Optimizing Performance for SQL Based Applications

Lab 2 – Optimizing Queries

Overview

While investigating the general slowness of a SQL Server database, you have come across some workloads that are running slowly. You decide to analyze execution plans for those workloads. In this lab, you will analyze execution plans and identify plan issues. You will then fix them to improve execution performance of workloads.

Before starting this lab, you should view **Module 2 – Optimizing Queries** in the course *Optimizing Performance for SQL Based Applications*. Then, if you have not already done so, follow the instructions in the **Getting Started** document for this course to set up the lab environment.

Please note that the setup script might take up to 15 minutes to run.

What You'll Need

To complete the labs, you will need the following:

- A SQL Server instance with the AdventureWorksLT sample database. Review the Getting Started document for information about how to provision this.
- The lab files for this course

Challenge 1: Analyzing Execution Plans

In this exercise, you will analyze execution plans.

Prepare the Lab Environment

1. Start Microsoft SQL Server Management Studio and connect to your database instance.
2. Download and open **Lab 02 - Setup.sql** from the **Setup** folder.
3. Execute the script in **Lab 02 - Setup.sql**.
4. This script will take a long time to run.

Collect an Actual Execution Plan

1. Download and open **Lab 02 - tuning 1.sql** from the **Setup** folder.
2. Select the query under the comment which begins **Task 1**. On the **Query** menu, click **Include Actual Execution Plan**.
3. Click **Execute**. Note the execution time.
4. In the **Results** pane, on the **Execution plan** tab, right-click the graphical execution plan, and then click **Save Execution Plan As**.
5. In the Save As dialog box, save the file as **plan1.sqlplan** on your desktop.
6. On the **Execution plan** tab, scroll to the far right-hand side of the actual query plan. Position the cursor over the query plan operator in the top right-hand corner, the name of which starts **Clustered Index Scan (Clustered)**. In the pop-up that appears, notice the difference between Estimated Number of Rows and Actual Number of Rows.

   This type of issue is caused by a poor estimate of cardinality. The table statistics indicate that the tables involved have very few rows, but in reality, the row count is much higher. Updating statistics for the tables involved will improve performance.

Rebuild Table Statistics

1. Select the query under the comment which begins **Task 2** and click **Execute**.

Compare the Execution Plan

1. Select the query under the comment which begins **Task 1**. On the **Query** menu, ensure **Include Actual Execution Plan** is selected.
2. Click **Execute**. Note the execution time.
3. On the **Execution plan** tab, scroll to the far right-hand side of the actual query plan. Position the cursor over the query plan operator in the top right-hand corner, the name of which starts **Clustered Index Scan (Clustered)**. In the pop-up that appears, notice the similarity between Estimated Number of Rows and Actual Number of Rows.

   Note that the estimated and actual row counts now match almost exactly. The query will execute faster than it did previously.

   The execution plan includes a suggestion for an index that would further improve query performance.

4. On the **Execution plan** tab, right-click the graphical execution plan and click **Compare Showplan**.
5. In the **Open** dialog box, browse to your desktop, select **plan1.sqlplan** and click **Open**. The new and old query execution plans will open side-by-side.
6. When you have finished your comparison, close the **Showplan Comparison** tab.