

最新版Web服务器项目详解 - 13 踩坑和面试题

原创 互联网猿 两猿社 2020-06-10 09:56

点击关注上方 "[两猿社](#)"
设为 "[置顶或星标](#)", 干货第一时间送达。

互联网猿 | 两猿社

本文内容

本篇是项目的最终篇，将介绍踩坑与面试题两部分。

踩坑，描述做项目过程中遇到的问题与解决方案。

面试题，介绍项目相关的知识点变种和真实面试题，**这里不会给出答案**，具体的，可以在项目微信群中讨论。

踩坑

做项目过程中，肯定会遇到形形色色、大大小小的问题，但并不是所有问题都值得列出来探讨，这里仅列出个人认为有意义的问题。

具体的，包括大文件传输。

大文件传输

先看下之前的大文件传输，也就是游双书上的代码，发送数据只调用了writev函数，并对其返回值是否异常做了处理。

```
1  bool http_conn::write()
2  {
3      int temp=0;
4      int bytes_have_send=0;
5      int bytes_to_send=m_write_idx;
6      if(bytes_to_send==0)
7      {
8          modfd(m_epollfd,m_sockfd,EPOLLIN);
```

```

9         init();
10        return true;
11    }
12    while(1)
13    {
14        temp=writev(m_sockfd,m_iv,m_iv_count);
15        if(temp<=-1)
16        {
17            if(errno==EAGAIN)
18            {
19                modfd(m_epollfd,m_sockfd,EPOLLOUT);
20                return true;
21            }
22            unmap();
23            return false;
24        }
25        bytes_to_send-=temp;
26        bytes_have_send+=temp;
27        if(bytes_to_send<=bytes_have_send)
28        {
29            unmap();
30            if(m_linger)
31            {
32                init();
33                modfd(m_epollfd,m_sockfd,EPLLIN);
34                return true;
35            }
36            else
37            {
38                modfd(m_epollfd,m_sockfd,EPLLIN);
39                return false;
40            }
41        }
42    }
43 }

```

在实际测试中发现，当请求小文件，也就是调用一次writev函数就可以将数据全部发送出去的时候，不会报错，此时不会再次进入while循环。

一旦请求服务器文件较大文件时，需要多次调用writev函数，便会出现问题，不是文件显示不全，就是无法显示。

对数据传输过程分析后，定位到writev的m_iv结构体成员有问题，每次传输后不会自动偏移文件指针和传输长度，还会按照原有指针和原有长度发送数据。

根据前面的基础API分析，我们知道writev以顺序iov[0]，iov[1]至iov[iovcnt-1]从缓冲区中聚集输出数据。项目中，申请了2个iov，其中iov[0]为存储报文状态行的缓冲区，iov[1]指向资源文件指针。

对上述代码做了修改如下：

- 由于报文消息报头较小，第一次传输后，需要更新m_iv[1].iov_base和iov_len，m_iv[0].iov_len置成0，只传输文件，不用传输响应消息头
- 每次传输后都要更新下次传输的文件起始位置和长度

更新后，大文件传输得到了解决。

```

1    bool http_conn::write()
2    {
3        int temp = 0;

```

```

4
5     int newadd = 0;
6
7     if (bytes_to_send == 0)
8     {
9         modfd(m_epollfd, m_sockfd, EPOLLIN, m_TRIGMode);
10        init();
11        return true;
12    }
13
14    while (1)
15    {
16        temp = writev(m_sockfd, m_iv, m_iv_count);
17
18        if (temp >= 0)
19        {
20            bytes_have_send += temp;
21            newadd = bytes_have_send - m_write_idx;
22        }
23        else
24        {
25            if (errno == EAGAIN)
26            {
27                if (bytes_have_send >= m_iv[0].iov_len)
28                {
29                    m_iv[0].iov_len = 0;
30                    m_iv[1].iov_base = m_file_address + newadd;
31                    m_iv[1].iov_len = bytes_to_send;
32                }
33                else
34                {
35                    m_iv[0].iov_base = m_write_buf + bytes_have_send;
36                    m_iv[0].iov_len = m_iv[0].iov_len - bytes_have_send;
37                }
38                modfd(m_epollfd, m_sockfd, EPOLLOUT, m_TRIGMode);
39                return true;
40            }
41            unmap();
42            return false;
43        }
44        bytes_to_send -= temp;
45        if (bytes_to_send <= 0)
46        {
47            {
48                unmap();
49                modfd(m_epollfd, m_sockfd, EPOLLIN, m_TRIGMode);
50
51                if (m_linger)
52                {
53                    init();
54                    return true;
55                }
56                else
57                {
58                    return false;
59                }
60            }
61        }
62    }

```

面试题

包括项目介绍，线程池相关，并发模型相关，HTTP报文解析相关，定时器相关，日志相关，压测相关，综合能力等。

项目介绍

- 为什么要做这样一个项目？
- 介绍下你的项目

线程池相关

- 手写线程池
- 线程的同步机制有哪些？
- 线程池中的工作线程是一直等待吗？
- 你的线程池工作线程处理完一个任务后的状态是什么？
- 如果同时1000个客户端进行访问请求，线程数不多，怎么能及时响应处理每一个呢？
- 如果一个客户请求需要占用线程很久的时间，会不会影响接下来的客户请求呢，有什么好的策略呢？

并发模型相关

- 简单说一下服务器使用的并发模型？
- reactor、proactor、主从reactor模型的区别？
- 你用了epoll，说一下为什么用epoll，还有其他复用方式吗？区别是什么？

HTTP报文解析相关

- 用了状态机啊，为什么要用状态机？
- 状态机的转移图画一下
- https协议为什么安全？
- https的ssl连接过程
- GET和POST的区别

数据库登录注册相关

- 登录说一下？
- 你这个保存状态了吗？如果要保存，你会怎么做？（cookie和session）
- 登录中的用户名和密码你是load到本地，然后使用map匹配的，如果有10亿数据，即使load到本地后hash，也是很耗时的，你要怎么优化？
- 用的mysql啊，redis了解吗？用过吗？

定时器相关

- 为什么要用定时器？
- 说一下定时器的工作原理
- 双向链表啊，删除和添加的时间复杂度说一下？还可以优化吗？

- 最小堆优化？说一下时间复杂度和工作原理

日志相关

- 说下你的日志系统的运行机制？
- 为什么要异步？和同步的区别是什么？
- 现在你要监控一台服务器的状态，输出监控日志，请问如何将该日志分发到不同的机器上？（消息队列）

压测相关

- 服务器并发量测试过吗？怎么测试的？
- webbench是什么？介绍一下原理
- 测试的时候有没有遇到问题？

综合能力

- 你的项目解决了哪些其他同类项目没有解决的问题？
- 说一下前端发送请求后，服务器处理的过程，中间涉及哪些协议？

如果本文对你有帮助，[阅读原文](#) star一下服务器项目，我们需要你的星星^_^.

完。