生产环境服务器性能优化指导手册

Guide for Production Environment Server Performance Optimization

版本： 1.0.0

拟制: 蒋 赞

修改记录

文件编号	版本号	拟制人/修改人	拟制/修改日期	更改理由	主要更改内容
	1.0.0	蒋赞	2015-09-26		起草
注 1：每次更改归档文档时必须填写该表 注 2：一次归档时，“更改理由”和“主要更改内容”填无					

目录

1. 系统的优化

服务器操作系统为 CentOS 64 位服务器版，版本一般选用 6.5 或 7.0，建议采用 7.0 版本。

CentOS 官方网址为：http://www.centos.org/

通常下载的 CentOS 版本如下：

http://mirrors.aliyun.com/centos/7/isos/x86_64/CentOS-7.0-1406-x86_64-Minimal.iso (推荐)

http://mirrors.aliyun.com/centos/7/isos/x86_64/CentOS-7.0-1406-x86_64-Everything.iso

1.1 磁盘分区

一般 CentOS 安装时，我们采用 CentOS-7.0-1406-x86_64-Minimal.iso 即可，我们可以将 ISO 利用 UltraISO 烧入 U 盘进行安装，也可以采用 VMWare 或 VirtualBox 进行虚拟化安装，在安装过程中我们采用默认方式安装即可，如英文、美式键盘等，需要特别注意的地方是磁盘的分区时，选择 Standard Partition 分区，分区大小及分配见下表：

分区名称	分区大小(M)	分区类型	备注
/	204800	ext4	根分区
/boot	512	ext4	启动分区，必须有，否则无法启动
/swap	与物理内存相当	swap	交换分区，当物理内存不够时，会启用交换分区
/data	所有剩余空间	ext4	独立一个分区做为数据盘，当系统损坏时，确保数据不丢失

对于 CentOS6.5，一般采用 ext4 的分区文件类型，对于 CentOS7 我们推荐采用 xfs 分区文件类型。

1.2 网络配置

```
#vi /etc/sysconfig/network-scripts/ifcfg-eth0
IPV6INIT="no"
IPV6_AUTOCONF="no"
IPADDR=192.168.128.222
NETMASK=255.255.255.0
GATEWAY=192.168.128.1
#NETWORK=192.168.128.0
#BROADCAST=192.168.128.255
```

1.3 SSH 服务配置

```
#vi /etc/ssh/sshd_config
```

```
IgnoreRhosts yes
UseDNS no
//然后重启 sshd 服务
//for centos6
#/etc/init.d/sshd restart
//for centos7
#systemctl restart sshd
```

1.4 系统配置

1.4.1 安装 sudo 及配置权限

```
#yum install sudo
#vi /etc/pam.d/su
auth required pam_wheel.so use_uid group=wheel
#visudo
%wheel    ALL=(ALL)    NOPASSWD: ALL
%www    ALL=/bin/vi, /usr/bin/vim, /usr/bin/tail, /bin/more, /bin/ls, /bin/rm, /bin/mv, /bin/cp, /bin/cat
%mis        ALL=/bin/vi, /usr/bin/vim, /usr/bin/fab, /usr/bin/tail, /bin/cat, /bin/more, /bin/rm, /bin/mv, /bin/ls, /bin/ping, /sbin/service, /bin/kill, /usr/bin/kill, /usr/bin/killall, /bin/chown, /bin/chmod, /bin/chgrp, /usr/bin/yum, /bin/mkdir, /bin/touch, /bin/cp, /bin/netstat, /usr/bin/python, /sbin/iptables
```

上面设置了两个组：

- 一个 wheel 组(管理员)，具备 sudo 权限，可以提权到 root 权限
- 一个 www 组(网站维护组)，只能拥有固定的命令

通常还会设置一个 dev 组(开发组)，让开发人员只能在自己的 home 目录进行开发操作

1.4.2 删除无用的帐号与群组

为了安全起见，这里把不需要使用的帐号与群组全部删除

```
userdel adm
userdel lp
userdel sync
userdel shutdown
userdel halt
userdel news
userdel uucp
userdel operator
userdel games
userdel gopher
userdel ftp
groupdel adm
groupdel lp
groupdel news
groupdel uucp
groupdel games
groupdel dip
groupdel pppusers
groupdel audio
groupdel video
```

1.4.3 修改登录欢迎语

修改欢迎语，提示登录用户不要乱操作服务器

```
#vi /etc/motd

    Welcome to Nubia Production Compute Service!
    ---------------------------------------------------------
    Please read the following words carefully.
    1. Don't reboot this machine. (It can't wake up automicly)
    2. Don't delete any others file.
    3. Please be sure to use your own account.
    4. Only members of the wheel group can run sudo.
    ---------------------------------------------------------
```

1.4.4 创建基本的用户组

创建 www 组及用户，及常用的用户名

```
groupadd www
useradd -g www www -s /sbin/nologin -d /var/www
mkdir -p /data/home
useradd -g wheel nubia -s /bin/bash -d /data/home/nubia
usermod -G wheel,root nubia
useradd -g wheel jiangz -s /bin/bash -d /data/home/jiangz
```

上面的命令大致解说如下：

(1) 创建一个 www 用户及 www 组，同时确保 www 用户无登录权限，并将 www 用户锁定在/var/www 目录；

(2) 创建一个 nubia 帐号，并加入 wheel 与 root 组，默认用户目录为/data/home/nubia；

(3) 创建一个 jiangz 帐号，并加入 wheel 组，默认用户目录为/data/home/jiangz；

1.4.5 修改 ulimit 限制

```
//查看系统最大的文件打开数
#cat /proc/sys/fs/file-max
#vi /etc/security/limits.conf
* soft nofile 204800
* hard nofile 204800
* soft nproc 65535
* hard nproc 65535
```

或者使用如下命令亦可

```
sudo sh -c 'echo "* soft nofile 204800" >> /etc/security/limits.conf'
sudo sh -c 'echo "* hard nofile 204800" >> /etc/security/limits.conf'
//for centos6
sudo sed -i 's/4096/65535/' /etc/security/limits.d/20-nproc.conf
//for centos7
sudo sed -i 's/1024/65535/' /etc/security/limits.d/90-nproc.conf
```

上面的命令设置系统单进程可以打开的文件数为 204800 个，系统可以开启 65535 个进程。

1.4.6 修改主机名

1) CentOS6 下修改主机名

```
# vi /etc/hosts
//在 127.0.0.1 与::1 后面增加主机名
# vi /etc/sysconfig/network
//修改主机名(HOSTNAME=)
//输入以下命令：
hostname  新主机名
```

2) CentOS7 下修改主机名

CentOS 7 里面修改 hostname 的方式有所改变，修改/etc/hosts 和/etc/sysconfig/network 两个文件已经不能生效。使用的新命令：

```
$ hostnamectl set-hostname <hostname>
```

主机名修改之后，然后用 ssh 客户端重新登录，就会显示新的主机名。

1.4.7 禁用 SELinux

```
#vi /etc/selinux/config
SELINUX=disabled
#setenforce 0 //临时生效
```

```
#getenforce //查看 selinux 状态
```

1.4.8 禁用不常用的服务

```
停止 postfix 服务
systemctl disable postfix
systemctl stop postfix
关闭防火墙
service firewalld stop
禁用防火墙
systemctl disable firewalld
```

1.5 内核优化

编辑/etc/sysctl.conf 内容如下：

```
net.ipv4.ip_forward = 0
net.ipv4.conf.all.rp_filter = 0
net.ipv4.conf.default.rp_filter = 0
net.ipv4.conf.default.accept_source_route = 0

kernel.sysrq = 0
kernel.core_uses_pid = 1
kernel.msgmnb = 65536
kernel.msgmax = 65536
kernel.shmmax = 68719476736
kernel.shmall = 4294967296

net.ipv4.neigh.default.gc_stale_time = 120
net.ipv4.conf.default.arp_announce = 2
net.ipv4.conf.all.arp_announce = 2
net.ipv4.conf.lo.arp_announce = 2

net.core.somaxconn = 81920
net.core.netdev_max_backlog = 81920
net.ipv4.ip_local_port_range = 1024 65000

net.ipv4.tcp_syncookies = 1
net.ipv4.tcp_keepalive_time = 300
net.ipv4.tcp_fin_timeout = 30

net.ipv4.tcp_timestamps = 1
net.ipv4.tcp_tw_reuse = 1
net.ipv4.tcp_tw_recycle = 1
#net.ipv4.tcp_tw_timeout = 60

net.ipv4.tcp_max_tw_buckets = 20000

net.ipv4.tcp_max_syn_backlog = 16384
net.ipv4.tcp_synack_retries = 3
net.ipv4.tcp_max_orphans = 131072
net.ipv4.tcp_no_metrics_save = 1
net.ipv4.tcp_sack = 1
net.ipv4.tcp_window_scaling = 1

#vm.overcommit_memory = 1
#net.ipv4.tcp_orphan_retries = 3
```

使之生效#sysctl -p
#cat /proc/sys/net/ipv4/tcp_tw_timeout
#cat /proc/sys/net/ipv4/tcp_timestamps

1.6 操作日志记录

```
#!/bin/sh

#history
#export HISTTIMEFORMAT="[%Y.%m.%d %H:%M:%S]"
export HISTTIMEFORMAT="`whoami` %F %T   "
USER_IP=`who -u am i 2>/dev/null| awk '{print $NF}'|sed -e 's/[()]//g'`
HISTDIR=/var/log/history

if [ -z $USER_IP   ]
then
   USER_IP=`hostname`
fi

if [ ! -d $HISTDIR ]
```

```
then
   sudo mkdir -p $HISTDIR
   sudo chmod 777 $HISTDIR
fi

if [ ! -d $HISTDIR/${LOGNAME} ]
then
   mkdir -p $HISTDIR/${LOGNAME}
   chmod 300 $HISTDIR/${LOGNAME}
fi

export HISTSIZE=4096
DT=`date +%Y%m%d_%H%M%S`
#export HISTFILE="$HISTDIR/${LOGNAME}/${USER_IP}_history.$DT"
export HISTFILE="$HISTDIR/${LOGNAME}/${USER_IP}_history.log"
chmod 600 $HISTDIR/${LOGNAME}/*_history* 2>/dev/null
```

2. MySQL 配置优化

这里的 MySQL 配置仅适应生产环境，以 8 核 16G 为例而优化的。MySQL 的优化主要通过 MySQL 的配置文件来设置。

```
 [mysqld]
user                  = mysql
port                  = 3306
socket                = /etc/mysql.sock
basedir          = /usr/local/mysql
datadir          = /data/database/mysql

slow_query_log        = 1
slow_query_log_file = /data/database/mysql/mysql_query_slow.log
long_query_time       = 1

server-id        = 1
log-bin               = master-bin
log-bin-index         = master-bin.index

#binlog-do-db=blog
#binlog-do-db=nubia_bbs
binlog-ignore-db=mysql,test,information_schema

#sql_mode=NO_ENGINE_SUBSTITUTION,STRICT_TRANS_TABLES
#sql_mode = NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION
sql_mode                      = ANSI

back_log                  = 511
sync_binlog               = 1
skip-name-resolve         = 1
lower_case_table_names    = 1
binlog_format             = mixed
binlog_cache_size         = 4M

max_connections       = 10000
max_connect_errors    = 10000

thread_cache_size     = 256    #1g:16 2g:32 4g:64 16g:256
thread_concurrency    = 8      #2*cpu cores
key_buffer_size  = 2048M       #1/8 RAM
```

```
innodb_data_file_path             = ibdata1:1024M:autoexten
innodb_buffer_pool_size       = 8G        #1/2

innodb_additional_mem_pool_size = 16M
innodb_log_file_size          = 256M
innodb_log_buffer_size            = 16M
innodb_log_files_in_group         = 3
innodb_lock_wait_timeout          = 60
innodb_file_per_table             = 1
innodb_flush_log_at_trx_commit    = 2
innodb_file_io_threads            = 8
innodb_max_dirty_pages_pct        = 90

default-storage-engine   = MyISAM
table_cache          = 512
external-locking     = FALSE
max_allowed_packet       = 32M
sort_buffer_size     = 2M
join_buffer_size     = 2M
query_cache_size         = 64M
query_cache_limit        = 4M
query_cache_min_res_unit = 2k
thread_stack             = 192K
transaction_isolation  = REPEATABLE-READ
tmp_table_size           = 256M
max_heap_table_size  = 256M
max_binlog_cache_size    = 8M
max_binlog_size      = 512M
expire_logs_days         = 7
read_buffer_size     = 1M
read_rnd_buffer_size  = 16M
bulk_insert_buffer_size = 64M

myisam_sort_buffer_size = 128M
myisam_max_sort_file_size = 10G
myisam_max_extra_sort_file_size = 10G
myisam_repair_threads    = 1
myisam_recover           = 1

[mysql]
no-auto-rehash
```

```
[mysqldump]
quick
max_allowed_packet = 256M

[mysqld_safe]
open-files-limit    = 10240
```

特别注意一下参数 innodb_flush_log_at_trx_commit 对性能的影响比较大，说明如下：

innodb_flush_log_at_trx_commit 参数解释：

0（延迟写）： log_buff --每隔 1 秒--> log_file —实时—> disk

1（实时写，实时刷）： log_buff —实时—> log_file —实时—> disk

2（实时写，延迟刷）： log_buff —实时—> log_file --每隔 1 秒--> disk

3. Tomcat 配置优化

生产环境的 Tomcat 请采用如下两个：

http://10.204.76.222/down/tomcat/tomcat-7.0.59.compiled.apr-1.1.32_for_centos7.zip

http://10.204.76.222/down/tomcat/tomcat-8.0.20.compiled.apr-1.1.32_for_centos7.zip

这两个 Tomcat 是经过定制与特定优化的，对于性能的提升非常显著，同时单个应用采用独立 JVM，应用之间互相隔离。

3.1 属性文件

```
handlers = 1catalina.org.apache.juli.FileHandler, 2localhost.org.apache.juli.FileHandler,
java.util.logging.ConsoleHandler

.handlers = 1catalina.org.apache.juli.FileHandler, java.util.logging.ConsoleHandler

1catalina.org.apache.juli.FileHandler.level = FINE
1catalina.org.apache.juli.FileHandler.directory = /data/logs/tomcat-8.0.12
1catalina.org.apache.juli.FileHandler.prefix = demo.catalina.

2localhost.org.apache.juli.FileHandler.level = FINE
2localhost.org.apache.juli.FileHandler.directory = /data/logs/tomcat-8.0.12
2localhost.org.apache.juli.FileHandler.prefix = demo.localhost.

java.util.logging.ConsoleHandler.level = FINE
java.util.logging.ConsoleHandler.formatter = java.util.logging.SimpleFormatter

org.apache.catalina.core.ContainerBase.[Catalina].[localhost].level = INFO
org.apache.catalina.core.ContainerBase.[Catalina].[localhost].handlers =
2localhost.org.apache.juli.FileHandler
```

3.2 配置文件

//TODO

```
<?xml version="1.0" encoding="utf-8"?>
<Server port="9001" shutdown="SHUTDOWN">
  <Listener className="org.apache.catalina.core.AprLifecycleListener" SSLEngine="off" />
  <!--Listener className="org.apache.catalina.core.JasperListener" /-->
  <Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener" />
  <Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener" />
  <Listener className="org.apache.catalina.core.ThreadLocalLeakPreventionListener" />

  <Service name="Catalina">
    <Executor name="tomcatThreadPool" namePrefix="catalina-exec-" maxThreads="1024"
minSpareThreads="64"/>

  <Connector      executor="tomcatThreadPool" port="8001"
protocol="org.apache.coyote.http11.Http11AprProtocol"    maxHttpHeaderSize="8192"
useBodyEncodingForURI="true" acceptCount="1024" keepAliveTimeout="60000"
maxKeepAliveRequests="256" enableLookups="false" connectionTimeout="20000"
disableUploadTimeout="true" URIEncoding="UTF-8" address="localhost"/>

    <Engine name="Catalina" defaultHost="localhost">
      <Host name="localhost"    appBase="webapps" unpackWARs="false" autoDeploy="false">
              <Context path="/" docBase="/data/wwwroot/demo" reloadable="false"
crossContext="false" privileged="true" workDir="work/demo_8001"/>
              <!--Valve className="org.apache.catalina.valves.RemoteAddrValve" allow="10\..*" /-->
              <!--Valve className="org.apache.catalina.valves.AccessLogValve"
directory="/data/logs/tomcat-8.0.11/"
                               prefix="open.nubia.cn_8001_access_log" suffix=".txt"
                                         pattern="%h %l %u %t &quot;%r&quot; %s %b" /-->
      </Host>
    </Engine>
  </Service>
</Server>
```

3.3 应用启动

```
sudo -E -u www ./demo_8001.sh start
```

3.4 应用停止

```
sudo -E -u www ./demo_8001.sh stop
```

3.5 应用调试

```
sudo -E -u www ./demo_8001.sh run
```

4. PHP 配置优化

4.1 php.ini 优化

第三方扩展只保留

redis.so

其它 mongo.so,memcached.so,gd.so,opcache.so 全注释掉

session 存储在 redis 中,减少 IO 读写

session.save_handler = redis

session.save_path = tcp://127.0.0.1:6379

date.timezone = "Asia/Shanghai"

extension_dir = /data/program/php-5.5.15/lib/php/extensions/no-debug-non-zts-20121212/

extension=gd.so

extension=redis.so

4.2 Opcache 优化

注意：保留独立的 opcache 模块配置

```
[opcache]
zend_extension=/data/program/php-5.5.15/lib/php/extensions/no-debug-non-zts-20121212/opcache.so
opcache.memory_consumption=128
opcache.optimization_level=1
opcache.interned_strings_buffer=8
opcache.max_accelerated_files=4096
opcache.revalidate_freq=60
opcache.fast_shutdown=1
opcache.enable=1
opcache.enable_cli=1
```

4.3 php-fpmX.conf 优化

将 listen = /tmp/php55-cgi1.socket 替换为 listen = /dev/shm/php55-cgi1.socket

4.4 集群服务

```
upstream phpbackend {
  server unix:/tmp/php53-cgi1.socket weight=100 max_fails=3 fail_timeout=30s;
  server unix:/tmp/php53-cgi2.socket weight=100 max_fails=3 fail_timeout=30s;
  server unix:/tmp/php53-cgi3.socket weight=100 max_fails=3 fail_timeout=30s;
  server unix:/tmp/php53-cgi4.socket weight=100 max_fails=3 fail_timeout=30s;
 }

upstream phpbackend55 {
        server unix:/dev/shm/php55-cgi1.socket weight=100 max_fails=3 fail_timeout=30s;
        server unix:/dev/shm/php55-cgi2.socket weight=100 max_fails=3 fail_timeout=30s;
        server unix:/dev/shm/php55-cgi3.socket weight=100 max_fails=3 fail_timeout=30s;
        server unix:/dev/shm/php55-cgi4.socket weight=100 max_fails=3 fail_timeout=30s;
}
```

4.5 压测数据

单压nginx静态页面 1W/s (采用ab -k)

单压nginx+ 4个php主进程 1060/s (采用ab -k，listen=/tmp/php55-cgi1.socket)

单压nginx+ 1个php主进程 1038/s，(采用ab -k，listen=/dev/shm/php55-cgi1.socket)

单压nginx+ 1个php主进程 1369/s，(采用ab，listen=127.0.0.1:9001)

单压nginx+ 1个php主进程 ，(采用ab -k，listen=/dev/shm/php55-cgi1.socket)

单压nginx+ 4个php主进程 1600/s，(采用jmeter 600/50，listen=/dev/shm/php55-cgi1.socket, session=/dev/shm/session)

单压nginx+ 4个php主进程 1200/s，(采用jmeter 600/50，listen=/dev/shm/php55-cgi1.socket, session=tcp.redis)

5. Nginx 配置优化

这里以 4 核 8G 服务器为例，下面的各配置仅做生产环境配置的参考，不特指具体的项目。

5.1 主配置

nginx.conf 通用配置

```
user   www www;
worker_processes 4;
worker_cpu_affinity 0001 0010 0100 1000;

error_log   /data/logs/nginx-1.7.9/error.log error;

#error_log logs/error.log notice;
#error_log logs/error.log info;
pid /tmp/nginx.pid;

worker_rlimit_nofile 40960;

events {
   use epoll;
   worker_connections 10240;
   multi_accept on;
   accept_mutex off;
}

http {
      include          mime.types;
      default_type   application/octet-stream;

  log_format   access   '$http_x_forwarded_for $remote_addr [$time_local] "$request" '
                          '$status $body_bytes_sent "$http_referer" '
                          '"$http_user_agent"';

      sendfile           on;

      tcp_nodelay        on;
      tcp_nopush         on;

      server_tokens off;        #关闭版本显示
```

```
    fastcgi_connect_timeout 300;
    fastcgi_send_timeout 300;
    fastcgi_read_timeout 300;
    fastcgi_buffer_size 256k;
    fastcgi_buffers 16 256k;
    fastcgi_busy_buffers_size 512k;
    fastcgi_temp_file_write_size 512k;

    #设定请求缓冲
    server_names_hash_bucket_size 128;

    client_body_buffer_size    128K;
    client_header_buffer_size       16k;
    large_client_header_buffers    4 64k;
    client_max_body_size 8m;

    client_body_timeout      30;
    client_header_timeout 10;
    keepalive_timeout         65;
    send_timeout                   30;

    proxy_connect_timeout      90;
    proxy_send_timeout            90;
    proxy_read_timeout           90;
    proxy_buffer_size            128k;
    proxy_buffers                   4 256k;
    proxy_busy_buffers_size 256k;
    proxy_temp_file_write_size 64m;
    proxy_ignore_client_abort on;

    open_file_cache max=204800 inactive=20s;
    open_file_cache_min_uses 1;
    open_file_cache_valid 30s;

    #开启 gzip 模块
    gzip on;
    gzip_min_length    1k;
    gzip_http_version 1.1;
    gzip_buffers         8 32k;
    gzip_comp_level 2; #0-9  默认值为 1，值越大压缩率越高，消耗的 cpu 资源越多，传输量减小
    gzip_types                    text/plain  text/css  application/json  application/x-javascript  text/xml
application/xml application/xml+rss text/javascript;
```

```
    gzip_vary on;

    #gzip_static on;
    gzip_proxied             any;
    gzip_disable       "MSIE [1-6]\.";

    output_buffers      1 32k;
    postpone_output    1460;

    #删除掉 kEDH 算法
    ssl_ciphers ALL:!kEDH!ADH:RC4+RSA:+HIGH:+EXP;

    server {
        listen           80 default_server;
        access_log      off;
        server_name      _;      #default
        return 444;
    }


    upstream jee_backend {
        hash $http_x_forwarded_for;
        server localhost:8001 weight=100 max_fails=3 fail_timeout=30s;
        server localhost:8002 weight=100 max_fails=3 fail_timeout=30s;
        server localhost:8003 weight=100 max_fails=3 fail_timeout=30s;
        server localhost:8004 weight=100 max_fails=3 fail_timeout=30s;
        #ip_hash;
        keepalive 1024;
    }

    upstream php_backend {
        #hash $http_x_forwarded_for;
    server unix:/dev/shm/php-5.7-dev_fpm-dev1.socket weight=100 max_fails=3 fail_timeout=30s;
    server unix:/dev/shm/php-5.7-dev_fpm-dev2.socket weight=100 max_fails=3 fail_timeout=30s;
    server unix:/dev/shm/php-5.7-dev_fpm-dev3.socket weight=100 max_fails=3 fail_timeout=30s;
    server unix:/dev/shm/php-5.7-dev_fpm-dev4.socket weight=100 max_fails=3 fail_timeout=30s;
        #ip_hash;
        keepalive 80;
    }


#     server {
```

```
#           listen          6428;
#           access_log    off;
#
#           location / {
#                 status 200;
#                 header Content-Type = text/html;
#                 body ~ "Welcome to nginx!";
#           }
#     }

      include site/*.conf;
}
```

5.2针对 php-cgi 的配置

```
server {
      listen          80;
      server_name nubia.com www.nubia.com;
      access_log    off;
      root /nowhere;
      rewrite ^ https://nubia.com$request_uri permanent;
}

server {
      listen 443;
      server_name        nubia.com;
      index                  index.php;
      root                    /data/wwwroot/virtual_server;
      #           include                rewrite_nubia.conf;

      rewrite ^(.*)/([a-z0-9_]*)\.html$ $1/?act=$2&%1;
      rewrite ^(.*)/([a-z0-9_]*)/([a-z0-9_\-]+)\.html$ $1/?act=$2!$3&%1;

      #access_log        /data/logs/nginx-1.7.9/virtual_server_access.log;
      access_log off;

      location ~ ^/status/ {
            stub_status on;
```

```
    }

    ssl on;
    ssl_certificate ssl_key/sso.nubia.com.crt;
    #openssl rsa -in sso.nubia.com.key -out sso.nubia.com.unsecure.key
    ssl_certificate_key ssl_key/sso.nubia.com.unsecure.key;
    ssl_client_certificate ssl_key/ca.pem;

    #ssl_protocols  SSLv3 TLSv1;
    ##ssl_ciphers AES:HIGH:!ADH:!MD5;
    #ssl_ciphers      ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP;
    #ssl_prefer_server_ciphers    on;
    ##self_signed_cert true;

    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
    ssl_ciphers
ECDHE-RSA-AES256-SHA384:AES256-SHA256:RC4:HIGH:!MD5:!aNULL:!eNULL:!NULL:!DH:!EDH:!AESGCM;
    #ssl_ciphers
ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP:!aNULL:!MD5:!EXP;
    #ssl_ciphers
ECDH+AESGCM:DH+AESGCM:ECDH+AES256:DH+AES256:ECDH+AES128:DH+AES:RSA+AES:!ADH:!AECDH:!MD5:!DSS:!3DES;
    ssl_prefer_server_ciphers on;
    ssl_session_cache shared:SSL:10m;
    ssl_session_timeout 10m;

    access_log   off;
    error_log    off;

   location ~ \.php$ {
      fastcgi_pass   php_backend;
      fastcgi_keep_conn on;
      #fastcgi_pass   unix:/tmp/php-cgi.socket;
      fastcgi_param SCRIPT_FILENAME /data/wwwroot/virtual_server$fastcgi_script_name;
      chunked_transfer_encoding        off;
      include          fastcgi_params;
    }

    location ~ .*\.(gif|jpg|jpeg|png|bmp|swf)$ {
      expires        30d;
    }
    location ~ .*\.(js|css)?$ {
      expires        1h;
    }

    error_page     500 502 503 504    /50x.html;
    location = /50x.html {
        root     html;
    }

    error_page     404    /404.html;
    location = /404.html {
        root     html;
    }
}
```

5.3 针对 php-proxy 的配置

```
# GITLAB
# Maintainer: @yin8086
# App Version: 4.1

# Modified from nginx http version
# Modified from http://blog.phusion.nl/2012/04/21/tutorial-setting-up-gitlab-on-debian-6/

# You need from run openssl to generate the ssl certificate.
# $ sudo openssl req -new -x509 -nodes -days 3560 -out gitlab.crt -keyout gitlab.key
# $ sudo chmod o-r gitlab.key

upstream gitlab {
  server unix:/home/git/gitlab/tmp/sockets/gitlab.socket;
}

# This is a normal HTTP host which redirects all traffic to the HTTPS host.
server {
    listen          80;
    server_name git.nubia.org;
    root /nowhere;
    rewrite ^ https://git.nubia.org$request_uri permanent;
```

```
}

server {
      listen 443;
      server_name git.nubia.org;
      root /home/git/gitlab/public;

      ssl on;
      ssl_certificate ssl_key/git.nubia.org.crt;
      #openssl rsa -in git.nubia.org.key -out git.nubia.org.unsecure.key
      ssl_certificate_key ssl_key/git.nubia.org.unsecure.key;
      ssl_client_certificate ssl_key/ca.pem;
      #ssl_protocols   SSLv3 TLSv1 TLSv2;
      ssl_protocols   SSLv3 TLSv1;
      #ssl_ciphers AES:HIGH:!ADH:!MD5;
      ssl_ciphers        ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP;
      ssl_prefer_server_ciphers    on;
      #self_signed_cert true;

      # individual nginx logs for this gitlab vhost
      access_log   /data/logs/nginx-1.7.4/gitlab_access.log;
      error_log    /data/logs/nginx-1.7.4/gitlab_error.log;

      location / {
            # serve static files from defined root folder;.
            # @gitlab is a named location for the upstream fallback, see below
            try_files $uri $uri/index.html $uri.html @gitlab;
      }

      # if a file, which is not found in the root folder is requested,
      # then the proxy pass the request to the upsteam (gitlab unicorn)
      location @gitlab {
            proxy_read_timeout 300; # https://github.com/gitlabhq/gitlabhq/issues/694
            proxy_connect_timeout 300; # https://github.com/gitlabhq/gitlabhq/issues/694
            proxy_http_version 1.1;
            proxy_set_header Connection "";
            proxy_redirect        off;

            proxy_set_header    X-Forwarded-Proto https;
            proxy_set_header    X-Forwarded-Ssl on;
            proxy_set_header    Host                $http_host;
            proxy_set_header    X-Real-IP           $remote_addr;
            proxy_pass http://gitlab;
        }
}
```

5.4 针对 Tomcat 的配置

```
server {
#       listen                  80;
        server_name         open.nubia.cn;
#      access_log           /data/logs/nginx-1.7.4/open.nubia.cn.log access;
        access_log      off;

          #rewrite ^(.*)\;JSESSIONID=(.*)$    $1      break;

          location ~ ^/status/ {
                stub_status on;
                access_log    off;
        allow 10.161.151.167;
        allow 218.104.236.138;
        deny all;
        }

# include site/blacklist.txt;

#       if ($host = 'nubia.cn' ) {
#             rewrite ^/(.*)$ http://www.nubia.cn/$1 permanent;
#       }
        #if ($http_user_agent ~ "Chrome/31.0.1650.63 Safari/537.36") {
        #       return 444;
        #       }
#       if ($http_user_agent ~ "Gecko/20120216 Firefox/3.6.27") {    return 444; }

            #if                                        ($http_user_agent                                        ~*
"qihoobot|Baiduspider|Googlebot|Googlebot-Mobile|Googlebot-Image|Mediapartners-Google|Adsb
ot-Google|Feedfetcher-Google|Yahoo!     Slurp|Yahoo!     Slurp     China|YoudaoBot|Sosospider|Sogou
spider|Sogou web spider|MSNBot|ia_archiver|Tomato Bot") {
              #              return 444;
              #}

#         if ($http_user_agent ~ ApacheBench|WebBench|Java/|http_load|wget) {
```

```
#     return 444;
#         }

        if ( $http_user_agent ~ "^((.*MIDP.*)|(.*WAP.*)|(.*UP.Browser.*)|(.*Smartphone.*)|(.*Obigo.*)|(.*Mobile.*)|(.*AU.Browser.*)|(.*wxd.Mms.*)|(.*WxdB.Browser.*)|(.*CLDC.*)|(.*UP.Link.*)|(.*KM.Browser.*)|(.*UC.*)|(.*SEMC\-Browser.*)|(.*Mini.*)|(.*Symbian.*)|(.*Palm.*)|(.*Nokia.*)|(.*Panasonic.*)|(.*MOT\-.*)|(.*SonyEricsson.*)|(.*NEC\-.*)|(.*Alcatel.*)|(.*Ericsson.*)|(.*BENQ.*)|(.*BenQ.*)|(.*Amoisonic.*)|(.*Amoi\-.*)|(.*Capitel.*)|(.*PHILIPS.*)|(.*SAMSUNG.*)|(.*Lenovo.*)|(.*Mitsu.*)|(.*Motorola.*)|(.*SHARP.*)|(.*WAPPER.*)|(.*LG\-.*)|(.*LG/.*)|(.*EG900.*)|(.*CECT.*)|(.*Compal.*)|(.*kejian.*)|(.*Bird.*)|(.*BIRD.*)|(.*G900/V1.0.*)|(.*Arima.*)|(.*CTL.*)|(.*TDG.*)|(.*Daxian.*)|(.*DAXIAN.*)|(.*DBTEL.*)|(.*Eastcom.*)|(.*EASTCOM.*)|(.*PANTECH.*)|(.*Dopod.*)|(.*Haier.*)|(.*HAIER.*)|(.*KONKA.*)|(.*KEJIAN.*)|(.*LENOVO.*)|(.*Soutec.*)|(.*SOUTEC.*)|(.*SAGEM.*)|(.*SEC\-.*)|(.*SED\-.*)|(.*EMOL\-.*)|(.*INNO55.*)|(.*ZTE.*)|(.*iPhone.*)|(.*Android.*)|(.*Windows CE.*)|(Wget.*)|(Java.*)|(curl.*)|(Opera.*))$") { rewrite "^/$" /mobile_index.html   last;}



    location / {
        index   index.html openbuy.html index.jsp;
        root    /data/wwwroot/nubia_open;
        expires       30d;
    }

    location /api/uc {
        rewrite ^/api/uc$ /api/uc.action last;
    }


    location ~ \.(jsp|action|do|zte|json|info)$ {
    #location / {
        proxy_pass http://jee_backend;
        proxy_http_version 1.1;
        proxy_set_header Connection "";
        proxy_redirect   off;
        #proxy_set_header Accept-Encoding 'gzip';
        proxy_set_header Host                  $http_host;
        proxy_set_header X-Real-IP        $proxy_add_x_forwarded_for;
        #$http_x_forwarded_for;
        #$remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header     Cookie       $http_cookie;
        chunked_transfer_encoding              off;
    }
```

```
     location ~ ^/(WEB-INF)/ {
          deny all;
     }

     error_page 403 404 500 502 503 504   /error.html;

     location = /error.html {
        root     /data/wwwroot/nubia_open;
     }

}

    # for SLB
    server {
        listen          6428;
        access_log     off;

        location / {
            index   index.html;
            root /data/wwwroot/slb;
            expires         30d;
        }

        error_page     500 502 503 504   /50x.html;
        location = /50x.html {
            root     html;
        }
    }
}
```