Automated backups leading to Performance issue

Last updated by | Radhika Shah | Sep 28, 2022 at 1:57 PM PDT

Contents

- Issue
- Investigation/Analysis
- Mitigation
- Internal Reference
- Public Doc Reference

Issue

Customers note that their queries/jobs are running slow. When they further monitor automated backups, they notice that the backups are running in parallel to their jobs and possibly causing performance issues.

Investigation/Analysis

Check the automated backup schedule and confirm if it indeed coincides with the timeframe that customer experiences performance degradation.

```
MonBackup
| where originalEventTimestamp >= ago(10d)
| where LogicalServerName contains "nerve-primary-db"
| where event startswith 'database_backup'
| where event_type == 'BACKUP_METADATA_DETAILS'
| where backup_type != 'Log'
| where logical_database_id =~ {logical_database_id}    //can be fetched from ASC Resource Explorer under MI/D
| extend backup_size_gb = todouble(uncompressed_backup_size) / 1024.0 /1024.0 /1024.0
| extend start_time = todatetime(backup_start_date)
| extend end_time = todatetime(backup_end_date)
| extend backup_duration_h = todouble(datetime_diff('second', end_time, start_time)) / 60.0 / 60.0
| project originalEventTimestamp, backup_type, backup_start_date, backup_end_date, backup_size_gb, backup_dura
| order by originalEventTimestamp desc
```

originalEventTimestamp	backup_type	backup_start_date	backup_end_date	backup_size_gb
2022-09-27 10:07:48.5822039	Diff	9/27/2022 3:55:19 AM	9/27/2022 10:07:12 AM	697.374145507813
2022-09-26 22:50:06.2541295	Diff	9/26/2022 3:55:14 PM	9/26/2022 10:49:30 PM	700.014282226563
2022-09-26 09:09:54.8209293	Diff	9/26/2022 3:55:38 AM	9/26/2022 9:09:17 AM	523.572570800781
2022-09-25 21:03:43.2283586	Diff	9/25/2022 2:22:09 PM	9/25/2022 9:02:59 PM	458.240173339844
2022-09-25 12:59:29.1570027	Full	9/24/2022 11:37:55 PM	9/25/2022 12:59:01 PM	4280.67950439453
2022-09-24 21:31:41.5844423	Diff	9/24/2022 2:32:05 PM	9/24/2022 9:31:09 PM	895.898864746094
2022-09-24 07:57:58.9577149	Diff	9/24/2022 2:32:15 AM	9/24/2022 7:57:35 AM	771.427978515625

We can see that the Full backup took over 13 hours (avg speed of 320gb/hr) while the differential backups taken every 12 hours (even though big enough) are taking ~6-7 hrs and slower than full backups (~128gb/hr - 142gb/hr). We expect the full and differential backups to have similar avg backup speed.

Further check if the database has an even file distribution:

```
MonDmIoVirtualFileStats
| where AppName =~ {AppName}
| where db_name =~ {logical_database_id}    //can be fetched from ASC Resource Explorer under MI/DB properties
| where type_desc == "ROWS"
| filter db_name !in ("master", "tempdb", "model", "msdb")
| filter end_utc_date >= ago(1h)
| filter is_primary_replica == 1
| extend SizeMB = todouble(size_on_disk_bytes/1024/1024)
| extend SizeGB = bin(SizeMB/1024, 0.01)
| extend SpaceUsedGB = bin(todouble(spaceused_mb)/1024, 0.01)
| project end_utc_date, file_id, type_desc, SizeMB, SizeGB, SpaceUsedGB
| sort by end_utc_date asc
```

Sample Output:

end_utc_date	file_id	type_desc	SizeMB	SizeGB	SpaceUsedGB
2022-09-28 19:55:49.3570000	1	ROWS	3625106	3540.14	3540.13
2022-09-28 19:55:49.3570000	3	ROWS	1566	1.52	0.84
2022-09-28 19:55:49.3570000	4	ROWS	2424	2.36	0.92
2022-09-28 19:55:49.3570000	5	ROWS	3624	3.53	1.44
2022-09-28 19:55:49.3570000	6	ROWS	4224	4.12	1.75
2022-09-28 19:55:49.3570000	7	ROWS	5216	5.09	2.26
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
2022-09-28 19:55:49.3570000	30	ROWS	16	0.01	0
2022-09-28 19:55:49.3570000	31	ROWS	7700	7.51	0.78

From the sample output above, we see that the DB is heavily imbalanced with the data file with file id 1 containing 3.5TB of the total ~3.7TB even though there are 30 files in the database. A more balanced layout can be achieved for example by moving tables to different files, rebuilding indexes on different files etc. This would be the best way for the Customer to increase the IOPS resources usage by balancing the database file layout.

Each differential backup also contains all changes made in the database since the last full backup. So, the total size of all differential backups gradually increases over the course of a week. Then it drops sharply after an older set of full, differential, and log backups ages out.

For example, assume that a heavy write activity, such as index rebuild, runs just after a full backup is completed. The modifications that the index rebuild makes will then be included:

- In the transaction log backups taken over the duration of the rebuild.
- In the next differential backup.
- In every differential backup taken until the next full backup occurs.

For the last scenario in larger databases, an optimization in the service creates a full backup instead of a differential backup if a differential backup would be excessively large otherwise. This reduces the size of all differential backups until the following full backup.

Mitigation

To mitigate the performance impact caused due to large backups, we can offer below options to customer:

- Ask customer if they'd like to reschedule the time of day (for differential) or the day of week (for full backup) for the backups to be taken such that the backups would not impact their workload. If so, create an ICM with the Backup Restore team to reschedule the backup window.
- In case of uneven file distribution, suggest customer to achieve a more balanced layout by moving tables to different files, rebuilding indexes on different files etc.
- Have customer optimize their workload to minimize the size of differential backups by avoiding large write operations, like index rebuilds, more frequently than they need to. For large data load operations, consider using clustered columnstore indexes and following related best practices. Also consider reducing the number of non-clustered indexes.

Internal Reference

ICM 320527148

Public Doc Reference

Fine-tune backup storage consumption

How good have you found this content?