Resumable Table Add Constraints

Last updated by | Subbu Kandhaswamy | Sep 20, 2022 at 9:14 AM PDT

Feature is currently in Private Preview. Tentative ETA on Public preview - Week of 09/19.

- We will be able to allow list customer's subscription to enable this feature during private preview. Contact SQL EEE's Or engineering thru ICM, if your customer is interested.

Contents

Resumable Table Add Constraints

Currently SQL Server supports resumable operations for online index creation and rebuild. The resumable operations allow index operations executed online (ONLINE=ON) and to:

- Pause and restart an index create or rebuild operation multiple times to fit a maintenance window
- Recover from index create or rebuild failures (such as database failovers or running out of disk space)
- Enable truncation of transaction logs during an index create or rebuild operation.
- When an index operation is paused, both the original index and the newly created one require disk space and need to be updated during DML operations.

What's new?

The new SQL Server extensions allow a resumable operation for the T-SQL DDL command "ALTER TABLE ADD CONSTRAINT" adding a primary/unique key. Currently this operation can be executed online (with ONLNE=ON), however, it may take many hours for a large table to complete, and it consumes a great number of resources not to mention a possible (although unlikely) failure/interruption during such execution. To allow to pause such operation during a maintenance window or to restart it from where it was interrupted during an execution failure without restarting it from the beginning, we introduced a resumable capability for ALTER TABLE ADD CONSTRAINT.

The new resumable capability for ALTER TABLE ADD CONSTRAINT supports the following customer scenarios:

- Pause/Resume running ADD CONSTRAINT operation to perform it during maintenance windows
- Resume such operation after failovers and system failures
- Executing such operation on a large table despite the small log size available.

[! NOTE] The resumable operation for ALTER TABLE ADD CONSTRAINT requires the ALTER command to be executed online (WITH ONLINE = ON).

Permissions

Requires ALTER permission on the table.

No new permissions for resumable ALTER TABLE ADD CONSTRAINT are required.

How to check resumable operation status

To view the status on for the resumable table constraint operation use the existing view sys.index_resumable_operations. This is a system view that monitors and checks the current execution status for resumable Index rebuild or creation.

Using Resumable options

T-SQL Syntax for ALTER TABLE

See the public documentation on ALTER TABLE (Transact-SQL)

Below is the current syntax with resumable extensions indicated in (yellow)

```
ALTER TABLE { database_name.schema_name.table_name | schema_name.table_name |table_name}
{
  . . .
   | ADD
      { ..
         | <table_constraint>
        ..
       } [ ,...n ]
. . .
}
[ ; ]

< table_constraint > ::=
 [ CONSTRAINT constraint_name ]
{
   {PRIMARY KEY | UNIQUE }
     {
       CLUSTERED (column [ ASC | DESC ] [ ,... n ])
       | NONCLUSTERED (column [ ASC | DESC ] [ ,... n ])
       | NONCLUSTERED HASH (column [ ,... n ] ) WITH ( BUCKET_COUNT = bucket_count )
      }
   [WITH <resumable_options> [ ; ]]
}
<resumable_options>]::=
{
    | ONLINE = { ON [ ( <low_priority_lock_wait> ) ] | OFF }
    | MAXDOP = max_degree_of_parallelism
    | RESUMABLE = { ON | OFF }
    | MAX_DURATION = <time> [MINUTES}
}

<low_priority_lock_wait>::=
{
    WAIT_AT_LOW_PRIORITY ( MAX_DURATION = <time> [ MINUTES ] ,
                          ABORT_AFTER_WAIT = { NONE | SELF | BLOCKERS } )
}
```

Remarks for ALTER TABLE

- A new clause **WITH <resumable_options** (in yellow) have been added to the current T-SQL syntax.
- Option RESUMABLE and MAX_DURATION are new and have been added to the existing syntax.
- RESUMABLE **=** { ON | **OFF**} -- specifies whether an online add constraint operation is resumable.
  ON -- add table constraint operation is resumable.
  OFF -- add table constraint operation is not resumable.

  Default for RESUMABLE is OFF
- MAX_DURATION = *time* [MINUTES] used with RESUMABLE = ON (requires ONLINE = ON).\
  Indicates time (an integer value specified in minutes) that a resumable online add constraint operation is executed before being paused. If not specified, the operation continues until completion.
- Options **ONLINE** and **MAXDOP** are already exist in previous versions
- The <low_priority_lock_wait> clause already exist in previous versions

To pause, resume or abort the resumable table constraint operation for ALTER TABLE ADD CONSTRAINT, use the following T-SQL syntax the exists today for resumable index operations. See ALTER INDEX (Transact-SQL) for more details.

T-SQL syntax for ALTER INDEX

For resumable constraints the existing ALTER INDEX ALL command is used

```
ALTER INDEX ALL ON <table_name>
      { RESUME [WITH (<resumable_index_options>,[...n])]
        | PAUSE
        | ABORT
      }
<resumable_index_option> ::=
 {
    MAXDOP = max_degree_of_parallelism
    | MAX_DURATION =<time> [MINUTES]
    | <low_priority_lock_wait>
 }
 <low_priority_lock_wait>::=
{
    WAIT_AT_LOW_PRIORITY ( MAX_DURATION = <time> [ MINUTES ] ,
                          ABORT_AFTER_WAIT = { NONE | SELF | BLOCKERS } )
}
```

Remarks for ALTER INDEX

```
ALTER INDEX ALL ON <Table> PAUSE
```

- Pause a resumable online add table constraint operation being executed

```
ALTER INDEX ALL ON <Table> RESUME [WITH (<resumable_index_options>,[...n])]
```

- Resume an add table constraint operation that is paused manually or due to a failure.
- MAX_DURATION used with RESUMABLE=ON
  The time (an integer value specified in minutes) that the resumable add table constraint operation is executed after being resumed. Once the time expires, the resumable operation is paused if it is still running.
- WAIT_AT_LOW_PRIORITY used with RESUMABLE=ON and ONLINE = ON.\
  Resuming an online add table constraint operation after a pause must wait for blocking operations on this table. WAIT_AT_LOW_PRIORITY indicates that the add table constraint operation will wait for low priority locks, allowing other operations to proceed while the resumable operation is waiting. Omitting the WAIT_AT_LOW_PRIORITY option is equivalent to WAIT_AT_LOW_PRIORITY (MAX_DURATION = 0 minutes, ABORT_AFTER_WAIT = NONE). For more information, see WAIT_AT_LOW_PRIORITY.

```
ALTER INDEX ALL ON <Table> ABORT
```

- Abort a running or paused add table constraint operation that was declared as resumable. The abort operation must be explicitly execute an ABORT command to terminate a resumable constraint operation. Failure or pausing a resumable table constraint operation does not terminate its execution; rather, it leaves the operation in an indefinite pause state.

See ALTER INDEX (Transact-SQL) for more details on PAUSE/RESUME/ABORT available for resumable operations.

View the status for resumable operation

To view the status on for the resumable table constraint operation use the existing view sys.index_resumable_operations.

Permissions

Requires `ALTER` permission on the table.

No new permissions for resumable ALTER TABLE ADD CONSTRAINT are required.

Examples

1. Resumable ALTER TABLE operation for adding a primary key clustered on column (a) with max_duration of 240 minutes

```
ALTER TABLE table1
ADD CONSTRAINT PK_Constrain PRIMARY KEY CLUSTERED (a)
WITH (ONLINE = ON, MAXDOP = 2, RESUMABLE = ON, MAX_DURATION = 240);
```

2. Resumable ALTER TABLE operation for adding a unique constraint on two columns (a and b) with max_duration of 240 minutes

```
ALTER TABLE table2
ADD CONSTRAINT PK_Constrain UNIQUE CLUSTERED (a,b)
WITH (ONLINE = ON, MAXDOP = 2, RESUMABLE = ON, MAX_DURATION = 240);
```

3. ALTER TABLE operation for adding a primary key clustered being paused and resumed

The table below shows two sessions Session#1 and Session#2 being executed chronologically using the following T-SQL statements. Session #1 executes a resumable ADD CONSTRAINT operation creating a primary key on column Col1. Session#2 check the execution status for the running constraint. After some time, it pauses the reusable operation. Check the status for the paused constraint. Finally, Session#1 resumes the paused constraint and session#2 check the status again.

<table>
<tr><th>Session #1</th><th>Session #2</th></tr>
<tr><td>Execute resumable add constraint

ALTER TABLE TestConstraint
ADD CONSTRAINT PK TestConstraint PRIMARY KEY (Col1)
WITH (ONLINE = ON, MAXDOP = 2, RESUMABLE = ON, MAX_DURATION = 30);</td><td></td></tr>
<tr><td></td><td>Check the constraint status

SELECT sql_text, state_desc, percent_complete
FROM sys.index_resumable_operations;</td></tr>
<tr><td>Output showing the operation

<table><tr><th>sql_text</th><th>state_desc</th><th>percent_complete</th></tr><tr><td>ALTER TABLE TestConstraint (...)</td><td>RUNNING</td><td>43.552</td></tr></table></td><td></td></tr>
<tr><td></td><td>Pause the resumable constraint

ALTER INDEX ALL ON TestConstraint PAUSE;</td></tr>
<tr><td><b>Error</b>

Msg 1219, Level 16, State 1, Line 6
Your session has been disconnected because of a high priority DDL operation.

Msg 1750, Level 16, State 1, Line 6
Could not create constraint or index. See previous errors.

Msg 0, Level 20, State 0, Line 5
A severe error occurred on the current command.
The results, if any, should be discarded.</td><td></td></tr>
<tr><td></td><td>Check the constraint status

SELECT sql_text, state_desc, percent_complete
FROM sys.index_resumable_operations;</td></tr>
<tr><td>Output showing the operation</td><td></td></tr>
</table>

<table>
<tr><th>Session #1</th><th>Session #2</th></tr>
<tr><td><table><tr><th>sql_text</th><th>state_desc</th><th>percent_complete</th></tr><tr><td><code>ALTER TABLE TestConstraint (...)</code></td><td>PAUSED</td><td>65.339</td></tr></table></td><td></td></tr>
<tr><td><code>ALTER INDEX ALL ON TestConstraint RESUME;</code></td><td></td></tr>
<tr><td></td><td>Check the constraint status
<pre>SELECT sql text, state_desc,
percent complete
 FROM sys.index_resumable_operations;</pre></td></tr>
<tr><td>Output showing the operation
<table><tr><th>sql_text</th><th>state_desc</th><th>percent_complete</th></tr><tr><td><code>ALTER TABLE TestConstraint (...)</code></td><td>RUNNING</td><td>90.238</td></tr></table></td><td></td></tr>
</table>

Once the operation is completed execute the following T-SQL statement to check the constraint.

```
SELECT constraint_name, table_name, constraint_type
FROM INFORMATION_SCHEMA.TABLE_CONSTRAINTS WHERE CONSTRAINT_TYPE='PRIMARY KEY';
GO
```

Here is the result set

constraint_name	table_name	constraint_type
PK_Constraint	TestConstraint	PRIMARY KEY

References

- Guidelines for Online Index Operations
- CREATE INDEX (Transact-SQL)
- ALTER INDEX (Transact-SQL)
- sys.index_resumable_operations
- WAIT_AT_LOW_PRIORITY

https://supportability.visualstudio.com/AzureSQLDB/_wiki/wikis/AzureSQLDB.wiki/640755/Resumable-Table-Add-Constraints

How good have you found this content?