Nginx Notes

Paolo D'Apice

iZENEsoft (Shanghai) Co. Ltd.

January 19, 2012

NGiИX™

Apache is like Microsoft Word, it has a million options but you only need six.
Nginx does those six things, and it does five of them 50 times faster than Apache.

Chris Lea

What's Nginx

Batman is fast. Nginx is fast.
Batman fights crime. Nginx fights wasted CPU cycles and memory leaks.
Batman performs well under pressure. Nginx, for its part, excels under heavy server loads.

But Batman would be almost nothing without the Batman utility belt. Instead of a utility belt, Nginx has a module chain.

Evan Miller

How it works

Module types:

handlers process a request and produce an output

filters manipulate the output produced by a handler

load-balancers forward a request choosing between multiple backend servers

Components

1. configuration
2. directives
3. context
4. definition
5. installation

Module Configuration Struct(s)

- up to three
  - main
  - server
  - location
- elements in the structs are populated by module directives

Module Directives

- array of `ngx_command_t` structs
- used for module configuration

Module Context

- an `ngx_http_module_t` struct
- contains function references for creating and merging configurations

Module Definition

- an `ngx_module_t` struct named
  `ngx_<module name>_module`
- contains references to context, directives and other callbacks

Module installation

- depends on the module type
- see later . . .

Handlers

Typically do four things:

1. get location configuration
2. generate appropriate response
3. send response header
4. send response body

Handlers

- do little "real work"
- callbacks invoked when the upstream is ready

Handlers
Installation

- define the directive that enables the module
- get the "core" struct for the location
- assign a handler to it

Filters

Manipulate responses generated by handlers

- header filters
- body filters

Filters
Header filters

Three basic steps:

1. decide whether to operate on the response
2. operate on the response
3. call the next filter

Filters
Body filters

- operate only on one buffer (chain link) at a time
  - overwrite
  - replace
  - insert new buffer before/after
- support incomplete buffer chain
- no nice high-level API

Filters

- ▶ post-configuration step
- ▶ insert the filter on the top of filter chain (LIFO)

Load-Balancer

TODO

Writing Nginx Modules

- no or little documentation
  - mostly in Russian or Chinese
  - *self documented* source code
- learn by example

Writing Nginx Modules

- write you code
- write a `config` file
- compile Nginx with your module

Writing Nginx Modules

- take a look at the hello-module
  - autotools adapted to CMake
  - C++ support[1]
  - Test::Nginx Perl framework
- stick to the naming standards

---

[1]Nginx is written in plain C

References

📄 Nginx Wiki
  `http://wiki.nginx.org/Main`

📄 Evan Miller's *definitive* guide
  `http://www.evanmiller.org/nginx-modules-guide.htm`

📄 Antoine Bonavita's blog about Nginx
  `http://www.nginx-discovery.com`

More References

📄 Taobao's Tengine
http://tengine.taobao.org

📄 agentz's blog
http://agentzh.org

📄 chaoslawful's blog
http://chaoslawful.iteye.com

📄 Joshua Zhu's blog
http://blog.zhuzhaoyuan.com