Tuning nginx

nginx Tuning

Nach Änderungen nginx neu laden (nginx -t && nginx -s reload):

```
# This number should be, at maximum, the number of CPU cores on your system.
worker_processes 16;

# Number of file descriptors used for Nginx. Should also be set in
/etc/security/limits.conf
worker_rlimit_nofile 200000;

# only log critical errors
error_log /var/log/nginx/error.log crit

# Determines how many clients will be served by each worker process.
# (Max clients = worker_connections * worker_processes)
# "Max clients" is also limited by the number of socket connections
available on the system (~64k)
worker_connections 4000;

# essential for linux, optmized to serve many clients with each thread
use epoll;

# Accept as many connections as possible, after nginx gets notification
about a new connection.
# May flood worker_connections, if that option is set too low.
multi_accept on;

# Caches information about frequently accessed files. Try to experiment with
those values.
open_file_cache max=200000 inactive=20s;
open_file_cache_valid 30s;
open_file_cache_min_uses 2;
open_file_cache_errors on;

# Disable access logs
access_log off;

# Sendfile copies data between one FD and other from within the kernel.
# More efficient than read() + write(), since the requires transferring data
to and from the user space.
sendfile on;

# Tcp_nopush causes nginx to attempt to send its HTTP response head in one
packet,
# instead of using partial frames. This is useful for prepending headers
```

```
before calling sendfile,
# or for throughput optimization.
tcp_nopush on;

# don't buffer data-sends (disable Nagle algorithm). Good for sending
frequent small bursts of data in real time.
tcp_nodelay on;

# Timeout for keep-alive connections. Server will close connections after
this time.
keepalive_timeout 30;

# Number of requests a client can make over the keep-alive connection. This
is set high for testing.
keepalive_requests 100000;

# allow the server to close the connection after a client stops responding.
Frees up socket-associated memory.
reset_timedout_connection on;

# send the client a "request timed out" if the body is not loaded by this
time. Default 60.
client_body_timeout 10;

# If the client stops reading data, free up the stale client connection
after this much time. Default 60.
send_timeout 2;

# Compression. Reduces the amount of data that needs to be transferred over
the network
gzip on;
gzip_min_length 10240;
gzip_proxied expired no-cache no-store private auth;
gzip_types text/plain text/css text/xml text/javascript application/x-
javascript application/xml;
gzip_disable "MSIE [1-6]\.";
```

TCP-Stack Tuning

Nach Änderungen „sysctl –system“ ausführen:

```
# Increase system IP port limits to allow for more connections
net.ipv4.ip_local_port_range = 2000 65000

net.ipv4.tcp_window_scaling = 1

# number of packets to keep in backlog before the kernel starts dropping
them
net.ipv4.tcp_max_syn_backlog = 3240000
```

```
# increase socket listen backlog
net.core.somaxconn = 3240000
net.ipv4.tcp_max_tw_buckets = 1440000

# Increase TCP buffer sizes
net.core.rmem_default = 8388608
net.core.rmem_max = 16777216
net.core.wmem_max = 16777216
net.ipv4.tcp_rmem = 4096 87380 16777216
net.ipv4.tcp_wmem = 4096 65536 16777216

net.core.default_qdisc=fq
net.ipv4.tcp_congestion_control=bbr
```