

# Case Study

## Financial Services Business



## 65% time save on data collection

**Service Type:** SQL Consultation | Low Concurrency

**Duration:** 1 Month

### Summary

A critical report was working very slowly when 10 or more concurrent users loading that while it was supposed to work with 50 concurrent users. A complicated stored procedure which was the source of the issue got tuned and some other recommendations got applied.

### Requirements

There is a bank using a web application written by ASP.NET and SQL Server 2005 collecting electronic documents from its three main branches through an internal network. Along the time, the

## Why SQLMax?

### EXPERIENCE

- 1,900+ SQL Jobs, Stored Procedures, Functions and SSIS Packages Tuned
- 100+ DBAs and Developers Have Relied on Our SQLMax Library
- 400+ Consultations Performed
- 40+ Databases Designed and Developed

### SOLUTIONS

- Complete Set of SQL Server-Focused Solutions
- Comprehensive Technical Coverage

### SUPPORT

- Full-time Services: 365\*7\*24
- Dedicated Account Managers
- World-Class Customer Service

### OUR TEAM

- Experienced & Knowledgeable Dedicated Team
- Clear Focus on MS SQL Server
- 100% US-Based

requirement to transfer data directly from all 500 branches to the head quarter had been felt and that was the beginning of some scalability issues. The response time of the main report in the web application was unacceptable when more than 10 concurrent users asking for it concurrently. Considering the 500 users are supposed to enter data during 3 pm to 4 pm, it was natural having around 50 users requesting the report at about the same time. In consequence, the response time was too long. Some users should have been waiting for a long time and some other receiving a timeout and just a few was almost fine. That's why it was required to decrease the response time to be able to collect all data before 4 pm.

## Situations

To gather data from 500 branches within an hour while there was possibility to have 50 concurrent users

## Approaches

Regarding the customer request, we sent an on-site resource to investigate. We established two probe tools to log some main information on the two levels of SQL Server and Windows during the data collecting operation. After 3 hours of log file analysis, It's figured out a complicated stored procedure written in a not optimized way was the main cause of the issue. It took us about 45 hours to tune the stored procedure and fully test it. After applying, still some "SOS\_SCHEDULER\_YIELD" wait type were raising up indicating CPU lack. To overcome that, we recommended adding one more physical CPU with 4 cores and upgrading the SQL Server to 2008R2 Enterprise to be able to support 8 concurrent CPU cores; moreover, we forced the stored procedure to be ran on a single CPU core so that other CPUs are available for other users.

## Results

- Supported concurrent users extended from 3 to 50 users
- Maximum response time from ~6 minutes was turned to 15 seconds at the worst case scenario
- Timeouts were totally wiped out
- Response time for a single user came down from ~7 sec to ~300 millisecond
- Upgrade to SQL Server 2008R2 improved the security and overall performance

## Benefits

- Bank's related managers are able to get their desired reports daily before 4 pm and decisions are taken prompt

## PROCESSES

- Clear Work Flow
- Customizable Service Level Agreements (SLAs)
- Transparent Terms & Conditions
- At Will Contracts Termination

## SECURITY & CONFIDENTIALITY

- Customer Data Security Made by Most Advanced Technologies
- Customer Information Confidentiality

## CUSTOMER SATISFACTION

- 80% Repeat Businesses
- Customer Satisfaction Gaurantee
- Free Cost/Time Estimation
- Free Initial Consultation

- Around \$30k money was not wasted to switch to a RAID storage

**The Bank's IT Manager**

"Not only the concurrency improved very quickly but an efficient road map were made to us which would be applicable on future similar scalability issues"

*www.SQLMax.com / Info@SQLMax.com / 1.855.Try.SQLMax*