Data Aging and Job Schedule Conflict Research

Date: 2025-11-14 **Objective:** Understand conflicts between job schedules and aging policies that delay storage space reclamation

Problem Statement

Job schedules may conflict with or delay aging policies, preventing storage space from being freed up. This research identifies the root causes and provides analysis methodology.

How Data Aging Works in Commvault

Retention Cycle Definition

> ***Retention Cycle:*** *A complete full (or synthetic full) backup followed by all subsequent incremental, differential, or transactional log backups that depend on that full backup.*

Aging Eligibility Requirements

Data becomes eligible for aging when **BOTH** conditions are met: 1. **Days retention exceeded** - Calendar days have passed 2. **Cycle retention exceeded** - Required number of backup cycles completed

Formula:

```
Data Eligible for Aging = (Current Date - Job Date > Retention Days)
                          AND
                          (Completed Cycles > Retention Cycles)
```

Common Conflicts Between Schedules and Aging

Conflict #1: Incomplete Backup Cycles

Problem: - Retention policy requires 2 cycles + 30 days - Full backups scheduled weekly - If full backup fails or is disabled, cycle never completes - Data remains on storage even after 30 days have passed

Example:

```
Week 1: Full Backup ✓
Week 2: Full Backup ✗ FAILED
Week 3: Full Backup ✗ FAILED
Week 4: Full Backup ✗ FAILED
Result: Only 1 cycle completed, data retention NOT met, aging BLOCKED
```

Impact: Data from Week 1 cannot be aged because cycle count (1) < required cycles (2)

Conflict #2: Backup Schedule Too Infrequent

Problem: - Retention: 14 days + 1 cycle - Full backup schedule: Monthly - Aging cannot occur for 30+ days even though days requirement is 14

Example:

```
Day 1: Full Backup
Day 15: Incremental (14 days passed, but cycle not complete)
Day 30: Full Backup (NOW cycle 1 completes, aging can begin)
Result: 30 days to reclaim storage, not 14 days
```

Impact: Storage space held 2x longer than days setting suggests

Conflict #3: Disabled Subclients with Cycle Retention

Problem: - Subclient is disabled (no longer backing up) - Retention: 30 days + 1 cycle - Since no new backups run, cycles never increment - Data NEVER ages out

Example:

```
Last Full Backup: 90 days ago
Subclient Status: DISABLED
Cycles Completed Since: 0
Result: Data aged based on days? NO - cycle requirement blocks aging
```

Impact: Disabled clients consume storage indefinitely

Solution: Commvault setting to "Ignore cycle retention on backup activity disabled subclients"

Conflict #4: Failed Full Backups

Problem: - Incremental backups succeed - Full backups consistently fail - New cycle never starts - Old data cannot age because cycle count doesn't increase

Example:

```
Cycle 1: Full ✓ → Inc ✓ → Inc ✓ → Inc ✓
Cycle 2: Full ✗ → Inc ✓ → Inc ✓ → Inc ✗
Cycle 3: Full ✗ → Inc ✓ → Inc ✗ → Inc ✓
Result: Still on Cycle 1, aging blocked for months
```

Impact: Critical storage bloat

Conflict #5: Auxiliary Copy Dependencies

Problem: - Primary copy data is eligible for aging - Auxiliary copy (cloud/tape) still depends on this data - Primary data cannot be pruned until aux copy is independent

Example:

```
Primary Copy: 60 days old, eligible for aging
Aux Copy: Created via incremental forever, depends on 60-day-old data
Result: Primary data CANNOT be aged/pruned
```

Impact: Storage space not reclaimed despite meeting retention

Conflict #6: Aggressive Backup Schedules

Problem: - Multiple full backups per day - Retention: 30 days + 2 cycles - 2 cycles complete in 2 days - But 30 days haven't passed - Storage full of recent, ineligible data

Example:

```
Day 1: Full (Cycle 1) + Full (Cycle 2)
Day 2: Full (Cycle 3) ← Now have 3 cycles
Day 3-30: All data still retained (days requirement not met)
Result: Rapid cycle completion doesn't help storage
```

Impact: Cannot age data faster than days setting allows

Conflict #7: Data Aging Job Schedule Conflicts

Problem: - Data aging job runs daily at 12:00 PM - Backup jobs running at 12:00 PM - DDB verification running at 12:00 PM - Resource contention delays aging

Example:

```
12:00 PM: Data Aging job starts
12:05 PM: Heavy backup job starts
12:10 PM: DDB verification starts
Result: Aging job slowed down or postponed
```

Impact: Aging delayed by hours or skipped entirely

Solution: Stagger schedules (aging at 2 AM, backups at 8 PM, verification at 5 AM)

Aging Policy Analysis Methodology

Step 1: Identify Retention Policies

```
SELECT
    parentName AS PlanName,
    retainBackupDataForDays AS Days,
    retainBackupDataForCycles AS Cycles,
    enableDataAging
FROM retention_rules
WHERE enableDataAging = 1
ORDER BY Days, Cycles;
```

Step 2: Identify Plans with Short Retention

```
-- Plans that should age quickly (14 days or less)
SELECT
    parentName AS PlanName,
    entityName AS CopyName,
    retainBackupDataForDays AS Days,
    retainBackupDataForCycles AS Cycles
FROM retention_rules
WHERE retainBackupDataForDays <= 14
  AND retainBackupDataForDays > 0
ORDER BY Days;
```

Step 3: Calculate Effective Retention

```
-- Assuming 7-day average cycle duration
SELECT
    parentName AS PlanName,
    retainBackupDataForDays AS ConfiguredDays,
    retainBackupDataForCycles AS ConfiguredCycles,
    (retainBackupDataForCycles * 7) AS CycleDays,
    CASE
        WHEN retainBackupDataForDays > (retainBackupDataForCycles * 7)
        THEN retainBackupDataForDays
        ELSE (retainBackupDataForCycles * 7)
    END AS EffectiveDays
FROM retention_rules
WHERE enableDataAging = 1
ORDER BY EffectiveDays;
```

Step 4: Identify Potential Conflicts

Long Cycle Retention:

```
-- Plans where cycles extend retention significantly beyond days
SELECT
    parentName,
    retainBackupDataForDays AS Days,
    retainBackupDataForCycles AS Cycles,
    (retainBackupDataForCycles * 7) AS EstimatedCycleDays,
    ((retainBackupDataForCycles * 7) - retainBackupDataForDays) AS ExtraDaysFromCycle
FROM retention_rules
WHERE (retainBackupDataForCycles * 7) > retainBackupDataForDays
  AND retainBackupDataForDays > 0
ORDER BY ExtraDaysFromCycles DESC;
```

Mismatched Retention:

```
-- Plans with 1 cycle but many days (vulnerable to backup failures)
SELECT
    parentName,
    retainBackupDataForDays AS Days,
    retainBackupDataForCycles AS Cycles
FROM retention_rules
WHERE retainBackupDataForCycles = 1
  AND retainBackupDataForDays >= 30
ORDER BY Days DESC;
```

Storage Space Issues - Root Causes

1. Backup Cycle Not Completing

- **Symptom:** Days exceeded but data not aging
- **Cause:** Full backups failing or not running
- **Solution:** Fix backup job failures, verify full backup schedule

2. Disabled Clients Holding Space

- **Symptom:** Old data from inactive clients not aging
- **Cause:** Cycle requirement on disabled subclients
- **Solution:** Enable "Ignore cycle retention on disabled subclients" setting

3. Retention Cycles Too High

- **Symptom:** Storage fills up quickly despite short days setting
- **Cause:** Multiple cycles required = data held for weeks/months
- **Solution:** Reduce cycle retention to 1 or 2 for most policies

4. Incremental Forever Strategy

- **Symptom:** Very old data still on primary storage
- **Cause:** No synthetic fulls, cycles never close
- **Solution:** Schedule synthetic full backups to complete cycles

5. Auxiliary Copy Dependencies

- **Symptom:** Eligible data not pruning from primary
- **Cause:** Aux copy hasn't created independent copy
- **Solution:** Run aux copy jobs to create independent full backups

6. Job Schedule Overlaps

- **Symptom:** Aging jobs taking very long or skipping
- **Cause:** Resource contention with backups/verification
- **Solution:** Stagger schedules to avoid conflicts

Recommendations for Storage Optimization

Immediate Actions

1. **Review Failed Backup Jobs**
2. Identify plans with consecutive full backup failures
3. Fix issues preventing cycle completion
4. **Audit Disabled Subclients**
5. Find subclients no longer backing up

6. Enable days-only aging for disabled clients
7. **Reduce Cycle Retention**
8. Change from 2+ cycles to 1 cycle where possible
9. Rely more on days retention
10. **Stagger Job Schedules**
11. Aging: 2:00 AM (low activity time)
12. Backups: 8:00 PM - 6:00 AM
13. Verification: 5:00 AM

Long-term Improvements

1. **Standardize Retention Policies**
2. Short: 14 days + 1 cycle
3. Medium: 30 days + 1 cycle
4. Long: 365 days + 2 cycles
5. **Implement Synthetic Fulls**
6. Schedule weekly synthetic fulls
7. Ensures cycles complete even if full backup fails
8. **Monitor Aging Effectiveness**
9. Track storage space freed per aging job
10. Alert on aging job failures
11. **Review Auxiliary Copy Strategy**
12. Ensure aux copies create independent fulls
13. Avoid indefinite dependencies on primary data

Analysis Queries for Your Environment

Query 1: Plans Likely Affected by Cycle Issues

```sql
-- Plans with high cycle retention that could block aging
SELECT
    parentName AS PlanName,
    COUNT(*) AS CopiesCount,
    AVG(retainBackupDataForCycles) AS AvgCycles,
    MAX(retainBackupDataForCycles) AS MaxCycles,
    AVG(retainBackupDataForDays) AS AvgDays
FROM retention_rules
WHERE retainBackupDataForCycles > 1
  AND enableDataAging = 1
GROUP BY parentName
ORDER BY AvgCycles DESC, AvgDays;
```

Query 2: Effective vs Configured Retention Gap

```sql
-- Show where cycle retention extends data retention
SELECT
    parentName,
    entityName,
    retainBackupDataForDays AS ConfigDays,
    retainBackupDataForCycles AS Cycles,
    (retainBackupDataForCycles * 7) AS CyclesDays,
    MAX(retainBackupDataForDays, retainBackupDataForCycles * 7) AS EffectiveDays,
    (MAX(retainBackupDataForDays, retainBackupDataForCycles * 7) - retainBackupDataFo
FROM retention_rules
WHERE retainBackupDataForDays > 0
  AND retainBackupDataForCycles > 0
  AND enableDataAging = 1
ORDER BY ExtraRetention DESC;
```

Query 3: Storage Space Risk Assessment

```
-- Plans with short days but multiple cycles (high bloat risk)
SELECT
    parentName AS PlanName,
    retainBackupDataForDays AS Days,
    retainBackupDataForCycles AS Cycles,
    CASE
        WHEN retainBackupDataForCycles > 2 THEN 'HIGH RISK'
        WHEN retainBackupDataForCycles = 2 THEN 'MEDIUM RISK'
        ELSE 'LOW RISK'
    END AS StorageRisk
FROM retention_rules
WHERE retainBackupDataForDays <= 30
  AND retainBackupDataForCycles >= 1
  AND enableDataAging = 1
ORDER BY retainBackupDataForCycles DESC, retainBackupDataForDays;
```

Next Steps for Analysis

1. ✅ Run queries against retention_rules table
2. ⏳ Collect job schedule data (need to implement)
3. ⏳ Identify plans with backup job failures
4. ⏳ Calculate actual vs expected storage reclamation
5. ⏳ Generate conflict report with recommendations

Key Metrics to Track

Metric	Good	Warning	Critical
Cycle Retention	1	2	3+

Metric	Good	Warning	Critical
Days Retention	≤30	31-90	>90
Effective Retention	≤30 days	31-60 days	>60 days
Extra Days from Cycles	0-7	8-21	>21
Backup Success Rate	>95%	85-95%	<85%

References

- Commvault Documentation: Data Aging Troubleshooting
- Commvault Community: Aging old backup due to full storage
- Commvault Documentation: Schedule Data Aging Jobs