


Yoara的备忘录

云端。

目录视图摘要视图RSS 订阅

个人资料



Yoara

访问：170772次

积分：2623

等级：

BLDG

5

排名：第10218名

原创：88篇

转载：50篇

译文：4篇

评论：13条

文章搜索

文章分类

hibernate (4)

HTML/JSP (9)

Javascript (7)

Java基础 (30)

linux (10)

Spring (10)

视图层框架技术 (11)

其他 (11)

设计模式 (12)

mysql (2)

今日Point (1)

学习mina (7)

Android (3)

分布式集群相关 (6)

activemq (5)

架构 (2)

【CSDN技术主题月】深度学习框架的重构与思考

深度学习代码专栏

【观点】有了深度学习，你还学传统机器学习算法么？

【知识库】深度学习知识图谱上线啦

OpenResty初步使用

标签：

nginx

openresty

2016-09-09 09:5955人阅读评论(0)收藏举报

分类：

nginx (1)

目录(?)

+

OpenResty

OpenResty™ 是一个基于 Nginx 与 Lua 的高性能 Web 平台，其内部集成了大量精良的 Lua 库、第三方模块以及大多数的依赖项。用于方便地搭建能够处理超高并发、扩展性极高的动态 Web 应用、Web 服务和动态网关。

OpenResty 通过汇聚各种设计精良的 Nginx 模块（主要由 OpenResty 团队自主开发），从而将 Nginx 有效地变成一个强大的通用 Web 应用平台。这样，Web 开发人员和系统工程师可以使用 Lua 脚本语言调动 Nginx 支持的各种 C 以及 Lua 模块，快速构造出足以胜任 10K 乃至 1000K 以上单机并发连接的高性能 Web 应用系统。

OpenResty 的目标是让你的Web服务直接跑在 Nginx 服务内部，充分利用 Nginx 的非阻塞 I/O 模型，不仅仅对 HTTP 客户端请求,甚至于对远程后端诸如 **MySQL**、PostgreSQL、Memcached 以及 **Redis** 等都进行一致的高性能响应。

目前openresty的组件包括：<http://openresty.org/en/components.html>

安装

框架 (1)

node (1)

solr (1)

angularjs (1)

git (1)

lucene (1)

nginx (2)

文章存档

2016年09月 (2)

2016年06月 (3)

2016年05月 (1)

2016年03月 (7)

2015年06月 (1)

展开

阅读排行

从Spring For Android开始 (12047)

Ubuntu下安装Zabbix (11932)

mina学习笔记四：交互的 (8759)

struts1.2 bean标签的用法 (5094)

Spring for Android 参考 (5019)

Maven常用指令 (4737)

spring 的延迟加载 (4588)

mina学习笔记三：一切的 (4538)

BodyContent揭秘及定制 (4233)

mina学习笔记五：做嫁衣 (3807)

推荐文章

* 2016 年最受欢迎的编程语言是什么？

* Chromium扩展（Extension）的页面（Page）加载过程分析

* Android Studio 2.2 来啦

* 手把手教你做音乐播放器（二）技术原理与框架设计

* JVM 性能调优实战之：使用阿里开源工具 TProfiler 在海量业务代码中精确定位性能代码

俄罗斯八日游

企业管理软件

德云社票价

婴儿游泳

您必须将这些库 perl 5.6.1+, libreadline, libpcres, libssl安装在您的电脑之中。 对于 Linux来说, 您需要确认使用 ldconfig 命令, 让其在您的系统环境路径中能找到它们。

Debian 和 Ubuntu 用户

推荐您使用 apt-get安装以下的开发库:

```
[html] 01. apt-get install libreadline-dev libncurses5-dev libpcre3-dev libssl-dev perl make build-essential
```

Fedora 和 RedHat 用户

推荐您使用yum安装以下的开发库:

```
[html] 01. yum install readline-devel pcre-devel openssl-devel gcc
```

./configure

下载**预编译包**, 然后在进入 openresty-VERSION/ 目录, 然后输入以下命令配置:

./configure
默认, --prefix=/usr/local/openresty 程序会被安装到/usr/local/openresty目录。

您可以指定各种选项, 比如

```
[html] 01. ./configure --prefix=/opt/openresty \
02.             --with-luajit \
03.             --without-http_redis2_module \
04.             --with-http_iconv_module \
05.             --with-http_postgres_module
```

试着使用 ./configure --help 查看更多的选项。

配置文件 (./configure script) 运行出错可以到 build/nginx-VERSION/objs/autoconf.err 找到。 VERSION 的地方必须与OpenResty版本号相对应, 比如 1.11.2.1。

接着 :

```
[html] 01. make
02.
03. make install
```

安装完成后的目录大概是这样的

```
[html] 01. drwxr-xr-x.  2 root root   4096  9月  1 15:45 bin
02. drwxr-xr-x.  6 root root   4096  9月  1 15:45 luajit
```

```
03. drwxr-xr-x. 6 root root 4096 9月 1 15:45 lualib
04. drwxr-xr-x. 11 root root 4096 9月 1 15:46 nginx
05. drwxr-xr-x. 41 root root 4096 9月 1 15:45 pod
06. -rw-r--r--. 1 root root 157757 9月 1 15:45 resty.index
07. drwxr-xr-x. 3 root root 4096 9月 1 15:45 site
```

配置nginx

进入xx/openresty/nginx/conf目录

编辑nginx.conf文件

1.在http模块下加入：（路径替换成自己的）

```
[html] C P
01. lua_package_path "/opt/openresty/lualib/?.lua;;"; #lua 模块
02. lua_package_cpath "/opt/openresty/lualib/?.so;;"; #c模块
```

2.在server模块下引入，将所有需要lua处理的单独放在一个文件中

```
[html] C P
01. include ../conf/myconf/luatest.conf;
```

luatest.conf配置如下

```
[html] C P
01. location ~ /openapi/test/. * {
02.     default_type 'text/html';
03.     # content_by_lua 'ngx.say("hello")';
04.     content_by_lua_file conf/myconf/lua/first.lua;
05. }
```

first.lua 脚本处理了一个非常常见的业务场景，就是使用nginx直接访问redis，脚本文件first.lua内容如下

```
[html] C P
01. local function close_redis(red)
02.     if not red then
03.         return
04.     end
05.
06.     -- local ok, err = red:close()
07.     -- if not ok then
08.     --     ngx.say("close redis error : ", err)
09.     -- end
10.     -- 释放连接(连接池实现)
11.     ngx.say("aaa")
12.     local pool_max_idle_time = 10000 --毫秒
13.     local pool_size = 100 --连接池大小
14.     local ok, err = red:set_keepalive(pool_max_idle_time, pool_size)
15.     if not ok then
16.         ngx.say("set keepalive error : ", err)
17.     end
18. end
19. --
20. local redis = require("resty.redis")
21. -- 创建实例
22. local red = redis:new()
23. -- 设置超时 (毫秒)
24. red:set_timeout(1000)
25. -- 建立连接
26. local ip = "192.168.0.43"
```

```
27. local port = 6381
28. local ok, err = red:connect(ip, port)
29. --调用API获取数据
30. local resp, err = red:get("DB_SMS:openresty")
31. if not resp then
32.     ngx.say("get msg error : ", err)
33.     return close_redis(red)
34. end
35. ngx.say(resp)
36. close_redis(red)
```

这样，openresty就算入门了。

使用

性能对比，直接通过nginx访问redis和直接访问应用服务器

通过nginx

```
Server Software:      openresty/1.11.2.1
Server Hostname:      192.168.0.195
Server Port:          9011

Document Path:        /openapi/test/redis/getRedis
Document Length:      26 bytes

Concurrency Level:    500
Time taken for tests:  0.899 seconds
Complete requests:    10000
Failed requests:       0
Write errors:         0
Total transferred:    1740000 bytes
HTML transferred:     260000 bytes
Requests per second:  11128.57 [#/sec] (mean)
Time per request:     44.929 [ms] (mean)
Time per request:     0.090 [ms] (mean, across all concurrent requests)
Transfer rate:        1890.99 [Kbytes/sec] received
```

直接访问应用服务器

```
Server Software:      Apache-Coyote/1.1
Server Hostname:      192.168.0.195
Server Port:          11110

Document Path:        /qfang-sms-web/openapi/test/redis/getRedis
Document Length:      14 bytes

Concurrency Level:    500
Time taken for tests:  3.015 seconds
Complete requests:    10000
Failed requests:       0
Write errors:         0
Total transferred:    3850752 bytes
HTML transferred:     140392 bytes
Requests per second:  3316.79 [#/sec] (mean)
Time per request:     150.748 [ms] (mean)
Time per request:     0.301 [ms] (mean, across all concurrent requests)
Transfer rate:        1247.28 [Kbytes/sec] received
```


顶 踩
0 0

上一篇 [看在nginx使用之前](#)


我的同类文章

nginx (1)		
• 看nginx使用之前	2016-09-09	阅读 13


参考知识库



Linux知识库
3922 关注 | 3042 收录



Redis知识库
2626 关注 | 716 收录



MySQL知识库
13267 关注 | 1414 收录

猜你在找

企业Web服务器Nginx应用实战	Nginx-openresty+mysql+redis使用篇一
大型分布式redis+solr+Linux+nginx+springmvc	使用Nginx+LuaOpenResty开发高性能Web应用
用redis 搭建大数据 热门排行榜, 用户推荐系统	Openresty nginx使用 luaJIT 配置
[颠覆传统所学]web开发级Centos颠覆实战(上)	使用muduo编译链接错误 undefined reference to
[颠覆传统所学]web级mysql颠覆实战	使用g++编译CC++程序链接时出现undefined

查看评论

暂无评论

您还没有登录,请[登录](#)或[注册](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题

Hadoop

AWS

移动游戏

Java

Android

iOS

Swift

智能硬件

Docker

OpenStack

VPN

Spark

ERP

IE10

Eclipse

CRM

JavaScript

数据库

Ubuntu

NFC

WAP

jQuery

BI

HTML5

Spring

Apache

.NET

API

HTML

SDK

IIS

Fedora

XML

LBS

Unity

Splashtop

UML

components

Windows Mobile

Rails

QEMU

KDE

Cassandra

CloudStack

FTC

coremail

OPhone

CouchBase

云计算

iOS6

Rackspace

Web App

SpringSide

Maemo

Compuware

大数据

aptech

Perl

Tornado

Ruby

Hibernate

ThinkPHP

HBase

Pure

Solr

Angular

Cloud Foundry

Redis

Scala

Django

Bootstrap

公司简介 | 招贤纳士 | 广告服务 | 银行汇款帐号 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

网站客服 杂志客服 微博客服 webmaster@csdn.net 400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 |

江苏乐知网络技术有限公司 提供商务支持

京 ICP 证 09002463 号 | Copyright © 1999-2016, CSDN.NET, All Rights Reserved 