Restore taking too long - Roll-Forward

Last updated by | Vitor Tomaz | Jun 8, 2022 at 5:35 AM PDT

Contents

Issue

Low performance on a restore operation or a restore operation taking more time than expected can be the reason for a customer to request an RCA and\or ask how can the restore be done faster.

A restore is a multiphase process. The possible phases of a restore include the data copy, redo (roll forward), and undo (roll back) phases:

- The data copy phase involves copying all the data, log, and index pages from the backup media of a database to the database files.
- The redo phase applies the logged transactions to the data copied from the backup to roll forward that data to the recovery point. At this point, a database typically has uncommitted transactions and is in an unusable state. In that case, an undo phase is required as part of recovering the database.
- The undo phase, which is the first part of recovery, rolls back any uncommitted transactions and makes the database available to users. After the roll back phase, subsequent backups cannot be restored.

The roll-forward process (redo) is basically applying the transaction that were commited until the point in time specified by the customer, the roll-back (undo) is reverting the transactions that didn't commited at the point in time specified by the customer.

Any of this two processes, roll-forward and roll-back, can take more or less time depending on the amount of transactions being executed at the point in time the customer selected.

Investigation/Analysis

To confirm how much time the roll-forward took or if it's still running when did it started you can run the following kusto queries.

Run this query to find the database id and the app name if you don't know them.

```
MonRestoreEvents
|where TIMESTAMP >=datetime(2022-02-22 11:09:08.8770000)
|where TIMESTAMP <datetime(2022-02-22 13:09:08.8770000)
//|where restore_request_id contains "26fddf01-c552-4600-9f49-859925a37eb2" //If we have the restore request i
| where LogicalServerName == "<instance name>"
//| where restore_database_progress contains "Executing restore query RESTORE DATABASE" //uncomment this line
| project originalEventTimestamp, LogicalServerName, AppName, event, restore_database_progress, message, resto
```

With the information (AppName and database id) obtained from the previous query we can run the followin query

Example on where to find the database id from the ouput of the previous query: "Executing restore query RESTORE DATABASE [xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx] FROM URL = N'..."

```
MonSQLSystemHealth
| where AppName =="xxxxx"
|where TIMESTAMP >=datetime(2022-03-14 11:09:08.8770000)
|where TIMESTAMP <datetime(2022-03-15 13:09:08.8770000)
|where message contains "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx" //database name while restoring, get it from Mo
| where message contains "Offline roll-forward begins" or message contains "Offline roll-forward is complete"
```

With the output of this last query we can see when the roll-forward has started and when it has finished, if it has already finished, and confirm that the roll-forward process had some contribution to the time the restore took or is taking.

Another contribution for the restore process to take more time is if in the point in time specified the database was using lower files, P10 for example. When restoring the files created will be the same size has they were in the point in time, it's not possible to increase the files during the restore or specify the size has a restore parameter.

Customers can also see when the roll-forward process started and finished by looking at the Erro Log.
As the Error Log sometimes constains too much information we can use the sp_mirestoreinfo script to filter and just view the relevant information regarding restores.

Mitigation

There is no mitigation, the only work around is to select a different point in time where the workload the database was facing was much lower.

Public Doc Reference (optional)

Understanding How Restore and Recovery of Backups Work in SQL Server

Root Cause Classification

Cases resolved by this TSG should be coded to the following root cause:
<Root cause path>

How good have you found this content?