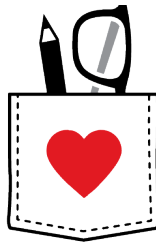


SQL Server Critical Care: A Sample Case



BRENT OZAR
UNLIMITED

What's it like to work with us in a SQL Critical Care® session?

In three or four days we identify your pain points, collect health indicators from your database environment and subsystems, and map out an action plan to address your pains. Along the way we train your team to take the pulse and vital stats of your SQL Server. We deliver recommendations for changes you should make within the next week, month, and quarter. We couple this with customized training, prescribed and delivered for your team and environment.

Each SQL Critical Care® Session is different: we focus on your unique configuration and pain points. Here's the story of one client's case.

Pain Points

"We want to get more out of our SQL Server. We've made some improvements, but we think we can do better."

Our client came to us with some significant short term pains. Their users were experiencing daily application timeouts. Database administrators had made performance tuning changes to reduce overall CPU usage, but timeouts continued. The company's business plan included doubling their user load within the next year. Combined with the timeouts, this had everyone on edge.

Our client also needed to provide high availability within their local datacenter for SQL Server databases. They wanted help to identify the technology that best fit their applications and their team.

We set out in our SQL Critical Care® session to answer two big questions: Could their current SQL Server be tuned to keep up? And how should they architect their new environment?

Health Indicators

This client has the ability to make changes in their code to improve performance. Since they control the code, we worked through a four-day process to performance tune their SQL Server configuration and identify ways to improve database code.

"We want to get more out of our SQL Server."

On the first day we talked through the current storage configuration, OS configuration, and SQL Server configuration and bottlenecks. Several issues surfaced: we found very high lock waits and CPU contention. We walked through different ways to collect bottleneck information and showed how to interpret

these bottlenecks in the context of the storage and server configuration.

On the second day we dove into database schema, index configuration and query tuning. We showed how to isolate the tables involved in lock waits and identify the query text and plans involved in blocking.

We identified high lock waits caused by a query pattern gone bad.

The database administrators were already actively tuning their indexes, so we delved into advanced indexing techniques. We walked through examples of tuning indexes and shared tactics to refine their process and make it more powerful. We then identified the queries that were having the highest CPU impact. We used the query plans for those statements to design an index tuning plan specifically for those heavy-hitters.

Diagnosis

This client had a knowledgeable team of database administrators and developers ready to make changes to tune their SQL Server configuration and queries. Health indicators showed that their current hardware had some capacity to spare, if the right changes were made in database structure and query patterns.

To identify the best strategy for high availability we discussed Recovery Point Objectives and Recovery Time Objectives. We captured high level projections for the next three years for performance requirements and budgets for licensing, hardware, storage, and IT Operations.

On the third day of the health check we documented our findings and recommendations for short, medium, and long term actions to address the client's pain points. We compiled a training deck on advanced configuration and SQL Server tuning topics to support the recommendations and provide references for technical staff.

Action Plan

Short Term: One Week

This client needed to quickly address their locking problem. Individual lock waits were brief but nearly constant. The high cumulative lock overhead was contributing to high CPU usage and client timeouts. We identified two statements as the primary sources of blocking. These statements shared a single problematic pattern.

We coded and demonstrated three alternatives to the current query pattern in production. For each pattern, we showed how to analyze strengths and weaknesses of the pattern using multiple active threads against a test schema. We recommended the pattern that would allow the greatest concurrency while meeting application requirements. We also addressed new ways to solve this concurrency problem in SQL Server 2012.

**Recommended
changes eliminated
daily timeouts.**

Medium Term: One Month

Recommendations for the first month focused on index tuning and parallelism settings in SQL Server. These iterations were designed to further drive down CPU use, improve concurrency, and speed up expensive queries.

Long Term: One Quarter

Limiting licensing costs while enabling an upgrade to SQL Server 2012 and fast failover within the current datacenter were the highest long term priorities. The client was ready to invest significant software development resources in tuning applications and was in the process of purchasing a robust shared storage appliance. This combination of factors indicated they were a good fit for a new two-socket server with processors optimized for OLTP workloads. Windows Failover Clustering was identified as the best fit for their immediate high availability needs.

Follow Up: Patient Results

How did these recommendations turn out when implemented? We checked in with this client several weeks after the health check to find out. The team had successfully implemented changes to query patterns on the problematic statements that drastically reduced lock waits. The database administrators had also performed two iterations of index tuning and had begun adjusting parallelism settings.

These changes eliminated the daily application timeouts and critical short-term pain points had been resolved.

The team was moving toward a new hardware environment with confidence. With the current environment performing acceptably, they were no longer at risk of being forced into a rushed implementation and migration.

Does Your SQL Server Need SQL Critical Care®?

Contact us at <http://BrentOzar.com/contact>, or email help@BrentOzar.com.