

Creating large indexes

Last updated by | Vitor Tomaz | Feb 24, 2023 at 3:27 AM PST

Contents

- [Issue](#)
- [Investigation / Analysis](#)
- [Mitigation](#)

Creating Large Indexes

This TSG describes steps for troubleshooting failures when creating large indexes.

Issue

The customer tried to create a nonclustered index on a large table, but each attempt failed after running for a few hours. After the issue was resolved later, the index had a size of 117 GB.

This was the command attempted:

```
CREATE NONCLUSTERED INDEX [table_column1_column2] ON [dbo].[table]
(
    [column1] ASC,
    [column2] ASC
)
WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF,
```

Error messages reported to the application:

The service has encountered an error processing your request. Please try again, error code 9001
The service has encountered an error processing your request. Please try again, error code 3314

These were confirmed through `MonSQLSystemHealth` in Kusto:

2018-05-25 02:07:48.42 spid20s Error: 9001, Severity: 21, State: 33.
2018-05-25 02:07:48.42 spid20s The log for database '4ed9950e-0354-45ad-9b73-dbdd2067d126' is not available. Check the operating system error log for related error messages. Resolve any errors and restart the database.
2018-05-25 02:07:48.42 spid20s Error: 3314, Severity: 21, State: 1.
2018-05-25 02:07:48.42 spid20s During undoing of a logged operation in database '4ed9950e-0354-45ad-9b73-dbdd2067d126' (page (1:53749871) if any), an error occurred at log record ID (59995:27016:26). Typically, the specific failure is logged previously as an error in the operating system error log. Restore the database or file from a backup, or repair the database.

Investigation / Analysis

The first step is to check ASC for any events that might have happened at the time of the failure:

- planned or unplanned failovers, e.g. related to scaling, maintenance, or crash dump
- performance issues, e.g. CPU, I/O, memory, long-running transactions, blocking
- storage-related issues, e.g. I/O errors or running out of space
- any other errors that point to issues with the data files or the transaction log

You can get additional details by checking the SQL errorlog through `MonSQLSystemHealth` in Kusto:

```
let srv = "servername";
let db = "databasename";
let startTime = datetime(2022-09-30 09:30:00Z);
let endTime = datetime(2022-09-30 15:30:00Z);
let timeRange = ago(7d);
let AppNames = MonAnalyticsDBSnapshot
| where logical_server_name =~ srv
| where logical_database_name =~ db
| extend AppName = sql_instance_name
| distinct AppName;
MonSQLSystemHealth
//| where TIMESTAMP > timeRange
| where TIMESTAMP >= startTime
| where TIMESTAMP <= endTime
| where AppName in~ ( AppNames )
project TIMESTAMP, NodeName, AppName, error_id, message
```

This can help confirming the health of the database instance:

- Database corruption would show itself through errors 823 or similar, with the clear hint to run DBCC CHECKDB. See [Database corruption on Azure SQL Database](#) to check further.
- A failure to backup the transaction log would have recorded a corresponding error.
- A reconfiguration/failover/scaling operation would be visible through a change in NodeName or AppName.

In the specific issue mentioned above, it turned out that the Create Index operation was interrupted by failovers for planned maintenance. The customer ran the operation on the weekend to avoid user impact, but this had also been the period for the monthly upgrades.

Mitigation

The best mitigation step depend on the actual cause that you have found in the investigation:

- Retry one more time, e.g. if the failure had been caused by maintenance.
- Scale the database to a higher SLO - which provides more space for transaction log and tempdb. It also speeds up the create process due to better CPU and I/O capacity.
- Use the RESUMABLE option on the Create Index command. See the mitigation section in [Indexes](#) for further details.

How good have you found this content?

