The Quest for Database Performance:

Trimming Data Processing Time by Months; Saving Millions by Keeping Software Releases on Track

Major travel and hospitality organizations often have extensive technology footprints, from complex online booking sites to multi-location sales tracking and calculation platforms. Yet, many of these systems have been coded in pieces over many years, and errors, miscalculations, and other missteps have crept in along the way. In some cases, the distribution of relevant, tracked data over myriad systems has become so extensive that it is nearly impossible for discrepancies to be identified—and unlikely to be addressed, if they are discovered.

Such was the case for one global organization, which had long been operating with a legacy sales reporting solution. The system had grown over time, morphing into a behemoth into which data was fed by five separate accounting systems. Everyone knew the solution had a few glitches, but they were considered minor—and development and testing resources were never sufficient to track down and eradicate the issues.

When management requested some changes to how the system was reporting, it was discovered that the legacy platform didn't have the feature set to accomplish the goals. The organization purchased another system to achieve the functionality, but it was necessary to incorporate the historical data from the legacy solution.

Before that could happen, prudence dictated software teams should ensure the two systems were operating at parity—both would output the same results when working with identical data.

What happened next was alarming.

Teams soon discovered that, in many cases, not only were the two systems not producing identical results with the same data; the new system was generating flawed results with verifiably accurate input. It was evident that the problems had to be resolved, so company leadership hired an outside consulting team to assist in-house developers and testers.

The in-house and third-party consulting teams began working to eliminate processing delays and errors, along with calculation and database inconsistencies, between two reporting platforms. After two years of effort by the combined teams, errors that had been identified had not been effectively remediated. Attempts at database integration between the two systems had been unsuccessful, and database test performance was unacceptable.

When it became evident that the effort had failed—database integration between the two systems was impossible and even fundamental tasks such as loading test data were unacceptably inefficient—taking months rather than weeks—Orasi consultants took over the effort.

The Quest for Performance

In the area of performance tuning and optimization, Orasi teams were able to reduce historical data processing from 120 days to fewer than 21 days and cut by two-thirds report processing time for specific resources such as the Sales database. The overall effort, including both performance tuning and test data optimization and management, is saving the company millions of dollars, each year.

Orasi consultants were brought in to achieve two goals:

- Build new, optimized test cases (1400 total), run and analyze tests, remediate critical code flaws and deploy the clean code.
- Engage in performance tuning to improve data processing efficiency for the firm's 3.5 years of historical sales data (1300 sales records). The extensive effort eventually went beyond eliminating processing inefficiencies and restraints to incorporate deployment and configuration of new, faster processing servers to achieve business goals.

The results, which Orasi shares in this technical brief and a companion project success story, were dramatic. In the area of database performance, specifically, historical data processing has been cut from four months to 21 days. To request the project success story on test data optimization, click here. To explore the processes and solutions by which the performance team achieved significant performance improvements, read on.

Determining Performance Requirements

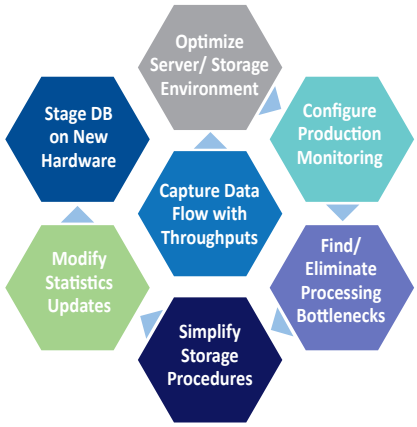
The original ROI justification for more accurate sales reporting was based on a loss in value that measured in the tens of millions, year over year. Specifically, the value of each day missed in the schedule for release was roughly $10MM/365 or $28K per day. The opportunity for improvement was $10MM per year/four months or roughly $3.3MM.

- The number of days to process was roughly 3.5 years or 1300 days of sales records.
- The prior solution could then be estimated to process 1300 daily records / 4 months => 11 records per day.
- During the prior execution, the client did not capture any monitoring data nor logging records.

The value of each day missed in the schedule for release was roughly $10MM/365 or $28K per day. The opportunity for improvement was $10MM per year/four months or roughly $3.3MM.

Orasi Activities for Performance Tuning

Project Outcome: 80% Reduction in Processing Time



Project Activities...and Risks

The existing (in-house and consultant) teams did not have a robust testing process for validation. Thus, any algorithmic change would have to be weighed against the cost of functional validation. Additionally, even if the testing could be done, the team did not have any resources allocated for doing this, because all of the testing was done as User Acceptance Testing without a documented plan or process.

For the project, there were several core activities in which the Orasi team engaged, all of which addressed operations that were being impacted by existing shortcomings with, or omissions in, current processes.

1. Capture Data Flow with Throughputs

There were three major steps to process one day of records:

1. Historical Sales Record (HSR) DB: Process existing historical data in the HSR relational database to create a flat file.
   - This phase read four tables and created one new one.
   - The processing time for this phase on the shared development server was one hour.
2. File Staging: Move the flat files to the next server.
   - There were three files in the package.
   - The SFTP process averaged 10 minutes.
3. Sales Reporting Analysis (SRA) DB: Process flat files into Sales Reporting Analysis relational database.
   - There were more than 30 steps in the data processing.
   - The processing time for this phase ranged from 45 minutes to 3 hours.

- During daily operations in production, this phase would hang indefinitely and need to be manually adjusted. The sustainment team had not fixed this issue over the prior year, and it was not recorded as either an incident or a defect.

2. Design an Optimized Server/Storage Environment

The overall project required improvements and changes to the data entering the historical sales record relational database. Additionally, the processing code at each tier was being improved to reduce errors in numerous unplanned ways. As a result, the performance optimization swim lane would be blocked by these project demands and vice versa.

The Orasi team performed a cost-benefit analysis based on laying out a temporary suite of virtual machines and storage to do performance optimization and final data processing. The results indicated the effort would provide significant benefits and be worth the effort.

HSR Database

The team determined that the production database instance could not be used for the final processing for numerous reasons:

- The production DB server handled 14+ hours of processing daily to meet other demands.
- Any production issue or update would have a higher priority than this batch process.
- The team could not perform optimizations to the Informix DB configuration due to production risks.
- The team could not perform optimizations to the Itanium Micro Focus UX OS due to production risks.

Because of those constraints, the team had to use the development database instance for the processing for several reasons:

- Itanium Micro Focus UX servers were not available in the datacenter.
- The Informix licensing costs were too high to create another instance.
- The solution was only validated on this specific Informix and Micro Focus UX version.

During this phase, the team discovered that the SAN storage solution for both development and production needs was Tier 3, which was appropriate for the daily workload of these servers. They had a 24-hour time window to perform processing, which was adequate based upon business requirements.

Tier 1 Storage

The IT organization had recently purchased a new Tier 1 storage solution, the EMC Extreme IO with integrated SSD. This storage tier had a maximum iOPS rate more than 100 times that of Tier 3. The IT organization was also transitioning a broad range of data from legacy tiers to this one, so they had an active and ready set of teams and queues available.

Staging Files

The current implementation staged the files onto a Tier 3 SAN using a default Windows VM instance. This Windows instance was shared for development, testing, and other uses. The server was inadequate, in terms of both durability across the processing window and performance.

The team transferred the staging location onto a drive collocated with the SRA DB.

SRA Database

The SRA DB was a Microsoft SQL Server running a recent version, and it was also up to date on the OS. Thus, the team could use the standard IT template for a new virtual machine instance. This request and cost was easy to capture and process.

The storage sub-system was housed on a Tier 3 SAN in the current test and production solution. The team proactively defined the new storage allocation on a Tier 1 SAN. This system would be retired after the optimization phases and the three-year processing window were completed.

3. Configure Monitoring

The default monitoring solution resided at the hardware level. It captured CPU, IO and network utilization. There were no dashboards or reports configured to capture any faults or alerts to the support queue. There were no signatures or definitions to describe this business system in any reporting or awareness module. The database performance was not tracked for index tuning, query optimization or other means.

There was no written history of any prior optimization efforts during the design phase. The team relied upon oral history for all prior information.

As the new servers were identified, the team configured an improved monitoring plan for production based upon the changes the team incorporated into the performance test environment. The new solution captured alerts at each phase of processing if they exceeded the current latencies by three or more standards of deviation from the mean processing time.

4. Find and Eliminate Bottlenecks via Hypothesis and Validation

Prior to the availability of the dedicated environments, the team captured processing times for a period of 10 days.

Legacy Shared Environment Baseline

During an open window between functional updates, the team captured data over a few days to establish a baseline of performance. The team also captured explain plans for the Informix and SQL DB.

HSR DB

Ranged between 45-90 minutes in the re-test, which was consistent with the production processing times.

The explain plan showed one complex sub-query, which was reused in many WHERE clauses. This was essentially a Boolean filter. The team proposed a new field in the schema during import, which would flag this record for use during the first phase of processing. According to the explain plan analysis, this would reduce the processing time by more than 30%.

Staging

The end-to-end process for SFTP of the files was 15-20 minutes. The IO workload was difficult to measure, but the files were less than 200MB so the overall time was out of balance with expectations.

SRA DB

Processing time ranged from 45-90 minutes with bigger spikes. The IO wait time generated alerts on the VMware management system when the team ran the tests. For approximately 10 minutes during each processing period, the average IO latency was over 300ms, compared to a target of <10ms.

For approximately 10 minutes during each processing period, the average IO latency was over 300ms, compared to a target of <10ms.

The SQL explain plans were not very useful due to the system design. The solution consisted of more than 30 complex stored procedures. Each stored procedure had more than 20 branches. In essence the SQL optimizer could not really assist with Index suggestions.

New Environments

After the new environments were finished, the team captured a baseline of performance on the new hardware for the three phases. The team started by processing 10 days of new files to capture a mean time to process over a larger sample set.

HSR DB

After the transition to the new hardware, the processing time dropped from 45-90 minutes to 15-20 minutes. This was entirely attributed to the faster Tier 1 SAN.

In addition, the new hardware allowed us to process the historical data 24 hours per day instead of 12-14 hours per day. The net improvement was considerable:

- Legacy environment: 10-12 historical days per 24 hours => 120 days.
- New environment: 60+ per 24 hours => less than 21 days This improvement met the business goals.

ASR DB

Per the earlier risk analysis mentioned earlier, the team were pressed to try hardware and DB-level tuning as a lower risk improvement to the system.

SQL MAXDOP

The MAXDOP was set to 1 as part of the configuration on this server. The team tested it on 1, 2, 4, 8, and 12. The performance per SPROC improved up to 8, which was the VCPU count of the server.

Disk IO Utilization

The SQL database transaction logs, DB files, and indices were not designed and laid out according to workload. Thus, the team re-arranged these files to minimize contention during the data processing phases. This took approximately five groups of changes to reduce IO latency and improve overall throughput.

5. Simplify Storage Procedures

Ultimately these changes offered the most benefit. The core technical resource who owned the design and the risk broke the SPROCS into smaller pieces. This allowed the SQL optimizer to work more efficiently. In addition, after the team fixed five of the most complex elements, the SQL index tuning wizard yielded a series of more tangible recommendations.

6. Modify Statistics Update Approach

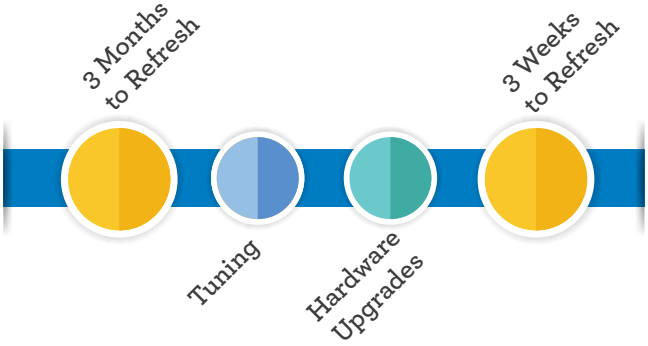
The team also modified the place as well as the level of statistics the team would update after processing each day of files. This modification was significant, because the overall processing time would vary as much as 50% depending upon the level of fixup the team applied. However, a 100% update took 20 minutes to process, so tuning this to the demand was critical.

7. Stage DB on New Hardware

After shifting the location to the faster SAN and collocating it on the SQL server, the file transfer time held steady at less than one minute. In addition, since the first two phases were now significantly faster than the final data processing time, the actual cost of the transfer was not in critical path. However, processing the files locally improved the first phase of data import into the SQL staging tables.

Final Analysis

Although many other ideas and hypotheses were not explored, mainly due to their impacts on one or more elements critical to maintaining the project deadline, the results achieved with the completed effort were significant.



- Sales Reporting Analysis (SRA) processing no longer hangs, requiring manual intervention. Specifically, after the project was complete—including the new Tier 1 SAN—processing time has dropped from 45-90 minutes to 15-20 minutes.
- Historical data processing can now take place 24/7, reducing historical processing time from 120 to fewer than 21 days.
- The performance optimization effort, in tandem with the associated test data management project, is saving the company millions of dollars, per year, due to the single achievement of keeping releases on track.

Other ROI, such as that from eliminating calculation errors between and within the two systems, is discussed in the companion paper to this piece, “Redesigning a Failing Reporting System.”