

]

The state of the art of nginx.conf scripting

The state of the art of **nginx.conf** scripting

☺*agentzh@gmail.com*☺

章亦春 (*agentzh*)

2010.10

```
$ nginx -c /path/to/nginx.conf
```



\$ ps aux | grep nginx

root 2003 0.0 0.0 25208 412 ? Ss 10:08 0:00 nginx: **master process nginx**

nobody 2004 0.0 0.0 25608 1044 ? S 10:08 0:00 nginx: **worker process**

nobody 2005 0.0 0.0 25608 1044 ? S 10:08 0:00 nginx: **worker process**

```
# nginx.conf
worker_processes 2;
events {
    worker_connections 1024;
}
http {
    ...
    server {
        listen 80;
        server_name localhost;
        ...

        location / {
            root /var/www;
            index index.html index.htm;
        }
    }
}
```

♡ *Hello World* on the *nginx* land

```
0100000000010100011011000000100101100011
1100010111010001000111111111110100000100
00101001011000011010111011010110110010001
01101100000101011001000100001110001001111
0100110010110100110110100111101111011110
0001101000100011101001010001101000011010
0100100110100010001010001110
10001001int main()
010101001{
1110011000    printf("Hello World");
001000001111    return 42;
00011010001000111010010110001101000011010
01001001101111010111011110000001010001110
1000100100010101100100111011101000101111
010101001110011010101110001010100011000
11100110000011011111010100111110001100
00100000111111101010010010011010101110110
```

enable the ngx_echo module in your nginx build

**\$./configure --prefix=/opt/nginx **

--add-module=/path/to/echo-nginx-module

```
location = '/hello' {  
  echo "hello, world!";  
}
```



```
$ curl 'http://localhost/hello'  
hello, world!
```

```
# enable the ngx_set_misc module and  
# Marcus Clyne's ngx_devel_kit in your nginx build  
$ ./configure --prefix=/opt/nginx \  
    --add-module=/path/to/echo-nginx-module \  
    --add-module=/path/to/ngx_devel_kit \  
    --add-module=/path/to/set-misc-nginx-module
```

```
location = '/hello' {  
    set_unescape_uri $person $arg_person;  
    set_if_empty $person 'anonymous';  
    echo "hello, $person!";  
}
```

```
$ curl 'http://localhost/hello?person=agentzh'  
hello, agentzh!
```

```
$ curl 'http://localhost/hello'  
hello, anonymous!
```

♥ Using *subrequests* to do mashup

```
location = '/merge' {  
    echo '[';  
    echo_location_async /moon;  
    echo ',';  
    echo_location_async /earth;  
    echo ']';  
}  
location /moon {  
    echo '"moon"';  
}  
location /earth {  
    echo '"earth"';  
}
```

```
$ curl 'http://localhost/merge'  
[  
"moon"  
,  
"earth"  
]
```

```
# (not quite) REST interface to our memcached server
# at 127.0.0.1:11211
location = /memc {
    set $memc_cmd $arg_cmd;
    set $memc_key $arg_key;
    set $memc_value $arg_val;
    set $memc_exptime $arg_exptime;

    memc_pass 127.0.0.1:11211;
}
```



```
$ curl 'http://localhost/memc?cmd=flush_all';
```

OK

```
$ curl 'http://localhost/memc?cmd=replace&key=foo&val=FOO';
```

NOT_STORED

```
$ curl 'http://localhost/memc?cmd=add&key=foo&val=Bar&exptime=60';
```

STORED

```
$ curl 'http://localhost/memc?cmd=replace&key=foo&val=Foo';
```

STORED

```
$ curl 'http://localhost/memc?cmd=set&key=foo&val=Hello';
```

STORED

```
$ curl 'http://localhost/memc?cmd=get&key=foo';
```

Hello

```
$ curl 'http://localhost/memc?cmd=delete&key=foo';
```

DELETED

♥ Memcached **connection pool** support



enable Maxim Dounin's ngx_http_upstream_keepalive module

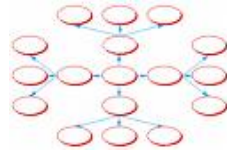
in your nginx build

```
$ ./configure --prefix=/opt/nginx \  
    --add-module=/path/to/echo-nginx-module \  
    --add-module=/path/to/memc-nginx-module \  
    --add-module=/path/to/ngx_http_upstream_keepalive
```

```
http {  
    ...  
    upstream my_memc_backend {  
        server 127.0.0.1:11211;  
  
        # a connection pool that can cache  
        #   up to 1024 connections  
        keepalive 1024 single;  
    }  
    ...  
}
```

```
location = /memc {  
    ...  
    memc_pass my_memc_backend;  
}
```

♥ Memcached server *hashing* based on user keys
(Hey, memcached **cluster**!)



enable the ngx_set_misc module and Marcus Clyne's

ngx_devel_kit again in your nginx build

**\$./configure --prefix=/opt/nginx **

**--add-module=/path/to/memc-nginx-module **

**--add-module=/path/to/ngx_devel_kit **

--add-module=/path/to/set-misc-nginx-module

```
http {  
    upstream A {  
        server 10.32.110.5:11211;  
    }  
    upstream B {  
        server 10.32.110.16:11211;  
    }  
    upstream C {  
        server 10.32.110.27:11211;  
    }  
    upstream_list my_cluster A B C;  
    ...  
}
```

```
location = /memc {  
    set $memc_cmd $arg_cmd;  
    set $memc_key $arg_key;  
    set $memc_value $arg_val;  
    set $memc_exptime $arg_exptime;  
  
    # hashing the $arg_key to an upstream backend  
    # in the my_cluster upstream list, and set $backend:  
    set_hashed_upstream $backend my_cluster $arg_key;  
  
    # pass $backend to memc_pass:  
    memc_pass $backend;  
}
```

♡ Some *non-blocking* MySQL love



```
# install libdrizzle first and then
# enable the ngx_drizzle and ngx_rds_json
# modules in your nginx build
$ ./configure --prefix=/opt/nginx \
    --add-module=/path/to/drizzle-nginx-module \
    --add-module=/path/to/rds-json-nginx-module
```

```
http {  
    upstream my_mysql_backend {  
        drizzle_server 127.0.0.1:3306 dbname=test  
        password=some_pass user=monty  
        protocol=mysql;  
    }  
    ...  
}
```

```
location = /cats {  
    drizzle_query 'select * from cats';  
    drizzle_pass my_mysql_backend;  
    rds_json on;  
}
```

```
$ curl 'http://localhost/cats'
```

```
[{"name":"Jerry","age":1},{"name":"Tom","age":3}]
```


♥ mysql connection pool support



```
http {  
    upstream my_mysql_backend {  
        drizzle_server 127.0.0.1:3306 dbname=test  
        password=some_pass user=monty  
        protocol=mysql;  
  
        # a connection pool that can cache up to  
        # 200 mysql TCP connections  
        drizzle_keepalive max=200 overflow=reject;  
    }  
    ...  
}
```

♥ Mysql **cluster** *hashing* love



re-enable the ngx_set_misc module and Marcus Clyne's

ngx_devel_kit in your nginx build

```
$ ./configure --prefix=/opt/nginx \  
    --add-module=/path/to/drizzle-nginx-module \  
    --add-module=/path/to/rds-json-nginx-module \  
    --add-module=/path/to/ngx_devel_kit \  
    --add-module=/path/to/set-misc-nginx-module
```

```
http {  
    upstream A {  
        drizzle_server ...;  
    }  
    upstream B {  
        drizzle_server ...;  
    }  
    upstream C {  
        drizzle_server ...;  
    }  
    upstream_list my_cluster A B C;  
    ...  
}
```

```
location ~ '^/cat/(.*)' {  
    set $name $1;  
    set_quote_sql_str $quoted_name $name;  
    drizzle_query "select *  
        from cats  
        where name=$quoted_name";  
  
    set_hashed_upstream $backend my_cluster $name;  
    drizzle_pass $backend;  
  
    rds_json on;  
}
```

♡ *ngx_postgres* has **already** landed.

Thanks Piotr Sikora!

http://github.com/FRiCKLE/ngx_postgres



```
# configure the PostgreSQL upstream backend
upstream my_pg_backend {
    postgres_server 10.62.136.3:5432 dbname=test
    user=someone password=123456;
}
```



```
location /cats {  
  postgres_query 'select * from cats';  
  postgres_pass my_pg_backend;  
  rds_json on;  
}
```

```
$ curl 'localhost/cats'
```

```
[{"name":"Marry","age":32},{"name":"Bob","age":12}]
```

♡ Everything is also *non-blocking* as ngx_drizzle.

Thanks to libpq's nonblocking API!

♡ Construct fully *RESTful* queries
in a **single** location

```
location ~ '^/cat/(\d+)' {  
    set $id $1;  
    set_form_input $name;  
    set_quote_sql_str $quoted_name $name;  
  
    postgres_query GET "select * from cats where id=$id";  
    postgres_query DELETE "delete from cats where id=$id";  
    postgres_query POST "insert into cats (id, name) values($id, $quoted_name)";  
  
    postgres_pass my_pg_backend;  
}
```

♡ Qunar.com is running *ngx_postgres* + *ngx_rds_json*
in **production**.

Thanks Liseen Wan's promotion!

北京儒家酒店_北京儒家酒店预订/价格查询-去哪儿Qunar.com - Windows Internet Explorer

http://hotel.qunar.com/city/beijing_city/q-%E5%84%B2%E5%AE%B6#fromDate=2010-06-20&toDate=2010-06-23

文件(F) 编辑(E) 查看(V) 收藏夹(A) 工具(T) 帮助(H)

北京儒家酒店_北京儒家酒店预订/价格查询-去哪儿

☐ ¥500以上
 ☐ 中关村、上地
 [更多商圈](#)
☐ 五星级/豪华
 ☐ 新开业/装修
 ☐ 会议室

93家酒店满足条件：评分从高到低 | 儒家 清空所有条件



Qunar城市百科：酒店品牌推荐>如家

【大众住宿领导者】国内排名第一的经济型连锁酒店，中国经济型酒店的先行者与领导者。08年第一家和颐酒店（上海漕宝路店）在上海开业，如家开始进军中高端商务酒店领域。作为上海的本土酒店品牌，如家依托上海世博的辐射效应，在上海和长三角周边地区都取得了非常高的入住率。同时还推出了世博绿色通道，世博门票、住宿一站搞定。

酒店名/地址	用户点评	最低报价
1 如家快捷北京天坛店 连锁经济型 商旅推荐 位于崇文区天坛东路，地处南二环以里，毗邻享誉世界的中国古代建筑明珠——天坛公园。网友评价说“小吃也挺多，价格比较合理，装潢很好”。	4.5分/209条评论  在该城市排39位	¥159起 查看详情
2 如家快捷酒店北京西客站店 连锁经济型 商旅推荐 位于北京西站北广场，毗邻军事博物馆、中央电视塔、玉渊潭公园、北京茶叶一条街、白云观、中华世纪坛等旅游景点。网友评价说“尤其是环境卫生，周边小吃多”。	4.5分/44条评论  在该城市排39位	¥261起 查看详情
3 如家快捷酒店北京景泰桥珐琅厂店 连锁经济型	4.5分/113条评论	¥169起

大图模式 ☒ 随屏幕浮动 ☐ 随地图缩放



地图数据 ©2010 M

http://hotel.qunar.com/city/beijing_city/dt-3828/#fromDate=2010-06-20&toDate=2010-06-23

Internet 100%



♡ *Caching* database responses using **memcached**
via ngx_srcache and ngx_memc.

<http://github.com/agentzh/srcache-nginx-module>

enable the ngx_srcache and other

modules in your nginx build

**\$./configure --prefix=/opt/nginx **

**--add-module=/path/to/srcache-nginx-module **

**--add-module=/path/to/rds-json-nginx-module **

**--add-module=/path/to/memc-nginx-module **

--add-module=/path/to/drizzle-nginx-module

♡ It's very important to put `ngx_srcache` *before* `ngx_rds_json` during nginx configure so that we cache the final JSON rather than RDS.

configure the mysql upstream backend

upstream mysql_backend {

drizzle_server 127.0.0.1:3306 dbname=test

password=some_pass user=monty protocol=mysql;

}

configure the cache storage location

location /memc {

internal;

set \$memc_key \$query_string;

set \$memc_exptime 300;

memc_pass 127.0.0.1:11211;

}

```
location /cats {  
    srcache_fetch GET /memc $uri;  
    srcache_store PUT /memc $uri;  
  
    default_type application/json;  
  
    drizzle_pass mysql_backend;  
    drizzle_query 'select * from cats';  
  
    rds_json on;  
}
```

```
$ curl 'localhost/cats'
```

```
[{"name":"Marry","age":32}, {"name":"Bob","age":12}]
```

if it is a cache miss

\$ memcached -vvv -p 11211

...

<10 new client connection

<10 get /cats

> NOT FOUND /cats

>10 END

<10 connection closed.

<10 new client connection

<10 set /cats 0 300 44

> NOT FOUND /cats

>10 STORED

<10 connection closed.

if it is a cache hit

\$ memcached -vvv -p 11211

...

<10 new client connection

<10 get /cats

> FOUND KEY /cats

>10 sending key /cats

>10 END

<10 connection closed.

♡ ngx_lua is quite usable *now*!

<http://github.com/chaoslawful/lua-nginx-module>



♡ chaoslawful is *crazy*!



first install lua (or even luajit) into your system...

enable the ngx_lua module in your nginx build

```
$ ./configure --prefix=/opt/nginx \  
    --add-module=/path/to/ngx_devel_kit \  
    --add-module=/path/to/echo-nginx-module \  
    --add-module=/path/to/lua-nginx-module
```

```
location = /adder {  
    set_by_lua $res  
        "local a = tonumber(ngx.arg[1])  
        local b = tonumber(ngx.arg[2])  
        return a + b" $arg_a $arg_b;  
  
    echo $res;  
}
```

```
$ curl 'localhost/adder?a=25&b=75'  
100
```

```
location = /fib {  
    set_by_lua $res "  
        function fib(n)  
            if n > 2 then  
                return fib(n-1) + fib(n-2)  
            else  
                return 1  
            end  
        end  
    end  
    local num = tonumber(ngx.arg[1])  
    return fib(num)  
" $arg_n;  
  
    echo $res;  
}
```

```
$ curl 'localhost/fib?n=10'
```

```
55
```


♥ or use *external* Lua script file...

```
location = /fib {  
    set_by_lua_file $res "conf/fib.lua" $arg_n;  
  
    echo $res;  
}
```

-- conf/fib.lua file

function fib(n)

if n > 2 then

return fib(n-1) + fib(n-2)

else

return 1

end

end

local num = tonumber(ngx.arg[1])

return fib(num)

\$ *Complex* database cluster hashing
can also be done in Lua

```
http {  
    upstream A {  
        drizzle_server ...;  
    }  
    upstream B {  
        drizzle_server ...;  
    }  
    upstream C {  
        drizzle_server ...;  
    }  
    ...  
}
```

```
location ~ '^/user/(\d+)' {  
    set $uid $1;  
    set_by_lua_file $backend "conf/hash.lua" $uid;  
    if ($backend = "") {  
        return 400; break;  
    }  
  
    drizzle_query "select * from users  
        where uid=$uid";  
    drizzle_pass $backend;  
    rds_json on;  
}
```

-- hash.lua

function hash(uid)

if uid > 0 and uid <= 1200 then return 'A' end

if uid > 1200 and uid <= 5300 then return 'B' end

if uid > 5300 and uid <= 7100 then return 'C' end

return ''

end

return hash(tonumber(ngx.arg[1]))

♥ Use **Lua** to code up nginx *content handler* directly


```
location = /lua {  
    content_by_lua 'ngx.say("Hello, Lua!");'  
}
```

```
$ curl 'localhost/lua'
```

Hello, Lua!

♡ ...and we can read arbitrary *nginx variables*
from within our Lua **content** handler!

```
location = /hello {  
    content_by_lua 'local who = ngx.var.arg_who  
        ngx.say("Hello, ", who, "!");  
    }
```

```
$ curl 'localhost/hello?who=agentzh'  
Hello, agentzh!
```

♡ We can also put Lua code into
external .lua **file** to eliminate escaping nightmare.

```
location /foo {  
    ...  
    content_by_lua_file /path/to/your/lua-file.lua;  
}
```

♥ We can also issue nginx *subrequests* directly from within Lua content handler *now*!


```
location /other {  
    echo "hello, world";  
}
```

transparent non-blocking I/O in Lua

```
location /lua {  
    content_by_lua '  
        local res = ngx.location.capture("/other")  
        if res.status == 200 then  
            ngx.print(res.body)  
        end;  
    '
```

```
$ curl 'localhost/lua'  
hello, world
```

♡ We'd call this whole set of nginx modules *ngx_openresty*
and our work is heavily **funded** by *Taobao.com*.

淘宝网

♥ It is already **powering** lz.taobao.com.





首页

产品介绍

最新动态

客户展示



了解并提高流量、转化率，直接提高销售额
了解上万广告费，到底效果如何
探索流行趋势，推广人气宝贝
对您的访客了若指掌，看看他们到底在干嘛？

活动介绍

量子统计登录

用户名:
密码: [忘记密码?](#)

登录

使用淘宝账号登录，即可免费试用。

使用心得

- 量子店铺统计宝典
- 挖掘报表数据，指导经营
- 2个月上4星，量子是最大的功臣
- 量子店铺统计功能，增强我的信心

我们一直在进步

- 淘宝量子店铺统计V3.0于5月17日全新升级，诚邀免费体验!
- 量子店铺统计3月16日火爆升级!
- 量子服务器扩容
- 详细来源功能升级了!

我们的客户

已经有416,950人使用过量子统计...



柠檬绿茶



妖精 emma



易酷生活



朵朵云





淘宝量子店铺统计V3.0 - 流量来源构成 - Windows Internet Explorer

http://lz.taobao.com/#srccomp/1/1

文件(E) 编辑(E) 查看(V) 收藏夹(A) 工具(T) 帮助(H)

淘宝量子店铺统计V3.0 - 流量来源构成

您好, 退出

淘宝网首页 | 我的淘宝 | 收藏量子 | 搜索 | 网站导航 | 帮助

量子统计

lz.taobao.com

店铺基本

店铺概况

实时客户访问

按小时流量分析

按天流量分析

宝贝被访排行

分类页被访排行

店内搜索关键词

营销效果

流量来源构成

淘宝搜索关键词

客户分析

访客地区分析

百宝箱

个性化统计图标

添加功能模块

管理功能模块

最新公告: 淘宝量子店铺统计V3.0测试版震撼登场, 诚邀免费试用!

跟淘宝说几句 去论坛提问

选择时间段: 2010-06-18 到 2010-06-18 查看

快速选择: 今天 昨天 前天 最近7天 最近30天 本周 上周 本月 上月

今天来源构成报表

下载

打印

来源	详细	到达页浏览量	百分比	趋势
搜索引擎	114	1	1.89%	查看
	合计	1	1.89%	
淘宝站内	淘宝站内其他	18	33.96%	查看
	淘宝店铺搜索	6	11.32%	查看
	淘宝其他店铺	6	11.32%	查看
	淘宝信用评价	3	5.66%	查看
	淘宝搜索	2	3.77%	查看
	淘宝收藏	1	1.89%	查看
	淘宝管理后台	1	1.89%	查看
	合计	37	69.81%	
直接访问	合计	10	18.87%	查看
其他	dr02.taobao.fenxi001.com	2	3.77%	查看
	club.baby.sina.com.cn	1	1.89%	查看
	sheetzoho.com	1	1.89%	查看
	blog.bbcare.cn	1	1.89%	查看
	合计	5	9.43%	

Internet 100%

♥ Generate nginx.conf from Perl TT2 templates

-- META-conf.lua

```
apiproxy = {  
    enable = true,  
    enable_lightface = true,  
    enable_admin = false, -- MUST disable in production  
    enable_devel = false, -- MUST disable in production  
    host = 'api.linezing.com',  
    port = 80,  
    log_path = '/opt/apiproxy/logs',  
    conf_path = '/opt/apiproxy/conf',  
},
```

```
nginx = {
```

```
...
```

♡ *Generate* nginx.conf from Perl TT2 templates

-- nginx.conf.tt

...

http {

default_type text/plain;

keepalive_timeout [% nginx.keepalive_timeout %];

access_log [% nginx.enable_access_log ?

apiproxy.log_path _ '/access.log' : 'off' %];

gzip [% nginx.enable_gzip ? 'on' : 'off' %];

gzip_min_length 1000;

gzip_types application/x-javascript text/css application/json;

gzip_disable "msie6";

...

♡ *Generate* Lua code by our LZSQL *compiler*

```
$ lzsql-compile -c -O2 -n src/*.lzsql
```

```
$ lzsql-link -m lightface.core -o lightface/core.lua src/*.oul
```

--/=view/itemdailyflow/type/trend

```
int      $uid;  
text     $begin, $end, $today, $url_index;  
symbol   $db;  
location $lz_report;
```

```
@hist :=  
  select ...  
  from LZDB.dpunit_purl_result($db, $begin, $end, $uid) as a  
  ...  
  at $lz_report;
```

```
@rt :=  
  select name, count(name)  
  from LZRTI.getPurl($end as day, $uid)  
  group by name  
  ...
```

```
return select ... from @hist union all @rt ...
```

♡ *Join* us at the **OpenResty** Google Group

<http://groups.google.com/group/OpenResty>

and the **nginx-devel** mailing list

<http://nginx.org/mailman/listinfo/nginx-devel>

♡ or just *catch* us on IRC:

irc.freenode.net **#nginx #openresty**

😊 *Any questions?* 😊

