Shrink Database that have ColumnStore Indexes

Last updated by | Radhika Shah | Sep 5, 2022 at 2:10 PM PDT

Contents

Scenario

Customer is reaching 4TB storage on their Managed Instance and they run DBCC SHRINKFILE to reduce the size of the database file. However, the size was not reduced. They also tried reducing smaller chunk like 1GB, that did not work as well (no error reported).

Investigation/Analysis

Check for any shrink failure reported in the telemetry:

```
MonSqlShrinkInfo
| where TIMESTAMP >= ago(3d)
| where LogicalServerName contains {MIName}
| where isnotempty(failure_reason) or status  == "shrink_fail"
| project TIMESTAMP, LogicalServerName, AppName, database_id, file_id, shrink_id, tag, status, target_mb = (to
| order by TIMESTAMP asc
```

Sample output:

TIMESTAMP	LogicalServerName	AppName	database_id	file_id	shrink_id	tag	status	failure_reason	code_package_version
2022-01-26 08:03:36.8190	[redacted]	[redacted]	7	2	895FA20A-E8B3-4886-95B2-386CDB33C728	shrink_failure_reasons		log backup sync failure	15.0.2255.112-WASDA3-f65d69ef
2022-01-26 09:21:36.9250	[redacted]	[redacted]	7	1	43CE8F7D-4860-4887-B446-CB7CF71B8A8C	shrink_failure_reasons		shrink hit exception 3617	15.0.2255.112-WASDA3-f65d69ef
2022-01-26 09:21:36.9250	[redacted]	[redacted]	7	0	43CE8F7D-4860-4887-B446-CB7CF71B8A8C	shrink_failure_reasons		shrink hit exception 3617	15.0.2255.112-WASDA3-f65d69ef

Sample output above shows that out of 3 shrinks that failed, two were canceled by user (error 3617), and one was blocked by log backup.

There might be shrinks that failed without specific error message. Note that all of the shrinks that failed were with bigger chunk (1028GB). For shrinks with smaller chunks, it seems that they didn't explicitly fail, but more like total reclaimed space was 0.

Sample output:

TIMESTAMP	LogicalServerName	AppName	database_id	file_id	shrink_id	tag	status	target_mb	file_size_mb	diffingb	percent_complete	failure_reason
2022-01-26 08:13:01.2520			7	1	1F03A954-50A3-457F-AC46-20AA7F17C8C3	shrink_sizes	shrink_fail	1,031,147	2,083,951	1,028		
2022-01-26 08:14:30.3930			7	1	9B889272-8AB9-4B01-BDA9-9A9607100FB9	shrink_sizes	shrink_fail	1,031,147	2,083,951	1,028		
2022-01-26 08:15:30.4560			7	1	8354D562-D0BB-40A7-88C9-30B07127A84E	shrink_sizes	shrink_fail	1,031,163	2,083,951	1,028		
2022-01-26 09:21:36.9250			7	1	43CE8F7D-4860-4887-B446-CB7CF71B8A8C	shrink_sizes	shrink_fail	1,031,166	2,083,951	1,028		
2022-01-26 09:22:06.8790			7	1	B777E7E4-7E37-41A0-AAEA-4C2ABC8866F3	shrink_sizes	shrink_fail	1,032,232	2,083,951	1,027		
2022-01-26 09:26:01.1760			7	1	D9ADE29C-0006-4E86-82C2-B50F504E1E1A	shrink_sizes	shrink_fail	1,032,216	2,083,951	1,027		
2022-01-26 09:29:01.0360			7	1	3D50844A-7FD5-4E3D-8A21-D11A460B1283	shrink_sizes	shrink_fail	1,031,150	2,083,951	1,028		
2022-01-26 09:31:01.0840			7	1	9BDBDEE6-96DC-4860-9C5E-5CAFBA6C8ED1	shrink_sizes	shrink_fail	1,031,150	2,083,951	1,028		

To confirm if the reason for these shrink failures was due to columnstore indexes:

```
MonSqlShrinkInfo
| where TIMESTAMP >= ago(3d)
| where LogicalServerName contains {MIName}
| where status contains "shrink_last_pages_columnstore"
| project TIMESTAMP, LogicalServerName, AppName, database_id, file_id, shrink_id, tag, status, target_mb = (to
| order by TIMESTAMP asc
```

Sample Output:

TIMESTAMP	LogicalServerName	AppName	data...	file_id	shrink_id	tag	status	target_mb	file_size_mb	diffingb	percent_complete	failure_reason	code_package_version
2022-09-03 00:04:21.7921085			7	1	5585C736-ECFD-49BD-9B50-BAE75307391E	shrink_status	shrink_last_pages_columnstore						16.0.537.221-RelDB-2e9b7b04
2022-09-03 00:26:21.9338888			7	1	6CFEAF66-C08C-4D5F-891F-6EAABAA60865	shrink_status	shrink_last_pages_columnstore						16.0.537.221-RelDB-2e9b7b04
2022-09-04 00:04:40.1771823			7	1	43BA92BC-3CD3-4CFD-A990-EAF846F654A1	shrink_status	shrink_last_pages_columnstore						16.0.537.221-RelDB-2e9b7b04
2022-09-04 00:25:40.2673249			7	1	245CD335-244D-4314-BD35-B2E83A710100	shrink_status	shrink_last_pages_columnstore						16.0.537.221-RelDB-2e9b7b04
2022-09-05 00:04:17.2804924			7	1	E6DBDB3E-608B-4745-A598-C4D6EB361DF7	shrink_status	shrink_last_pages_columnstore						16.0.537.221-RelDB-2e9b7b04
2022-09-05 00:25:47.4060271			7	1	876E9CB2-A73E-41C5-B8AE-2FE904DF4C46	shrink_status	shrink_last_pages_columnstore						16.0.537.221-RelDB-2e9b7b04

To find out the Columnstore index that is preventing shrink:

```
MonSQLSystemHealth
| where TIMESTAMP >= {StartTime} and TIMESTAMP <= {EndTime}
| where LogicalServerName == {MIName}
| where message contains "Page information during shrink" and message contains "IsColumnstore 1"
| project TIMESTAMP, message
```

Sample Output:

TIMESTAMP	message
2022-09-03 00:04:21.7452411	2022-09-03 00:04:10.73 spid103 [DbId: 7][PageId: 1:1069448] Page information during shrink. AllocUnitId: 72057595543289856, PartitionId: 72057595345960960, IndexId: 1, ObjectId: 349309100, AllocUnitType LOB_DATA, IsColumnstore 1
2022-09-03 00:04:21.7452411	2022-09-03 00:04:10.73 spid103 [DbId: 7][PageId: 1:1069449] Page information during shrink. AllocUnitId: 72057595543289856, PartitionId: 72057595345960960, IndexId: 1, ObjectId: 349309100, AllocUnitType LOB_DATA, IsColumnstore 1
2022-09-03 00:04:21.7452411	2022-09-03 00:04:10.73 spid103 [DbId: 7][PageId: 1:1069450] Page information during shrink. AllocUnitId: 72057595543289856, PartitionId: 72057595345960960, IndexId: 1, ObjectId: 349309100, AllocUnitType LOB_DATA, IsColumnstore 1
2022-09-03 00:04:21.7452411	2022-09-03 00:04:10.73 spid103 [DbId: 7][PageId: 1:1069451] Page information during shrink. AllocUnitId: 72057595543289856, PartitionId: 72057595345960960, IndexId: 1, ObjectId: 349309100, AllocUnitType LOB_DATA, IsColumnstore 1
2022-09-03 00:04:21.7452411	2022-09-03 00:04:10.73 spid103 [DbId: 7][PageId: 1:1069452] Page information during shrink. AllocUnitId: 72057595543289856, PartitionId: 72057595345960960, IndexId: 1, ObjectId: 349309100, AllocUnitType LOB_DATA, IsColumnstore 1
2022-09-03 00:04:21.7452411	2022-09-03 00:04:10.73 spid103 [DbId: 7][PageId: 1:1069453] Page information during shrink. AllocUnitId: 72057595543289856, PartitionId: 72057595345960960, IndexId: 1, ObjectId: 349309100, AllocUnitType LOB_DATA, IsColumnstore 1
2022-09-03 00:04:21.7452411	2022-09-03 00:04:10.73 spid103 [DbId: 7][PageId: 1:1069454] Page information during shrink. AllocUnitId: 72057595543289856, PartitionId: 72057595345960960, IndexId: 1, ObjectId: 349309100, AllocUnitType LOB_DATA, IsColumnstore 1

Customer can identify problematic index with following query:

```
Select * from sys.indexes
where object_id = <ObjectId> and index_id = <IndexId>
```

Mitigation

The Root cause in this case was that the last page in the file belongs to the columnstore index which was blocking file shrink.

Mitigation: Customer needs to rebuild the problematic columnstore index with index_id and object_id (as identified above) after which shrink will be successful.

Note: Even with the above mitigation, we recommend shrinking in smaller chunks since shrink is single threaded operation and is very IO intensive.

Internal Reference

ICM 285739510

How good have you found this content?