Bug report as following:

- Nginx -1.10.2
- Uname -Linux 2.6.32-642.13.1.el6.x86_64

We use nginx as a reverse proxy server , and enable ngx_http_slice_module.

We set up nginx with the following configuration:

```
http
{       include                         mime.types;
        default_type                    application/octet-stream;
        server_tokens                   off;
        server_names_hash_bucket_size   512;
        client_header_buffer_size       32k;
        large_client_header_buffers     4 32k;
        client_max_body_size            5000m;
        sendfile                        on;
        server_names_hash_bucket_size   512;
        client_header_buffer_size       32k;
        large_client_header_buffers     4 32k;
        client_max_body_size            5000m;
        sendfile                        on;
        tcp_nopush                      on;
        keepalive_timeout               90;
        tcp_nodelay                     on;
        limit_rate_after 5m;
        limit_rate       500k;
        client_body_buffer_size         512k;
        proxy_connect_timeout           10;
        proxy_read_timeout              600;
        proxy_send_timeout              600;
        proxy_buffer_size               32k;
        proxy_buffers                   2 32k;
        proxy_max_temp_file_size        0;
        proxy_busy_buffers_size         32k;
        proxy_temp_file_write_size      256k;
        proxy_set_header                J-Forwarded-For $remote_addr;
        proxy_set_header                Host $host;
        ssl_session_cache               shared:SSL:300m;

        server {
            listen                  80;
            server_name             test.nginx.com;
            include method.conf;
            include gzip.conf;
            include setheaders.conf;
            proxy_set_header                Host $host;

            slice 4m;
            proxy_set_header  Range $slice_range;

            location / {
                proxy_pass http://jcs;
            }
        }
}
```

We have a problem:

When the client receives the first slice body, nginx no longer sends the second slice body, and the connection (for client) is hanging .

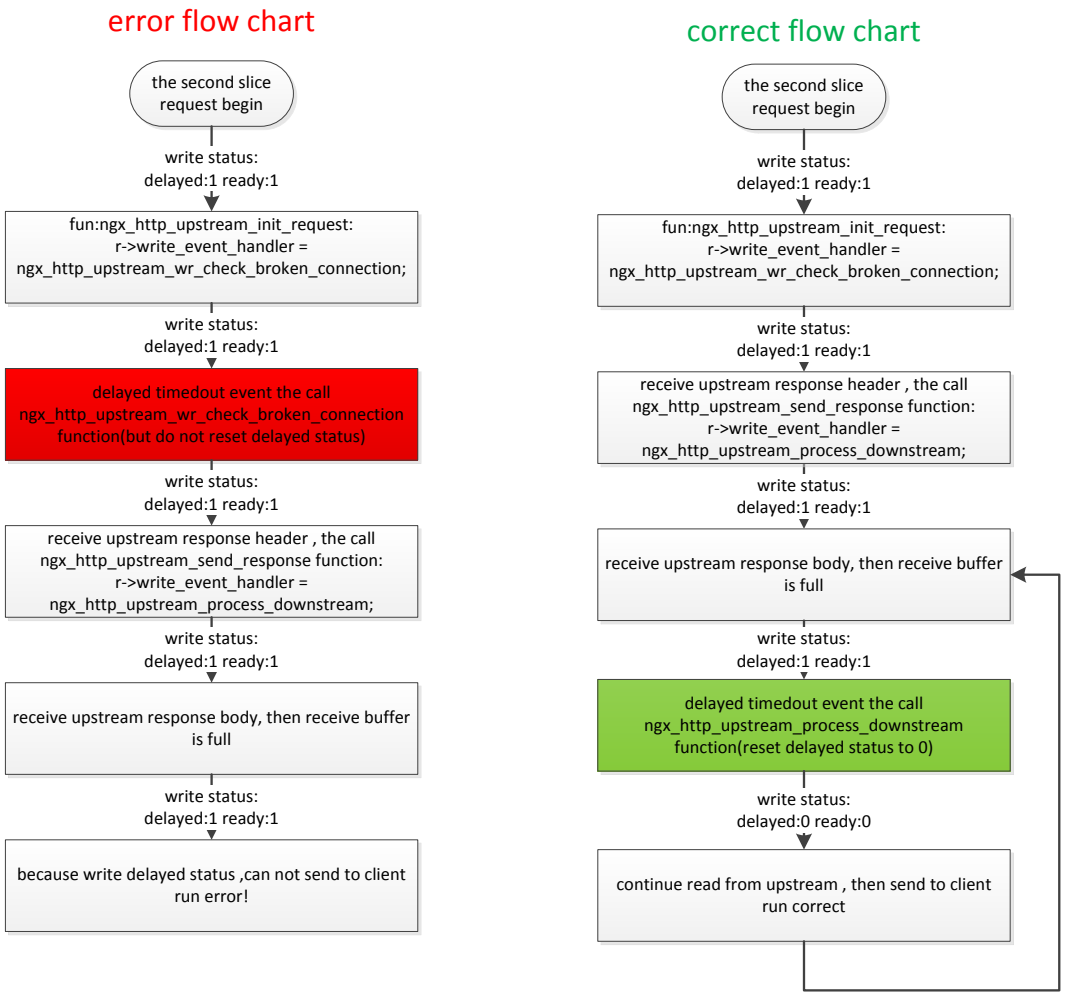
Then we debug the problem, we found that before the second slice body sent to client , the connection's(for client) write's status as following:

ready is 1 , delayed is 1

After upstream_module use ngx_event_pipe_read_upstream function to read some body response from upstream, then the buffer is full, but due to the write's delayed status, the response cannot be sent to the client. Because there is no event to call write_event_handler to reset delayed status and timedout status. The upstream_module step in error cycle.

We suspect that this problem is due to execute the wrong timer event handler.

Flow chart as following:



We look forward your early reply.

Thank you for your attention to this matter.

Reporter personal information as following:

Full name: Yanbin Zhai

Email: zhaiyanbin@jd.com

Corporation: Beijing Jingdong Shangke Information Technology