Standard Operating Procedure (SOP) for configuring and managing **CTP (Cost Threshold for Parallelism)** on Microsoft SQL Server, aligned with industry best practices for **SQL Server 2017 through SQL Server 2025**.
This SOP is designed for use in an enterprise environment and assumes familiarity with SQL Server configuration, performance monitoring, and change control processes.

Standard Operating Procedure

Title: Configuration and Management of Cost Threshold for Parallelism (CTP)
System: Microsoft SQL Server 2017 through SQL Server 2025
Scope: All SQL Server instances within the enterprise
Owner: Database Administration Team
Effective Date: 25th December, 2025 - Thursday
Revision: 1.0
Approvals: [DBA Lead / IT Manager / QA / Security]

1. Purpose

This SOP defines the process for configuring, reviewing, monitoring, and adjusting the **Cost Threshold for Parallelism (CTP)** setting on Microsoft SQL Server instances. The goal is to ensure consistent application of performance best practices, improve query performance, and maintain system stability across supported SQL Server versions (2017–2025).

2. Definitions

Term	**Definition**
CTP (Cost Threshold for Parallelism)	A SQL Server configuration option that specifies the threshold at which the query optimizer considers parallel execution plans.
MAXDOP	Maximum Degree of Parallelism — the maximum number of CPUs used in parallel plan execution.
DMV	Dynamic Management View — server-side views reporting internal SQL Server state.
Baseline	A performance snapshot used for comparison over time.

3. Applicability

This SOP applies to all SQL Server instances in production, pre-production, QA, and staging environments. It is intended for use by Database Administrators tasked with performance tuning and configuration management.

4. Responsibilities

Role	**Responsibilities**
DBA Team	Implement and maintain CTP settings per this SOP, monitor performance impacts, document changes.
DBA Lead	Approve changes and ensure adherence to procedures.
Change Control Board (CCB)	Review and approve configuration change requests.

5. Preconditions

Before adjusting CTP:

- The SQL Server instance must be documented in the configuration management database (CMDB).
- Performance baselines must be available from monitoring tools (e.g., Query Store, PerfMon, Extended Events).
- Recent workload patterns must be analyzed to justify threshold changes.

6. Tools Required

Tool	Purpose
SQL Server Management Studio (SSMS)	Execute T-SQL and inspect server state.
Query Store	Analyze query performance over time.
Performance Monitoring Tools	PerfMon / Extended Events / third-party monitoring.
PowerShell with SqlServer module	Optional automation.

7. Procedure

7.1 Establish Baseline Performance

1. Document current CTP and MAXDOP values:
2. EXEC sp_configure 'cost threshold for parallelism';
3. EXEC sp_configure 'max degree of parallelism';
4. Capture recent workload data using Query Store:

```
SELECT
    qsqt.query_sql_text,
    SUM(qsrs.count_executions) AS total_executions,
    SUM(qsrs.avg_duration * qsrs.count_executions) / NULLIF(SUM(qsrs.count_executions), 0) AS avg_duration
FROM sys.query_store_runtime_stats AS qsrs
JOIN sys.query_store_plan AS qsp ON qsrs.plan_id = qsp.plan_id
JOIN sys.query_store_query AS qsq ON qsp.query_id = qsq.query_id
JOIN sys.query_store_query_text AS qsqt ON qsq.query_text_id = qsqt.query_text_id
GROUP BY qsqt.query_sql_text
ORDER BY avg_duration DESC;
```

5. Analyze performance metrics (CPU, duration, waits) around high-cost queries.

7.2 Determine Appropriate CTP Value

1. The **default** CTP is 5. However, default should rarely be used in production.
2. Typical industry practice sets CTP between **25 and 50** for OLTP workloads; **50 to 100+** where complex queries dominate.
3. Review system characteristics:
   - High CPU contention → consider raising CTP.
   - Frequent inappropriate parallel plans → raise CTP gradually.
   - Low throughput with many single-threaded queries → lower CTP carefully.
4. Document the targeted CTP value and justification before implementation.

7.3 Change Implementation

1. Generate the T-SQL to apply the new CTP value:

```
EXEC sp_configure 'cost threshold for parallelism', 50;
RECONFIGURE;
```

2. Validate that the change is included in the upcoming maintenance window or change control ticket.
3. Execute the change during approved maintenance window.
4. After applying, verify the setting:

```
EXEC sp_configure 'cost threshold for parallelism';
```

7.4 Post-Change Monitoring

1. For at least **48–72 hours**:
   - Monitor performance counters: CXPACKET, SOS_SCHEDULER_YIELD, CPU utilization.
   - Review Query Store for regressions or shifts in execution plans.
2. Use baselines captured earlier to determine whether backsliding is necessary.
3. Document performance improvements or regressions.

8. Adjustment Guidelines

Scenario	Recommended Action
Increased query duration on OLTP workloads	Consider lowering CTP slightly (incremental adjustments)
CPU starvation due to excessive parallelism	Consider raising CTP
Mixed workload with inconsistent behavior	Adjust and model against baselines

Adjustment increment:

- Change CTP by **5–10 at a time**, not large leaps.

9. Audit and Review

1. Conduct quarterly configuration reviews.
2. Compare CTP setting against performance trends.
3. Log all CTP adjustments in change control logs.

10. Rollback Procedure

If a change adversely impacts performance:

1. Identify prior known good CTP value.
2. Revert configuration:
3. EXEC sp_configure 'cost threshold for parallelism', <previous_value>;
4. RECONFIGURE;
5. Monitor system to ensure stability.

11. Compliance and Reporting

All changes must:

- Be documented in the change control system.
- Include justification and test results.
- Reference performance baselines.

Produce periodic compliance reports:

- Current CTP values across all environments
- Trend analysis of query performance
- Exceptions and remediations

12. References

Reference	Description
Microsoft Docs	Official documentation for SQL Server CTP and MAXDOP
Industry Best Practices	DBA performance tuning guidelines

13. Revision History

Revision	Date	Description	Author
1.0	[Date]	Initial draft and release	DBA Team

Structured guide for **calculating and setting the Cost Threshold for Parallelism (CTP)** in **OLTP** and **OLAP** environments, aligned with SQL Server best practices (2017–2025). I'll include **step-by-step methodology, formulas, and practical recommendations**.

1. Understanding CTP

- **CTP** determines the **estimated query cost** at which SQL Server will consider using a **parallel execution plan**.
- Cost is measured in **query optimizer units** (based on estimated CPU, I/O, and memory).
- Default **CTP = 5**, which is **too low for most production workloads** and may cause excessive parallelism.

Key principle:

- **Low CTP** → More queries run in parallel → High CPU usage → Potential context switching and CXPACKET waits.
- **High CTP** → Only expensive queries run in parallel → Better CPU utilization → Single-threaded small queries run efficiently.

2. Methodology to Calculate CTP

Step 1: Collect workload metrics

1. Enable **Query Store** if not already enabled:

```
ALTER DATABASE [YourDB] SET QUERY_STORE = ON;
ALTER DATABASE [YourDB] SET QUERY_STORE (OPERATION_MODE = READ_WRITE);
```

2. Identify **top resource-consuming queries**:

```
SELECT TOP 20
  qsqt.query_sql_text,
  SUM(qsrs.count_executions) AS exec_count,
  SUM(qsrs.avg_duration * qsrs.count_executions) / NULLIF(SUM(qsrs.count_executions),0) AS
avg_duration_us
FROM sys.query_store_runtime_stats AS qsrs
JOIN sys.query_store_plan AS qsp ON qsrs.plan_id = qsp.plan_id
JOIN sys.query_store_query AS qsq ON qsp.query_id = qsq.query_id
JOIN sys.query_store_query_text AS qsqt ON qsq.query_text_id = qsqt.query_text_id
GROUP BY qsqt.query_sql_text
ORDER BY avg_duration_us DESC;
```

3. Identify queries that **run frequently and with low duration** (small OLTP queries) and **large queries with high duration** (OLAP queries).

Step 2: Define thresholds

- **Cost units** are **query optimizer estimated costs**.
- Query optimizer exposes the estimated cost in **Execution Plan properties**.
- Steps:

1. For OLTP:
   - Find queries with **execution time < 500 ms** and **high frequency**.
   - Calculate average **estimated cost** (visible in SSMS execution plan or Query Store).
   - Example: Most small queries cost 10–20 units → set CTP **higher than their cost** to avoid unnecessary parallelism.
2. For OLAP:
   - Large analytical queries may cost **hundreds to thousands of units**.
   - Set CTP **slightly below typical large query costs** to allow parallel execution.

Step 3: Formula / heuristic for initial CTP

Environment	Initial CTP (units)	Rationale
OLTP	25–50	Avoid parallelism for short, frequent queries; improve concurrency.
Mixed	35–60	Balance between OLTP responsiveness and analytical queries.
OLAP	50–100+	Allow expensive queries to run in parallel for faster completion.

Note: Always tune based on **Query Store execution plan cost**.

Step 4: Apply CTP in SQL Server

1. Check current CTP:
   ```
   EXEC sp_configure 'cost threshold for parallelism';
   ```
2. Apply new CTP:
   ```
   EXEC sp_configure 'cost threshold for parallelism', 50;  -- Example
   RECONFIGURE;
   ```
3. Verify:
   ```
   EXEC sp_configure 'cost threshold for parallelism';
   ```

Step 5: Monitoring and fine-tuning

1. Monitor **wait stats**:
   ```
   SELECT wait_type, wait_time_ms, waiting_tasks_count
   FROM sys.dm_os_wait_stats
   WHERE wait_type LIKE 'CXPACKET';
   ```
2. Observe **CPU utilization**:
   ```
   SELECT sqlserver_start_time, cpu_count, hyperthread_ratio, numa_node_count
   FROM sys.dm_os_sys_info;
   ```
3. Adjust CTP in **increments of 5–10** units based on:
   - High CXPACKET waits → increase CTP
   - Underutilized CPU / long-running large queries → decrease CTP

Step 6: OLTP vs OLAP Guidance

Feature	OLTP Recommendation	OLAP Recommendation
Short queries (<500 ms)	Avoid parallelism → CTP higher	N/A
Long queries (>5 s)	Can allow parallelism	Encourage parallelism → lower CTP
CPU utilization	Keep CPU spikes low	Maximize CPU usage for batch queries
MAXDOP	Follow NUMA & CPU guidance (1–8 cores)	Can be higher if many cores available
Tuning frequency	Quarterly or on workload change	After batch schedule or quarterly review

Step 7: Automating CTP Calculation (Optional)

- Use Query Store to **compute 90th percentile query cost**:
  ```
  SELECT
     PERCENTILE_CONT(0.9) WITHIN GROUP (ORDER BY avg_cpu_time)
        OVER () AS cost_90th
  FROM sys.query_store_runtime_stats;
  ```

- Set **CTP slightly below the 90th percentile** to allow most expensive queries to go parallel while keeping frequent small queries single-threaded.

Step 8: Documentation & Change Control

- Always document:
    - Old and new CTP values
    - Date, DBA, environment
    - Monitoring metrics
    - Justification for change
- Apply changes through **change management** with rollback plan.

Concise summary for **calculating and setting CTP** in OLTP vs OLAP environments, without a script:

OLTP Environment (Transactional)

- **Goal:** Minimize parallelism for short, frequent queries to reduce CPU contention and context switching.
- **Step 1:** Identify high-frequency, short-duration queries using Query Store or execution plans.
- **Step 2:** Determine average **query cost** of these queries.
- **Step 3:** Set **CTP** slightly above these query costs (typical: **25–50**).
- **Step 4:** Monitor **CXPACKET waits** and CPU usage; adjust CTP in increments of 5–10 if necessary.
- **Step 5:** Ensure MAXDOP aligns with NUMA/core guidance (usually 1–8 for OLTP).

OLAP Environment (Analytical/Batch)

- **Goal:** Allow parallelism for large, long-running queries to reduce execution time.
- **Step 1:** Identify long-duration, high-cost queries.
- **Step 2:** Determine average **query cost** of these queries.
- **Step 3:** Set **CTP** slightly below the 90th percentile of expensive queries (typical: **50–100+**).
- **Step 4:** Monitor **CPU utilization** and query completion times.
- **Step 5:** MAXDOP can be higher if sufficient cores are available (follow NUMA guidance).

General Best Practices

1. Always **incrementally adjust** CTP; avoid large jumps.
2. Use **Query Store** to validate that small queries remain single-threaded.
3. Document all changes with baseline metrics.
4. Monitor **wait stats** (CXPACKET, SOS_SCHEDULER_YIELD) after any change.
5. Review settings quarterly or after workload changes.

This method ensures **balanced parallelism**, prevents CPU overload in OLTP, and accelerates analytical queries in OLAP while maintaining system stability.

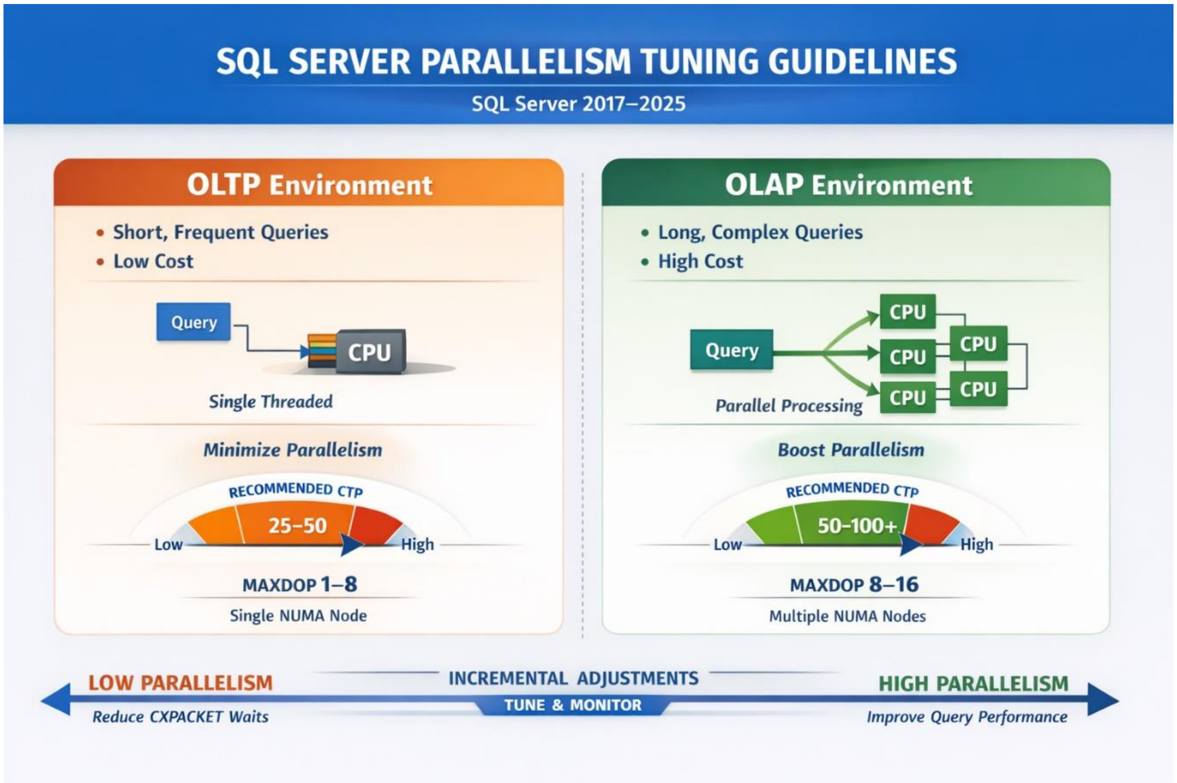
Guideline table for **CTP and MAXDOP settings** for OLTP vs OLAP workloads on SQL Server 2017–2025.
This is based on industry best practices and Microsoft recommendations.

Environment	Query Type	Typical Query Cost (Optimizer Units)	Recommended CTP	MAXDOP Guidance	Notes
OLTP	Short, frequent (<500 ms)	5–20	25–50	1–8 (follow NUMA & CPU)	Avoid parallelism for small queries to reduce CXPACKET waits.
OLTP	Mixed/medium	20–50	30–60	1–8	Incrementally adjust based on CPU and wait stats.
OLAP	Long-running, high-cost (>5 s)	50–500+	50–100+	Up to number of physical cores per NUMA node, max 16 per Microsoft guidance	Encourage parallelism to improve query performance.
Mixed	Varied workloads	25–100	35–60	1–16	Balance between OLTP responsiveness and analytical queries.

Key Recommendations

1. **Incremental Changes:** Adjust CTP in steps of 5–10 units, monitor CPU and wait stats.
2. **Use Query Store:** Determine 90th percentile query cost for large queries as a reference point.
3. **MAXDOP:** Always follow NUMA node guidance:
   - Single NUMA node → MAXDOP ≤ 8
   - Multiple NUMA nodes → MAXDOP ≤ 16 or cores per NUMA node, whichever is smaller.
4. **Monitor:** Check CXPACKET and SOS_SCHEDULER_YIELD waits after CTP changes.
5. **Document:** Capture baseline, CTP changes, MAXDOP, and query performance metrics.

This table gives a **quick reference for tuning** SQL Server parallelism according to workload type while maintaining best practices.


SQL SERVER PARALLELISM TUNING GUIDELINES
SQL Server 2017–2025
OLTP Environment
• Short, Frequent Queries
• Low Cost
Query
CPU
Single Threaded
Minimize Parallelism
RECOMMENDED CTP
25–50
Low
High
MAXDOP 1–8
Single NUMA Node
OLAP Environment
• Long, Complex Queries
• High Cost
Query
CPU
CPU
CPU
CPU
CPU
Parallel Processing
Boost Parallelism
RECOMMENDED CTP
50–100+
Low
High
MAXDOP 8–16
Multiple NUMA Nodes
LOW PARALLELISM
Reduce CXPACKET Waits
INCREMENTAL ADJUSTMENTS
TUNE & MONITOR
HIGH PARALLELISM
Improve Query Performance