

blade2001的专栏

目录视图

摘要视图

RSS 订阅

个人资料



blade2001

访问: 1049340次

积分: 12918

等级:

BLOG > 7

排名: 第622名

原创: 160篇 转载: 562篇

译文: 3篇 评论: 97条

文章搜索

文章分类

C/C++ (30)

Learn from Bugs (5)

Linux (135)

OO/SE (3)

P2P (1)

US (17)

VOIP (76)

Web (49)

Windows (23)

创业 (24)

大规模分布计算(云、搜索引擎) (75)

安全 (19)

手机软件(J2ME&Android) (57)

数据库 (mysql/sqlserver/oracle) (21)

数据结构及算法 (9)

游戏开发相关 (15)

生活 (16)

网络/通信 (25)

脚本语言 (15)

软件 (8)

网络/通信 (96)

测试运维 (117)

产品设计&用户体验 (7)

Java (22)

Jabber/XMPP (12)

英语 (2)

关于使用libcurl的注意事项

标签: curl

2014-02-26 14:39 3020人阅读 评论(0) 收藏 举报

分类: Linux (134) C/C++ (29)

1. libcurl与CLOSE_WAIT

转载自: <http://blog.sunshow.net/2010/03/libcurl-and-close-wait/>

调用libcurl下载, 然后使用netstat查看发现有大量的TCP连接保持在CLOSE_WAIT状态

查看libcurl的文档说明, 有这样一个选项:

CURLOPT_FORBID_REUSE

Pass a long. Set to 1 to make the next transfer explicitly close the connection when done. Normally, libcurl keeps all connections alive when done with one transfer in case a succeeding one follows that can re-use them. This option should be used with caution and only if you understand what it does. Set to 0 to have libcurl keep the connection open for possible later re-use (default behavior).

也就是说, 默认情况下libcurl完成一个任务以后, 出于重用连接的考虑不会马上关闭

如果没有新的TCP请求来重用这个连接, 那么只能等到CLOSE_WAIT超时, 这个时间默认在7200秒甚至更高, 太多的CLOSE_WAIT连接会导致性能问题

解决方法:

curl_easy_setopt(curl, CURLOPT_FORBID_REUSE, 1);

最好再修改一下TCP参数调低CLOSE_WAIT和TIME_WAIT的超时时间

2. libcurl 使用笔记

转载自: <http://gcoder.blogbus.com/logs/54871550.html>

libcurl 是一个很不错的库, 支持http, ftp等很多的协议。使用库最大的心得就是, 不仔细看文档, 仅仅看着例子就写程序, 是一件危险的事情。我的程序崩溃了, 我怀疑是自己代码写的问题, 后来发现是库没用对。不仔细看文档 (有时候文档本身也比较差劲, 这时除了看仔细外, 还要多动脑子, 考虑它是怎么实现的), 后果很严重。不加思索的使用别人的库或者代码, 有时候很惬意, 但是出问题时, 却是寝食难安的。

1. CURLcode curl_global_init(long flags); 在多线程应用中, 需要在主线程中调用这个函数。这个函数设置libcurl所需的环境。通常情况, 如果不显式的调用它, 第一次调用curl_easy_init()时, curl_easy_init 会调用 curl_global_init, 在单线程环境下, 这不是问题。但是多线程下就不行了, 因为curl_global_init不是线程安全的。在多个线程中调用curl_easy_int, 然后如果两个线程同时发现curl_global_init还没有被调用, 同时调用 curl_global_init, 悲剧就发生了。这种情况发生的概率很小, 但可能性是存在的。

2. libcurl 有个很好的特性, 它甚至可以控制域名解析的超时。但是在默认情况下, 它是使用alarm + siglongjmp 实现的。用alarm在多线程下做超时, 本身就几乎不可能。如果只是使用alarm, 并不会导致程序崩溃, 但是, 再加上 siglongjmp, 就要命了 (程序崩溃的很可怕, core中几乎看不出有用信息), 因为其需要一个sigjmp_buf型的全局

<http://blog.csdn.net/blade2001/article/details/19978527>

1/4

视频 (14)

文章存档

2016年06月 (3)

2016年05月 (10)

2016年04月 (3)

2016年03月 (14)

2016年02月 (4)

展开

阅读排行

浅谈MD5加密算法中的加

Java线上应用故障排查之

告诉你一个真实的中国互

用python获取本地的ip地

AWS使用小记之EC2(Ele

虚拟化技术比较 PV HVM

分布式消息队列(消息中

美国奶酪（芝士） - chee

linux 查看即时网速 /流量

Linux Shell中判断某个环

(38432)

(14333)

(14234)

(12411)

(11333)

(10861)

(10028)

(10014)

(9254)

(8894)

评论排行

浅谈MD5加密算法中的加

开发Openfire聊天记录插

我和计算机技术—在迅雷

openfire服务器性能测试

利用Hudson构建c/c++持

Java线上应用故障排查之

python的GUI测试工具---

网游脚本浅谈

Linux的内存管理（free i

XMPP、HTTP 模拟压力；

(9)

(7)

(6)

(5)

(4)

(4)

(3)

(3)

(3)

(3)

推荐文章

*浅谈android中异步加载之"取消异步加载"二

*笑谈Android图表-----MPAndroidChart

*Nginx正反向代理、负载均衡等功能实现配置

*浅析ZeroMQ工作原理及其特点

*Android官方开发文档Training系列课程中文版：多样屏幕之支持不同的屏幕尺寸

*Spring Boot 实践折腾记（三）：三板斧，Spring Boot下使用Mybatis

最新评论

python的GUI测试工具---autolX

chenmiu2015: 死活搞不懂,dll应该是加载成功了,但是报AttributeError: function 'XXX...

Java线上应用故障排查之一：高f

j991349114: 真心不错，帮我发现了一个大问题产生的原因。谢

变量，多线程修改它。（通常情况下，可以每个线程一个 sigjmp_buf 型的变量，这种情况下，多线程中使用 siglongjmp 是没有问题的，但是libcurl只有一个全局变量，所有的线程都会用）。

具体是类似 curl_easy_setopt(curl, CURLOPT_TIMEOUT, 30L) 的超时设置，导致alarm的使用（估计发生在域名解析阶段），如前所述，这在线程中是不行的。解决方式是禁用掉alarm这种超时， curl_easy_setopt(curl, CURLOPT_NOSIGNAL, 1L)。

这样，多线程中使用超时就安全了。但是域名解析就没了超时机制，碰到很慢的域名解析，也很麻烦。文档的建议是 Consider building libcurl with c-ares support to enable asynchronous DNS lookups, which enables nice timeouts for name resolves without signals. c-ares 是异步的 DNS 解决方案。

3. libcurl 多线程使用注意事项

转载自:<http://blog.csdn.net/jaylong35/article/details/6439549>

1、问题来源，多线程使用Libcurl导致程序跑一段时间后自己退出，没有明显的异常。找不到合适的BUG。

最后通过查看资料和网上找的一些文章，发现，原来是信号处理的问题：

CURLOPT_NOSIGNAL

Pass a long. If it is 1, libcurl will not use any functions that install signal handlers or any functions that cause signals to be sent to the process. This option is mainly here to allow multi-threaded unix applications to still set/use all timeout options etc, without risking getting signals. (Added in 7.10)

If this option is set and libcurl has been built with the standard name resolver, timeouts will not occur while the name resolve takes place. Consider building libcurl with c-ares support to enable asynchronous DNS lookups, which enables nice timeouts for name resolves without signals.

Setting CURLOPT_NOSIGNAL to 1 makes libcurl NOT ask the system to ignore SIGPIPE signals, which otherwise are sent by the system when trying to send data to a socket which is closed in the other end. libcurl makes an effort to never cause such SIGPIPEs to trigger, but some operating systems have no way to avoid them and even on those that have there are some corner cases when they may still happen, contrary to our desire.

就是当多个线程都使用超时处理的时候，同时主线程中有sleep或是wait等操作。如果不设置这个选项，libcurl将会发信号打断这个wait从而导致程序退出。

所以，在使用的时候把这个选项设置成1就可以了。

curl_setopt(curl, CURLOPT_NOSIGNAL, 1L);

2、关于libcurl库的初始化和关闭：curl_global_init()和curl_global_cleanup()

这两个函数并不是线程安全的。所以只能在线程中进行一次的初始化和清除。

虽然这个不是一定就会有问题，但是如果这样处理还是有概率发生的。

问题: 多线程libcurl运行一段时间后出现崩掉，没有确定的点，没有确定的URL。一直查看源代码没有问题，最后通过debug跟踪发现是在访问SSL的时候出现的crash。<http://blog.csdn.net/jaylong35/article/details/6988690>

才想起来openssl是不支持多线程的，要自己做加锁处理。而且libcurl中并没有支持相关的加锁操作。

解决办法：

在初始化libcurl的时候为openssl创建一个互斥锁函数，一个回调函数传给openssl

openssl锁函数原形：void (* func)(int,int, const char *,int)

设置方式：CRYPTO_set_locking_callback(void (* func)(int,int, const char *,int));

设置这样一个函数还不够，另外还要配置一个锁id回调函数，这个可以参考openssl多线程下的使用相关。

谢

openfire服务器性能测试的几种方法
qq_34714678: 你好 我想问下
tsung的管理端口怎么配置
啊???? 急急
急!!!!!!!!!!!!!!

Java线上应用故障排查之一: 高可用
zhu19774279: @ceekay_ Linux
的shell

开发Openfire聊天记录插件
kore:
interceptorManager.addInterceptor()
开发Openfire聊天记录插件
kore:
interceptorManager.addInterceptor()

HTML Encode 和URL Encode的
cniiaomingjian: 我看到的效果
是: 空格不是编码为加号
(+), 而是%20

AWS使用小记之EC2(Elastic Cloud
甘蓝: 太棒了

向前兼容与向后兼容
epy007: 很不错, 解释的很清楚。

openfire服务器性能测试的几种方法
yzx200712256: 你好, , 想请问
一下, 一般openfire服务器的测
试方案有哪些啊?

Blog

良少的专栏

大宝(sodme) (RSS)

李学凌 (RSS)

云风

fishman

传奇游戏源码深入分析 (RSS)

雷军: 人因梦想而伟大

开发网站

flipcode.com

gametutorials

game engine

CodeProject.com

GameDev.net

RFC官方

永远的UNIX

游戏资源网

业界时事

IT 社区 & 媒体平台

思维的乐趣

id函数原形: unsigned int (*func)(void)

设置方式: CRYPTO_set_id_callback(unsigned int (*func)(void));

通过这两个设置就可以解决HTTPS多线程请求出现crash的问题。

问题: curl_easy_perform 一直阻塞很长时间没返回,

#0 0xf7746430 in __kernel_vsyscall ()

#1 0x00580fd6 in poll () from /lib/libc.so.6

#2 0x081e77fa in Curl_socket_check ()

#3 0x081ff1ca in Curl_do_perform ()

解决办法: 设超时: curl_easy_setopt(curl, CURLOPT_TIMEOUT, 30L)

顶

0

踩

0

上一篇 linux下共享文件夹给windows

下一篇 iperf测量 TCP 和 UDP 的网络性能

我的同类文章

Linux (134)

C/C++ (29)

• How To Debug and Troubleshoot... 2016-06-14 阅读 38

• 定制centos自动化安装ISO 2016-05-31 阅读 84

• Linux把内存挂载成硬盘提... 2016-05-20 阅读 76

• gdb常用用法 2015-07-20 阅读 190

• GDB调试release程序 2015-06-19 阅读 601

• netstat监控大量ESTABLIS... 2015-05-30 阅读 321

• Bash中关于日期时间操作... 2016-06-05 阅读 58

• 性能调优攻略 2016-05-23 阅读 113

• Linux Coredump信息收集 2015-12-08 阅读 136

• select、poll、epoll之间的... 2015-07-14 阅读 246

• Linux的IO性能监控工具iost... 2015-05-30 阅读 222

更多文章

猜你在找

iOS8-Swift开发教程

Windows Server 2012 DNS Server 管理

CCIE专家讲解思科认证网络认证CCNA v2.0网络工程师和

名称解析

全能项目经理训练营

libcurl 多线程使用注意事项补充HTTPSopenssl多线程

友盟统计

Ubuntu 上搭建 Ruby on Rails 生产环境

Tornado 学习

Tornado 概览

查看评论

暂无评论

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

http://blog.csdn.net/blade2001/article/details/19978527

3/4

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