Boosting SQL Server Performance Cheat Sheet

by [deleted] via cheatography.com/2754/cs/6194/

Introduction

When working with SQL Server, there are correct and incorrect ways to set up, configure and maintain systems. Traditionally, systems are expected to work properly after the initial set-up. However, this is rarely the case.
A system’s performance decreases over time, often due to unforeseen stress loads imposed on the system. These stress loads include increased data volumes, variety and velocity levels that exceed the initial design of the hardware.
Here are five performance techniques and best practices designed to successfully aid DBAs in their system implementations:

DATA & LOG FILES

Data and log files are the fundamental building blocks for any DBA activity. Begin by placing data and log files in their proper locations. SQL Server has two distinct I/O patterns for data and log I/O. In general, data I/O patterns are random in nature, but SQL Server spends a great deal of time writing transaction logs sequentially. Therefore, as a best practice, data and log files should reside on separate drives to prevent interference with one another.

TUNING QUERIES

Once the I/O load is generated by SQL Server, developers should focus on tuning the queries. Often time-consuming, this process may involve a significant level of experience depending on the design of the application.

CHECKPOINTS

Checkpoints are designed to run periodically and are controlled by a recovery interval. You can measure the Checkpoint I/O by looking at the performance counter “Buffer Manager: Checkpoint pages/sec.” It is important to note, that developers will need to take this number and multiply it by eight and then divide by 1024 to get to Megabytes.

INDIRECT CHECKPOINTS

New to SQL Server 2012, indirect checkpoints provide configurable database-level alternatives to automatic checkpoints. In the event of a system crash, indirect checkpoints provide potentially faster, more predictable recovery time than automatic checkpoints.

DB Artisan

DBArtisan is a tool that can help with your turning. Learn more at http://www.embarcadero.com/

FILTERED STATISTICS

Filtered Statistics are computed over a subset of rows in a table, contain histograms, densities and tries. Filtered Statistics with highly selective predicates will be faster to build and update, and end up more accurate when they are created with a full scan sampling option.

LATCH CONTENTION FACTOR

Latch contention that hinders performance in OLTP environments is usually caused by high concurrency related to one or more of the following factors:

- **Schema Design & Access Patterns**
- **High Degree of Concurrency at the Application Level**
- **Layout of Logical Files Used by SQL Server Databases**
- **I/O Subsystem Performance**

Excessive page latch contention typically occurs in conjunction with a high level of concurrent requests from the application tier. There are certain programming practices that can also introduce a high number of requests for a specific page.
Whether you are a seasoned DBA or just starting out, these five techniques will add value to your overall performance optimization process.

MAXIMIZE YOUR DATABASE PERFORMANCE

As a leader in database administration, DBArtisan® arms you with the toolset necessary to monitor cross-platform DBMSs from a single UI while performing intuitive analytics to proactively identify issues paving the way for fast and efficient performance resolutions.