**TUNING TOOLS**

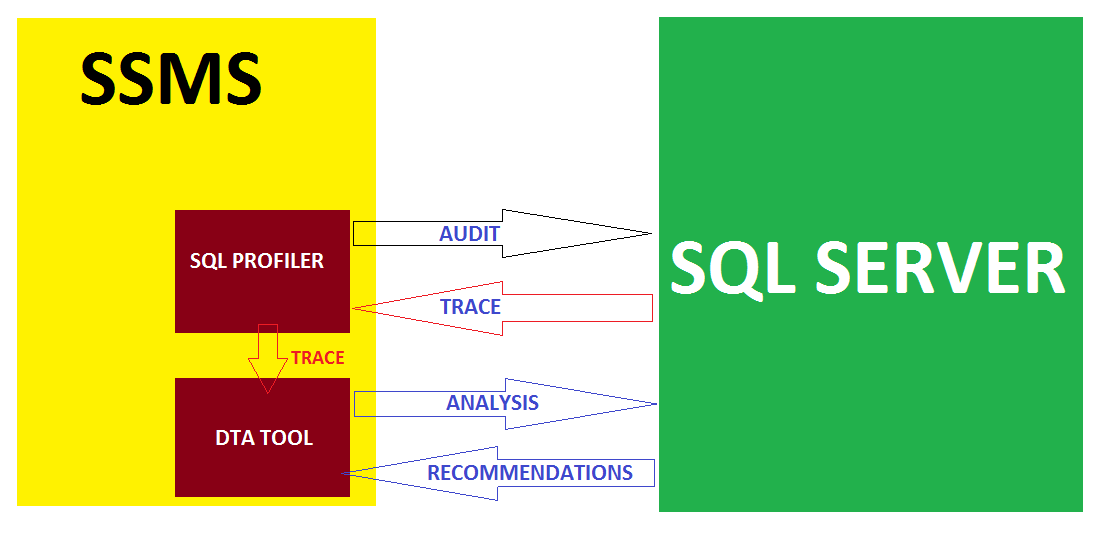
STEP 1: LAUNCH SQL SERVER PROFILER TOOL & AUDIT THE QUERIES. TEMPLATE = TUNING TEMPLATE.

SAVE THE QUERIES INTO A **TRACE** FILE = **WORKLOAD FILE**.

STEP 2: LAUNCH DATABASE ENGINE TUNING ADVISOR TOOL AND INPUT ABOVE TRACE FILE.

START THE ADVISOR TOOL.

THIS GIVES A SET OF RECOMMENDATIONS ON USAGE OF PARTITIONS, INDEXES, STATISTICS.

****

ALL RECOMMENDATIONS GIVEN BY DTA TOOL SHOULD BE EXECUTED IN THE SAME ORDER.

THESE INDEX RECOMMENDATIONS ARE ACTUALLY GENERATED BY QUERY OPTIMIZER COMPONENT.

THIS DTA TOOL IS NOT APPLICABLE FOR MEMORY TABLES (MEMORY OPTIMIZED TABLES).

**CAUTION:**

PREFER USING PROFILER TOOL AND DTA TOOL DURING NON-BIZ HOURS.

PREFERABLY DURING MIDNIGHT OR LATE EVENING WHEN THE DATABASE TRAFFIC IS COMPARATIVELY LOW.

REASON: PROFILER & DTA TOOL DEMANDS ADDITION CONNECTIONS TO THE SERVER. THIS INVOLVES MORE USAGE OF MEMORY & TEMPDB SPACE.

**EXECUTION PLAN ANALYSIS**

**EXECUTION PLAN TYPES:**

1. ESTIMATED EXECUTION PLAN : PLAN GENERATED BEFORE QUERY EXECUTION

2. LIVE EXECUTION PLAN : PLAN GENERATED DURING QUERY EXECUTION

3. ACTUAL EXECUTION PLAN : PLAN GENERATED AFTER QUERY EXECUTION

PURPOSE: USING EXECUTION PLANS, WE IDENTIFY MISSING INDEXES AND CREATE THEM. ALSO ANALYSE QUERY COSTS.

**EXECUTION PLAN ANALYSIS:**

**1. PHYSICAL SCAN:** REFERS TO DISK LEVEL READS & WRITES. INVOLVES MORE TIME, COST.

**2. LOGICAL SCAN:** REFERS TO MEMORY LEVEL READS & WRITES. FASTER, CHEAPER.

**3. ESTIMATED EXECUTION MODE:** REFERS TO THE STORAGE ENGINE ARCHITECTURE FOR A GIVEN TABLE. ROW LEVEL STORAGE OR COLUMN LEVEL STORAGE.

4. **ACTUAL NUMBER OF ROWS** : REFERS TO THE NUMBER OF ROWS TO BE PROCESSED.

**5. QUERY COST TYPES:**

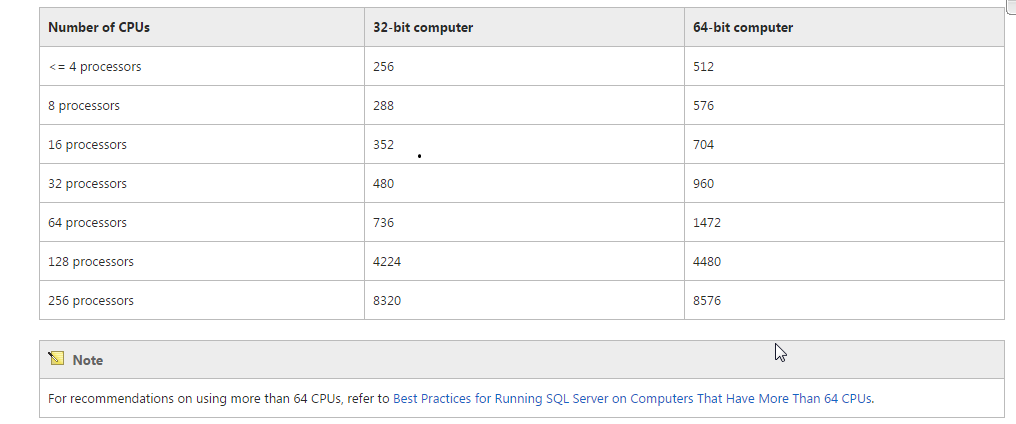
1. **IO COST** : REFERS TO DISK IO FOR READS & WRITES.

FOR HIGHER IO COST : IDENTIFY, RESOLVE MISSING INDEXES @ **DTA**

2. **CPU COST**: REFERS TO THREAD MANAGEMENT FACTOR BASED ON "NUMA" NODES ON THE PROCESSOR. ADDING NEW PROCESSOR NODE=HOT CPU

FOR HIGHER **CPU COST** VALUES SET PROPER THREAD COUNT.

SQL SERVER > PROPERTIES > PROCESSOR > set Processor **Thread Count**

****

3. **SUB TREE COST**: REFERS TO COST INVOLVED IN ANALYSING PARSE, COMPILE PLANS.

**RECOMPILE** (REFRESH) THE DATABASE OBJECTS [LIKE SPs, VIEWS, ETC.].

SP\_RECOMPILE ‘PROCEDURENAME’

SP\_REFRESHVIEW ‘VIEWNAME’

**4. OPERATOR COST**: REFERS TO KEYWORDS & OPERATIONS WITHIN SQL QUERIES.

FOR HIGHER OPERATOR COST, UPDATE **STATISTICS @ MAINT" PLANS**

IF QUERY INVOLVES MORE VOLUME DATA, SPOOLING TAKES PLACE.

**SPOOLING : TO LOAD DATA FROM MEMORY & DATA FILE TO TEMPDB.**

HIGHER SPOOLING ALSO REQUIRES INDEX CREATION & STATS UPDATES.

TYPES OF SPOOLING:

**LAZY SPOOL:** DATA IS LOADED INTO TEMPDB AND THEN QUERY EXECUTION STARTS. QO USES THIS FORMODERATE VOLUME OF DATA.

**EAGER SPOOL:** DATA IS LOADED INTO TEMPDB AND THE QUERY GETS EXECUTED AT THE SAME TIME. QO USES THIS FOR HUGE VOLUME OF DATA. PREFERABLY IN GBs, TBs.

**FINAL TUNING CHECKLIST:**

1. ANALYSE THE QUERIES USING ESTIMATED EXECUTION PLANS. IMPLEMENT INDEXES

2. CONSIDER TO USE PARTITIONS IF POSSIBLE (FOR HISTORICAL DATA)

3. CONSIDER TO IMPLEMENT COMPRESSIONS IF POSSIBLE.

4. ENSURE TIMELY REORGANIZATION OF INDEXES

5. ENSURE TIMELY UPDATE OF STATISTICS

6. IMPLEMENT DTA TOOL [TRACE FILE / TRACE TABLE / PLAN CACHE / QUERY STORE]

7. ANALYSE QUERY EXECUTION PLANS & QUERY COSTS

8. IF SPOOLING IS INVOVLED, THEN INCREASE THE SIZE OF TEMPDB.

ALTER DATABASE TEMPDB MODIFY FILE (NAME = ‘TEMPDEV’, SIZE = 1250 GB)