CT Cleanup Session Blocking User Session

Last updated by | Akio Hose | May 30, 2022 at 7:27 PM PDT

Author	Date
Akio Hose	May 31, 2022

Contents

Issue Background

The Change Tracking Cleanup session was blocking other user session. A T-SQL KILL command was attempted but failed due to the session being system initiated.

Possible root cause

Locks on the index pages associated with the internal Change Tracking table (a.k.a Side Table with the naming convention of 'change_tracking<#>') were held in a waiting state preventing the change tracking cleanup task (thread) to read index pages. However, root cause of what was holding the locks on the index pages was not identified. The executing query was attempting to acquire locks on index pages, and it was possible that there was a racing condition to acquire locks on the resources (i.e. data pages in the change tracking side table and syscommittab table) causing the query to fall under a 'self-deadlock' state (See RCA section of IcM 295241121 for details).

Identify the issue

Check if there is a block session (from sys.dm_os_waiting_tasks or sp_who2). In this incident (IcM 295241121) the customer noticed that one of there user session was blocked for 34 hours which they identified the blocking session from sp_who2.

Ensure to run the ASC Troubleshooting report

Possible root cause was based on analysis of the SQL process dump file. However, there were not enough information to identify the root cause. The thread associated with the change tracking cleanup showed it was reading index pages indicating disk IO and other IO related issue might have been the root cause.

Mitigation

Restart of the SQL instance is required to terminate the blocking session. Users can perform a manual failover which is effectively initiating a SQL restart.

Public Doc reference

Change Tracking Cleanup–Part 1

Change Tracking Cleanup – Part 2

About Change Tracking

Internal Reference

IcM 295241121

How good have you found this content?