

]

Applications of ngx_openresty and perl at lz.taobao.com

Applications of **ngx_openresty** and **perl** at lz.taobao.com

☺*agentzh@gmail.com*☺
章亦春 (*agentzh*)

2011.07

♥ We are lz.taobao.com

<http://lz.taobao.com>



我们是淘宝量子统计

➡ we offer web/sale/ads *analitcal* services for our *Taobao sellers*.

淘宝网

➡ 我们为我们的淘宝卖家提供
网络/销售/广告的*统计分析*服务

♡ We're analyzing *tera bytes* of raw data every day

我们每天分析 *TB* 级别的数据

♡ We do have *1,800,000+* of
Taobao seller users **registered** for our web app.

我们的 web 应用有一百八十多万的淘宝卖家注册用户

File Edit View History Bookmarks Tools Help

← → ↺ ⓧ 🏠

http://lz.taobao.com/

☆

🌐

Google

🔍

量子恒道统计

+



量子恒道统计

http://lz.taobao.com

淘宝官方

量子数科院

量子帮派

帮助中心

🔗 设为桌面快捷

★ 收藏量子

关注量子微博

首页

产品攻略

最新动态

客户展示

NEW



2011

量子恒道统计(淘宝官方出品)

正式推出!

真实 精准 直观 效果 解读

立即订购 ▶ 新手入门指南

量子恒道统计登录

账户名

手机号/会员名/邮箱

密码

☐ 安全控件登录

登录

忘记密码?

使用淘宝账号登录, 即可免费试用。

使用心得

我们一直在进步

我们的客户 已有1,853,235人使用量子恒

Done

🐞

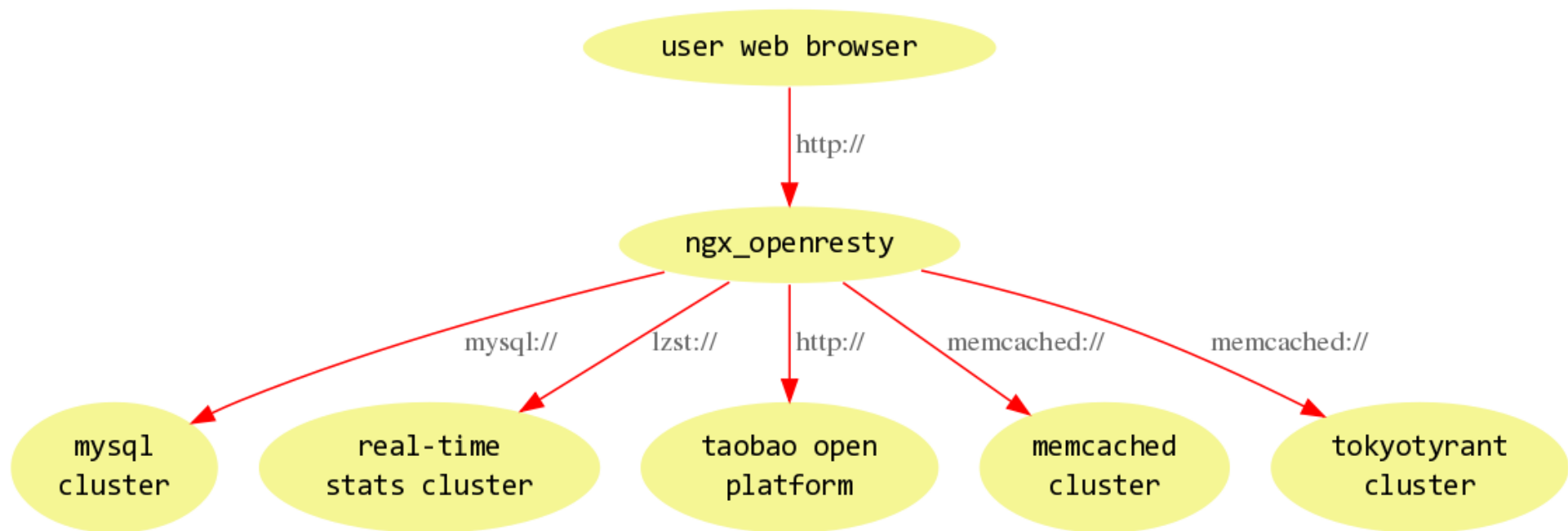
FoxyProxy: Disabled

🌐

110.75.2.3



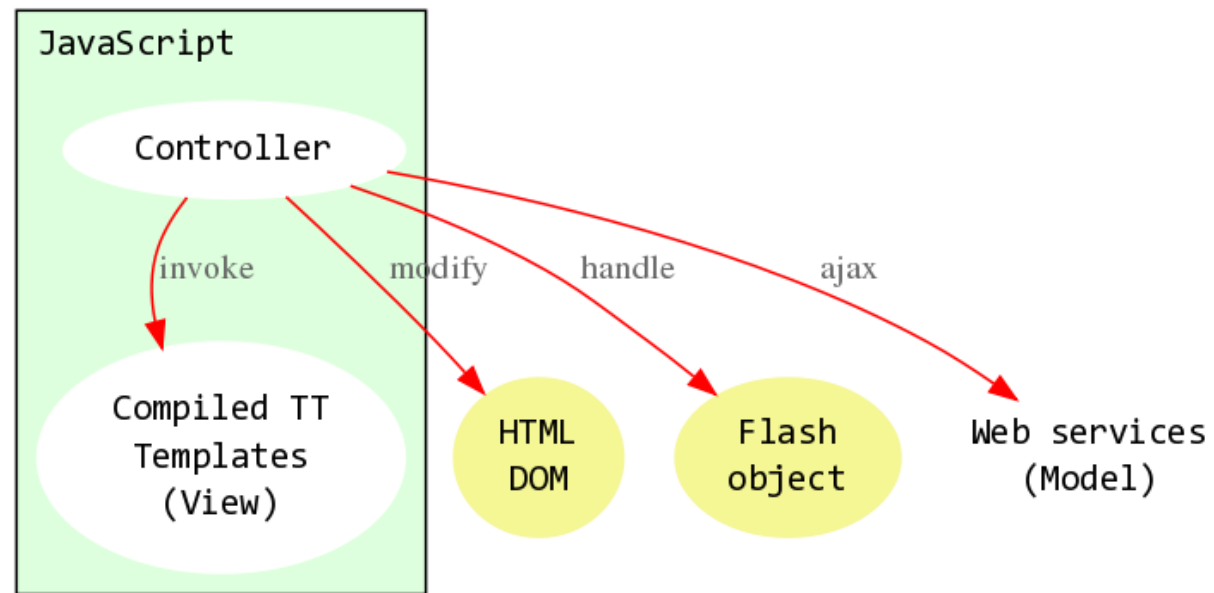




Architecture Diagram for `lz.taobao.com`

♥ Our frontend web app is running directly
in your *web browser*!

我们的前端 web 应用是直接运行在
你的**网络浏览器**里的！



What's inside the user web browser

♡ TT == Perl *Template Toolkit*

<http://search.cpan.org/perldoc?Template::Manual>

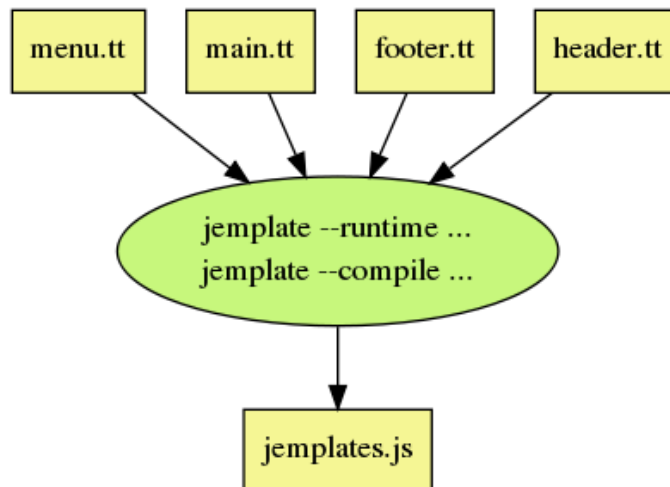
♡ We only use the TT *templating language*,
not the Perl TT engine itself

我们只是使用了 TT 这门模版语言,
而不是 Perl TT 引擎本身

♥ We use the **Jemplate** compiler to
compile our TT templates to
standalone **JavaScript** code

<http://search.cpan.org/perldoc?Jemplate>

我们使用 Jemplate 编译器把我们的 TT
模版编译为独立的 JavaScript 代码



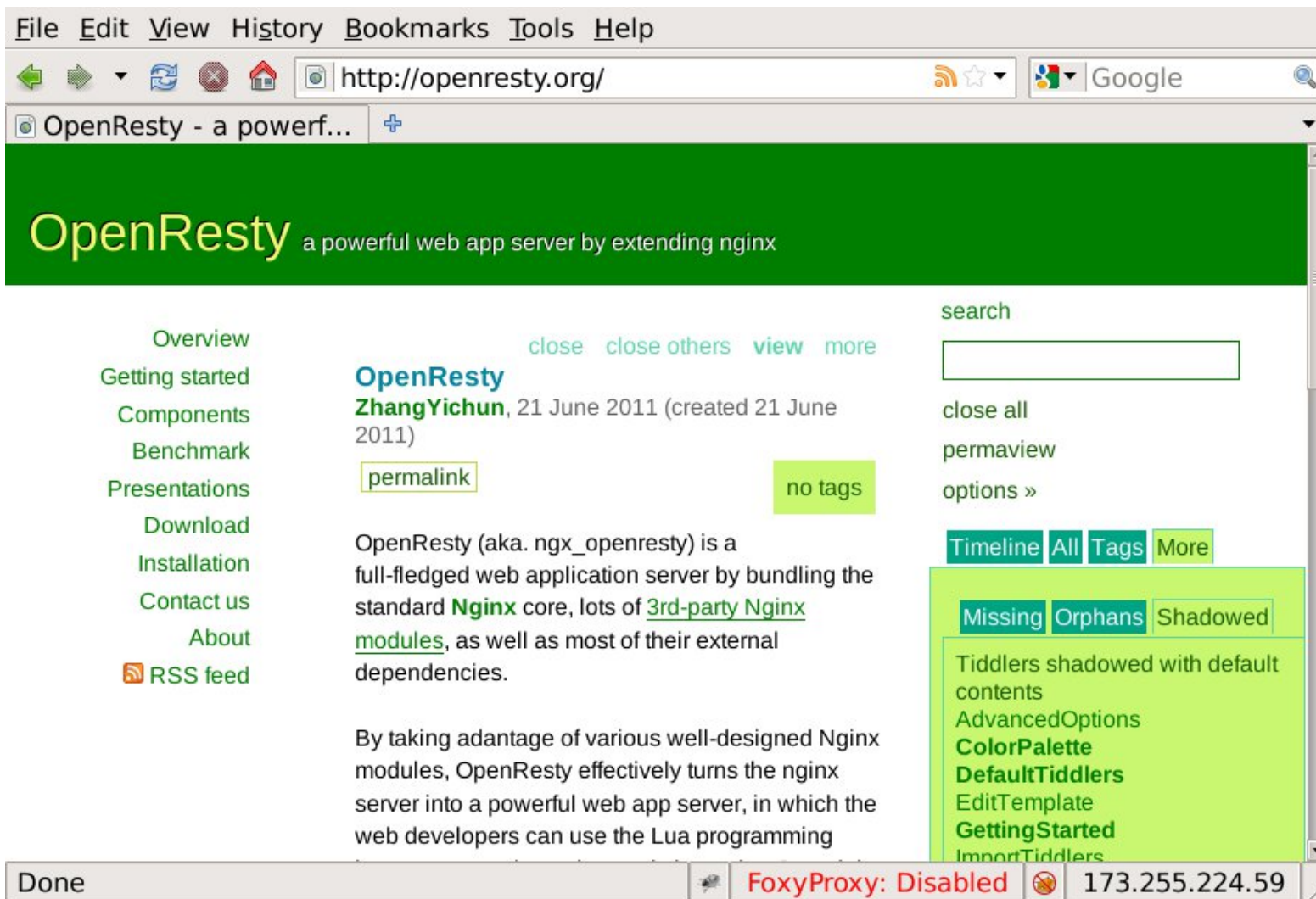
♡ *Client-side* templating is a lot of fun!

客户端模版编程非常有趣！

♡ ngx_openresty is a massively
enhanced and *extended* version of **nginx**

<http://openresty.org>

ngx_openresty 是 nginx 的一个
进行进行过大量增强和扩展后的版本



♥ The **original idea** of ngx_openresty
comes from the man
simply known as *chaoslawful*

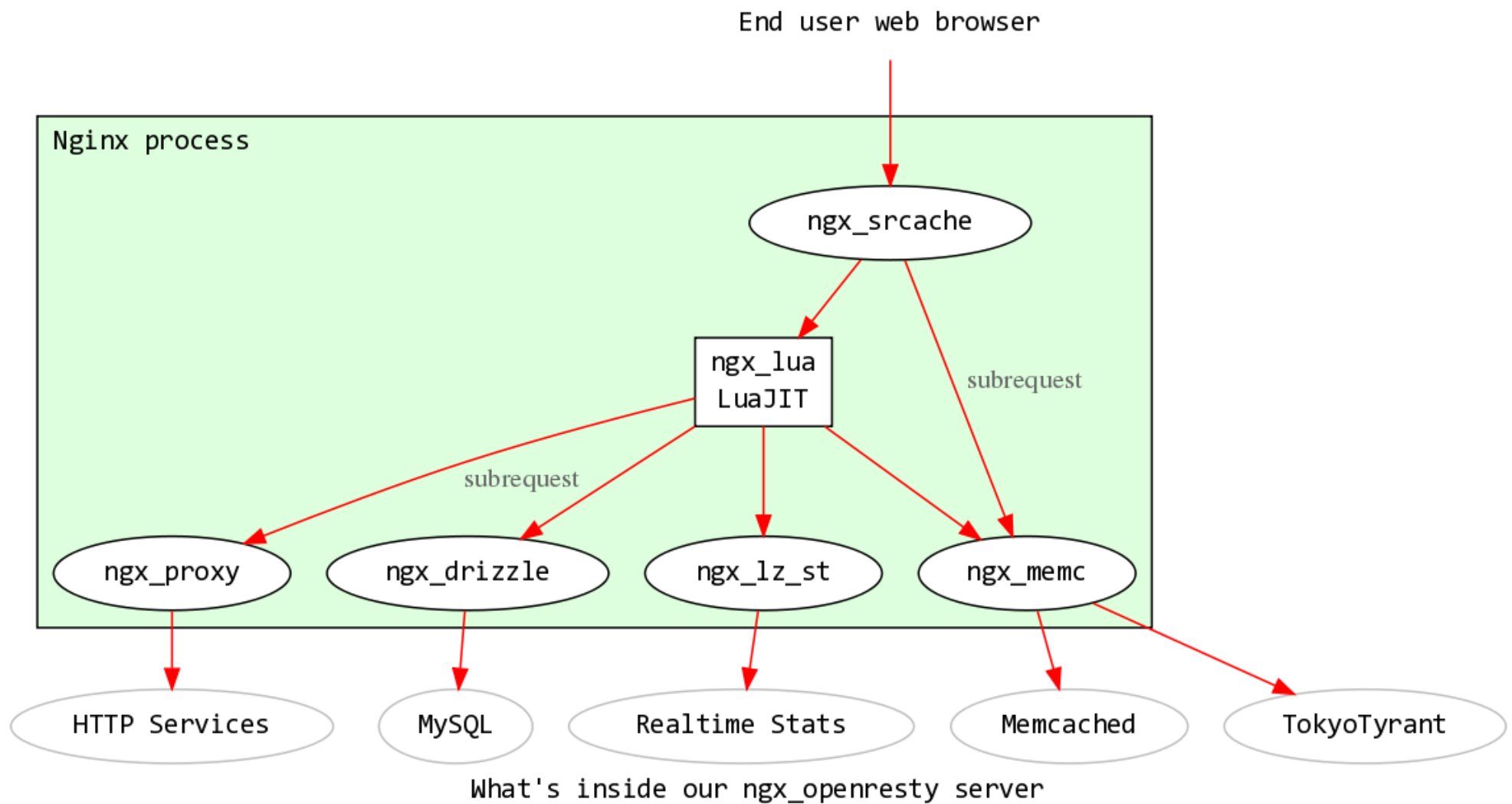
<https://github.com/chaoslawful/>

ngx_openresty 的最初的想法来自一个
只知道叫做 chaoslawful 的男人



♡ Our server-side code just provides
web services and *security protection*.

我们的服务器端代码只是提供
支持性的数据服务和安全保护。



♡ There's 0 lines of Perl
in the online ngx_openresty server

在线的 ngx_openresty 服务器中
有 0 行 Perl 代码

♡ Synchronous *non-blocking* Lua code

同步的非阻塞的 Lua 代码

```
upstream main_db {  
    drizzle_server 127.0.0.1:3306  
        user=monty password=some_pass dbname=test  
        protocol=mysql;  
  
    drizzle_keepalive max=10 overflow=reject  
        mode=single;  
}
```

```
location /mysql {  
    set_unescape_uri $sql $arg_sql;  
    set_unescape_uri $backend $arg_backend;  
  
    drizzle_query $sql;  
    drizzle_pass $backend;  
}
```

```
location /main {  
    content_by_lua_file conf/my-app.lua;  
}
```

-- my-app.lua

```
local res = ngx.location.capture("/mysql", {  
    args = {  
        sql = "select count(*) from cats",  
        backend = "main_db"  
    }  
})
```

♡ But almost *all* of our Lua code
for the business logic
is *generated* and optimized by **Perl**

但我们几乎所有用于业务逻辑的 Lua 代码
都是由 Perl 生成和优化的

♡ We *invented* the **LZSQL** little language
for our business.

<http://agentzh.org/misc/nginx/lzsql-manual.html>

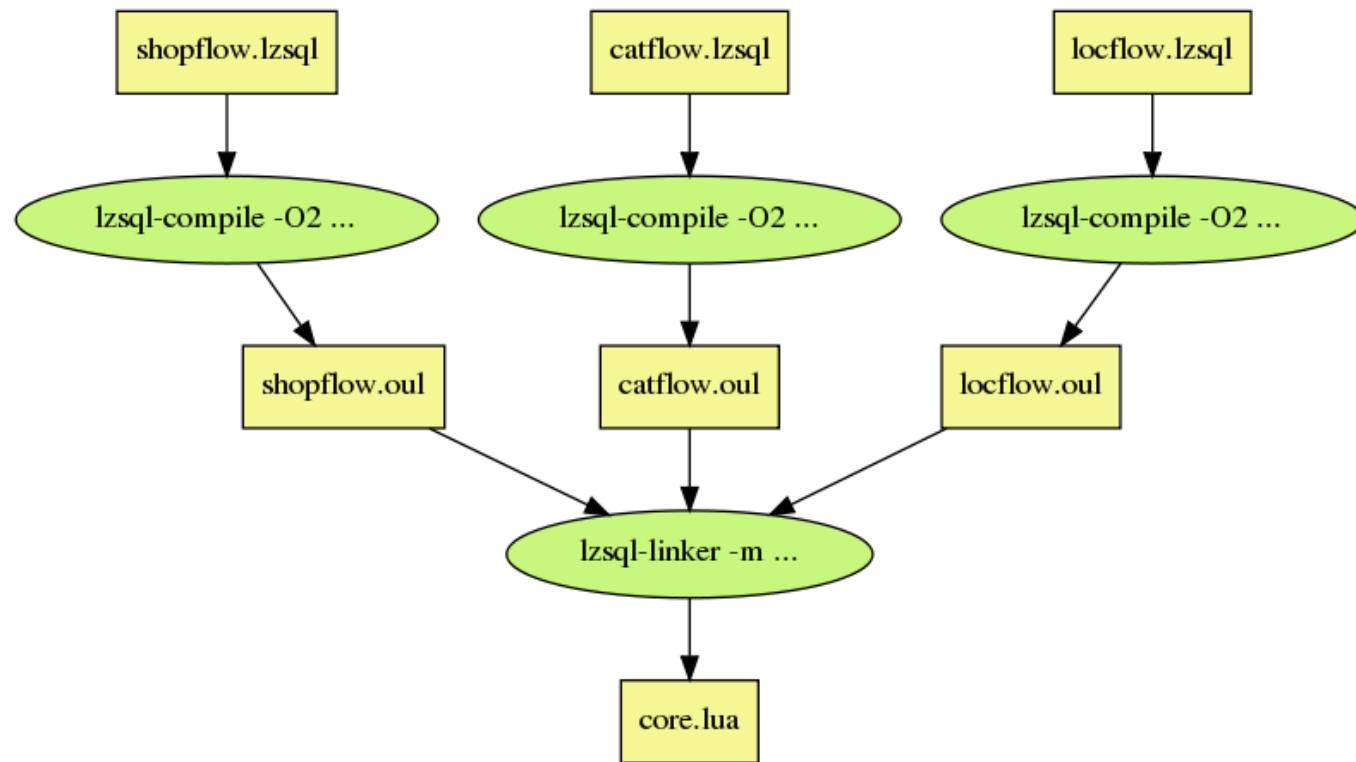
我们为我们的业务发明了 LZSQL 这种小语言

```
-- /=/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 ...
```

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

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

♥ Our LZSQL *compiler* is written in Perl!

我们的 LZSQL 编译器是用 Perl 编写的！

☺ Parse::RecDescent -> LZSQL *Parser*

☺ Moose -> LZSQL *AST*

☺ Moose -> LZSQL *Optimizer*

☺ Moose -> LZSQL *Code Emitter*

♥ We *can* generate **Lua** code,
so why not *C*?

我们既然可以生成 Lua 代码，
那为什么不能生成 C 呢？

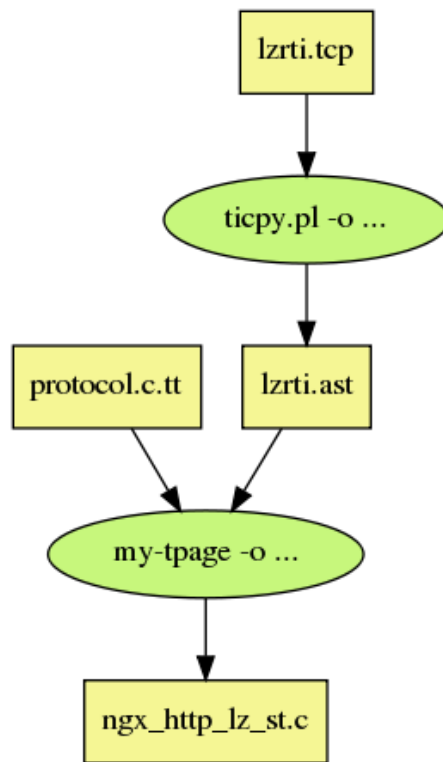
♡ Our ngx_lz_st module is generated *automatically* from our real-time stats engine's TCP protocol **documentation**!

我们的 ngx_lz_st 模块是从我们的
实时统计引擎的文档自动生成的！

♡ The documentation is actually written in
a declarative little *language*
named **Ticpy** designed by myself!

这个文档其实是用我自己设计的声明性的
名叫 Ticpy 的小语言来表达的!

```
// File lzrti.tcp
...
// fid=7
// @desc: 获取指定店铺的，指定访客的信息
// @param uid: unit_id
// @param uv: 访客cookie
// @param all_sessions: 是否只返回p4p点击：1 是 0 否
getUv(uid:4, uv:s, all_sessions:4): (
    login_time:8, // session开始时间
    login_url:s, // session开始页面
    login_title:s, // session开始页面标题
    stay_time:4, // 当前session的时长
    page_deep:4, // 当前session的访问深度
    im:s, // 用户的旺旺
    ip:s, // ip地址
    location_id:4, // 地址id
    ...
)
```

```
$ wc -l lzrti.tcp protocol.c.tt ngx_http_lz_st.c
```

```
345 lzrti.tcp
```

```
464 protocol.c.tt
```

```
12478 ngx_http_lz_st.c
```

```
13287 total
```

♡ So...I'd rather write
programs to write programs
to write **programs**...

所以。。。我宁可写
写程序的程序，
也不写程序。。。。

♥ *Test* our nginx **C modules** by
our opensource *Perl* module *Test::Nginx*!

<http://search.cpan.org/perldoc?Test::Nginx>

利用我们开源的 Perl 模块 Test::Nginx
来测试我们的 Nginx C 模块！

File Edit View History Bookmarks Tools Help

Test::Nginx - search.c... +

CPAN

Home · Authors · Recent · News ·
Mirrors · FAQ · Feedback

in All CPAN Search

[agentzh \(章亦春\)](#) > [Test-Nginx-0.15](#) > Test::Nginx [permlink](#)

Module Version: 0.15 [Source](#)



NAME

DESCRIPTION

etcproxy integration

Test::Nginx <=> nginx

nginx <=> memcached

valgrind integration

Nginx C modules that use Test::Nginx to drive their test suites

Download:
[Test-Nginx-0.15.tar.gz](#)

[Dependencies](#)

[Annotate this POD](#)

Done

FoxyProxy: Disabled 199.15.176.161

```
use Test::Nginx::Socket;  
plan tests => 2 * blocks();  
run_tests();  
__DATA__
```

```
=== TEST 1: sanity
```

```
--- config
```

```
location /main {  
    echo_subrequest GET /sub;  
}  
location /sub {  
    echo hello;  
}
```

```
--- request
```

```
GET /main
```

```
--- response_body
```

```
hello
```

♥ *Test* our **web serives** by *Perl*!

用 Perl 来测试我们的 web 服务！

```
use t::LZ;  
plan tests => 1 * blocks();  
run_tests();  
__DATA__
```

```
=== TEST 1: lzers
```

```
--- url
```

```
/=/view/lzers/~/~
```

```
--- res
```

```
[
```

```
{
```

```
  "cnt" : 911619
```

```
}
```

```
]
```


=== TEST 5: sellerbasic

--- uid: 1509

--- url

/=/view/sellerbasic/~/~

--- res

[

{

"uv_times" : 257

},

{

"login_title" : "店内搜索页",

"login_url" : "http://xiaowangpu...",

...

♥ Generate *test databases* from rules
by our opensource **Cheater** utility
written by *Perl*!

<http://search.cpan.org/perldoc?Cheater>

利用我们开源的 Cheater 实用程序，从规则
自动生成测试数据库！

File Edit View History Bookmarks Tools Help

http://search.cpan.org/~agent/Cheater-0.10/lib/Cheater-0.10/ Cheater - search.cpan.org

CPAN Home · Authors · Recent · News · Mirrors · FAQ · Feedback

in All CPAN Search

[agentzh \(章亦春\)](#) > [Cheater-0.10](#) > Cheater [permalink](#)

Module Version: 0.10 [Source](#)

NAME
VERSION
DESCRIPTION
INSTALLATION
AUTHOR
COPYRIGHT & LICENSE

 Download:
[Cheater-0.10.tar.gz](#)
[Dependencies](#)

Done FoxyProxy: Disabled 199.15.176.161

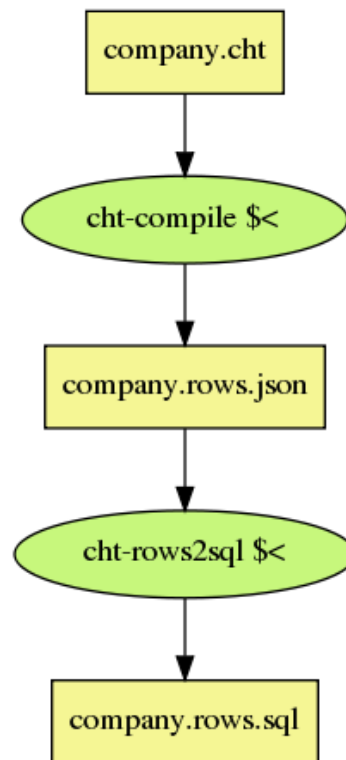
company.cht

Empolyee table

```
table employees (  
    id serial;  
    name text /[A-Z]a-z{2,5} [A-Z]a-z{2,7}/ not null unique;  
    age integer 18..60 not null;  
    birthday date;  
    height real 1.50 .. 1.90 not null;  
    grades text {'A','B','C','D'} not null;  
    department references departments.id;  
)
```

Department table

```
table departments (  
    id serial;  
    name text /\w{2,10}/ not null;  
)
```



♥ I've been hacking on *Git*Hub!

<http://github.com/agentzh>

我在 GitHub 上玩开源！

♡ *Follow* me on Sina Weibo!

<http://weibo.com/agentzh/>

在新浪微博上关注我！

😊 *Any questions?* 😊

