**Performance Tuning for High Availability using Transactional Replication**

**(SQL Server 2022)**

**1) Business Scenario (CSC-style)**

**System:** CSC Compliance Management (OLTP)

**Need:** Keep a downstream system updated in near real-time:

- Read-only reporting DB
- DR/standby integration feed
- Third-party downstream consumer

**Topology:**

- **Publisher:** CSC_OLTP (primary app DB)
- **Distributor:** CSC_DIST (central distribution DB)
- **Subscriber:** CSC_REPORTING (read-only/reporting)

**Incident symptoms (very common):**

- "Reporting data is 30–90 minutes behind"
- Undistributed Commands grows
- Log file on publisher grows unexpectedly
- Distribution DB grows fast
- High CPU on Distributor
- High latency during month-end peaks

---

**2) Replication Internals (Performance Lens)**

Transactional replication pipeline:

1. **Log Reader Agent**

- Reads Publisher transaction log
- Converts changes into replication commands
- Writes into **distribution DB** (MSrepl_commands, MSrepl_transactions)

2. **Distribution Agent**

- Reads distribution DB commands
- Applies to Subscriber
- Affected by network/latency/indexing/subscriber locks

**Where performance breaks (3 choke points)**

| Layer | What breaks | Symptoms |
|---|---|---|
| Publisher / Log Reader | Log can't be scanned fast enough | Log grows, undistributed commands rise |
| Distributor | Distribution DB hot + MSrepl tables overloaded | Distributor CPU/IO spikes |
| Subscriber / Distribution Agent | Apply can't keep up | Latency increases even if log reader is fine |

---

**3) Golden Metrics (What DBAs must measure)**

You always measure 4 things:

**A) End-to-end latency**

- **Tracer tokens** (best "true latency" indicator)

**B) Backlog volume**

- Undistributed commands / transactions

**C) Agent health + throughput**

- Log Reader: "commands/sec"
- Distribution Agent: "delivered commands/sec"

**D) Bottleneck resources**

- CPU/IO on Distributor
- Network RTT between Distributor ↔ Subscriber
- Blocking on Subscriber

---

**4) CSC DBA – "Single Pane" Troubleshooting Queries (Copy/Paste)**

**4.1 Tracer token latency (End-to-end truth)**

Run on **Publisher** (in publication DB context usually okay):

```
-- Insert a tracer token to measure latency
EXEC sys.sp_posttracertoken @publication =
N'CSC_Compliance_Pub';
GO

-- View tracer token history (publisher -> distributor ->
subscriber)
```

```
EXEC sys.sp_helptracertokenhistory
    @publication = N'CSC_Compliance_Pub';
GO
```

**How to explain output to DBAs**

- Publisher → Distributor time = Log Reader health
- Distributor → Subscriber time = Distribution Agent / Subscriber apply health

---

### 4.2 Identify backlog / undistributed commands

Run on **Distributor**:

```
-- Backlog summary by subscription
EXEC sys.sp_replcounters;
GO
```

Typical counters to watch:

- Undistributed Commands
- Undistributed Transactions
- Delivery Latency

---

### 4.3 Agent status + last errors (Distributor)

```
-- Distribution agent history
SELECT TOP (50)
    h.time,
    h.runstatus,
    h.comments,
    h.delivered_commands,
    h.delivery_rate,
    h.error_id
FROM distribution.dbo.MSdistribution_history h
ORDER BY h.time DESC;
GO

-- Log reader history
SELECT TOP (50)
    h.time,
    h.runstatus,
    h.comments,
    h.delivered_commands,
    h.delivery_rate,
```

```
    h.error_id
FROM distribution.dbo.MSlogreader_history h
ORDER BY h.time DESC;
GO
```

**Meaning**

- delivery_rate near zero with growing backlog = your choke point
- frequent errors / retries = network/subscriber constraints

---

### 4.4 Find "hot articles" (which table is flooding replication)

Run on **Distributor**:

```
-- Identify articles contributing most commands (approx by
joining transaction/command tables)
SELECT TOP (20)
    a.article,
    COUNT_BIG(*) AS approx_commands
FROM distribution.dbo.MSrepl_commands c
JOIN distribution.dbo.MSarticles a
    ON c.article_id = a.artid
GROUP BY a.article
ORDER BY approx_commands DESC;
GO
```

**Learning**

- You'll usually find 1–3 tables causing 80% of pressure (e.g., Audit/Log tables).

---

## 5) Performance Tuning Playbook (CSC DBA Decision Tree)

### Step 1: Is the bottleneck Log Reader or Distribution Agent?

Use tracer tokens:

- If Publisher→Distributor is high ⇒ **Log Reader / Publisher log IO**
- If Distributor→Subscriber is high ⇒ **Distribution Agent / network / subscriber**

---

## 6) Tuning: Publisher + Log Reader (When Publisher→Distributor is slow)

### A) Fix publisher log & IO

**Best practices**

- Put **publisher LOG file** on fastest storage
- Avoid huge transactions (batch writes)
- Don't run heavy log backups that compete (coordinate schedules)

### B) Increase Log Reader throughput (agent profile)

In Replication Monitor, use a "High Volume" profile or adjust parameters:

Common tuning knobs:

- -ReadBatchSize
- -ReadBatchThreshold
- -MaxCmdsInTran (careful)

**Conceptual guidance**

- Increase batch sizes to reduce overhead, but avoid memory spikes.

---

## 7) Tuning: Distributor (When distribution DB is the choke point)

Distributor is often the real bottleneck.

### A) Place distribution DB on fast IO

- **distribution data + log** on SSD / high IOPS
- Pre-size files (avoid autogrowth thrash)

**B) Index & cleanup health**

- Ensure distribution cleanup job is healthy
- Tune retention (don't keep too much history)

Key jobs to check:

- "Distribution clean up"
- "Agent history clean up"

**C) Don't overload distributor**

- Avoid colocating heavy SSIS/reporting on distributor
- Avoid antivirus scanning on distribution file paths

---

**8) Tuning: Subscriber + Distribution Agent (When Distributor→Subscriber is slow)**

**A) Apply performance: subscriber indexing**

- Too many indexes on subscriber can slow apply
- But missing indexes can cause downstream query pressure and blocking

**B) Reduce blocking on subscriber**

Check blocking quickly on subscriber:

```
SELECT
    r.session_id,
    r.status,
    r.wait_type,
    r.wait_time,
    r.blocking_session_id,
    t.text
FROM sys.dm_exec_requests r
CROSS APPLY sys.dm_exec_sql_text(r.sql_handle) t
WHERE r.session_id > 50
```

```
ORDER BY r.wait_time DESC;
GO
```

**C) Distribution agent knobs (apply faster)**

- -CommitBatchSize
- -CommitBatchThreshold
- -SubscriptionStreams (parallel apply; great when many independent commands)

**Rule of thumb**

- **SubscriptionStreams** helps when backlog is huge and subscriber can handle parallelism.

---

**9) COMPLETE HANDS-ON LAB (Instructor-friendly)**

**Lab Goal**

Simulate:

1. normal replication
2. backlog buildup
3. diagnose whether log reader or distribution agent is bottleneck
4. tune agent settings
5. verify latency improvements

**Lab Setup (assumes replication already configured)**

**Window A (Workload generator on Publisher):**

```
USE CSC_OLTP;
GO

-- Simulate OLTP surge: many status changes
DECLARE @i INT = 0;

WHILE @i < 20000
BEGIN
```

```
    UPDATE TOP (50) dbo.ComplianceFilings
    SET filing_status = CASE WHEN filing_status = 'OK' THEN
'OVERDUE' ELSE 'OK' END
    WHERE due_date < DATEADD(DAY, -(@i % 30), CAST(GETDATE()
AS DATE));

    SET @i += 1;
END
GO
```

**Window B (DBA monitoring on Distributor):**

```
EXEC sys.sp_replcounters;
GO

EXEC sys.sp_helptracertokenhistory
    @publication = N'CSC_Compliance_Pub';
GO
```

**Induce Subscriber slowness (optional lab trick)**

Run a heavy report on Subscriber to create apply contention:

```
USE CSC_REPORTING;
GO
-- Example heavy query
SELECT COUNT_BIG(*)
FROM dbo.ComplianceFilings cf
JOIN dbo.Entities e ON e.entity_id = cf.entity_id
GROUP BY e.country_code;
GO
```

**What DBAs should observe**

- Backlog rises
- Tracer token Distributor→Subscriber latency increases
- MSdistribution_history delivery_rate drops

---

**10) "Complete CSC DBA Script Pack" (Single File Structure)**

Use this structure in your Module-7/8 style demo packs:

**A) Baseline snapshot**

- sp_replcounters
- msdistribution_history top rows
- tracer token history

**B) Workload generator**

- update burst loop

**C) Bottleneck identification**

- tracer tokens (split latencies)
- agent histories
- top articles by command count

**D) Tuning actions checklist**

- agent profile changes
- commit batch tuning
- streams tuning
- distribution cleanup verification

**E) Post-fix validation**

- tracer token latency improvement
- backlog decrease rate

---

**11) CSC DBA Replication Performance Checklist (Print-ready)**

**Publisher**

- Log disk latency low
- No mega-transactions
- Log backups scheduled smartly

**Distributor**

- distribution DB pre-sized
- Cleanup jobs healthy
- IO fast; no other heavy workloads

**Subscriber**

- No blocking from reports
- Apply parallelism tuned
- Index strategy balanced

**Monitoring**

- Tracer tokens used during incidents
- sp_replcounters tracked hourly during peaks
- Agent history checked for retry loops

---

**12) What CSC DBAs should say on a bridge call**

- "Latency is mainly **Publisher→Distributor**, indicating Log Reader or publisher log IO."
- "Latency is mainly **Distributor→Subscriber**, indicating apply throughput or subscriber contention."
- "Undistributed commands are growing at X/min; delivery rate is Y/sec, so we're falling behind."
- "We'll tune Distribution Agent batching / streams and validate using tracer tokens."

---