

wdt3385的专栏

目录视图

摘要视图

RSS 订阅

个人资料



wdt3385



访问：338971次

积分：5988分

排名：第1052名

原创：78篇 转载：1184篇

译文：0篇 评论：21条

文章搜索

文章分类

- Linux (287)
- 线程 (5)
- C++ (8)
- 调试工具 (3)
- 网络编程 (36)
- 编译 (46)
- 编码 (11)
- VC (28)
- CAPI (20)
- C/C++ (8)
- 网络通信 (18)
- WINDOWS (21)
- API (4)
- Socket (24)
- 界面库 (1)
- 调试 (2)
- cygwin (2)
- openssl (24)
- PKI (2)
- 证书 (15)
- SSL (16)
- C语言 (80)
- 进程 (26)
- vim (1)

有奖征资源，博文分享有内涵

人气博主的资源共享：老罗的Android之旅

关注CSDN社区微信，福利多多

《Hadoop高级编程》有奖试读

keepalived配置文件解析系列之(二)keyword存储的设计与实现

分类：keepalived

2013-06-27 10:52

144人阅读

评论(0)

收藏

举报

与关键字相关的数据结构和操作主要在lib/parser.c文件中，下面介绍其中的核心内容。

一、关键字相关数据结构

作为关键字，首先需要有一个名字name来区分，当解析配置文件时遇到该关键字要进行什么操作则由函数handler来表达，最后单独的一个关键字是作为整个层次关键字中的一员，需要有相应的成员去关联下一级别的关键字，这个任务由 sub指针完成。具体的关键字定义如下：

```
1 /* keyword definition */
2 struct keyword {
3     char *string;
4     void (*handler) (vector);
5     vector sub;
6 };
```

位于同一层次的关键字构成一个列表(vector,向量)，C语言中没有直接的vector定义，keepalived采用定义了自己的列表(vector)，如下：

```
1 /* vector definition */
2 struct _vector {
3     unsigned int allocated;
4     void **slot;
5 };
6 typedef struct _vector *vector;
```

上面的定义是很常见的，allocated对应列表的大小，而slot则对应列表的内容。留意到slot的类型为void**，这意味着这样的列表存储的内容可以是任意数据类型的指针（在下文中，我们将看到，在层次关键字结构中，它用来存储指向关键字结构体keyword的指针）。

关键字的层次是用变量sublevel进行组织的，sublevel初始为0，表示关键字的层次从0开始，添加下一层次的关键字之前需要递增sublevel变量，需要回到上一层关键字之前要递减sublevel变量。

每层关键字对应一个或多个关键列表(取决于该列表由哪些关键字派生出来)。第零层关键字是必定存在的，所以定义了一个全局的关键字列表keywords(注意keepalived中以带s或者_vec结尾的名字区分单个关键字和关键字列表)

二、关键字层次组织

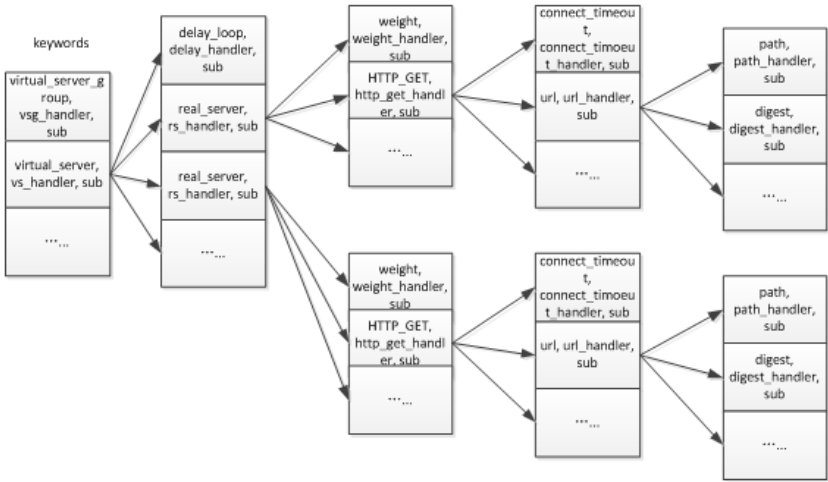
下面分别从check_init_keywords(void) (位于check_parser.c文件中) 和 install_http_check_keyword(void)函数中摘取相关语句组成一个分层的关键字结构。摘取的语句如下：

```
1 install_keyword_root("virtual_server_group", &vsg_handler);
2 install_keyword_root("virtual_server", &vs_handler);
3 install_keyword("delay_loop", &delay_handler);
4 install_keyword("real_server", &rs_handler);
5 install_sublevel();
6 install_keyword("weight", &weight_handler);
```

- 内核 (1)
- bkwin (16)
- 金山开源 (18)
- Lua (4)
- BOLT (15)
- CSP (6)
- MFC (2)
- Oracle (1)
- SHELL (8)
- Install Shield (37)
- xml (5)
- 开源 (2)
- PKCS#11 (4)
- 工具 (8)
- 服务 (5)
- 脚本 (4)
- USB (1)
- mysql (67)
- 开发 (3)
- ubuntu (2)
- apache (5)
- 网络Intenet (3)
- 知识 (6)
- web (3)
- CGI (3)
- 报文 (3)
- ESQL (10)
- java (1)
- 云计算 (2)
- php (2)
- ftp (1)
- javascript (6)
- jqplot (4)
- SQLite (6)
- 测试 (9)
- OPEN (0)
- OPENDPI (6)
- iptables (10)
- ModSecurity (1)
- LVS (51)
- 集群 (34)
- HA (11)
- Nginx (29)
- TUN (2)
- keepalived (31)
- Piranha (3)
- 负载均衡 (22)
- ADC (5)
- shhttpd (1)
- mongoose (13)
- HTTP (2)
- Django (37)
- HAproxy (1)
- zeroshell (2)
- DNS (5)
- route (5)
- BIND (3)
- 平时工作 (1)
- easyUI (4)
- RRD (18)
- SNMP (1)
- 正则表达式 (3)
- 源码解析 (22)
- matplotlib (12)
- highcharts (1)

```
7 install_keyword("HTTP_GET", &http_get_handler);
8 install_sublevel();
9 install_keyword("connect_timeout", &connect_to_handler);
10 install_keyword("url", &url_handler);
11 install_sublevel();
12 install_keyword("path", &path_handler);
13 install_keyword("digest", &digest_handler);
14 install_sublevel_end();
15 install_sublevel_end();
16 install_sublevel_end();
```

对应的关键字层次组织图如下:



上图中层次1中的第2个real_server在代码中对应的语句并没有写出来，在这里画出是为了展现同一层次的关键字如何关联各自的下一层关键字。此外，关键字的sub指针实际上指向一个列表（vector），为了更形象地表达这个点，图中采用了由sub指针指向列表中的各个元素的记法。keywords对应全局的关键字列表（即层次0的关键字），delay_loop、real_server等位于层次1，weight、HTTP_GET等位于层次2，connect_timeout、url等位于层次3，path、digest位于层次4。

三、关键字操作介绍

如上所介绍，keepalived通过install_keyword_root()、install_keyword()、Install_sublevel()和install_sublevel_end()等操作构建关键字层次组织。下面对这些关键字操作及它们内部调用的函数进行介绍。

install_keyword_root(char *string, void (*handler) (vector))为层次0（对应全局变量keywords）的关键字列表添加一个名字为string，处理函数为handler的关键字。具体的实现是通过调用keyword_alloc(keywords, string, handler)来实现的。

install_keyword(char *string, void (*handler) (vector))为层次为sublevel的关键字列表添加一个名字为string，处理函数为handler的关键字。具体的实现是通过调用keyword_alloc_sub(keywords, string, handler)来实现的。

install_sublevel(void)表示将处理下一层次的关键字列表，具体的实现是递增sublevel。

install_sublevel_end(void)表示回到上一层关键字列表，具体的实现是递减sublevel。

keyword_alloc(vector keywords_vec, char *string, void (*handler) (vector))为关键字列表keywords_vec添加一个名字为string，处理函数为handler的关键字。具体的实现是先调用vector_alloc_slot(keywords_vec)为要新增加的关键字分配内存空间，再用string和handler构造一个关键字，最后调用vector_set_slot(keywords_vec, keyword)添加新的关键字到关键字列表keywords_vec中。

keyword_alloc_sub(vector keywords_vec, char *string, void (*handler) (vector))为当前关键字列表keywords_vec的最后一个关键字的第sublevel层关键字列表的最后一个元素的下一层关键字列表添加一个名字为string，处理函数为handler的关键字。具体的实现是先判断当前关键字列表keywords_vec的最后一个关键字的第sublevel层关键字列表的最后一个元素的下一层关键字列表是否存在，若不存在的话，调用vector_alloc()分配下一层的关键字列表的内存空间。最后调用keyword_alloc(keyword->sub, string, handler)为该层添加新的关键字。

四、小结

如上，keepalived的关键字采用图的方式进行组织，每层关键字的数量是不限的，任意一个关键字都可以有下一层次的关键字，具有很大的灵活性，所以平时在C类项目中若有解析具有这样特点的配置文件时，可以考虑复用keepalived的这一设计。

下面的文章将会介绍如何根据这些关键字的组织解析配置文件。

- 管理 (1)
- QT (46)
- Eric4 (4)
- S (0)
- 性能 (4)
- 代码走查 (1)
- WAF (1)
- 邮件 (1)
- ARP (2)
- ACM (5)
- 网络 (0)
- log (0)
- logstash (3)
- 防篡改 (1)
- PyRTF (1)
- Tengine (1)
- IP地址库 (3)
- 防盗链 (1)

文章存档

2014年06月 (1)

2014年05月 (3)

2014年04月 (4)

2014年03月 (3)

2014年02月 (6)

展开

阅读排行

一个完整的安装程序实例 (3608)

一个完整的安装程序实例 (3451)

一个完整的安装程序实例 (3164)

关于Installshield里一些? (3157)

一个完整的安装程序实例 (3109)

一个完整的安装程序实例 (2933)

如何为工程添加一个Inst (2448)

PKI/CA (1896)

Linux进程间通信之共享F (1642)

Bolt引擎内置的元对象介 (1609)

评论排行

Bolt引擎内置的元对象介 (3)

Thinkphp的RBAC，基于 (2)

(转)什么是套接字(Socket (2)

Mongoose笔记——mair (2)

PyQt4 精彩实例分析* 实 (2)

金山开源安全卫士全套代 (1)

套接字编程中,select:bad (1)

关于Installshield里一些? (1)

头文件.h和源文件.cpp的 (1)

linux修改系统时间 (1)

推荐文章

最新评论

套接字编程中,select:bad file des
xiaoyan19891227: 貌似没有解
决我的问题啊

Thinkphp的RBAC，基于角色的
yellowxiaotian: 3Q

上一篇 keepalived配置文件解析系列之(一)keepalived简介及配置文件介绍

下一篇 keepalived配置文件解析系列之(三)配置文件解析过程

主题推荐

存储

设计

数据结构

全局变量

c语言

猜你在找

c语言中的各种变量是如何存储的

C语言学习之变量存储

c语言 变量存储位置的区别。

北航 991数据结构与C语言程序设计考试大纲 (2014

c数据结构的顺序存储的实现

数据结构之栈的push与pop操作 (顺序存储结构的c实

C程序设计语言 chapter1 1.10 变量和作用域

数据结构C语言实现系列——单链表

全局变量与局部变量定义及存储小结

keepalived配置文件解析系列之(三)配置文件解析过程

www.uuwise.com

验证码识别平台行业领导者

0秒极速识别90%图片,要速度找优优! !!!单机200图片/S线程随便跑

极速、稳定、大开发支持

30台集群服务器5千名工人7x24小时不间断工作

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

Kibana+Logstash+Elasticsearch
Keep-Thinking: 大神，真心求教
logstash，这几天项目继续这三
种日志技术，还望指点一二，本
人扣扣：4663753...

opendpi 源码分析（二）
outin0508: 请问下 那个demo代
码能不能直接用来处理pcap包

Mongoose笔记——main
ywwnuds: MONGOOSE_DLL int
mg_get_var(const char *data,
size...

Mongoose笔记——main
ywwnuds: "mg_get_var": 函数不
接受 2 个参数

winrar 4.2 64位注册码
fazwh: 好用。

PyQt4 精彩实例分析* 实例9 利用
897841089: "然后使用pyuic4命
令生成一个py文件，如下图所
示"这个过程啥意思，真实有点不
懂了

PyQt4 精彩实例分析* 实例9 利用
897841089: 写得不错，很实用

关于Installshield里一些常见问题
cvbcbv: nice

[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

京 ICP 证 070598 号

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

Copyright © 1999-2014, CSDN.NET, All Rights Reserved

