Log Full

Last updated by | Amie Coleman | Nov 30, 2022 at 6:40 AM PST

Contents

Issue

This TSG is designed to help you understand the different causes of transaction log full issues and how to investigate them using our internal telemetry and techniques on customer remote sessions.

Each SQL Database has a transaction log which records all transactions and modifications made by each transaction. If the transaction log becomes full it will not allow further data modifications (INSERT, UPDATE, DELETE), thus impacting the customers environment and activity significantly.

In Azure SQL Database, the customer may encounter error 9002 or 40552 when the transaction log is full. These errors occur when the database transaction log, managed by Azure SQL Database, exceeds thresholds for space and cannot continue to accept transactions.

Error

When you encounter a log full scenario, you can observe variations of the below error. The error indicates what is preventing the reuse of transaction log space:
The transaction log for database '' is full due to '' and the holdup lsn is ().'

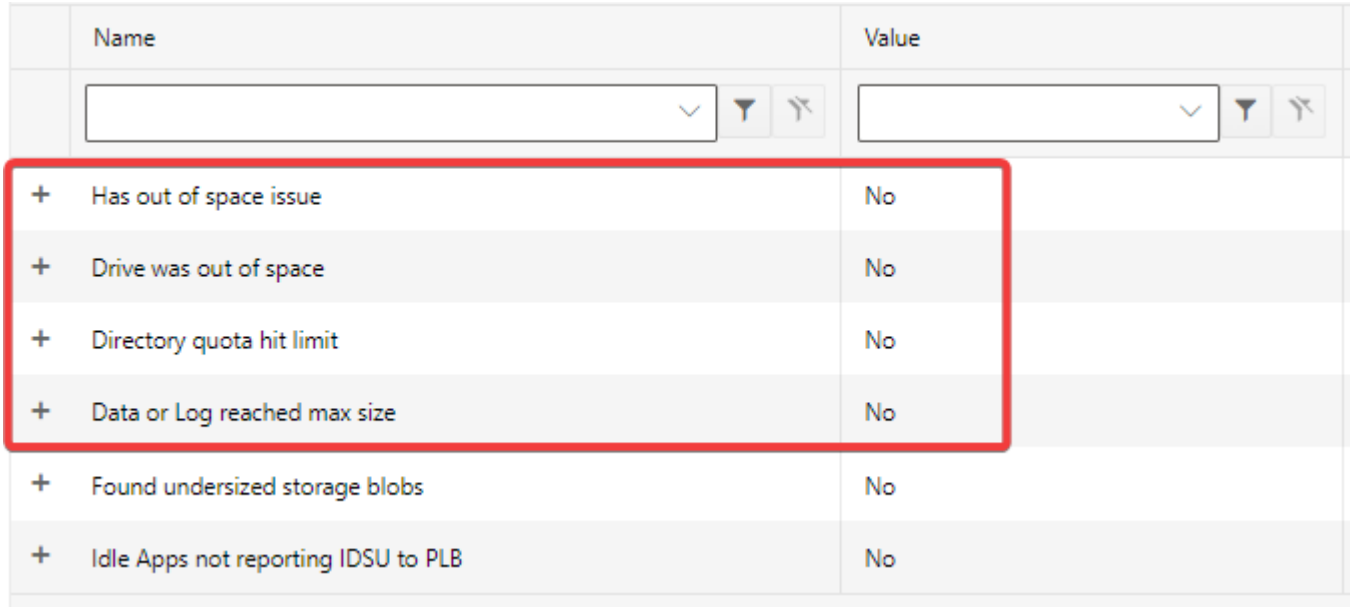
You may also observe the error 40552:
40552: The session has been terminated because of excessive transaction log space usage. Try modifying fewer rows in a single transaction

Investigation/Analysis

ASC/Kusto/XTS

Check for space related issues
Go to ASC >> SQL Troubleshooter >> Performance >> Space Issues

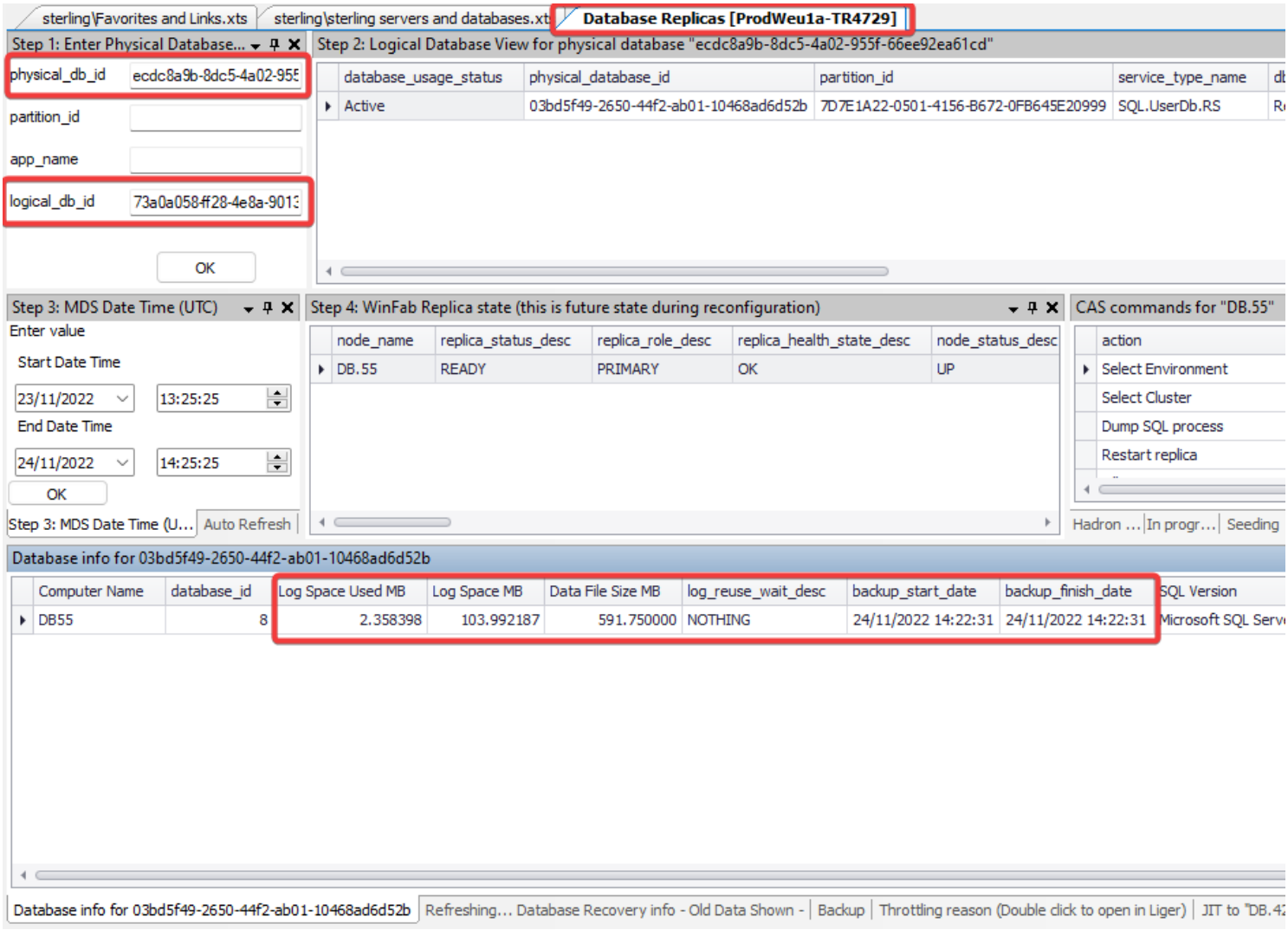
The Space Troubleshooter will provide you with an overview/summary to quickly identify if the customer is facing any space related issues:



Name	Value
Has out of space issue	No
Drive was out of space	No
Directory quota hit limit	No
Data or Log reached max size	No
Found undersized storage blobs	No
Idle Apps not reporting IDSU to PLB	No

Database Replicas view in XTS

Using the database replicas view in XTS, review the log utilisation and log reuse information. The database replicas view requires the physical DB ID or logical DB ID, which you can get from the Sterling Servers and Databases view:



Check Historical Log Usage

Run the below Kusto query to see historical log size information (space used/max size). This can help you understand the usage trend:

```
MonDmIoVirtualFileStats
| where TIMESTAMP >= ago(1d)
| where LogicalServerName =~ ""
| where db_name =~ ""
| where type_desc == "LOG" //looking only for "LOG"
| where is_primary_replica == 1
| project TIMESTAMP, file_id, type_desc, spaceused_mb, max_size_mb, size_on_disk_mb =(size_on_disk_bytes/1024/
//| render timechart
```

Automated Backup Information

ASC contains information on the Automated Backups for Azure SQL Database, and while it is not common to see an issue with missing log backups, you can utilise this telemetry to confirm that backups are happening as expected:

In ASC >> Sql Troubleshooter Report >> Backup/Restore >> Short-Term Retention Backups

Log Backup

Show log backup data within the specified time frame

Drag a column header and drop it here to group by that column

originalEventTimestamp	LogicalServerName	logical_database_name	logical_database_id	backup_start_date	backup_end_date	BackupSizeInMb
2022-11-25 11:29:29	prod-we-001	PRDDB		11/25/2022 11:29:27 AM	11/25/2022 11:29:29 AM	126.71
2022-11-25 11:24:30				11/25/2022 11:24:27 AM	11/25/2022 11:24:29 AM	125.96
2022-11-25 11:19:30				11/25/2022 11:19:27 AM	11/25/2022 11:19:29 AM	129.03
2022-11-25 11:14:29				11/25/2022 11:14:27 AM	11/25/2022 11:14:28 AM	133.7
2022-11-25 11:09:29				11/25/2022 11:09:27 AM	11/25/2022 11:09:29 AM	147.68
2022-11-25 11:04:29				11/25/2022 11:04:27 AM	11/25/2022 11:04:29 AM	140.8
2022-11-25 10:59:29				11/25/2022 10:59:27 AM	11/25/2022 10:59:28 AM	142.13

Customer-side checks

The below customer-side checks can be provided to the customer or reveiwed during a remote session. We will look at the current log usage percent and what is preventing log trunction from happening:

Check log usage percent

```
dbcc sqlperf(logspace)
```

	Database Name	Log Size (MB)	Log Space Used (%)	Status
1	master	1.492188	55.49738	0
2	tempdb	15.99219	23.54665	0
3	model	7.992188	15.44477	0
4	msdb	8.992188	14.3788	0
5	MetadataDB	103.9922	12.03892	0

Check log reuse wait

```
select name, log_reuse_wait_desc from sys.databases
```

	name	log_reuse_wait_desc
1	master	NOTHING
2	amcolematest	NOTHING
3	metadatadbnew	NOTHING
4	AdentureWorksDB	NOTHING
5	MetadataDB	NOTHING

In the above example, we can see the log_reuse_wait_desc is "NOTHING", which means that there is nothing blocking the log truncation process.

The below table contains descriptions of the other potential reasons for what is preventing log truncation (as of the last checkpoint). The log reuse wait informs you to what conditions or causes are preventing the transaction log from being truncated by a regular log backup:

logreuse_wait_desc	Diagnosis	Response required
NOTHING	Typical state. There is nothing blocking the log from truncating	No
CHECKPOINT	A checkpoint is needed for log truncation. Rare.	No response required unless sustained.
LOG BACKUP	A log backup is in progress.	No response required unless sustained.
ACTIVE BACKUP OR RESTORE	A database backup is in progress.	No response required unless sustained.
ACTIVE TRANSACTION	An ongoing transaction is preventing log truncation.	The log file cannot be truncated due to active and/or uncommitted transactions.
REPLICATION	In Azure SQL Database, likely due to change data capture (CDC) feature.	In Azure SQL Database, query sys.dm_cdc_errors and resolve errors.
AVAILABILITY_REPLICA	Synchronization to the secondary replica is in progress.	No response required unless sustained.

Check for uncommitted or active transactions

Run this sample query to find uncommitted or active transactions and their properties.

Returns information about transaction properties, from sys.dm_tran_active_transactions.
Returns session connection information, from sys.dm_exec_sessions.
Returns request information (for active requests), from sys.dm_exec_requests.
Returns the current request's text or input buffer text, using the sys.dm_exec_sql_text or sys.dm_exec_input_buffer DMVs. If the data returned by the text field of sys.dm_exec_sql_text is NULL, the request is not active but has an outstanding transaction. In that case, the event_info field of sys.dm_exec_input_buffer will contain the last command string passed to the database engine.

```
SELECT [database_name] = db_name(s.database_id)
, tat.transaction_id, tat.transaction_begin_time, tst.session_id
, session_open_transaction_count = tst.open_transaction_count --uncommitted and unrolled back transactions ope
, transaction_duration_s = datediff(s, tat.transaction_begin_time, sysdatetime())
, input_buffer = ib.event_info
, request_text = CASE  WHEN r.statement_start_offset = 0 and r.statement_end_offset= 0 THEN left(est.text, 400
                       ELSE    SUBSTRING ( est.[text],    r.statement_start_offset/2 + 1,
                                           CASE WHEN r.statement_end_offset = -1 THEN LEN (CONVERT(nvarchar(ma
                                                ELSE r.statement_end_offset/2 - r.statement_start_offset/2 + 1
                                           END  )  END
, request_status = r.status
, request_blocked_by = r.blocking_session_id
, transaction_state = CASE tat.transaction_state
                      WHEN 0 THEN 'The transaction has not been completely initialized yet.'
                      WHEN 1 THEN 'The transaction has been initialized but has not started.'
                      WHEN 2 THEN 'The transaction is active - has not been committed or rolled back.'
                      WHEN 3 THEN 'The transaction has ended. This is used for read-only transactions.'
                      WHEN 4 THEN 'The commit process has been initiated on the distributed transaction. This i
                      WHEN 5 THEN 'The transaction is in a prepared state and waiting resolution.'
                      WHEN 6 THEN 'The transaction has been committed.'
                      WHEN 7 THEN 'The transaction is being rolled back.'
                      WHEN 8 THEN 'The transaction has been rolled back.' END
, transaction_name = tat.name
, azure_dtc_state    --Applies to: Azure SQL Database only
              =    CASE tat.dtc_state
                   WHEN 1 THEN 'ACTIVE'
                   WHEN 2 THEN 'PREPARED'
                   WHEN 3 THEN 'COMMITTED'
                   WHEN 4 THEN 'ABORTED'
                   WHEN 5 THEN 'RECOVERED' END
, transaction_type = CASE tat.transaction_type    WHEN 1 THEN 'Read/write transaction'
                                                  WHEN 2 THEN 'Read-only transaction'
                                                  WHEN 3 THEN 'System transaction'
                                                  WHEN 4 THEN 'Distributed transaction' END
, tst.is_user_transaction
, local_or_distributed = CASE tst.is_local WHEN 1 THEN 'Local transaction, not distributed' WHEN 0 THEN 'Distr
, transaction_uow    --for distributed transactions.
, s.login_time, s.host_name, s.program_name, s.client_interface_name, s.login_name, s.is_user_process
, session_cpu_time = s.cpu_time, session_logical_reads = s.logical_reads, session_reads = s.reads, session_wri
, observed = sysdatetimeoffset()
FROM sys.dm_tran_active_transactions AS tat
INNER JOIN sys.dm_tran_session_transactions AS tst  on tat.transaction_id = tst.transaction_id
INNER JOIN Sys.dm_exec_sessions AS s on s.session_id = tst.session_id
LEFT OUTER JOIN sys.dm_exec_requests AS r on r.session_id = s.session_id
CROSS APPLY sys.dm_exec_input_buffer(s.session_id, null) AS ib
OUTER APPLY sys.dm_exec_sql_text (r.sql_handle) AS est;
```

Mitigation

Depending on the outcome of your investigation (i.e., if the issue is still occurring what is the log reuse wait), there are multiple paths to take to resolve. Information on each mitigation can be found in the following workflow for Transaction Log full

1. ACTIVE_TRANSACTION. If you found that there was an active transaction that caused the log to fill up and is preventing log truncation, follow this article
2. AVAILABILITY_REPLICA, can mean that the transaction log records are being applied on the Geo Secondary. Investigate the availability of the Secondary or the possibility of slow performance on the Secondary (you can accomplish this by creating an ASC Report for the Secondary resource). This could also indicate that there is a long running transaction on the primary (Using ASC, check Performance >> Blocking & Deadlocking >> Long Running Transactions Summary)

3. REPLICATION can indicate an issue related to CDC (Change Data Capture) or Transactional Replication. Review Change Data Capture or Transaction Replication
4. LOG_BACKUP would indicate that a log backup is in progress. Seeing the LOG_BACKUP reuse wait in Azure SQL DB is quite rare, but if it is preventing log truncation for a period of time and you cannot see log backups occurring (by reviewing MonBackup/ASC), open an IcM to the product group team for Backup/Restore

Public Doc Reference

Troubleshooting transaction log errors in Azure SQL DB
Error 9002 Troubleshooting
What is preventing Log Truncation

Internal Reference

Active Transaction TSG
Transactional Replication TSG
Change Data Capture TSG
262190811

Root Cause Classification

Cases resolved by this TSG should be coded to the following root cause:
Azure SQL v3\Backup/Restore\Automated Backups

How good have you found this content?