IDERA® | Precise Case Study

PRECISE FOR SQL SERVER

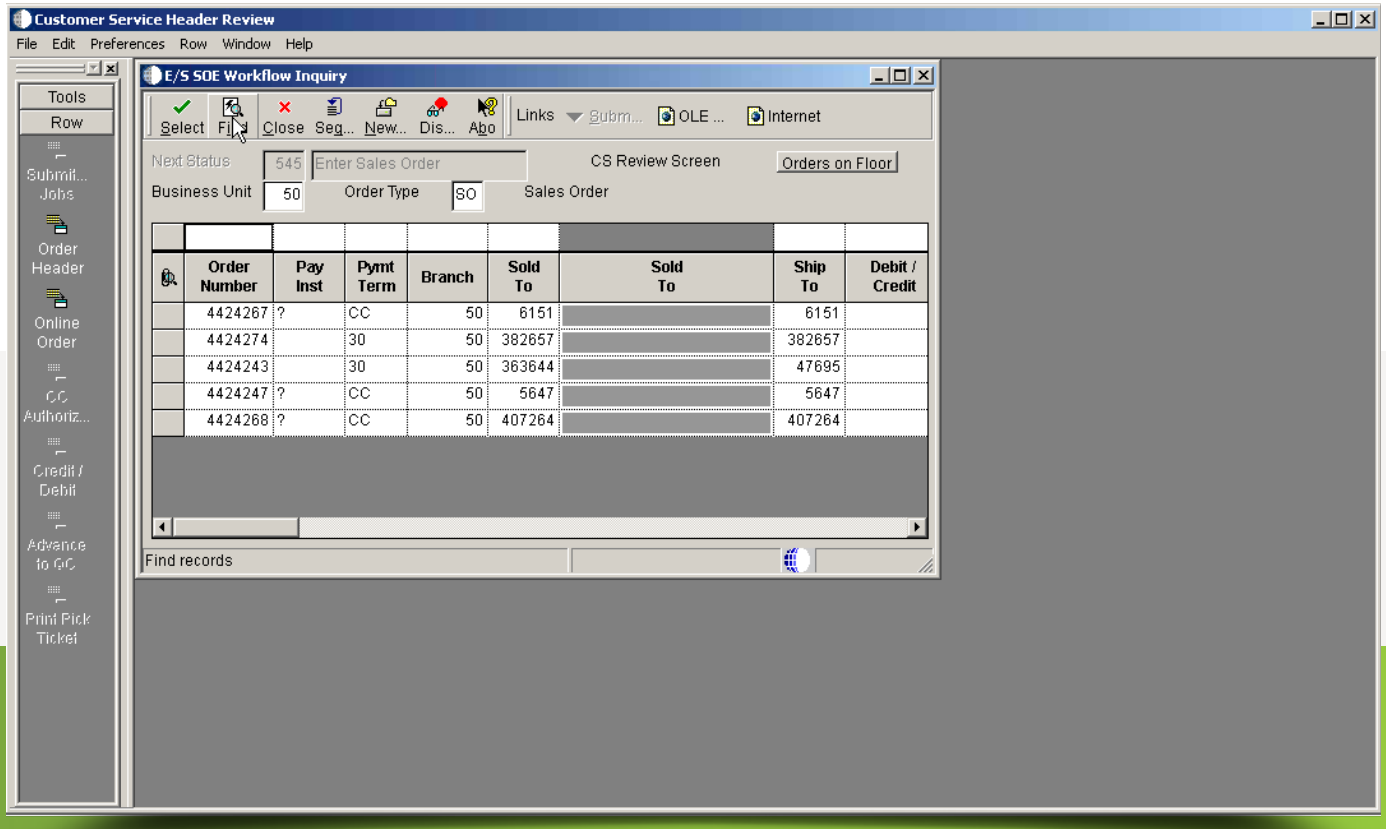
Wholesale Distribution Enterprise Fixes Business-Critical Application Error

PROBLEM BACKGROUND

A medium enterprise wholesale distribution company for nutritional supplements and drinks in the United States of America uses the third-party, multiple-tier JD Edwards Enterprise resource planning software to manage warehouse orders.

However, the business-critical application suffered from the ‘Sales Order’ screen freezing. These apparent screen freezes caused warehouse operations to miss a critical business service-level agreements to ship all orders received that day by 3 PM on the same day.

The service-level agreement is so important that the employees of the warehouse are paid to meet it.
The problematic application depends on Microsoft SQL Server.



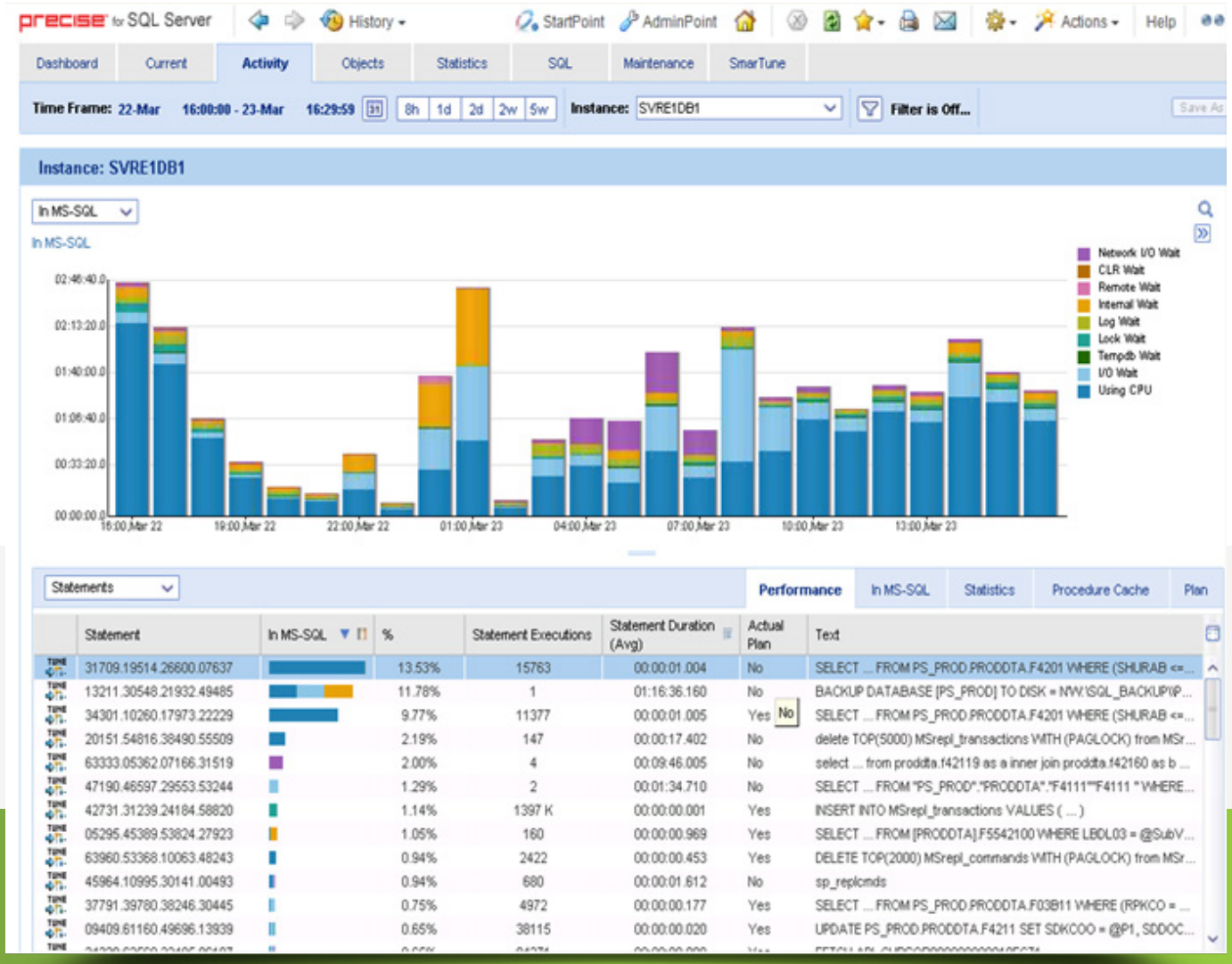
The business-critical application that suffered from apparent screen freezes.

INVESTIGATION

The ‘Statements’ area at the bottom of the ‘Activity’ workspace displayed in the third row the SQL statement identified as ‘34301’ for ‘SELECT FROM PS_PROD’ on which the ‘Sales Order’ screen of the application depends.

The third column displays the time in the SQL Server database. The fourth column displays the fraction of total time in the SQL Server database. The fifth column displays the number of executions. The sixth column displays the average duration of the SQL statement. The seventh column displays whether an actual execution plan is available. The eighth column displays the SQL text.

Selecting in the first column the ‘Tune’ button for this SQL statement displays the ‘SQL’ workspace in context of the SQL statement that is identified by ‘34301’.



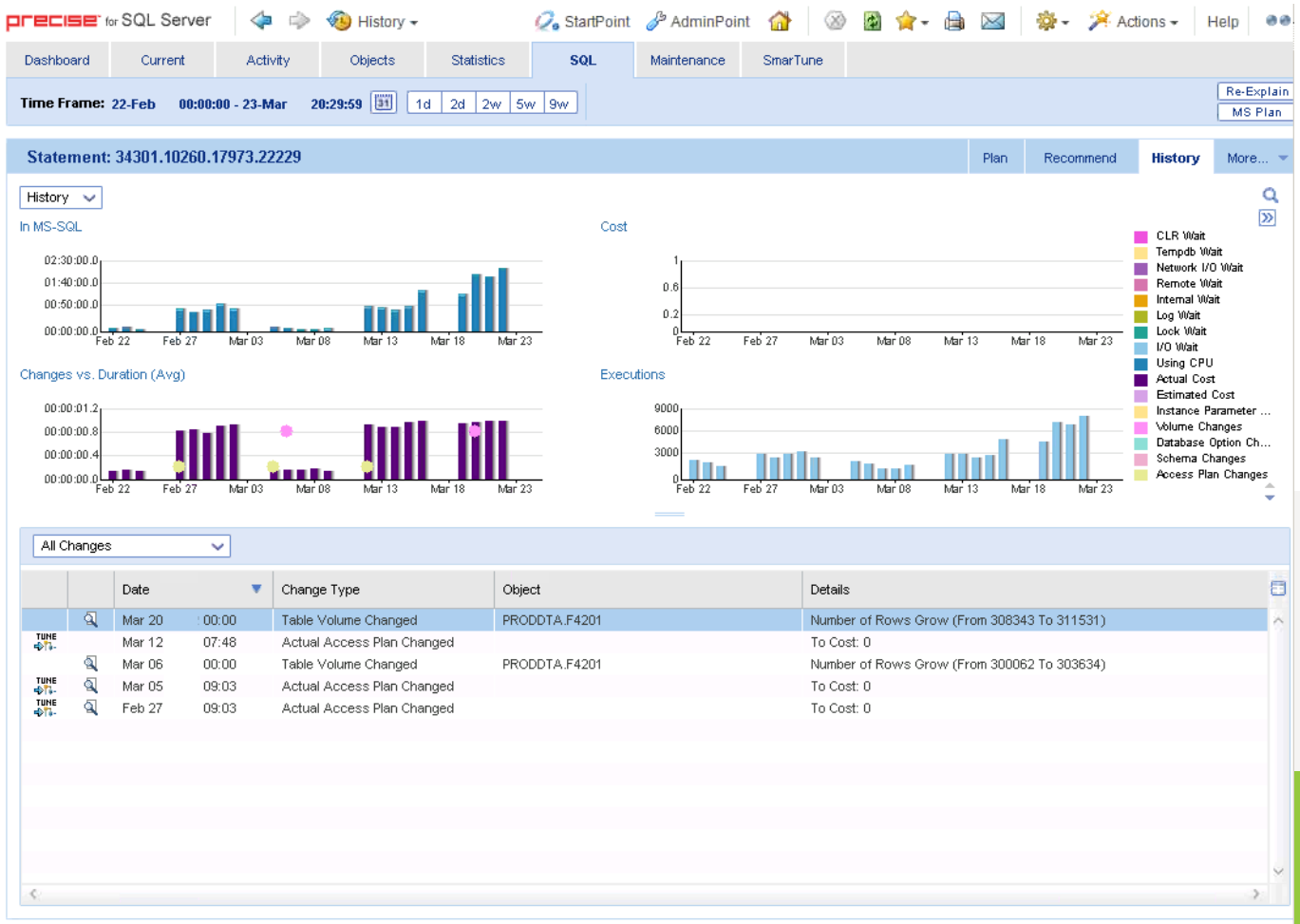
The ‘Statements’ area at the bottom of the ‘Activity’ workspace.

The ‘History’ area at the top of the ‘History’ tab of the ‘SQL’ workspace for the top SQL statement displays a chart for the average duration versus changes. The chart represents cause versus effect.

The chart shows that during the first three business days (that is, Wednesday to Friday), the average duration was some 0.15 seconds which refers to fast execution and few apparent screen freezes. In contrast, during the subsequent five business days (that is, Monday to Friday), the average duration was some 0.8 seconds which refers to sufficiently slow execution to appear as frequent apparent screen freezes.

The ‘Executions’ chart shows that during subsequent work weeks, the number of executions increases up to some three times because frustrated end-users resubmitted the ‘Sales Order’ screen. During these days, the application no longer meets the business service-level agreement.

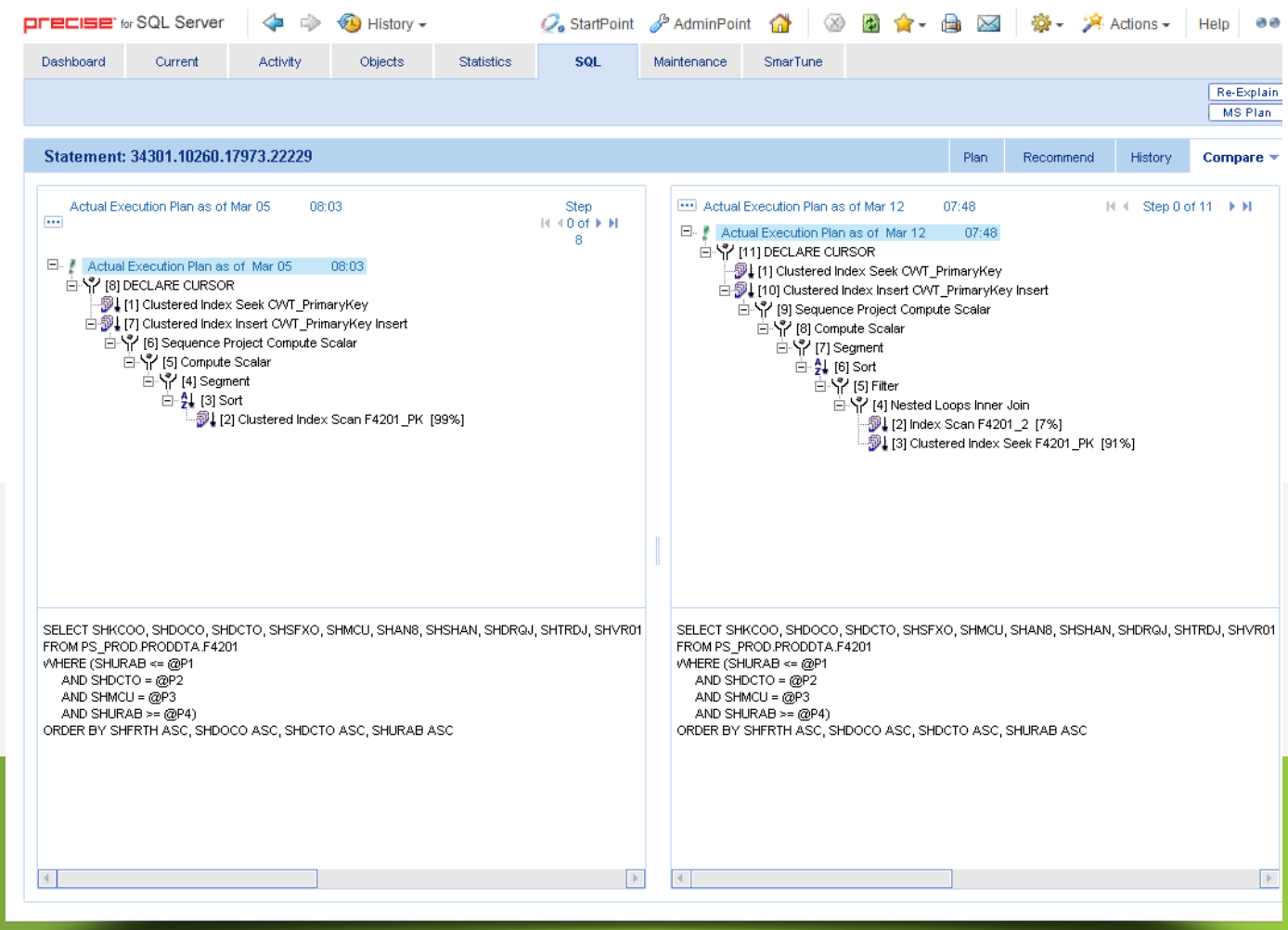
The “All Changes” area at the bottom part of the ‘History’ tab displays the changes. It appears that on March 5, SQL Server changed to a fast access plan. On March 6, the table volume changes which refers to significant table growth. On March 12, SQL Server changed to a slow access plan. That is, the SQL optimizer made different decisions. It is important to determine what decisions the SQL optimizer made.



The ‘History’ tab of the ‘SQL’ workspace for the top SQL statement.

The ‘Compare’ tab of the ‘SQL’ workspace for the top SQL statement compares the two actual execution plans. When comparing the two actual execution plans with the fast execution plan from March 5 on the left side and the slow execution plan from March 12 on the right side, it appears that the slow execution plan performed a ‘Filter’ join. The ‘Filter’ calls every row in the loop table which uses a large amount of CPU.

Both execution plans provide the same result, but one execution plan is more efficient than the other. Consequently, it is imperative to lock in the faster execution plan to improve the performance of the SQL statement and thus the performance of the ‘Sales Order’ screen of the application.



The ‘Compare’ tab of the ‘SQL’ workspace for the top SQL statement.

FINDINGS

The company pinpointed the problematic SQL statement with Precise. The detailed history showed very different execution times for the problematic SQL statement. The company tracked the problem to flip-flopping execution plans. One execution plan was fast, while another execution plan was slow. Locking in the fast execution plan resolved the apparent screen freezes.

SUMMARY

The company locked in the fast execution plan based on the results of the analysis with Precise to deliver consistent performance to the end-users of the business-critical third-party application.

The improved performance allowed the employees of the warehouse to meet their critical service-level agreement that directly affects their paycheck. Moreover, restoring the performance elevated the database administrator to be focused on technology and business transactions.

PRECISE FOR DATABASES

ACCELERATE BUSINESS PERFORMANCE

- Database Performance Fuels Company Performance
- Multiple Platform Database Monitoring and Alerting
- Performance Management Database
- Root Cause Identification
- Tuning Recommendations
- What-if Analysis
- Capacity Planning

SEE IT IN ACTION