2382421 - Optimizing the Network Configuration on HANA- and OS-Level

Version	38	**Type**	SAP Note
Language	English	**Master Language**	English
Priority	Recommendations / Additional Info	**Category**	Consulting
Release Status	Released for Customer	**Released On**	29.06.2020
Component	HAN-DB (SAP HANA Database)		

Please find the original document at **https://launchpad.support.sap.com/#/notes/ 2382421**

Symptom

In SAP HANA scale out environments, the single nodes and services need to communicate with each other for various purposes.
In single-node systems, at least services need to communicate with each other.

The default Linux and HANA configuration might not necessarily be optimized for the needs of the internal HANA communication.
Hence, it is recommended to optimize related parameters to allow for a smooth, error-free inter-node and inter-service communication.

The settings below are recommended for all HANA systems, especially those with a high number of processes, like scale-out systems or MDC systems with multiple tenant databases.

Other Terms

network, TrexNet, Timeout, incomplete communication, aborted connection, protocol mismatch, unknown type found, connection broken, scaleout

Reason and Prerequisites

Reason:

Optimization of network settings is required.

Prerequisites:

You are using one of the following SAP HANA Revisions:

- SAP HANA 1:
  - Revisions of SPS10
  - Revisions of SPS11
  - Revisions of SPS12
- or higher

Solution

Especially for large scale-out systems or multi database container (MDC) installations with many tenant databases, setting the following values for the network related Linux kernel and SAP HANA parameters is recommended. Deviations from below recommendations can be beneficial in system specific cases.
As single node systems installed as single database container (SDC) usually have a lower communication

need than scale-out or MDC environments as there is less internode-communication, no substantial improvements are expected by setting below parameters.

In order to simplify the SAP HANA parameter settings we suggest to set the parameters in general, regardless of the system type.

Linux Kernel Parameters

- *net.core.somaxconn >= 4096*
  This parameter limits the size of the accept backlog of a listening socket.
  The Linux default of 128 is not sufficient so you need to set the parameter to at least 4096 in order that the HANA system can use higher values.
  There is an interdependency between this parameter and HANA configuration parameter *tcp_backlog.*
  If *net.core.somaxconn* is set to a lower value than *tcp_backlog, tcp_backlog* will be silently truncated to the value set for *net.core.somaxconn.* Therefore, you need to ensure that *net.core.somaxconn* is always set to a value equal to or greater than *tcp_backlog*.

- *net.ipv4.tcp_max_syn_backlog >= 8192*
  This is the size of the SYN backlog.
  To prevent the kernel from using SYN cookies in a situation where lots of connection requests are sent in a short timeframe and to prevent a corresponding warning about a potential SYN flooding attack in the system log, the size of the SYN backlog should be set to a reasonably high value.
- *net.ipv4.ip_local_port_range*
  As HANA uses a considerable number of connections for the internal communication, it makes sense to have as many client ports available as possible for this purpose.
  At the same time, you need to ensure that you explicitly exclude the ports used by processes and applications which bind to specific ports by adjusting parameter *net.ipv4.ip_local_reserved_ports* accordingly.
  If configured correctly, the SAP Host Agent takes care of adjusting this parameter and setting it manually is neither recommended nor required.
  For further details about the SAP Host Agent please see the section below.
  The SAP Host Agent typically increases the port range typically to 9000-65499. If your port range is significantly different, for example when your lower port range starts with port 40000, please check the SAP Host Agent section below.
- *net.ipv4.ip_local_reserved_ports*
  This parameter specifies the ports which are reserved for known applications. You especially also have to specify the standard ports that are used by the SAP HANA. To find out which standard ports are used by your SAP HANA please refer to SAP Note 2477204.
  Ports listed in this parameter will not be used by automatic port assignment, while explicit port allocation behavior is unchanged.
  If configured correctly, the SAP Host Agent takes care of the standard ports used by SAP HANA if the instance numbers are provided accordingly. Setting this configuration manually is neither recommended nor required.
  For further details about the SAP Host Agent please see the section below.
- *net.ipv4.tcp_slow_start_after_idle = 0*
  This setting disables the need to scale-up incrementally the TCP window size for TCP connections which were idle for some time. Using this parameter it is ensured that the maximum speed is used from beginning also for previously idle TCP connections.
- *net.ipv4.tcp_wmem* and *net.ipv4.tcp_rmem*
  These parameters specify the minimum, default and maximum size of the TCP send and receive buffer.
  They are mostly relevant for system replication scenarios with a latency higher than usual.

The maximum value should be equal to at least the bandwidth delay product of the relevant connection.
Both, *tcp_wmem* and *tcp_rmem*, are specified as three values separated by blanks: minimum, default and maximum buffer size.
Preconditions for these settings to take effect are:

- *net.core.wmem_max* and *net.core.rmem_max* must not be lower than the respective maximum value.
- TCP window scaling has been enabled by setting *net.ipv4.tcp_window_scaling = 1*
  Example:
  net.ipv4.tcp_wmem = 4096 16384 4194304
  In this example, the current maximum is 4 MB. Given a 10 GBit/s connection with a latency of 1 ms, the required maximum would be 10 GBit/s * 1ms = 1.25 Mbytes, therefore the current setting is fine.
  If you want to saturate a 1 Gbit/s connection with a latency of 100 ms, the required maximum is 1 GBit/s * 100 ms = 12.5 Mbyte, so in this case the setting should be adjusted to at least 12.5 MByte.
  The minimum and the default buffer size do not need to be adjusted.

- *net.core.wmem_max* and *net.core.rmem_max*
  These settings define the maximum socket send and receive buffer size.
  To ensure complete functionality it must be ensured that the *wmem_max* and *rmem_max* values are at least the same as the respective maximum value of the parameters *net.ipv4.tcp_wmem* and *net.ipv4.tcp_rmem*.
- *net.ipv4.tcp_window_scaling = 1*
  This setting enables the TCP window scaling.
  On most systems it already should be active. Moreover, it is a prerequisite for *net.ipv4.tcp_wmem* and *net.ipv4.tcp_rmem*.

In landscapes where **TCP timestamps** are **enabled** please carefully evaluate if the following OS settings can be applied:

- *net.ipv4.tcp_timestamps = 1*
  This setting adds the timestamp field to the TCP header.
  It should already be active on most systems and is a prerequisite for *net.ipv4.tcp_tw_reuse* and *net.ipv4.tcp_tw_recycle*.
  If you are running on Microsoft Azure depending on your scenario the setting of this OS parameter might not be supported. Please refer to the documentation provided by Microsoft for details. In this case please adjust the OS parameters as recommended by Microsoft. Moreover, please keep in mind that also the previous two parameters *net.ipv4.tcp_tw_reuse* and *net.ipv4.tcp_tw_recycle* must not be enabled in such a case.

- *net.ipv4.tcp_tw_recycle = 1*
  This setting reduces the time a connection spends in the TIME_WAIT state.
  One precondition for it to take effect is that TCP timestamps are enabled, i.e. *net.ipv4.tcp_timestamps = 1*, which is the default on most modern systems.
  Please note that this setting must not be applied if the HANA node has to communicate with hosts behind a NAT firewall. Moreover, it must not be applied if not all hosts that use a TCP connection to communicate with the HANA node have TCP timestamps enabled. Otherwise you might encounter TCP connection issues after applying this configuration parameter.
  In case you are running ABAP Application Server instances on Windows, please refer to SAP Note 2789262 for further details on possible connection issues.
  As of Linux kernel version 4.12, this configuration parameter is removed without substitution. You can therefore not set it anymore for OS versions running on kernel versions >= 4.12.

This applies as of SUSE Linux Enterprise Server (SLES) 12 SP4, SLES 15 GA and RHEL 8.0.

- *net.ipv4.tcp_tw_reuse = 1*
  This setting allows HANA to reuse a client port immediately after the connection has been closed, even though the connection is still in TIME_WAIT state. A precondition for it to take effect is that TCP timestamps are enabled, i.e. *net.ipv4.tcp_timestamps = 1*, which is the default on most modern systems.
  Please note that this setting must not be applied if the HANA node needs to communicate with hosts behind a NAT firewall. Moreover, it must not be applied if not all hosts that use a TCP connection to communicate with the HANA node have TCP timestamps enabled. Otherwise you might encounter TCP connection issues after applying this configuration parameter.
  As of SLES 15 SP2 and RHEL 8.1 the new option "2" is available which is also the default one. In single node systems this new default value is sufficient. If you are therefore running on a high enough OS version you should only consider setting net.ipv4.tcp_tw_reuse = 1 in case you are running SAP HANA in a scale-out setup.

On **SAP HANA 1 Revisions <= 122.14** and on **all SAP HANA 2 Revisions of SPS00** you additionally need to set the following parameter:

- *net.ipv4.tcp_syn_retries = 8*
  The default value for this parameter is 5, which translates to a timeout of about 24 seconds.
  If the system is under load, a timeout of 24 seconds can be too short and lead to avoidable errors.
  It also prevents processes to set a longer timeout. The recommended value is 8, which translates into a timeout of 190 seconds.

Remarks:

- To check the current values for the parameters you can use:
  - single parameter check
    sysctl -a | grep <parameter_name>
  - complete check:
    sysctl -a| grep -E
    "net.core.somaxconn|net.ipv4.tcp_max_syn_backlog|net.ipv4.ip_local_port_range|net.ipv4.ip_local_reser
- For a persistent change, edit */etc/sysctl.conf*. If a required parameter is not yet listed in the file, you can append it.
- For a temporary change to take effect immediately until the next reboot, you can submit the corresponsing *sysctl* commands:
  sysctl -w <parameter_name>=<parameter_value>

SAP Host Agent Configuration

As described in SAP Note 401162 the SAP Host Agent can automatically optimize the following Linux Kernel Parameters:

- net.ipv4.ip_local_port_range
- net.ipv4.ip_local_reserved_ports

To correctly configure the SAP Host Agent please first make sure that the /etc/sysctl.conf configuration does not contain any of these two parameters.
Afterwards, you should configure the SAP Host Agent profile parameters as described in SAP Note 401162. You especially have to maintain an instance list. Additionally, in the SAP Host Agent configuration you can explicitly exclude ports that are used by other applications on the same host or you can explicitly specify the

overall port range. Changes to the latter one normally are not required.

Remarks:

- If you are using SAP HANA System Replication and SAP Host Agent version >= 7.21 PL39 please don't forget to specify the product HANAREP in your SAP Host Agent configuration.
- In case you are using SAP HANA System Replication and a SAP Host Agent version < 7.21 PL39 you do not have to specify the product HANAREP. Depending on the installed mode the following additional configuration changes have to be applied:
  - If you have installed your database in single database container (SDC) mode you should specify the instance numbers INSTANCE_NUMBER and (INSTANCE_NUMBER+1) in the SAP Host Agent configuration to ensure that all needed ports are reserved for SAP HANA System Replication.
  - When your database is running in multi database container (MDC) mode you only have to define the instance number INSTANCE_NUMBER in the SAP Host Agent configuration. Moreover, you should reserve the following additional ports in your hostagent configuration: reserved_port/additional = 4(INSTANCE_NUMBER)01-4(INSTANCE_NUMBER)99
- After changing the SAP Host Agent configuration please follow the procedure described in SAP Note 401162 such that the changed configuration takes effect.

SAP HANA Parameters

The SAP HANA network parameters depend on the exact database revision. In newer Revisions most of the parameters are not needed anymore since SAP HANA internally optimizes them.
Please be informed, that after setting SAP HANA network parameters, you need to **restart** the SAP HANA system for the changes to take effect.

On **all SAP HANA Revisions** you need to set the *tcp_backlog* parameter in your global.ini:

- *global.ini -> [communication] -> tcp_backlog >= 2048*
  If many requests are sent in a very short timeframe the accept backlog runs full. As all incoming requests are handled centrally it is possible that the default value of only 128 causes connection attempts to run into a timeout. In that case you will see *connection broken* errors on the client side. The upper limit for this parameter is 8192.
  Consider the interdependency between *tcp_backlog* and linux configuration parameter *net.core.somaxconn*. If *tcp_backlog* is set to a higher value than *net.core.somaxconn,* it is silently truncated to the value of *net.core.somaxconn*. Therefore, make sure to check *net.core.somaxconn* when changing *tcp_backlog* and adjust it accordingly if required.
  To check whether the setting has been applied correctly, you can switch the tracelevel of the trace component *stream* to *debug* before you start your instance. Then, look for the string *Requested backlog* in the trace.
  As this will generate a lot of trace output, do not forget to switch back to the default value after the startup phase. This parameter is only applied when it is added to the global.ini file - adding it to the service specific ini files will not have an effect.
  This parameter must not be confused with the backlog size available in the SQL connection layer which is traced during every HANA startup, e.g. "*start the SQL listening port: 31015 with backlog size 128*". Adjusting the backklog size in the SQL connection layer is not required.

On **SAP HANA 1 Revisions <= 122.14** and on **all SAP HANA 2 Revisions of SPS00** you additionally need to set the following parameters:

- *<service>.ini -> [communication] -> maxchannels*
  The *maxchannels* parameter specifies how many incoming connections are kept open per SAP HANA service.

The maximum number of outgoing connections to other worker nodes to be kept open per SAP HANA service is implicitly derived from this setting using the following formula:
max_open_outgoing_per_service = 0.9 * *maxchannels* / #worker_nodes_without_standby
Neither the *maxchannels* setting nor the derived max_open_outgoing_per_service are hard limits - HANA will allow more connections to be created, but connections exceeding the configured limit will be closed after they have been processed instead of being kept open. In situations where the configured limits are constantly exceeded, e.g. by an enormous amount of requests, this will cause a lot of connections to be opened and closed again which will add to the size of the accept backlog and will in turn lead to increased latency. For this reason, it is important to optimize this setting for the respective SAP HANA service in order to avoid such situations.
The optimal value depends on the number of worker nodes and on the type and amount of load the respective SAP HANA service is put under. As each outgoing connection to another worker node consumes a port on operating system level, you need to make sure that the sum of max_open_outgoing_per_service values for all SAP HANA services must not exceed the number of usable ports on any worker host. You can calculate the number of available ports by calculating the difference between the number of ports defined by the linux kernel parameter net.ipv4.ip_local_port_range minus the number of ports defined by linux kernel parameter net.ipv4.ip_local_reserved_ports.
In most scenarios, the main workload goes to the indexserver. Therefore, a reasonable configuration to start with for a single-node system or a scale-out system with a small number of nodes could e.g. be:

indexserver.ini	[communication]	maxchannels	20000 up to 40000
nameserver.ini	[communication]	maxchannels	4000

In case you are using the xsengine frequently:

xsengine.ini	[communication]	maxchannels	4000 up to 20000

In case you are using a scriptserver:

scriptserver.ini	[communication]	maxchannels	4000

Especially in scale-out systems with a high number of nodes, it is beneficial to further optimize the *maxchannels* parameter per SAP HANA service.
EXAMPLE

System Architecture	Scale-Out system with 16 nodes + 1 standby node
System Usage	Mainly used as database system for a SAP system No additional use cases which create load for the xsengine No scriptserver usage
Linux Kernel Parameters	net.ipv4.ip_local_port_range is set to “1024 64999”, i.e. 63975+1 ports are in the defined range net.ipv4.ip_local_reserved_ports is set in a way that it covers a number of 200 ports
Maximum Possible Sum Value For maxchannels Of All SAP HANA Services	According to the above explanations, you can calculate the maximum possible sum over all SAP HANA services for maxchannels as: (((net.ipv4.ip_local_port_range_high_value - net.ipv4.ip_local_port_range_low_value + 1) - number_of_ports_covered_by_net.ipv4.ip_local_reser ved_ports) / 0.9) * #worker_nodes_without_standby i.e. MaxPossibleMaxchannels = (((64999 - 1024 + 1) - 200) / 0.9) * 16 = 1133795 This example is supposed to demonstrate how to calculate the absolute maximum possible value while it is not recommended at all to set the parameter to the absolute maximum value. If you still do, you will seize all available ports for connections to be kept open and will thereby lose the possibility of having an extra buffer of auto-close connections in case of situations with extreme load that is not expected. Even in systems with very high inter-node communication workload we have not seen any cases where it was

	reasonable or required to set maxchannels in a way that the sum for all services exceeds: max(40000, (MaxPossibleMaxchannels * 0.1)) Reasonable sum for maxchannels for all SAP HANA services in our example e.g. = max(40000, 1133795*0.1) = 113379 In this example there is no significant load on SAP HANA services other than the indexserver. For this reason you might want to mainly adjust the indexserver, make a minor increase for the nameserver and might not want to further increase the maxchannels parameter of other services like the scriptserver and the xsengine.

- *<service>.ini -> [communication] -> maxendpoints*

  The *maxendpoints* parameter specifies how many data structures to preallocate for storing endpoint information.
  It is recommended that the *maxendpoints* value corresponds to *maxchannels* value of the corresponding SAP HANA service, but no functional impairment results if it does not.

On **SAP HANA 1 Revisions <= 122.06** and on **SAP HANA 2 Revisions <= 001.00** you additionally need to set the following parameters:

- *indexserver.ini -> [communication] -> signal_send_timeout* and *signal_recv_timeout*

indexserver.ini	[communication]	signal_send_timeout	60000
indexserver.ini	[communication]	signal_recv_timeout	60000

  Each SAP HANA service creates a control connection. This control connection has a default timeout of 5 seconds, which can lead to timeout errors under high load conditions.
  To prevent these, it is advisable to increase the timeout to 60 seconds.

On **SAP HANA 1 Revisions <= 122.01** you additionally need to set the following parameter:

- *indexserver.ini -> [communication] -> handles*

indexserver.ini	[communication]	handles	40000

  The handles parameter specifies the size of the poll set.
  It must not be set to a value lower than the maximum number of idle channels, which corresponds to *maxchannels* of the indexserver.ini.

Remarks:

- After upgrading SAP HANA please remove the SAP HANA parameters which are not needed anymore on the new SAP HANA Revision
- Parameter *AcceptQueueLen* is obsolete as of SAP HANA 1 SPS10

Software Components

Software Component	Release
HDB	1.00 - 1.00
HDB	2.00 - 2.00

This document refers to

SAP Note/KBA	Title
2477204	FAQ: SAP HANA Services and Ports
401162	Linux: Avoiding TCP/IP port conflicts and start problems
2789262	Connection problems between Windows hosts and HANA database on Linux hosts
2380229	SAP HANA Platform 2.0 - Central Note
1523337	SAP HANA Database 1.00 - Central Note

This document is referenced by

SAP Note/KBA	Title
3000978	Replication stops due to network issues
2934640	HANA and Replication - Collecting Support Data for Replication / Network related Tickets
2932598	Resource registration failed to HANA Cockpit
2927380	After indexserver crash, Tenant database does not start due to network communication problems
2843393	"SQL code: -10709" occurred while accessing table "DDFTX"
2477204	FAQ: SAP HANA Services and Ports
1999993	How-To: Interpreting SAP HANA Mini Check Results
2772007	Error : system replication port could not be opened;port=40002
2771017	Replication does not complete due to nameserver error
2712064	SAP HANA System Replication Error port 4##### already in use
2748924	Enabling HANA System replication returns error after Hardware change on secondary site
2744521	Scheduling Crystal Report for Enterprise document based on JDBC connection is timed out
2727182	Solution Manager cannot connect to Diagnostics Agent on Hana database server - Solution Manager 7.2
2600030	Parameter Recommendations in SAP HANA Environments
2523310	ERROR: DBSL error 99 (db code -10807) System call 'recv' failed, Connection reset by peer

2509816	SAP HANA: false alert about restarted service
2455613	HANA: rc=15 connection broken errors
2399990	How-To: Analyzing ABAP Short Dumps in SAP HANA Environments
2684254	SAP HANA DB: Recommended OS settings for SLES 15 / SLES for SAP Applications 15
2652643	How to analyze '-10108' network errors on HANA landscape
2100566	Frequently Asked Questions for Lenovo/IBM saphana support script
2544949	Indexserver Crash During Table Redistribution Due to Network Communication Issues Causing Rollback of Subtransaction
2205917	SAP HANA DB: Recommended OS settings for SLES 12 / SLES for SAP Applications 12
401162	Linux: Avoiding TCP/IP port conflicts and start problems
2433474	Indexserver crashes in TrexNet::BufferedIO::flushBuffer
2427296	HANA TrexNet BadParam//channel not in list
2368186	Worker Nodes or Services Other Than the Master Indexserver do not Start up Properly
2299260	Internal Communication Failures in SAP HANA (Connection Broken)

