1849392 - HANA alerts related to locks, hanging sessions and long runners

Version 1 **Validity:** 20.04.2013 - active **Language** English

Symptom

Statisticserver alerts are triggered on HANA side:

HANA alert 39: Long-running statements
HANA alert 42: Long-running cursors
HANA alert 48: Long-running uncommitted write transactions
HANA alert 42: Long-running blocking situations

Environment

HANA

Resolution

These alerts are also reported into the alert trace of the indexserver (indexserver_alert_<host>.trc):

> long running uncommitted write transaction detected: CONNECTION ID = 200647, HOST = hanadb:30003, CLIENT HOST = hanaci, CLIENT PID
> TRANSACTION_ID = "902", TRANSACTION TOTAL EXECUTED TIME = 6321 sec
> The long running uncommitted write transaction possibly incurs performance degradation of HANA database. Please commit/rollback a transacti
> application or kill the connection by "ALTER SYSTEM DISCONNECT SESSION '202982' "

As a common quality, these alerts always contain information on the responsible client process. Before any action can be done (commit/rollback/kill), t
investigation has do be conducted on what the client process is doing.

Investigating client activity and application

In the first step, it has to be determined to which application the CLIENT PID on CLIENT host belongs to. Typically, this would be done using OS tools
like the 'ps' command in linux/unix or the taskmanager on windows systems. If the process is an SAP workprocess (disp+work.exe or dw), then further
be conducted using transaction SM50 on the relevant SAP application server. In case of an 'uncomitted write transaction', it might actually be the clien
blocking the commits. The SAP workprocess might be in state 'sleep' or waiting for locks/semaphore, thus preventing the execution of the commit.

Tools for analyzing SAP workprocess activity:

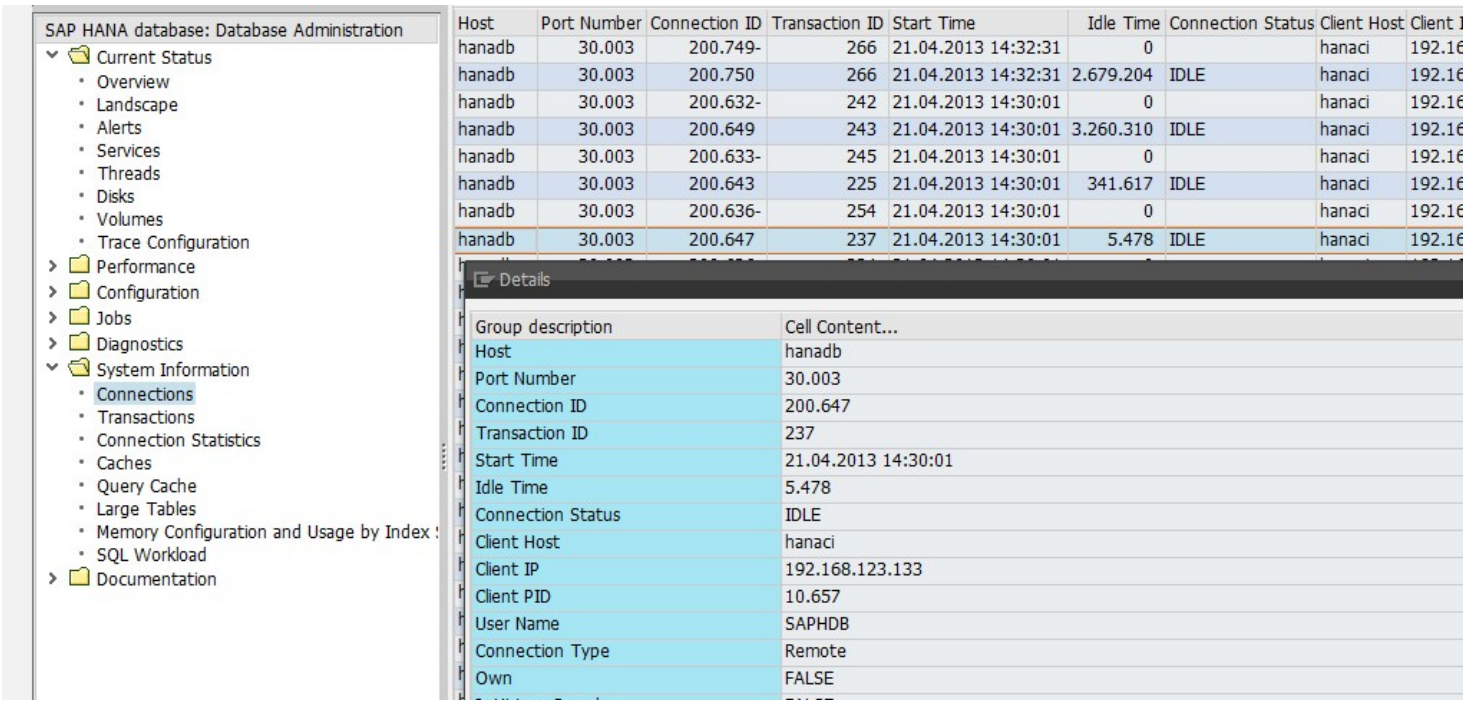
- Transactions SE30 or ST12
- In case of processes that appear to be hanging, call stacks can be generated into the developer traces (accessible using transaction ST11). Th
  commands to do so are 'kill -USR2 <pid>' (UNIX/LINUX) and 'sapntkill -USR2 <pid>' (Windows).

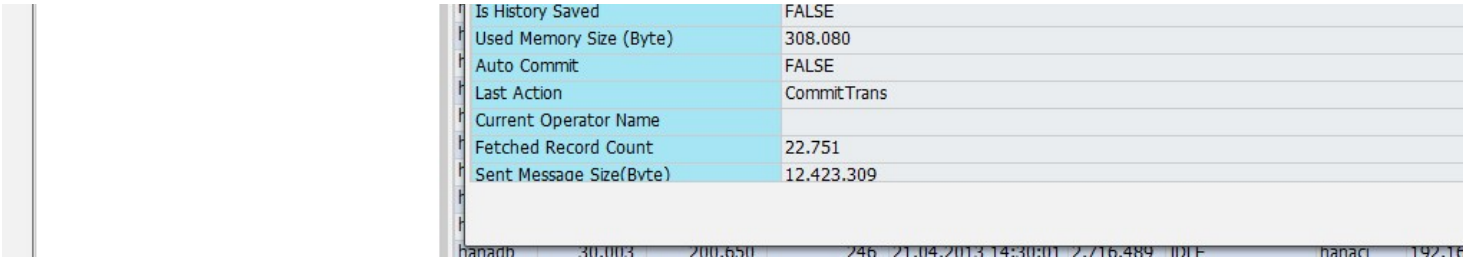
Analyzing database activity

Connections

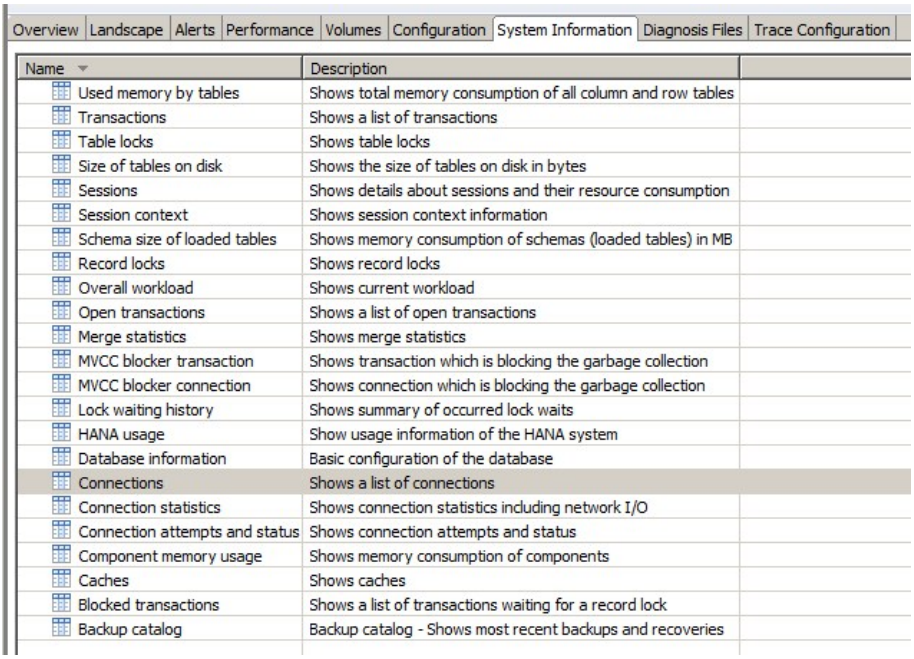
If it can be ruled out that the problem is located on client side, the activity on HANA side has to be analyzed next. As an entry point, the CONNECTION
used.

dbacockpit:





HANA Studio:



SQL Statement:

select * from m_connections where connection_id = <id>

Threads

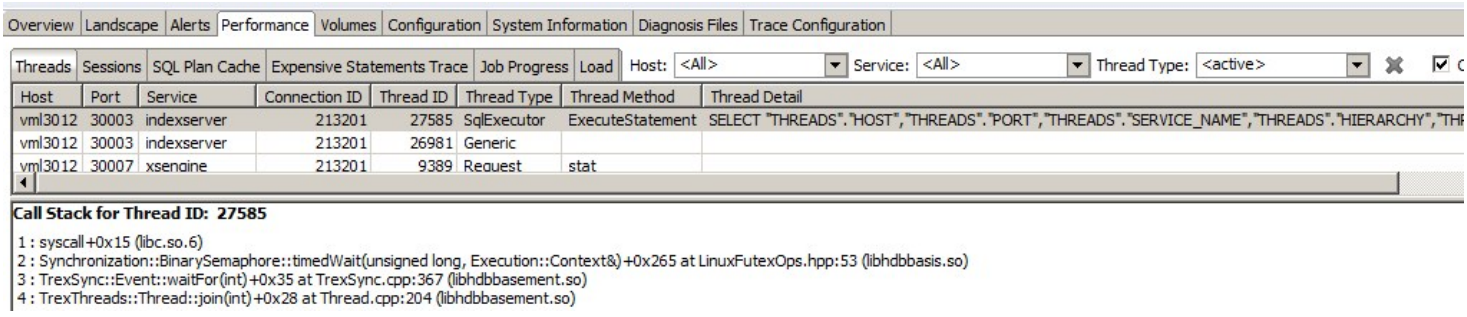
Since HANA is using multiple threads to process incoming request, an overview of all threads associated with a particular connection has to be creat

select * from m_service_threads where hierarchy like '<connection id>/%'

Tools to analyze HANA thread activity

Call stacks

The HANA Studio provides the functionality to display the call stacks of the HANA threads:



While this information is mainly for SAP internal purposes, the following function names being visible in the call stack could indicate known issues:

- TRexAPI::LateMatColumn: When observed during a seemingly hanging DSO migration, this function is an indication for the problem outlined in 1821785.

Header Data

Released On 14.05.2013 14:29:11
Release Status Released to Customer
Component BC-DB-HDB SAP HANA database
Priority Normal
Category Problem

Product

This document is not restricted to a product or product version

References

This document refers to:

CSS SAP Notes

1821785 DSO conversion is hanging due low number of latematcol pages