목차

네트워크 튜닝을 위한 커널 파라메터 3
Socket Layer/ Socket buffers 3
Socket queues 4
TCP parameters 5
UDP parameters 8
Interface Layer 8
Interface Layer [Hardware interrupts, Rx queues, Rx buffers] 9
Diagnostic Steps 11

네트워크 튜닝을 위한 커널 파라메터

원문 링크 : https://access.redhat.com/solutions/108513

Socket Layer/ Socket buffers

- **net.core.rmem_default**

The default receive socket buffer size in bytes. INTEGER - (The settable value range)-2147483647 - 2147483647 Default: 229376

- **net.core.rmem_max**

The maximum receive socket buffer size in bytes. INTEGER - (The settable value range)-2147483647 - 2147483647 Default: 229376

- **net.core.somaxconn**

Limit of socket listen() backlog, known in userspace as SOMAXCONN. See also tcp_max_syn_backlog for additional tuning for TCP sockets. Also see this solution on how to adjust it. INTEGER - (The settable value range)-2147483647 - 2147483647 Default: 128(defined as SOMAXCONN)

- **net.ipv4.tcp_adv_win_scale**

Count buffering overhead as bytes/2^tcp_adv_win_scale (if tcp_adv_win_scale > 0) or bytes-bytes/2^(-tcp_adv_win_scale), if it is ⇐ 0. INTEGER - (The settable value range)-2147483647 - 2147483647 Default: 2

- **net.ipv4.tcp_app_win**

Reserve max(window/2^tcp_app_win, mss) of window for application buffer. Value 0 is special, it means that nothing is reserved. INTEGER - (The settable value range)-2147483647 - 2147483647 Default: 31

- **net.ipv4.tcp_rmem**

This parameter has 3 INTEGERS: min, default, max - (The settable value range)-2147483647 - 2147483647 min: Minimal size of receive buffer used by TCP sockets.It is guaranteed to each TCP socket, even under moderate memory pressure. Default: 4096 default: The default size of the receive buffer for a TCP socket. Default: 87380 max: The maximum size of the receive buffer used by each TCP socket. Default: 4194394

- **net.ipv4.tcp_wmem**

This parameter has 3 INTEGERS: min, default, max - (The settable value range)-2147483647 - 2147483647 min: min: Amount of memory reserved for send buffers for TCP sockets. Each TCP socket has rights to use it due to fact of its birth. Default: 4096 default: The default size of the send buffer for a TCP socket. Default: 16384 max: The maximum size of the send buffer used by each TCP socket. Default: 4194394

- **net.core.wmem_default**

The default send socket buffer size in bytes. INTEGER - (The settable value range)-2147483647 - 2147483647 Default: 229376

- **net.core.wmem_max**

The maximum send socket buffer size in bytes. INTEGER - (The settable value range)-2147483647 - 2147483647 Default: 229376 The memory overhead for socket buffers is: buffer-size/2^tcp_adv_win_scale (tcp_adv_win_scale default is 2)

NOTE that use of setsockopt to set receive/send buffers will result in autotuning getting disabled.

Socket queues

- **net.ipv4.tcp_abort_on_overflow**

If listening service is too slow to accept new connections, reset them. It means that if overflow occurred due to a burst, connection will recover. Enable this option only if you are really sure that listening daemon cannot be tuned to accept connections faster. Enabling this option can harm clients of your server. BOOLEAN - TRUE(other than 0), FALSE(0) Default: 0(disabled)

- **net.ipv4.tcp_fin_timeout**

Time to hold socket in state FIN-WAIT-2, if it was closed by our side. Peer can be broken and never close its side, or even died unexpectedly. Default value is 60sec. Usual value used in 2.2 was 180 seconds, you may restore it, but remember that if your machine is even underloaded WEB server, you risk to overflow memory with kilotons of dead sockets, FIN-WAIT-2 sockets are less dangerous than FIN-WAIT-1, because they eat maximum 1.5K of memory, but they tend to live longer. Cf. tcp_max_orphans. INTEGER - (The settable value range)-2147483 - 2147483. Default: 60

- **net.ipv4.tcp_max_orphans**

Maximal number of TCP sockets not attached to any user file handle, held by system. If this number is exceeded orphaned connections are reset immediately and warning is printed. This limit exists only to prevent simple DoS attacks, you must not rely on this or lower the limit artificially, but rather increase it (probably, after increasing installed memory), if network conditions require more than default value, and tune network services to linger and kill such states more aggressively. Let me to remind again: each orphan eats up to ~64K of unswappable memory. INTEGER - (The settable value range)-2147483647 - 2147483647

- **net.ipv4.tcp_max_tw_buckets**

Maximal number of timewait sockets held by system simultaneously. If this number is exceeded time-wait socket is immediately destroyed and warning is printed. This limit exists only to prevent simple DoS attacks, you must not lower the limit artificially, but rather increase it (probably, after increasing installed memory), if network conditions require more than default value. INTEGER - (The settable value range)-2147483647 - 2147483647

- **net.ipv4.tcp_orphan_retries**

How may times to retry before killing TCP connection, closed by our side. Default value 7 corresponds to ~50sec-16min depending on RTO. If you machine is loaded WEB server, you should think about lowering this value, such sockets may consume significant resources. Cf. tcp_max_orphans. INTEGER - (The settable value range)-2147483647 - 2147483647 Default: 0

- **net.ipv4.tcp_tw_recycle**

Enable fast recycling TIME-WAIT sockets. It should not be changed without advice/request of technical experts. BOOLEAN - TRUE(other than 0), FALSE(0) Default: 0(disabled)

- **net.ipv4.tcp_tw_reuse**

Allow to reuse TIME_WAIT sockets for new connection when it is safe from protocol viewpoint. It should not be changed without advice/request of technical experts. BOOLEAN - TRUE(other than 0), FALSE(0) Default: 0(disabled)

TCP parameters

- **net.ipv4.tcp_abc**

Controls Appropriate Byte Count (ABC) defined in RFC3465. ABC is a way of increasing congestion window (cwnd) more slowly in response to partial acknowledgments. INTEGER - Possible values are: 0 : increase cwnd once per acknowledgment (no ABC) 1 : increase cwnd once per acknowledgment of full sized segment 2 : allow increase cwnd by two if acknowledgment is of two segments to compensate for delayed acknowledgments. Default: 0(off)

- **net.ipv4.tcp_syn_retries**

Number of times initial SYNs for an active TCP connection attempt will be retransmitted. Should not be higher than 255. INTEGER - (The settable value range)-2147483647 - 2147483647 Default: 5, corresponds to ~180 seconds.

- **net.ipv4.tcp_synack_retries**

Number of times SYNACKs for a passive TCP connection attempt will be retransmitted. Should not be higher than 255. INTEGER - (The settable value range)-2147483647 - 2147483647 Default: 5, which corresponds to ~180 seconds.

- **net.ipv4.tcp_keepalive_time**

How often TCP sends out keepalive messages when keepalive is enabled. INTEGER - (The settable value range)-2147483 - 2147483. Default: 7200, corresponds to 2hours.

- **net.ipv4.tcp_keepalive_probes**

How many keepalive probes TCP sends out, until it decides that the connection is broken. INTEGER - (The settable value range)-2147483647 - 2147483647 Default: 9

- **net.ipv4.tcp_keepalive_intvl**

How frequently the probes are send out. Multiplied by tcp_keepalive_probes it is time to kill not

responding connection, after probes started. INTEGER - (The settable value range)-2147483 - 2147483 Default: 75 sec(i.e. connection will be aborted after ~11 minutes of retries).

- **net.ipv4.tcp_retries1**

How many times to retry before deciding that something is wrong and it is necessary to report this suspicion to network layer. INTEGER - (The settable value range)-2147483647 - 255. Default: 3(Minimal RFC value), which corresponds to ~3sec-8min depending on RTO.

- **net.ipv4.tcp_retries2**

How may times to retry before killing alive TCP connection. RFC1122 says that the limit should be longer than 100 sec. It is too small number. INTEGER - (The settable value range)-2147483647 - 2147483647 Default: 15, which corresponds to ~13-30 min depending on RTO.

- **net.ipv4.tcp_max_syn_backlog**

Maximal number of remembered connection requests, which are still did not receive an acknowledgment from connecting client.If server suffers of overload, try to increase this number. INTEGER - (The settable value range)-2147483647 - 2147483647 Default: In RHEL5, 1024 for systems with more than 128Mb of memory, and 128 for low memory machines.

- **net.ipv4.tcp_window_scaling**

Enable window scaling as defined in RFC1323. BOOLEAN - TRUE(other than 0), FALSE(0) Default: 1(enabled)

- **net.ipv4.tcp_rfc1337**

Default: 0 BOOLEAN - TRUE(other than 0), FALSE(0) Default: 0(disabled)

- **net.ipv4.tcp_syncookies**

Enable tcp syncookies. Send out syncookies when the syn backlog queue of a socket overflows. The syncookies feature attempts to protect a socket from SYN flood attack. BOOLEAN - TRUE(other than 0), FALSE(0) Default: 0(disabled)

- **net.ipv4.tcp_timestamps**

Enable timestamps as defined in RFC1323. BOOLEAN - TRUE(other than 0), FALSE(0) Default: 1(enabled)

- **net.ipv4.tcp_sack**

Enable select acknowledgments (SACKS). BOOLEAN - TRUE(other than 0), FALSE(0) Default: 1(enabled)

- **net.ipv4.tcp_fack**

Enable FACK congestion avoidance and fast retransmission. The value is not used, if tcp_sack is not enabled. BOOLEAN - TRUE(other than 0), FALSE(0) Default: 1(enabled)

- **net.ipv4.tcp_dsack**

Allows TCP to send “duplicate” SACKs. BOOLEAN - TRUE(other than 0), FALSE(0) Default: 1(enabled)

- **net.ipv4.tcp_ecn**

Enable Explicit Congestion Notification in TCP. BOOLEAN - TRUE(other than 0), FALSE(0) Default: 0(disabled)

- **net.ipv4.tcp_reordering**

Maximal reordering of packets in a TCP stream. INTEGER - (The settable value range)-2147483647 - 2147483647 Default: 3

- **net.ipv4.tcp_low_latency**

If set, the TCP stack makes decisions that prefer lower latency as opposed to higher throughput. By default, this option is not set meaning that higher throughput is preferred. An example of an application where this default should be changed would be a Beowulf compute cluster. BOOLEAN - TRUE(other than 0), FALSE(0) Default: 0(disabled)

- **net.ipv4.tcp_tso_win_divisor**

This allows control over what percentage of the congestion window can be consumed by a single TSO frame. The setting of this parameter is a choice between burstiness and building larger TSO frames. INTEGER - (The settable value range)-2147483647 - 2147483647 Default: 3

- **net.ipv4.tcp_frto**

Enables F-RTO, an enhanced recovery algorithm for TCP retransmission timeouts. It is particularly beneficial in wireless environments where packet loss is typically due to random radio interference rather than intermediate router congestion. BOOLEAN - TRUE(other than 0), FALSE(0) Default: 0(disabled)

- **net.ipv4.tcp_congestion_control**

Set the congestion control algorithm to be used for new connections. The algorithm “reno” is always available, but additional choices may be available based on kernel configuration. STRING

- **net.ipv4.tcp_workaround_signed_windows**

If set, assume no receipt of a window scaling option means the remote TCP is broken and treats the window as a signed quantity. If unset, assume the remote TCP is not broken even if we do not receive a window scaling option from them. BOOLEAN - TRUE(other than 0), FALSE(0) Default: 0(disabled)

- **net.ipv4.tcp_slow_start_after_idle**

If set, provide RFC2861 behavior and time out the congestion window after an idle period. An idle period is defined at the current RTO. If unset, the congestion window will not be timed out after an idle period. BOOLEAN - TRUE(other than 0), FALSE(0) Default: 1(enabled)

UDP parameters

- **net.ipv4.udp_mem**

This parameter has 3 INTEGERS: min, pressue, max - (The settable value range)0 - 2147483647. Number of pages allowed for queueing by all UDP sockets. Default is calculated at boot time from amount of available memory: min: Below this number of pages UDP is not bothered about its memory appetite. When amount of memory allocated by UDP exceeds this number, UDP starts to moderate memory usage. pressure: This value was introduced to follow format of tcp_mem. max: Number of pages allowed for queueing by all UDP sockets.

- **net.ipv4.udp_rmem_min**

Minimal size of receive buffer used by UDP sockets in moderation. Each UDP socket is able to use the size for receiving data, even if total pages of UDP sockets exceed udp_mem pressure. The unit is byte. INTEGER - (The settable value range)0 - 2147483647 Default: 4096.

- **net.ipv4.udp_wmem_min**

Minimal size of send buffer used by UDP sockets in moderation. Each UDP socket is able to use the size for sending data, even if total pages of UDP sockets exceed udp_mem pressure. The unit is byte. INTEGER - (The settable value range)0 - 2147483647 Default: 4096.

Interface Layer

- **net.ipv4.ip_default_ttl**

Set the default time-to-live value of outgoing packets. INTEGER - (The settable value range)-2147483647 - 2147483647 Default: 64

- **net.ipv4.ip_forward**

Forward Packets between interfaces. This variable is special, its change resets all configuration parameters to their default state (RFC1122 for hosts, RFC1812 for routers) BOOLEAN - TRUE(other than 0), FALSE(0) Default: 0(disabled)

- **net.ipv4.ip_no_pmtu_disc**

Disable Path MTU Discovery. BOOLEAN - TRUE(other than 0), FALSE(0) Default: 0(disabled)

- **net.ipv4.inet_peer_gc_maxtime**

Mximum interval between garbage collection passes. This interval is in effect under low (or absent) memory pressure on the pool. Measured in jiffies. INTEGER - (The settable value range)-2147483 - 2147483 Default: 120

- **net.ipv4.inet_peer_gc_mintime**

Minimum interval between garbage collection passes. This interval is in effect under high memory

pressure on the pool. Measured in jiffies. INTEGER - (The settable value range)-2147483 - 2147483 Default: 10

- **net.ipv4.inet_peer_maxttl**

Maximum time-to-live of entries. Unused entries will expire after this period of time if there is no memory pressure on the pool (i.e. when the number of entries in the pool is very small). Measured in jiffies. INTEGER - (The settable value range)-2147483 - 2147483 Default: 600

- **net.ipv4.inet_peer_minttl**

Minimum time-to-live of entries. Should be enough to cover fragment time-to-live on the reassembling side. This minimum time-to-live is guaranteed if the pool size is less than inet_peer_threshold. Measured in jiffies. INTEGER - (The settable value range)-2147483 - 2147483 Default: 120

- **net.ipv4.inet_peer_threshold**

The approximate size of the storage. Starting from this threshold entries will be thrown aggressively. This threshold also determines entries' time-to-live and time intervals between garbage collection passes. More entries, less time-to-live, less GC interval. INTEGER - (The settable value range)-2147483647 - 2147483647 Default: 65664

- **net.ipv4.route.min_adv_mss**

The advertised MSS depends on the first hop route MTU, but will never be lower than this setting. INTEGER - (The settable value range)-2147483647 - 2147483647 Default: 256

- **net.ipv4.route.min_pmtu**

minimum discovered Path MTU. INTEGER - (The settable value range)-2147483647 - 2147483647 Default: 552

- **net.ipv4.route.mtu_expires**

Time, in seconds, that cached PMTU information is kept. INTEGER - (The settable value range)-2147483 - 2147483 Default: 600

Interface Layer [Hardware interrupts, Rx queues, Rx buffers]

- **net.ipv4.ipfrag_high_thresh**

Maximum memory used to reassemble IP fragments. When ipfrag_high_thresh bytes of memory is allocated for this purpose, the fragment handler will toss packets until ipfrag_low_thresh is reached. INTEGER - (The settable value range)-2147483647 - 2147483647 Default: 262144

- **net.ipv4.ipfrag_low_thresh**

See ipfrag_high_thresh INTEGER - (The settable value range)-2147483647 - 2147483647 Default: 196608

- **net.ipv4.ipfrag_max_dist**

ipfrag_max_dist is a non-negative integer value which defines the maximum “disorder” which is allowed among fragments which share a common IP source address. Note that reordering of packets is not unusual, but if a large number of fragments arrive from a source IP address while a particular fragment queue remains incomplete, it probably indicates that one or more fragments belonging to that queue have been lost. When ipfrag_max_dist is positive, an additional check is done on fragments before they are added to a reassembly queue - if ipfrag_max_dist (or more) fragments have arrived from a particular IP address between additions to any IP fragment queue using that source address, it's presumed that one or more fragments in the queue are lost. The existing fragment queue will be dropped, and a new one started. An ipfrag_max_dist value of zero disables this check. Using a very small value, e.g. 1 or 2, for ipfrag_max_dist can result in unnecessarily dropping fragment queues when normal reordering of packets occurs, which could lead to poor application performance. Using a very large value, e.g. 50000, increases the likelihood of incorrectly reassembling IP fragments that originate from different IP datagrams, which could result in data corruption. INTEGER - (The settable value range)0 - 2147483647 Default: 64

- **net.ipv4.ipfrag_time**

Time in seconds to keep an IP fragment in memory. INTEGER - (The settable value range)-2147483 - 2147483 Default: 30

- **net.ipv4.ipfrag_secret_interval**

Regeneration interval (in seconds) of the hash secret (or lifetime for the hash secret) for IP fragments. INTEGER - (The settable value range)-2147483 - 2147483 Default: 600

- **net.ipv4.conf.<DEV>.medium_id**

Integer value used to differentiate the devices by the medium they are attached to. Two devices can have different id values when the broadcast packets are received only on one of them. INTEGER - (The settable value range)-2147483647 - 2147483647

- **net.ipv4.conf.<DEV>.proxy_arp**

Do proxy arp. proxy_arp for the interface will be enabled if at least one of conf/{all,interface}/proxy_arp is set to TRUE, it will be disabled otherwise BOOLEAN - TRUE(other than 0), FALSE(0)

- **net.ipv4.conf.<DEV>.shared_media**

Send(router) or accept(host) RFC1620 shared media redirects. Overrides ip_secure_redirects. shared_media for the interface will be enabled if at least one of conf/{all,interface}/shared_media is set to TRUE, it will be disabled otherwise. BOOLEAN - TRUE(other than 0), FALSE(0) Default: 1(enabled)

- **net.ipv4.conf.<DEV>.arp_announce**

Define different restriction levels for announcing the local source IP address from IP packets in ARP requests sent on interface. The max value from conf/{all,interface}/arp_announce is used. Increasing the restriction level gives more chance for receiving answer from the resolved target while decreasing the level announces more valid sender's information. INTEGER - Possible values are: 0 : Use any local address, configured on any interface 1 : Try to avoid local addresses that are not in the target's

subnet for this interface. This mode is useful when target hosts reachable via this interface require the source IP address in ARP requests to be part of their logical network configured on the receiving interface. When we generate the request we will check all our subnets that include the target IP and will preserve the source address if it is from such subnet. If there is no such subnet we select source address according to the rules for level 2. 2 : Always use the best local address for this target. In this mode we ignore the source address in the IP packet and try to select local address that we prefer for talks with the target host. Such local address is selected by looking for primary IP addresses on all our subnets on the outgoing interface that include the target IP address. If no suitable local address is found we select the first local address we have on the outgoing interface or on all other interfaces, with the hope we will receive reply for our request and even sometimes no matter the source IP address we announce. Default: 0

- **net.ipv4.conf.<DEV>.accept_redirects**

Accept ICMP redirect messages. BOOLEAN - TRUE(other than 0), FALSE(0) Default: 1(enabled)

- **net.ipv4.conf.<DEV>.accept_source_route**

Accept packets with SRR option. BOOLEAN - TRUE(other than 0), FALSE(0) Default: 1(enabled)

- **net.ipv4.conf.<DEV>.rp_filter**

Enable Reverse Path Filter defined in RFC3704 INTEGER - Possible values are: 0 : No source validation. 1 : Strict mode as defined in RFC3704 Strict Reverse Path. Each incoming packet is tested against the FIB and if the interface is not the best reverse path the packet check will fail. By default failed packet are discarded. 2 : Loose mode as defined in RFC3704 Loose Reverse Path. Each incoming packet's source address is also tested against the FIB and if the source address is not reachable via any interface the packet check will fail. Default: 0(disabled)

- **net.ipv4.conf.<DEV>.send_redirects**

Send redirects, if router. BOOLEAN - TRUE(other than 0), FALSE(0) Default: 1(enabled)

Diagnostic Steps

Be extremely careful with the above tunables in a production scenario, as changing them live may have unintended consequences. The above tunables are meant to be adjusted on a case-by-case basis depending on the topology of the network, behavior of the applications, and purpose of the system. If you would like Red Hat's recommendation on how to best proceed in tuning a system for best network performance please do not hesitate to reach out to Red Hat Consulting or contact Red Hat Global Support Services if you have production-related issues you believe may be solved with the above tunables.