

一、前言

NSCD (Name Service Cache Daemon)是一个系统缓存服务，用于缓存常见的名称服务信息，例如用户、组、主机名和服务。它可以提高系统的性能，减少对名称服务的频繁查询和网络流量，从而加快系统的响应速度。

NSCD在许多 Linux 发行版中默认安装，可以使用 `systemctl status nscd` 命令检查其运行状态。如果需要启动、停止或重新启动 NSCD，可以使用 `systemctl start nscd`、`systemctl stop nscd` 或 `systemctl restart nscd` 命令。

如果你在/etc/Hosts指定某个IP解析域名，发现实际请求过程中并不生效，那么这时候就要考虑系统内是否存在nscd进程。

开启nscd的hosts缓存服务后，先会优先查找nscd缓存表，如果缓存表里没有，`/etc/hosts` 也没有，再正常发起dns query。

二、安装

某些发行版nscd在系统内内置（GNB Lib C库），如没有内置或采用的是最小化安装的系统(minimal)，则可以直接从软件源安装。

发行版	安装命令
CentOS/RedHat	<code>yum install -y nscd</code>
Debian/Ubuntu	<code>apt-get install -y nscd</code>
Arch/Manjaro	<code>pacman -Sy glibc</code>
Gentoo	<code>USE="nscd" emerge --ask glibc</code>

其中Gentoo注意要加上nscd的USE标记，否则默认不会携带nscd包。

```
o 17:40:49 ~ -> emerge -pv sys-libs/glibc 1539 0.03 20.03.23
* Last emerge --sync was 64d 19h 53s ago.

These are the packages that would be merged, in order:

Calculating dependencies... done!
[ebuild R ] sys-libs/glibc-2.36-r5:2.2::gentoo USE="multiarch (multilib) ssp_stack-realign (static-libs) systemd -audit -caps (-cet) -compile-locale (-crypt) (-custom-cflags) -doc -gd -hash-sysv-compat -headers-only -multilib-bootstrap [-nscd] -profile (-selinux) -suid -systemtap -test (-vanilla)" 18,264 KiB
Total: 1 package (1 reinstall), Size of downloads: 18,264 KiB
#号表示默认不启用
o 17:40:55 ~ ->
```

三、nscd.conf参数含义说明

默认配置文件路径在 `/etc/nscd.conf`，使用grep、sed过滤掉空行和注释后，可以看到默认配置，这里将每一行的含义都分别注释出来：

```

$ grep -vE '^$|^#' /etc/nscd.conf|sed -nr 's/^\s*//gp
debug-level          0
paranoia             no
enable-cache        passwd          yes
positive-time-to-live  passwd          600
negative-time-to-live  passwd          20
suggested-size       passwd          211
check-files          passwd          yes
persistent           passwd          yes
shared               passwd          yes
max-db-size          passwd          33554432
auto-propagate       passwd          yes
enable-cache        group           yes
positive-time-to-live  group          3600
negative-time-to-live  group          60
suggested-size       group          211
check-files          group          yes
persistent           group          yes
shared               group          yes
max-db-size          group          33554432
auto-propagate       group          yes
enable-cache        hosts          yes
positive-time-to-live  hosts          3600
negative-time-to-live  hosts          20
suggested-size       hosts          211
check-files          hosts          yes
persistent           hosts          yes
shared               hosts          yes
max-db-size          hosts          33554432
enable-cache        services       yes
positive-time-to-live  services       28800
negative-time-to-live  services       20
suggested-size       services       211
check-files          services       yes
persistent           services       yes
shared               services       yes
max-db-size          services       33554432
enable-cache        netgroup       yes
positive-time-to-live  netgroup       28800
negative-time-to-live  netgroup       20
suggested-size       netgroup       211
check-files          netgroup       yes
persistent           netgroup       yes
shared               netgroup       yes
max-db-size          netgroup       33554432

```

这些参数说明使用 `man nscd.conf` 就能看到各个参数的详细解释，以下是一些常用参数的释义：

参数	说明
debug-level	debug等级，0表示不输出任何信息。
paranoia	偏执模式，启用偏执模式会导致nscd周期性地重新启动。
enable-cache	开启缓存。
positive-time-to-live	TTL缓存时间，默认3600s。
negative-time-to-live	如果Name Service找不到条目，则会将其添加到缓存中并标记为“不存在”。此选项设置从缓存中删除不存在的条目的时间。password和host缓存默认为20秒，group缓存默认为60秒。
suggested-size	设置用于存储缓存项的哈希的大小。由于这是一个散列，正常应该大于预期缓存的最大条目数，并且应该是质数。默认为211个条目。
check-files	定期检查/etc/passwd、/etc/group或/etc/hosts的修改时间（分别用于passwd、group和host缓存），如果文件自上次检查以来已更改，则使缓存无效。
persistent	保留缓存内容，以便在nscd重新启动时提供服务。设置偏执模式时很有用。
shared	用于服务nscd数据库的内存映射与客户端共享，这样客户端就可以直接在其中搜索，而不必每次执行查找时都通过套接字询问守护进程。
max-db-size	用于配置nscd缓存数据库的最大大小，单位为Byte。
auto-propagate	会自动将缓存数据的变化广播到其他进程，以确保缓存数据的一致性。
threads	等待请求的线程数量。
logfile	日志文件路径。
server-user	运行用户，即使用哪个用户来运行nscd服务。

四、nscd命令选项说明

1、-d, --debug

在当前终端下以debug模式运行，可以看到输出的调试日志。

```
02:01:56 # ~ nscd -d
Mon 20 Mar 2023 02:01:57 AM CST - 2913046: monitoring file /etc/passwd for database passwd
Mon 20 Mar 2023 02:01:57 AM CST - 2913046: monitoring file `etc/passwd` (1)
Mon 20 Mar 2023 02:01:57 AM CST - 2913046: monitoring directory `etc` (2)
Mon 20 Mar 2023 02:01:57 AM CST - 2913046: monitoring file /etc/group for database group
Mon 20 Mar 2023 02:01:57 AM CST - 2913046: monitoring file `etc/group` (3)
Mon 20 Mar 2023 02:01:57 AM CST - 2913046: monitoring directory `etc` (2)
Mon 20 Mar 2023 02:01:57 AM CST - 2913046: monitoring file /etc/hosts for database hosts
Mon 20 Mar 2023 02:01:57 AM CST - 2913046: monitoring file `etc/hosts` (4)
Mon 20 Mar 2023 02:01:57 AM CST - 2913046: monitoring directory `etc` (2)
Mon 20 Mar 2023 02:01:57 AM CST - 2913046: monitoring file /etc/resolv.conf for database hosts
Mon 20 Mar 2023 02:01:57 AM CST - 2913046: monitoring file `etc/resolv.conf` (5)
Mon 20 Mar 2023 02:01:57 AM CST - 2913046: monitoring directory `etc` (2)
Mon 20 Mar 2023 02:01:57 AM CST - 2913046: monitoring file /etc/services for database services
Mon 20 Mar 2023 02:01:57 AM CST - 2913046: monitoring file `etc/services` (6)
Mon 20 Mar 2023 02:01:57 AM CST - 2913046: monitoring directory `etc` (2)
Mon 20 Mar 2023 02:01:57 AM CST - 2913046: monitoring file /etc/netgroup for database netgroup
Mon 20 Mar 2023 02:01:57 AM CST - 2913046: disabled inotify-based monitoring for file `etc/netgroup`:
Mon 20 Mar 2023 02:01:57 AM CST - 2913046: stat failed for file `etc/netgroup'; will try again later:
Mon 20 Mar 2023 02:02:02 AM CST - 2913046: handle request: request received (Version = 2) from PID 2913
Mon 20 Mar 2023 02:02:02 AM CST - 2913046: GETFDHST
Mon 20 Mar 2023 02:02:02 AM CST - 2913046: provide access to FD 9, for hosts
Mon 20 Mar 2023 02:02:02 AM CST - 2913046: handle request: request received (Version = 2) from PID 2913
Mon 20 Mar 2023 02:02:02 AM CST - 2913046: GETHOSTBYADDR (183.3.226.35)
Mon 20 Mar 2023 02:02:02 AM CST - 2913046: Haven't found "183.3.226.35" in hosts cache!
```

将会在当前终端运行nscd进程，如果后台已经nscd服务，使用 `systemctl stop nscd` 暂停下即可。

2、-f, --config-file=NAME

指定nscd的配置文件路径：

```
nscd -f /etc/nscd.conf
```

```
02:21:04 # ~ nscd -f /etc/nscd.conf
02:22:28 # ~ ps -ef|grep nscd
root    2916907      1  0 02:22 ?        00:00:00 nscd -f /etc/nscd.conf
root    2916920 2906677  0 02:22 pts/0    00:00:00 grep --color=auto --exclude-dir=.bzr --exclude-dir=.hg --exclude-dir=.svn --exclude-dir=.idea --exclude-dir=.tox nscd
```

3、-F, --foreground

前台运行，将不会打印任何输出信息到屏幕上。

```
nscd -F
```

```
02:28:56 # ~ nscd -F
Debian dnscrypt&dnsmasq&nscd (1) x
02:29:55 # ~ ps -ef|grep nscd
root    2917728 2917654  0 02:28 pts/0    00:00:00 nscd -F
root    2917874 2917739  0 02:29 pts/1    00:00:00 grep --color=auto --exclude-dir=.bzr --exclude-dir=CVS --exclude-dir=.hg --exclude-dir=.svn --exclude-dir=.idea --exclude-dir=.tox nscd
```

当然如果你想后台运行也可以配合nohup &或screen来实现，但nscd默认就已经由systemd掌管了，没必要多此一举，这里只是展示各个参数的作用。

4、-g, --statistics

输出当前的配置统计，将输出生效后的各个参数选项配置情况和缓存命中率统计等。

```
nscd -g
```

我这里只配置了hosts缓存：

```
$ grep -vE '^$|^#' /etc/nscd.conf | sed -nr 's/^\s*//gp'
debug-level          4
paranoia             no
logfile /var/log/nscd.log
enable-cache         hosts          yes
positive-time-to-live hosts          3600
negative-time-to-live hosts          200
suggested-size       hosts          211
check-files          hosts          no
persistent           hosts          yes
shared               hosts          yes
max-db-size          hosts          33554432
```

可以看到输出信息参数值都能对上：

```
02:44:53 ~ nscd -g | sed -n '1,10p;52,72p'
nscd configuration:
    4 server debug level
  9m 5s server runtime
    4 current number of threads
   32 maximum number of threads
    0 number of times clients had to wait
   no paranoia mode enabled
 3600 restart internal
    5 reload count
    0 maximum chain length searched
    0 number of delays on rdlock
    0 number of delays on wrlock
    0 memory allocations failed
   yes check /etc/group for changes

hosts cache:
   yes cache is enabled
   yes cache is persistent
   yes cache is shared
   211 suggested size
 216064 total data pool size
   208 used data pool size
  3600 seconds time to live for positive entries
   200 seconds time to live for negative entries
    23 cache hits on positive entries
     0 cache hits on negative entries
    11 cache misses on positive entries
     4 cache misses on negative entries
   60% cache hit rate
```

5、-i, --invalidate=TABLE

使指定的缓存失效，后面可以接：passwd、group、hosts、services、netgroup。

比如使hosts缓存失效：

```
nscd -i hosts
```

```
03:02:02 ping qq.com
PING qq.com (192.168.1.102[局域网 对方和您在同一内部网]) 56(84) bytes of data.
From 192.168.1.72[局域网 对方和您在同一内部网] (192.168.1.72[局域网 对方和您在同一内部网]) icmp_seq=1 Destination unreachable: Host is unreachable
From 192.168.1.72[局域网 对方和您在同一内部网] (192.168.1.72[局域网 对方和您在同一内部网]) icmp_seq=2 Destination unreachable: Host is unreachable
From 192.168.1.72[局域网 对方和您在同一内部网] (192.168.1.72[局域网 对方和您在同一内部网]) icmp_seq=3 Destination unreachable: Host is unreachable
From 192.168.1.72[局域网 对方和您在同一内部网] (192.168.1.72[局域网 对方和您在同一内部网]) icmp_seq=4 Destination unreachable: Host is unreachable
From 192.168.1.72[局域网 对方和您在同一内部网] (192.168.1.72[局域网 对方和您在同一内部网]) icmp_seq=5 Destination unreachable: Host is unreachable
From 192.168.1.72[局域网 对方和您在同一内部网] (192.168.1.72[局域网 对方和您在同一内部网]) icmp_seq=6 Destination unreachable: Host is unreachable
^C
```

```
03:02:14 nscd -i hosts
03:02:21 _
```

```
Debian dnscrypt&dnsmasq&nscd (1) x
Mon 20 Mar 2023 03:02:10 AM CST - 2920559: GETHOSTBYADDR (192.168.1.72)
Mon 20 Mar 2023 03:02:10 AM CST - 2920559: Haven't found "192.168.1.72" in hosts cache!
Mon 20 Mar 2023 03:02:10 AM CST - 2920559: add new entry "192.168.1.72" of type GETHOSTBYADDR for hosts to cache
Mon 20 Mar 2023 03:02:19 AM CST - 2920559: pruning hosts cache; time 1679252539
Mon 20 Mar 2023 03:02:19 AM CST - 2920559: considering GETHOSTBYADDR entry "192.168.1.72", timeout 1679252730
Mon 20 Mar 2023 03:02:19 AM CST - 2920559: considering GETAI entry "qq.com", timeout 1679255501
Mon 20 Mar 2023 03:02:19 AM CST - 2920559: considering GETAI entry "baidu.com", timeout 1679252662
Mon 20 Mar 2023 03:02:21 AM CST - 2920559: handle_request: request received (Version = 2) from PID 2920635
Mon 20 Mar 2023 03:02:21 AM CST - 2920559: INVALIDATE (hosts)
Mon 20 Mar 2023 03:02:21 AM CST - 2920559: pruning hosts cache; time 9223372036854775807
Mon 20 Mar 2023 03:02:21 AM CST - 2920559: considering GETHOSTBYADDR entry "192.168.1.72", timeout 1679252730
Mon 20 Mar 2023 03:02:21 AM CST - 2920559: considering GETAI entry "qq.com", timeout 1679255501
Mon 20 Mar 2023 03:02:21 AM CST - 2920559: considering GETAI entry "baidu.com", timeout 1679252662
Mon 20 Mar 2023 03:02:21 AM CST - 2920559: remove GETHOSTBYADDR entry "192.168.1.72"
Mon 20 Mar 2023 03:02:21 AM CST - 2920559: remove GETAI entry "qq.com"
Mon 20 Mar 2023 03:02:21 AM CST - 2920559: remove GETAI entry "baidu.com"
```

6、-K, --shutdown

终止nscd进程，此选项并不会影响systemd管理的nscd服务进程。

```
nscd -K
```

```
02:57:21 nscd -f /etc/nscd.conf
02:59:20 ps -ef|grep nscd
root 2920350 2917739 0 02:59 pts/1 00:00:00 tail -f /var/log/nscd.log
root 2920359 1 0 02:59 ? 00:00:00 nscd -f /etc/nscd.conf
root 2920370 2917654 0 02:59 pts/0 00:00:00 grep --color=auto --exclude-dir=.bzr --exclude-dir=CVS --exclude-dir=.git --exclude-dir=.hg --exclude-dir=.svn --exclude-dir=.idea --exclude-dir=.tox nscd
02:59:24 nscd -K
02:59:42 ps -ef|grep nscd
root 2920350 2917739 0 02:59 pts/1 00:00:00 tail -f /var/log/nscd.log
root 2920403 2917654 0 02:59 pts/0 00:00:00 grep --color=auto --exclude-dir=.bzr --exclude-dir=CVS --exclude-dir=.git --exclude-dir=.hg --exclude-dir=.svn --exclude-dir=.idea --exclude-dir=.tox nscd
02:59:50
```

```
Debian dnscrypt&dnsmasq&nscd (1) x
02:59:06 tail -f /var/log/nscd.log
Mon 20 Mar 2023 02:59:24 AM CST - 2920359: handle_request: request received (Version = 2) from PID 2920369
Mon 20 Mar 2023 02:59:24 AM CST - 2920359: GETFDPW
Mon 20 Mar 2023 02:59:24 AM CST - 2920359: handle_request: request received (Version = 2) from PID 2920369
Mon 20 Mar 2023 02:59:24 AM CST - 2920359: GETPWBYUID (0)
Mon 20 Mar 2023 02:59:37 AM CST - 2920359: pruning hosts cache; time 1679252377
Mon 20 Mar 2023 02:59:37 AM CST - 2920359: considering GETAI entry "qq.com", timeout 1679255501
Mon 20 Mar 2023 02:59:37 AM CST - 2920359: considering GETAI entry "baidu.com", timeout 1679252662
Mon 20 Mar 2023 02:59:37 AM CST - 2920359: considering GETHOSTBYADDR entry "39.156.66.10", timeout 1679252284
Mon 20 Mar 2023 02:59:37 AM CST - 2920359: remove GETHOSTBYADDR entry "39.156.66.10"
Mon 20 Mar 2023 02:59:42 AM CST - 2920359: handle_request: request received (Version = 2) from PID 2920393
Mon 20 Mar 2023 02:59:42 AM CST - 2920359: SHUTDOWN
```

7、-t, --nthreads=NUMBER

指定运行的的线程数。

```
nscd -t <线程数量>
```

在/etc/nscd.conf配置文件则通过 `threads` 选项来指定。

五、用法实战

以缓存hosts为例，其它tables缓存参数没有区别，/etc/nscd.conf配置如下：

```
debug-level          4
paranoia             no
logfile /var/log/nscd.log
enable-cache        hosts          yes
positive-time-to-live hosts        3600
negative-time-to-live hosts        200
suggested-size      hosts        211
check-files         hosts        no
persistent          hosts        yes
shared              hosts        yes
max-db-size         hosts        33554432
```

重启nscd服务：

```
systemctl restart nscd.service
```

首先验证下缓存是否可以命中，第一次ping，nscd没有相关的缓存条目，于是加入到缓存中，第二次ping，从缓存拿到结果。

```

16:25:47 ~ → ping qq.com -c 1
PING qq.com (183.3.226.35[广东省深圳市 深圳市腾讯计算机系统有限公司电信节点]) 56(84) bytes of data.
64 bytes from 183.3.226.35[广东省深圳市 深圳市腾讯计算机系统有限公司电信节点] (183.3.226.35[广东省深圳市 深圳市腾讯计算机系统有限公司电信节点]): icmp_seq=1 ttl=56 time=2.18 ms

--- qq.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 2.175/2.175/2.175/0.000 ms

16:25:53 ~ → ping qq.com -c 1
PING qq.com (183.3.226.35[广东省深圳市 深圳市腾讯计算机系统有限公司电信节点]) 56(84) bytes of data.
64 bytes from 183.3.226.35[广东省深圳市 深圳市腾讯计算机系统有限公司电信节点] (183.3.226.35[广东省深圳市 深圳市腾讯计算机系统有限公司电信节点]): icmp_seq=1 ttl=56 time=2.21 ms

--- qq.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 2.214/2.214/2.214/0.000 ms

16:25:54 ~ →

Debian dnscrypt&dnsmasq&nscd (1) x
16:25:38 ~ → tail -f /var/log/nscd.log
2023年03月20日 星期一 16时25分53秒 - 2974885: 处理请求: 已从进程ID 2975111 收到请求 (版本 = 2)
2023年03月20日 星期一 16时25分53秒 - 2974885: GETFDHST
2023年03月20日 星期一 16时25分53秒 - 2974885: 提供访问到 FD 7, 用于 hosts
2023年03月20日 星期一 16时25分53秒 - 2974885: 处理请求: 已从进程ID 2975111 收到请求 (版本 = 2)
2023年03月20日 星期一 16时25分53秒 - 2974885: GETAI (qq.com) 第一次, nscd db里没有缓存qq.com的条目, 之后加入到缓存中
2023年03月20日 星期一 16时25分53秒 - 2974885: 无法在主机缓冲区中找到“qq.com”!
2023年03月20日 星期一 16时25分53秒 - 2974885: 加入新的项目“qq.com”的型态 GETAI 用于 hosts 到 cache (首先)
2023年03月20日 星期一 16时25分53秒 - 2974885: 处理请求: 已从进程ID 2975111 收到请求 (版本 = 2)
2023年03月20日 星期一 16时25分53秒 - 2974885: GETHOSTBYADDR (183.3.226.35)
2023年03月20日 星期一 16时25分53秒 - 2974885: 无法在主机缓冲区中找到“183.3.226.35”!
2023年03月20日 星期一 16时25分53秒 - 2974885: 加入新的项目“183.3.226.35”的型态 GETHOSTBYADDR 用于 hosts 到 cache (首先)
2023年03月20日 星期一 16时25分54秒 - 2974885: 处理请求: 已从进程ID 2975125 收到请求 (版本 = 2)
2023年03月20日 星期一 16时25分54秒 - 2974885: GETFDHST 第二次ping, nscd命中缓存
2023年03月20日 星期一 16时25分54秒 - 2974885: 提供访问到 FD 7, 用于 hosts

```

1、check-files

上述配置中，check-files为关闭状态，即当我们修改hosts文件时，也不会触发更新缓存。当域名缓存到nscd后，尝试修改/etc/hosts，并再次访问域名，发现并不会走/etc/hosts：

```

16:34:46 ~ → vim /etc/hosts
16:34:58 ~ → tail -n1 /etc/hosts
192.168.1.101 qq.com
16:35:05 ~ → ping qq.com -c 1
PING qq.com (183.3.226.35[广东省深圳市 深圳市腾讯计算机系统有限公司电信节点]) 56(84) bytes of data.
64 bytes from 183.3.226.35[广东省深圳市 深圳市腾讯计算机系统有限公司电信节点] (183.3.226.35[广东省深圳市 深圳市腾讯计算机系统有限公司电信节点]): icmp_seq=1 ttl=56 time=2.19 ms

--- qq.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 2.187/2.187/2.187/0.000 ms

16:35:11 ~ →

Debian dnscrypt&dnsmasq&nscd (1) x
16:34:35 ~ → tail -f /var/log/nscd.log
2023年03月20日 星期一 16时34分40秒 - 2975678: 处理请求: 已从进程ID 2975813 收到请求 (版本 = 2)
2023年03月20日 星期一 16时34分40秒 - 2975678: GETFDHST
2023年03月20日 星期一 16时34分40秒 - 2975678: 提供访问到 FD 7, 用于 hosts
2023年03月20日 星期一 16时34分46秒 - 2975678: 处理请求: 已从进程ID 2975845 收到请求 (版本 = 2)
2023年03月20日 星期一 16时34分46秒 - 2975678: GETFDHST
2023年03月20日 星期一 16时34分46秒 - 2975678: 提供访问到 FD 7, 用于 hosts
2023年03月20日 星期一 16时34分51秒 - 2975678: 处理请求: 已从进程ID 2975863 收到请求 (版本 = 2)
2023年03月20日 星期一 16时34分51秒 - 2975678: GETFDPW
2023年03月20日 星期一 16时34分51秒 - 2975678: 处理请求: 已从进程ID 2975863 收到请求 (版本 = 2)
2023年03月20日 星期一 16时34分51秒 - 2975678: GETPWBYUID (0)
2023年03月20日 星期一 16时35分01秒 - 2975678: 处理请求: 已从进程ID 2975878 收到请求 (版本 = 2)
2023年03月20日 星期一 16时35分01秒 - 2975678: GETFDGR
2023年03月20日 星期一 16时35分01秒 - 2975678: 处理请求: 已从进程ID 2975878 收到请求 (版本 = 2)
2023年03月20日 星期一 16时35分01秒 - 2975678: INITGROUPS (root)
2023年03月20日 星期一 16时35分06秒 - 2975678: pruning hosts 缓存: 时间 1679301306
2023年03月20日 星期一 16时35分06秒 - 2975678: 认为中 GETAI 项目“qq.com”, 逾时 1679301712
2023年03月20日 星期一 16时35分06秒 - 2975678: 认为中 GETHOSTBYADDR 项目“183.3.226.35”, 逾时 1679301470
2023年03月20日 星期一 16时35分11秒 - 2975678: 处理请求: 已从进程ID 2975892 收到请求 (版本 = 2)
2023年03月20日 星期一 16时35分11秒 - 2975678: GETFDHST
2023年03月20日 星期一 16时35分11秒 - 2975678: 提供访问到 FD 7, 用于 hosts

```

此时我们把配置文件的check-files选项启用，并重启nscd服务，且清除hosts缓存：

```

$ cat /etc/nscd.conf
debug-level          4
paranoia             no
logfile              /var/log/nscd.log
enable-cache         hosts          yes
positive-time-to-live hosts          3600
negative-time-to-live hosts          200
suggested-size       hosts          211
check-files          hosts          yes
persistent           hosts          yes
shared               hosts          yes

$ systemctl restart nscd
$ nscd -i hosts

```

再尝试看看是否会走/etc/hosts：

```

192.168.1.101 qq.com 3553 0.02 20.03.23
192.168.1.101 qq.com 3554 0.02 20.03.23
PING qq.com (192.168.1.101[局域网 对方和您在同一内部网]) 56(84) bytes of data.
64 bytes from qq.com (192.168.1.101[局域网 对方和您在同一内部网]): icmp_seq=1 ttl=64 time=0.539 ms
^C
192.168.1.100 qq.com 3558 0.03 20.03.23
192.168.1.100 qq.com 3559 0.03 20.03.23
192.168.1.100 qq.com 3560 0.03 20.03.23
PING qq.com (192.168.1.100[局域网 对方和您在同一内部网]) 56(84) bytes of data.
64 bytes from qq.com (192.168.1.100[局域网 对方和您在同一内部网]): icmp_seq=1 ttl=64 time=0.563 ms
--- qq.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.563/0.563/0.563/0.000 ms
192.168.1.100 qq.com 3561 0.03 20.03.23

```

```

Debian dnscrypt&dnsmasq&nscd (1) x
2023年03月20日 星期一 16时42分15秒 - 2976262: GETFDPW
2023年03月20日 星期一 16时42分15秒 - 2976262: 处理请求: 已从进程ID 2976403 收到请求 (版本 = 2)
2023年03月20日 星期一 16时42分15秒 - 2976262: GETPWBYUID (0)
2023年03月20日 星期一 16时42分19秒 - 2976262: 监视文件 '/etc/hosts' 原为 moved, 移除监视
2023年03月20日 星期一 16时42分19秒 - 2976262: 监视文件 '/etc/hosts' 原为 created, 添加监视
2023年03月20日 星期一 16时42分19秒 - 2976262: pruning hosts 缓存; 时间 9223372036854775807
2023年03月20日 星期一 16时42分19秒 - 2976262: 认为中 GETAI 项目 "qq.com", 超时 1679305324
2023年03月20日 星期一 16时42分19秒 - 2976262: 认为中 GETHOSTBYADDR 项目 "192.168.1.101", 超时 1679305324
2023年03月20日 星期一 16时42分19秒 - 2976262: remove GETAI entry "qq.com"
2023年03月20日 星期一 16时42分19秒 - 2976262: remove GETHOSTBYADDR entry "192.168.1.101"
2023年03月20日 星期一 16时42分19秒 - 2976262: 监视的文件 '/etc/hosts' 被写入
2023年03月20日 星期一 16时42分19秒 - 2976262: pruning hosts 缓存; 时间 9223372036854775807
2023年03月20日 星期一 16时42分35秒 - 2976262: 处理请求: 已从进程ID 2976427 收到请求 (版本 = 2)
2023年03月20日 星期一 16时42分35秒 - 2976262: GETFDHST
2023年03月20日 星期一 16时42分35秒 - 2976262: 提供访问到 FD 7, 用于 hosts
2023年03月20日 星期一 16时42分35秒 - 2976262: 处理请求: 已从进程ID 2976427 收到请求 (版本 = 2)
2023年03月20日 星期一 16时42分35秒 - 2976262: GETAI (qq.com)
2023年03月20日 星期一 16时42分35秒 - 2976262: 无法在主机缓冲区中找到"qq.com"!
2023年03月20日 星期一 16时42分35秒 - 2976262: 加入新的项目 "qq.com" 的型态 GETAI 用于 hosts 到 cache (首先)
2023年03月20日 星期一 16时42分35秒 - 2976262: 处理请求: 已从进程ID 2976427 收到请求 (版本 = 2)
2023年03月20日 星期一 16时42分35秒 - 2976262: GETHOSTBYADDR (192.168.1.100)
2023年03月20日 星期一 16时42分35秒 - 2976262: 无法在主机缓冲区中找到"192.168.1.100"!
2023年03月20日 星期一 16时42分35秒 - 2976262: 加入新的项目 "192.168.1.100" 的型态 GETHOSTBYADDR 用于 hosts 到 cache (首先)

```

结果符合预期，以qq.com为例，nscd进程会监视/etc/hosts，当触发修改后，则将老的缓存条目清除，下次访问时发现缓存qq.com域名，那么先走/etc/hosts，能找到地址，把地址加入缓存。

2、dig、nslookup处理逻辑

这两个命令并不会读取/etc/hosts，也不会走nscd缓存，而是直接向/etc/resolv.conf配置的NS发送dns query。但实际访问域名，会走nscd的缓存，缓存没有再查找/etc/hosts，/etc/hosts没有则正常发起dns查询。

```
16:50:30 ~ dig qq.com +short
123.151.137.18[天津市 深圳市腾讯计算机系统有限公司电信节点]
183.3.226.35[广东省深圳市 深圳市腾讯计算机系统有限公司电信节点]
61.129.7.47[上海市 深圳市腾讯计算机系统有限公司电信节点]
16:51:02 ~ nslookup qq.com
Server: 192.168.1.1
Address: 192.168.1.1#53

Non-authoritative answer:
Name: qq.com
Address: 61.129.7.47
Name: qq.com
Address: 123.151.137.18
Name: qq.com
Address: 183.3.226.35

16:51:05 ~ ping qq.com -c 1
PING qq.com (192.168.1.100[局域网 对方和您在同一内部网]) 56(84) bytes of data:
64 bytes from qq.com (192.168.1.100[局域网 对方和您在同一内部网]): icmp_seq=1 ttl=64 time=0.245 ms

--- qq.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.245/0.245/0.245/0.000 ms

16:51:08 ~ _

Debian dnscrypt&dnsmasq&nscd (1) x
16:50:58 ~ tail -f /var/log/nscd.log
2023年03月20日 星期一 16时51分08秒 - 2977023: 处理请求: 已从进程ID 2977298 收到请求 (版本 = 2)
2023年03月20日 星期一 16时51分08秒 - 2977023: GETFDHST
2023年03月20日 星期一 16时51分08秒 - 2977023: 提供访问到 FD 7, 用于 hosts
```

使用getent命令可以测试域名实际在系统中会先解析到哪个地址:

```
getent <database> <key>
```

```
17:01:55 ~ grep qq.com /etc/hosts
192.168.1.100 qq.com
17:03:01 ~ dig qq.com +short
123.151.137.18[天津市 深圳市腾讯计算机系统有限公司电信节点]
61.129.7.47[上海市 深圳市腾讯计算机系统有限公司电信节点]
183.3.226.35[广东省深圳市 深圳市腾讯计算机系统有限公司电信节点]
17:03:06 ~ getent hosts qq.com
192.168.1.100 qq.com
17:03:10 ~

Debian dnscrypt&dnsmasq&nscd (1) x
17:02:55 ~ tail -f /var/log/nscd.log
2023年03月20日 星期一 17时03分10秒 - 2977023: 处理请求: 已从进程ID 2978171 收到请求 (版本 = 2)
2023年03月20日 星期一 17时03分10秒 - 2977023: GETFDHST
2023年03月20日 星期一 17时03分10秒 - 2977023: 提供访问到 FD 7, 用于 hosts
```

所以,当你发现dig、nslookup解析没有问题,但ping、curl等实际访问域名都是另一个IP,且指定/etc/hosts的情况下也没效果,此时就要考虑nscd缓存,以及check-files的启用状态。

3、抓包验证

- 首先通过 `nscd -i hosts` 清除缓存条目,之后使用getent或者其它任何访问域名的命令,来触发一次DNS解析拿到A记录(比如下图中的三个IP);
- 此时再修改/etc/hosts,由于nscd.conf的check-files是启用状态,检测到/etc/hosts文件变化后,清除qq.com的缓存条目;
- 此时我们使用ping命令再次触发解析,nscd没有qq.com的缓存(前面已经清除qq.com的缓存条目),读取/etc/hosts文件,匹配到192.168.1.100内网地址,nscd将内网地址加入到缓存中;
- 此时执行第二次getent,匹配nscd缓存到qq.com 192.168.1.100,不会再发起A记录的dns query请求,从下图中可以看到整个过程。

```
17:15:58 ~ # nscd -i hosts
17:16:42 ~ # getent hosts qq.com
61.129.7.47 qq.com
123.151.137.18 qq.com
183.3.226.35 qq.com
17:16:48 ~ # vim /etc/hosts
17:17:01 ~ # grep qq.com /etc/hosts
192.168.1.100 qq.com
17:17:04 ~ # ping qq.com -c 1
PING qq.com (192.168.1.100[局域网 对方和您在同一内部网]) 56(84) bytes of data.
64 bytes from qq.com (192.168.1.100[局域网 对方和您在同一内部网]): icmp_seq=1 ttl=64 time=0.449 ms
--- qq.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.449/0.449/0.449/0.000 ms
17:17:12 ~ # getent hosts qq.com
192.168.1.100 qq.com
17:20:19 ~ # -

Debian dnscrypt&dnsmasq&nscd (1) x
17:16:25 ~ # tcpdump -i any -nn -s 0 udp port 53
tcpdump: data link type LINUX_SLL2
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 262144 bytes
17:16:48.333287 ens192 Out IP 192.168.1.72.38761 > 192.168.1.1.53: 21091+ AAAA? qq.com. (24)
17:16:48.334137 ens192 In IP 192.168.1.1.53 > 192.168.1.72.38761: 21091 0/0/0 (24)
17:16:48.335157 ens192 Out IP 192.168.1.72.57665 > 192.168.1.1.53: 18641+ A? qq.com. (24)
17:16:48.335457 ens192 In IP 192.168.1.1.53 > 192.168.1.72.57665: 18641 3/0/0 A 61.129.7.47, A 123.151.137.18, A 183.3.226.35 (72)
17:20:19.582449 ens192 Out IP 192.168.1.72.33716 > 192.168.1.1.53: 52336+ AAAA? qq.com. (24)
17:20:19.594820 ens192 In IP 192.168.1.1.53 > 192.168.1.72.33716: 52336 0/1/0 (74)
```

六、总结

NSCD的主要作用是缓存名称服务查询的结果，可以加快名称服务查询的速度，查询结果不必每次都从网络或本地DB(/var/cache/nscd)中获取，所以能一定程度上减少对网络和本地数据库的负载。也正因为此特性，如果缓存的查询结果过时或不准确，可能会导致应用程序出现错误或安全问题。此外，如果nscd服务出现故障，可能会导致名称服务查询失败或变慢。

同时，nscd可以缓存多种名字服务数据库，包括hosts、passwd、group、services、netgroup等。对于每种数据库，nscd都可以配置其缓存大小、生存时间等参数，以满足不同的需求。

总之，nscd是一个优秀的Name Service缓存守护进程，可以提高系统的名字服务性能，但是在实际应用中需要注意配置参数，避免出现缓存不一致等问题。