Asset Monitoring and Diagnostics System for Locomotives

Success Story

Customer

Our customer has assembled the most comprehensive collection of radio remote control brands for locomotives, cranes, material handling equipment, mining machinery, mobile equipment and virtually any equipment where the operator can be moved to a safer, more efficient location.

With nearly 10,000 rail-related applications installed worldwide, this company offers the most comprehensive combination of experience, product quality, and technical support to its clients.

Additionally, this company provides the services of remote monitoring & diagnostic to improve safety, efficiency and productivity in following ways:
» Monitor equipment in real-time
» Reduce costly maintenance visits
» Customize reports on equipment usage
» Receive and monitor alerts from a computer, pda, or cell phone

Challenges

One of our client's products is a web based system for remote monitoring and diagnosing Remote Control Locomotive (RCL) systems. This product provides customized monitoring of rail equipment, productivity/statistical data analysis, and usage anytime, anywhere.

This client contacted us to provide connectivity to more than 1000 Locomotives at a given time along with downloading and processing more than 15 million records on daily basis for different reports/alerts.

There were many challenges inherent in this project. The primary challenge was to measure the application/system performance and throughput throughout the evolution of the product/project life cycle, while changes were being made to meet the customer's requirements.
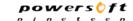
Another challenge was the regression testing of a big database, while keeping in view that multiple integrated applications were dependent on the same database/data.

Moreover, we had to develop and maintain an in-house test environment to test the application/system before deploying it on production.

Solutions

Benchmark testing was used to determine the performance characteristics of the application/system. These tests were repeatable and enabled single change to be made to the application in an attempt to determine whether there was a performance improvement or degradation. Goals of benchmark testing were to measure how a change affects the system's performance characteristics and then tune its performance in iterative cycles. 'Windows Performance Monitor' was used for performance counter data logging and analysis which provided a visual display of built-in Windows performance counters to review historical data. Required performance counters were added in Performance Monitor by creating custom Data Collector Sets.

It featured multiple graph views that enabled us to visually review performance log data. Custom views were also created in Performance Monitor. These views were exported as Data Collector Sets to be used with performance and logging features. We performed following steps as a part of the Benchmark testing process using 'Windows Performance Monitor':

>> Added relevant counters to collect the data
>> Executed the tests for a defined period (e.g. one day, week, etc.) under a certain load
>> Exported the data from Data Collector Sets in .csv and .xls on test completion
>> Analyzed the counter values
>> Drew the charts for comparison between different releases

This process was repeated whenever a relevant change was made in the application/system.

Synchronizing the database schema and database contents was another challenge as rapid changes were being made to the database in every release (i.e. data coverage, addition, manipulation processing, etc.). Red Gate SQL Toolbelt was used to minimize the regression time and to assure the data correctness so that there was no undesired change in the already tested schema/data contents.

Whenever regression was required, already existing benchmarked data was used in comparison with the latest changes to mitigate any anomalous state. Using 'Red Gate Toolbelt', we could easily compare a schema and huge amount of data in different tables. 'Red Gate Toolbelt' helped us to compare and deploy changes quickly and without any error.

We standardized the test environment components and its versions by using TFS version control system and checklists. This saved up to 50% time that is usually consumed by the test environment and configuration issues.

Achievements

The successful completion of the project included following achievements:

>> Designed Benchmarking Data (for application/system performance characteristics analysis)

>> Used 'Windows Performance Monitor' Tool to measure the performance by using this tool
>> Employed 'Red Gate SQL Toolbelt' to automate the database regression
>> Eliminated issues during database migration from test to production
>> Accelerated the deployment process
>> Generated T-SQL scripts
>> Used Team Foundation Server Tool (for effective version control)
>> Reduced cycle time, operation cost and manual dependency with automation & standardization
>> Created a local test environment to effectively analyze the root cause of incidents and defects

Business Results

>> Delivered the products/projects on-time and within budget
>> Utilized the available resources effectively to deliver multiple tasks/projects simultaneously
>> Our skill set and expertise are a source of inspiration for our customers

Contact Us

Explore ways to use our expertise in growing your business while establishing a valuable partnership with us.

Contact our consultants at:

E-mail: sqa@powersoft19.com
Website: www.powersoft19.com/sqa