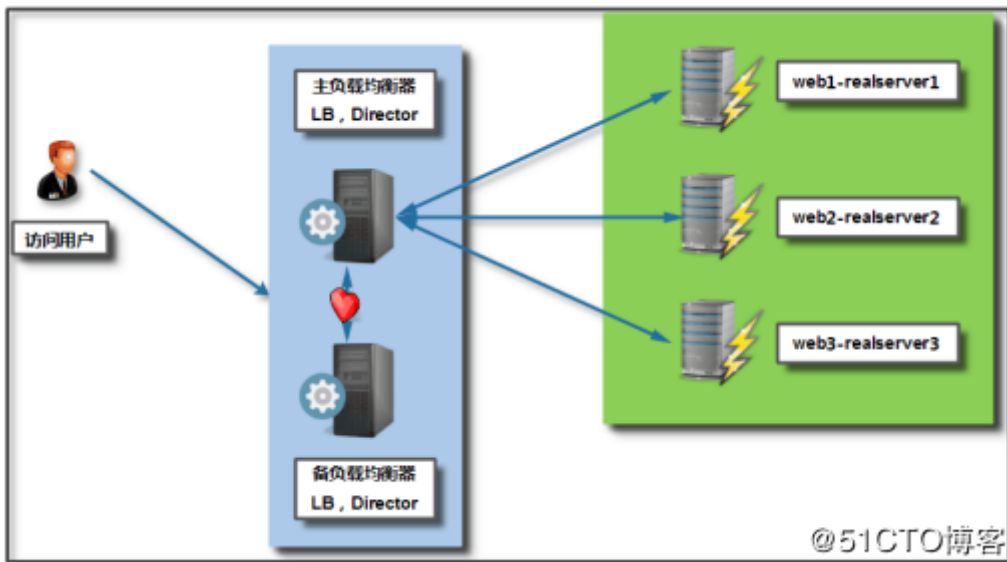
01. 处理高性能 (Performance)
02. 价格有效性 (Cost-effectiveness)
03. 可伸缩性 (Scalability)
04. 高可用性 (Availability)

集群概念的特点说明：高可用 高性能

1.1.2 负载均衡的作用

1. 实现用户访问数据的调度
2. 实现压力分担

Nginx基础知识-负载均衡概念介绍



1.1.3 负载均衡实现方法

1. 硬件实现负载均衡

1) F5

[点击查看源网页](#)



2) Netscaler

3) Radware

4) A10

2. 软件实现负载均衡

1) Nginx+Hearttbeat (高可用)

支持7层 (http https) 1.9以后也支持4层

2) LVS+keepalived (高可用)

只支持4层 (端口)

3) haproxy

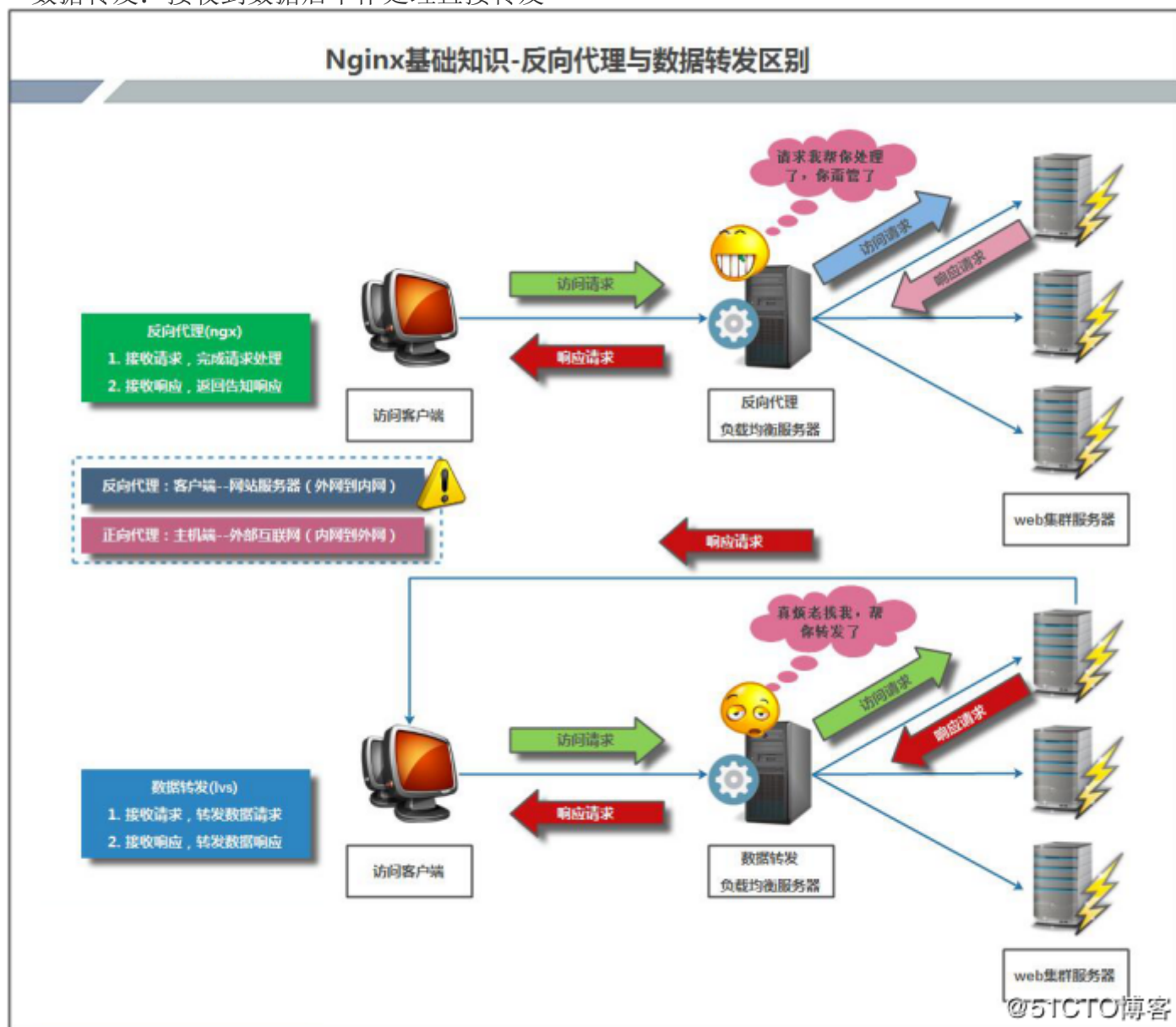
3. 方向代理概念说明

反向代理和数据转发的区别：

反向代理：把客户端请求发给服务端

正向代理：把服务端的请求发给客户端

数据转发：接收到数据后不作处理直接转发



1.2 部署nginx反向代理负载均衡服务

1.2.1 部署nginx网站集群服务器

第一部分：准备环境：部署nginx网站集群服务器（web01 web02 web03）

```
server {
    listen      80;
    server_name bbs.etiantian.org;
    root        html/bbs;
    index       index.html index.htm;
}

server {
    listen      80;
    server_name www.etiantian.org;
    root        html/www;
    index       index.html index.htm;
}
```

说明：将以上虚拟主机配置统一放置到web01 web02 web03服务器中

```
# 在站点目录下创建测试文件
for name in www bbs;do echo "$(hostname -i) $(hostname) $name"
>>/application/nginx/html/$name/george.html;done

for name in www bbs;do cat /application/nginx/html/$name/george.html;done
```

1.2.2 部署nginx反向代理服务器

1. 在db01服务器上——测试所有web服务节点是否能够正常访问（在命令行解析，不用添加到hosts文件）

```
[root@lb01 ~]# for name in www bbs;do curl -H host:${name}.etiantian.org
172.16.1.7/george.html;done
172.16.1.7 web01 www
172.16.1.7 web01 bbs
[root@lb01 ~]# for name in www bbs;do curl -H host:${name}.etiantian.org
172.16.1.8/george.html;done
172.16.1.8 web02 www
172.16.1.8 web02 bbs
[root@lb01 ~]# for name in www bbs;do curl -H host:${name}.etiantian.org
172.16.1.9/george.html;done
172.16.1.9 web03 www
172.16.1.9 web03 bbs
```

PS: curl -H host:www.etiantian.org 172.16.1.7/george.html 在命令行解析，不用添加到hosts文件

2. 配置nginx主配置文件编写

- ① upstream
- ② proxy_pass

类似于ansible:

```
ansible: hosts          nginx
[georgekai]            upstream georgekai {
172.16.1.31             server 172.16.1.31:80;
172.16.1.32             server 172.16.1.32:80;
172.16.1.33             server 172.16.1.33:80;
                        }

ansible georgekai       proxy_pass http://georgekai
```

```
[root@lb01 ~]# cat /application/nginx/conf/nginx.conf
worker_processes 1;
error_log /tmp/error.log error;
events {
    worker_connections 1024;
}
http {
    include mime.types;
    default_type application/octet-stream;
    sendfile on;
    log_format main '$remote_addr - $remote_user [$time_local] "$request" '
        '$status $body_bytes_sent "$http_referer" '
        '"$http_user_agent" "$http_x_forwarded_for"';
    access_log logs/access.log main;
    keepalive_timeout 65;
    upstream georgekai{
        server 172.16.1.7:80;
        server 172.16.1.8:80;
        server 172.16.1.9:80;
    }
    server {
        listen 80;
        server_name bbs.etiantian.org;
        root html/bbs;
        index index.html index.htm;
        location / {
            proxy_pass http://georgekai;
        }
    }
}
```

```
}
```

3. 进行负载均衡测试

```
[root@lb01 ~]# curl -H host:bbs.etiantian.org 10.0.0.5/george.htmlf
172.16.1.7 web01 bbs
[root@lb01 ~]# curl -H host:bbs.etiantian.org 10.0.0.5/george.html
172.16.1.8 web02 bbs
[root@lb01 ~]# curl -H host:bbs.etiantian.org 10.0.0.5/george.html
172.16.1.9 web03 bbs
```

1.2.3 负载均衡模块常用功能

weight #实现权重负载访问功能（能者多劳）

max_fails #定义后端访问的失败次数

fail_timeout #定义后端失败重试的间隔（单位是秒）

backup #定义后端服务的热备节点（其他负载节点服务器都挂了，使用备份）

配置：

```
worker_processes 1;
events {
    worker_connections 1024;
}
http {
    include mime.types;
    default_type application/octet-stream;
    sendfile on;
    keepalive_timeout 65;
    include extra/blog.conf;
    upstream georgekai {
        server 10.0.0.7:80 weight=3 max_fails=3 fail_timeout=10s;
        server 10.0.0.8:80 weight=1;
        server 10.0.0.9:80 weight=1 backup;
    }
    server {
        listen 80;
        server_name bbs.etiantian.org;
        root html/bbs;
        index index.html index.htm;
        location / {
            proxy_pass http://georgekai;
        }
    }
}
```

upstream 模块内参数	参数说明
server 10.0.10.8:80	负载均衡后面的 RS 配置，可以是 IP 或域名，如果端口不写，默认是 80 端口。 高并发场景下，IP 可换成域名，通过 DNS 做负载均衡
weight=1	代表服务器的权重，默认值是 1。权重数字越大表示接受的请求比例越大
max_fails=5	Nginx 尝试连接后端主机失败的次数，这个值是配合 proxy_next_upstream、fastcgi_next_upstream 和 memcached_next_upstream 这三个参数来使用的。 当 nginx 接收后端服务器返回这三个参数定义的状态码时，会将这个请求转发给正常工作的后端服务器，例如 404、502、503、Max_fails 的默认值是 1； 企业场景下建议 2-3 次。如京东 1 次，蓝汛 10 次，根据业务需求去配置
fail_timeout=10s	在 max_fails 定义的失败次数后，距离下次检查的间隔时间，默认是 10s；如果 max_fails 是 5，它就检测 5 次，如果 5 次都是 502，那么，它会根据 fail_timeout 的值，等待 10s 再去检查，还是只检查一次，如果持续 502，在不重新加载 Nginx 配置的情况下，每隔 10s 都只检查一次。常规业务 2~3 秒比较合理，比如京东 3 秒，蓝汛 3 秒，可根据业务需求去配置。
backup (演示说明)	热备配置（RS 节点的高可用），当前面激活的 RS 都失败后会自动启用热备 RS 这标志着这个服务器作为备份服务器，若主服务器全部宕机了，就会向它转发请求 注意：当负载均衡算法为 ip_hash 时，后端服务器在负载均衡调度中的状态不能是 weight 和 backup
down==#	这标志着服务器永远不可用，这个参数可配合 ip_hash 使用；类似 注@51CTO博客

1.2.4 模块调度算法：

1. 定义轮询调度算法-rr（默认调度算法）
2. 定义权重调度算法-wrr
3. 定义静态调度算法-ip_hash（访问多的话，会负载不均）

PS: 根据用户源地址算出一个范围，那么下次这个用户再次访问，会根据这个范围还分配给那个对应的固定web 服务器

配置方法：

```
upstream george {
ip_hash
    server 172.16.1.7:80 weight=3 max_fails=3 fail_timeout=10s;
    server 172.16.1.8:80 weight=1;
    server 172.16.1.9:80 weight=1 backup;
}
```

4. 定义最小的连接数-least_conn
哪个服务器连接数少，分配给它（谁闲着给谁）
5. fair（动态调度算法）
会根据后端服务端的实际情况来分配，灵活分配

1.2.5 实现反向代理服务器根据用户请求的虚拟主机信息 而显示页面内容（可查看请求报文）

```
worker_processes 1;
events {
    worker_connections 1024;
}
http {
    include mime.types;
    default_type application/octet-stream;
    sendfile on;
    keepalive_timeout 65;
```

```

include      extra/blog.conf;
    upstream george {
        #ip_hash;
        server 10.0.0.7:80 weight=3 max_fails=3 fail_timeout=10s;
        server 10.0.0.8:80 weight=1;
        server 10.0.0.9:80 weight=1;
    }
server {
    listen      80;
    server_name bbs.etiantian.org;
    root        html/bbs;
    index       index.html index.htm;
    location / {
        proxy_pass http://georgekai;
        proxy_set_header Host $host; --- 修改请求头里面host参数信息（curl -v可以查看请求头信息）
                                #不加这一条，默认只会访问第一个虚拟主机的站点信息
    }
}

server {
    listen      80;
    server_name www.etiantian.org;
    root        html/www;
    index       index.html index.htm;
    location / {
        proxy_pass http://georgekai;
        proxy_set_header Host $host;
    }
}

```

1.2.6 实现用户经过反向代理访问后端web服务显示真实用户IP地址信息

就是在web服务部上用`tail -f /application/nginx/logs/access.log`，会在后面显示出用户的真是IP地址

1. web服务器配置

```

log_format main '$remote_addr - $remote_user [$time_local] "$request" '
                '$status $body_bytes_sent "$http_referer" '
                '"$http_user_agent" "$http_x_forwarded_for"';
access_log logs/access.log main;

```

2. 反向代理服务器配置

```

[root@lb01 ~]# cat /application/nginx/conf/nginx.conf
worker_processes 1;
error_log /tmp/error.log error;
events {
    worker_connections 1024;
}
http {
    include      mime.types;
    default_type application/octet-stream;
    sendfile     on;
    log_format main '$remote_addr - $remote_user [$time_local] "$request" '
                    '$status $body_bytes_sent "$http_referer" '
                    '"$http_user_agent" "$http_x_forwarded_for"';
    access_log off;
    keepalive_timeout 65;
    upstream georgekai{
        server 10.0.0.7:80;
        server 10.0.0.8:80;
        server 10.0.0.9:80;
    }

    server {
        listen      80;
        server_name bbs.etiantian.org;
        location / {

```

```

        proxy_pass http://georgekai;;
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-For $remote_addr;    S ——$remote_addr就是客户访客的
IP
    }
}
server {
    listen      80;
    server_name www.etiantian.org;
    location / {
        proxy_pass http://georgekai;
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-For $remote_addr; ——$remote_addr就是客户访客的IP
    }
}
}

```

3. 测试

```
tail -f logs/access.log
```

```
172.16.1.6 - - [09/Feb/2018:00:13:49 +0800] "GET /george.html HTTP/1.0" 200 17 "-" "Mozilla/5.0
(Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.84
Safari/537.36" "10.0.0.1"
```

1.2.7 nginx反向代理常见问题:

1. DNS域名解析, 应该将域名解析为代理服务器地址
2. 区分nginx服务, 1b01上部署的是nginx代理服务器, 在web服务器上查看访问情况(日志信息)
3. 访问测试异常(浏览器软件造成测试效果不正确, 建议用谷歌)

1.2.8 复制均衡反向代理根据请求地址分配 (/static)

需求信息

```

www.etiantian.org/static 10.0.0.7:80 html/www/static static静态服务器
www.etiantian.org/upload 10.0.0.8:80 html/www/upload upload服务器
www.etiantian.org/       10.0.0.9:80 html/www      默认

```

部署web服务器测试环境:

1. 配置web01服务器环境:

```

cd /application/nginx
mkdir html/www/static
echo "10.0.0.7 web01 static" >>html/www/static/nana.html
cat html/www/static/nana.html

```

2. 配置web02服务器环境:

```

cd /application/nginx
mkdir html/www/upload
echo "10.0.0.8 web02 upload" >>html/www/upload/nana.html
cat html/www/upload/nana.html

```

3. 配置web03服务器环境:

```

cd /application/nginx
echo "10.0.0.9 web03 default" >>html/www/nana.html
cat html/www/nana.html

```

4. 利用nginx反向代理服务器进行测试访问

```

curl -H host:www.etiantian.org 10.0.0.7/static/nana.html
curl -H host:www.etiantian.org 10.0.0.8/upload/nana.html
curl -H host:www.etiantian.org 10.0.0.9/nana.html

```

5. 编写nginx反向代理配置文件

第一个部分: upstream配置


```
upstream static {
    server 10.0.0.7:80;
}
```

```
upstream upload {
    server 10.0.0.8:80;
}
```

```
upstream default {
    server 10.0.0.9:80;
}
```

第二个部分：proxy_pass配置

```
location ~* /static {
    proxy_pass http://static;
    proxy_set_header Host $host;
    proxy_set_header X-Forwarded-For $remote_addr;
}
```

```
location ~* /upload {
    proxy_pass http://upload;
    proxy_set_header Host $host;
    proxy_set_header X-Forwarded-For $remote_addr;
}
```

```
location / {
    proxy_pass http://default;
    proxy_set_header Host $host;
    proxy_set_header X-Forwarded-For $remote_addr;
}
```

```
worker_processes 1;
error_log /tmp/error.log error;
events {
    worker_connections 1024;
}
http {
    include mime.types;
    default_type application/octet-stream;
    sendfile on;
    log_format main '$remote_addr - $remote_user [$time_local] "$request" '
        '$status $body_bytes_sent "$http_referer" '
        '"$http_user_agent" "$http_x_forwarded_for"';
    access_log off;
    keepalive_timeout 65;
    upstream static {
        server 10.0.0.7:80;
    }

    upstream upload {
        server 10.0.0.8:80;
    }

    upstream default {
        server 10.0.0.9:80;
    }
}
```

```
server {
    listen 80;
    server_name www.etiantian.org;
    location /static/ {
        proxy_pass http://static;
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-For $remote_addr;
    }
}
```

```

    }

    location /upload/ {
        proxy_pass http://upload;
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-For $remote_addr;
    }
    location / {
        proxy_pass http://default;
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-For $remote_addr;
    }
}
}

```

3. 进行客户端访问测试

windows上测试:

```

http://www.etiantian.org/static/nana.html
http://www.etiantian.org/upload/nana.html
http://www.etiantian.org/nana.html

```

linux上测试:

```

[root@lb01 nginx]# curl -H host:www.etiantian.org 10.0.0.7/static/nana.html
10.0.0.7 web01 static
[root@lb01 nginx]# curl -H host:www.etiantian.org 10.0.0.8/upload/nana.html
10.0.0.8 web02 upload
[root@lb01 nginx]# curl -H host:www.etiantian.org 10.0.0.9/nana.html
10.0.0.9 web03 default

```

1.2.9 根据客户端的设备（user_agent）转发实践

1. 部署web服务器测试环境

配置web01服务器环境:

```

cd /application/nginx
echo "10.0.0.7 web01 mobile" >>html/www/nana.html      ---手机端访问
cat html/www/nana.html  ---检查

```

配置web02服务器环境:

```

cd /application/nginx
echo "10.0.0.8 web02 chrom" >>html/www/nana.html      --- 谷歌浏览器访问
cat html/www/upload/nana.html  ---检查

```

配置web03服务器环境:

```

cd /application/nginx
echo "10.0.0.9 web03 default" >>html/www/nana.html    --- 其他浏览器客户端访问
cat html/www/nana.html  ---检查

```

2. 利用nginx反向代理服务器进行测试访问

```

curl -H host:www.etiantian.org 10.0.0.7/nana.html

curl -H host:www.etiantian.org 10.0.0.8/nana.html

curl -H host:www.etiantian.org 10.0.0.9/nana.html

```

3. 编写nginx反向代理配置文件

```

worker_processes 1;
events {
    worker_connections 1024;
}
http {
    include mime.types;

```

```
default_type application/octet-stream;
sendfile on;
keepalive_timeout 65;
log_format main '$remote_addr - $remote_user [$time_local] "$request" '
                '$status $body_bytes_sent "$http_referer" '
                '"$http_user_agent" "$http_x_forwarded_for"';

upstream mobile {
    server 10.0.0.7:80 ;
}

upstream PC {
    server 10.0.0.8:80 ;
}

upstream default {
    server 10.0.0.9:80 ;
}

server {
    listen 80;
    server_name www.etiantian.org ;
    location / {
        if ($http_user_agent ~* "iphone")
        {
            proxy_pass http://mobile ;
        }
        if ($http_user_agent ~* "Chrome")
        {
            proxy_pass http://PC ;
        }
        proxy_pass http://default ;
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-For $remote_addr;
    }
    access_log logs/access_www.log
}
main;
```

4. 进行客户端访问测试

PS: -A: 表示客户端设备类型

```
[root@lb01 nginx]# curl -A iphone www.etiantian.org/nana.html
10.0.0.7 web01 mobile
```

```
[root@lb01 nginx]# curl -A chrome www.etiantian.org/nana.html
10.0.0.8 web02 chrom
```

```
[root@lb01 nginx]# curl -A georgekai www.etiantian.org/nana.html
10.0.0.9 web03 default
```

小伙伴们可以关注我的微信公众号: linux运维菜鸟之旅



关注“中国电信天津网厅”公众号，首次绑定可免费领2G流量，为你的学习提供流量！



版权声明：原创作品，如需转载，请注明出处。否则将追究法律责任