

关于视频课

- 目的：在实际项目中，我们学习的知识是如何被使用的
- 涉及到代码的部分，建议在 PC 上收看

35.ingress-nginx 中几个性能相关 PR

NGINX Ingress Controller

build passing codecov 57% go report A+ license Apache-2.0 stars 5k contributions welcome license scan passing

- NGINX Ingress Controller for Kubernetes: <https://github.com/kubernetes/ingress-nginx>, 是 k8s 的项目, 并非 NGINX 官方的项目
- 流量入口
- 基于 Go、NGINX 和 lua-nginx-module 开发

学习目的

- 如何发现实际项目中的性能问题
- 如何利用专栏的知识，来修复性能问题
- 如何通过 GitHub 协作

两个很小的 PR

 **used cJSON.safe instead of pcall.** ✓ **approved** **cncf-cla: yes** **lgTM** **size/M**

#3674 by moonming was merged on 19 Jan

 **used table functions of LuaJIT for better performance.** ✓ **approved** **cncf-cla: yes** **lgTM** **size/S**

#3673 by moonming was merged on 2 Feb

used cJSON.safe instead of pcall



```
▼ 8 rootfs/etc/nginx/lua/balancer.lua
... @@ -1,5 +1,5 @@
1     local ngx_balancer = require("ngx.balancer")
2 -local json = require("cjson")
3     local util = require("util")
4     local dns_util = require("util.dns")
5     local configuration = require("configuration")
@@ -114,9 +114,9 @@ local function sync_backends()
114         return
115     end
116
117 - local ok, new_backends = pcall(json.decode, backends_data)
118 - if not ok then
119 -     ngx.log(ngx.ERR, "could not parse backends data: " .. tostring(new_backends))
120     return
121 end
122
```

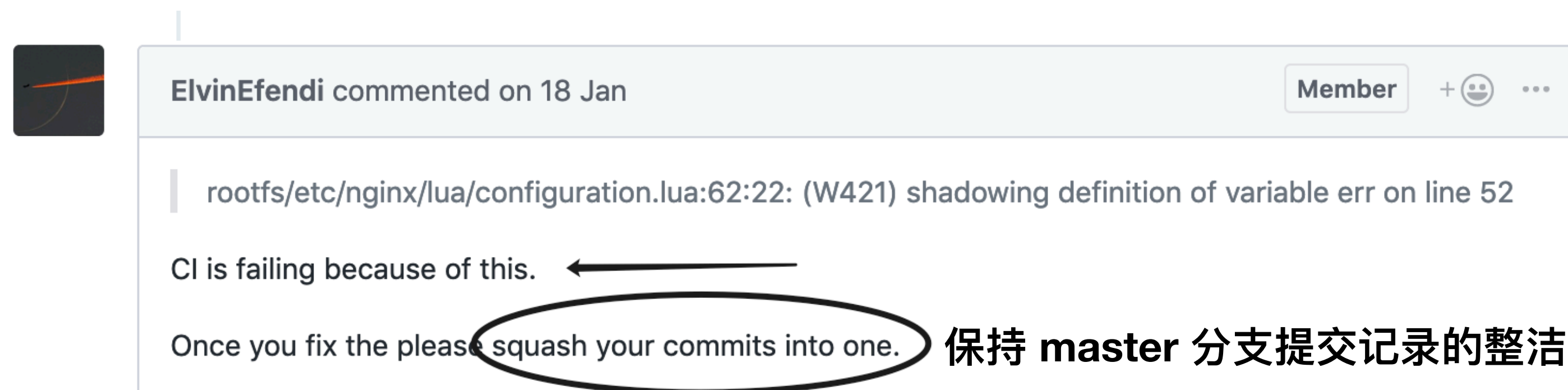
```
1  local ngx_balancer = require("ngx.balancer")  使用 safe 函数
2  +local cJSON = require("cjson.safe")
3  local util = require("util")
4  local dns_util = require("util.dns")
5  local configuration = require("configuration")
```

```
114      return
115      end
116
117  + local new_backends, err = cJSON.decode(backends_data)
118  + if not new_backends then
119  +     ngx.log(ngx.ERR, "could not parse backends_data, err: %s", err)
120      return
121      end
122
```

减少 pcall 调用，代码也更清爽

使用逗号，而非字符串拼接

CI 和合并 commits



变量的重复定义，这在 `err` 这种常用变量名时经常遇到
处理的方法就是单独 `local` 一下

```
62 + local success
63 + success, err = certificate_data:safe_set(server.hostname, server.sslCert.pemCertKey)
```


我更感兴趣的是： 它的 CI 是如何检测到的呢？

[.travis.yml](#)

Showing the top two matches Last indexed 5 hours ago

```
31     - secure: ZZlcwdr4X2ZeIuA4f5wiT04qNCpSi
      /X4RYKjXny7eixeHl5lgQ++IV9APvwWfsCiREFhiQFs
      /0nCJBGohgSqC8h/U0bB0HeehEWnF1zzfQPREzHwVkU
      /VbfY1HW4QkpIkPVSFBatciuePUnIkEX6+jVM+GEZ0h
      /NbPqUwBn5RbuD71L69vFZZLpU09V4PuablWW8ACQxg
32
33   jobs:
34     include:
35       - stage: Static Check
36         script:
37           - make static-check
```

Makefile

Showing the top six matches Last indexed 19 days ago

```
161          build/go-in-docker.sh build/build-plugin.sh
162
163  .PHONY: clean
164  clean:
165          rm -rf bin/ .gocache/ .env
166
167  .PHONY: static-check
168  static-check:
169          @$(DEF_VARS) \
170          build/go-in-docker.sh build/static-check.sh
171
```

Branch: master [ingress-nginx](#) / [build](#) / [static-check.sh](#) ElvinEfendi lua randomseed per worker2 contributors  

Executable File | 29 lines (23 sloc) | 777 Bytes

```
1  #!/bin/bash
2
3  # Copyright 2018 The Kubernetes Authors.
4  #
5  # Licensed under the Apache License, Version 2.0 (the "License");
6  # you may not use this file except in compliance with the License.
7  # You may obtain a copy of the License at
8  #
9  #     http://www.apache.org/licenses/LICENSE-2.0
10 #
11 # Unless required by applicable law or agreed to in writing, software
12 # distributed under the License is distributed on an "AS IS" BASIS,
13 # WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14 # See the License for the specific language governing permissions and
15 # limitations under the License.
16
17 set -o errexit
18 set -o nounset
19 set -o pipefail
20
21 if [ -z "${PKG}" ]; then
22     echo "PKG must be set"
23     exit 1
24 fi
25
26 hack/verify-all.sh
27
28 luacheck --codes -q rootfs/etc/nginx/lua/
```

熟悉的老朋友：luacheck

luacheck 的配置



Branch: master ▾ [ingress-nginx](#) / .luacheckrc

 ElvinEfendi lua randomseed per worker

3 contributors   

11 lines (10 sloc) | 268 Bytes

```
1 std = 'ngx_lua'
2 globals = {
3     '_TEST'
4 }
5 exclude_files = {'./rootfs/etc/nginx/lua/test/**/*.lua'}
6 files["rootfs/etc/nginx/lua/lua_ingress.lua"] = {
7     ignore = { "122" },
8     -- TODO(elvinefendi) figure out why this does not work
9     --read_globals = {"math.randomseed"},
10 }
```

used table functions of LuaJIT for better performance



```
4 +local new_tab = require "table.new"  
5 +local clear_tab = require "table.clear"  
6 +local clone_tab = require "table.clone"
```

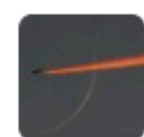
使用可以 LuaJIT 的 table 函数

```
7
```

```
8 -- if an Nginx worker processes more than (MAX_BATCH_SIZE/FLUSH_INTERVAL) RPS then  
9 local MAX_BATCH_SIZE = 10000  
10 local FLUSH_INTERVAL = 1 -- second
```

旧版本是 `local metrics_batch = {}`

```
12 +local metrics_batch = new_tab(MAX_BATCH_SIZE, 0)
```



ElvinEfendi on 18 Jan Member

What's the second argument for? Why 0? A link to docs would be sufficient too.



moonming on 18 Jan Author Contributor

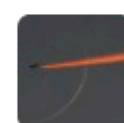
doc: <https://github.com/openresty/luajit2/blob/v2.1-agentzh/doc/extensions.html>

`table.new(narray, nhash)` , the second argument is for hash.

```
- table.insert(metrics_batch, metrics())
```

自己维护数组下标，替代内置的 `table.insert` 方法

```
8  
9 + metrics_batch[metrics_size + 1] = metrics()
```



ElvinEfendi on 18 Jan Member

+(

Does this custom table implementation not have `insert` ?



moonming on 18 Jan Author Contributor

+(

it has the `insert` method, but the `table.insert` is an $O(n)$ operation due to the `lj_tab_len` call inside LuaJIT.

We can avoid it in hot loops for better performance.

总结

- 对于问题要刨根问底
- 热路径中代码要万分小心，要考虑能否被 JIT
- 尽量使用 OpenResty 和 LuaJIT 的库
- OpenResty 在社区上的工作很不足

Q&A