PerformanceLab
Software testing company

Case study: Retail

MVideo is a leading retail chain that sells electronics and household appliances. It is one of Europe's largest players in this sector. MVideo was founded in 1993. Since then, 368 stores have been opened in 158 cities and more than 18,000 people employed. At the end of 2014, MVideo's annual sales were 3.5 billion dollars, with an offering of more than 25,000 different products.

A new automated sales and logistics system needed to be load tested before it could be deployed. Considering a planned expansion, MVideo needed to be convinced that the new system could withstand the load from 512 stores at peak periods.

Challenge

A new automated sales and logistics system needed to be load tested before it could be deployed. Considering a planned expansion, MVideo needed to be convinced that the new system could withstand the load from 512 stores at peak periods. There were also specific performance requirements given, such as: replication time, maximum replication queue size, and processing time for messages in the queue manager.

The main challenge was organizing the stores' 256 databases and feeding an intense load, comprised of more than 10 customer business processes and SOAP requests to a central database, from more than 30 web services. The load from 512 stores was to be supplied by doubling the intensity of operations from 256 stores.

Given a two-tiered architecture and a thick client, the hardest part of developing the load scripts was emulating the logic of the business processes, which was largely implemented on the client side, thus complicating the parametrization of the load scripts. What's more, this challenge was resolved without detailed documentation, a database schema, or assistance from the developers.

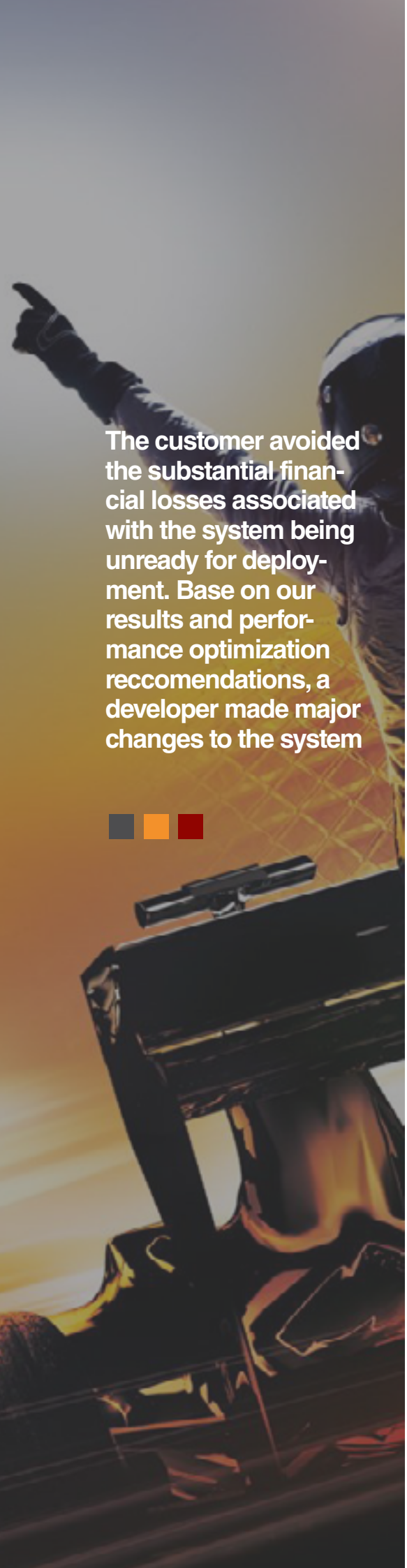
Solution

In order to make the load testing as close as possible to real operating conditions and to meet the requirements with respect to the number of stores, 256 Oracle RDBMS schemas were rolled out in several databases on different virtual machines.

To optimize the writing of load scenarios, our engineers developed a utility that parsed client logs and used them to create scripts for JMeter, a load testing tool.

Several technical challenges were resolved during test preparations and initial attempts to apply the load. For example, due to the extreme load and the complex computations involved in emulating the client's actions, the workstations allocated to the load testing were unable to supply the required number of operations. To achieve the required load intensity, the scenarios were parallelized among the available workstations, the trash collector was optimized for JMeter, superfluous SQL requests were eliminated, and the scripts were ported from Java to Groovy.

The primary object of testing was the central database, which receives the load resulting from replication of the stores' databases. Because the stores' databases lagged behind in processing the client's SQL requests, they were optimized.
Intermediate test results revealed problems in the central database



The customer avoided the substantial financial losses associated with the system being unready for deployment. Base on our results and performance optimization reccomendations, a developer made major changes to the system

and replicators. Accordingly, some performance optimization work was done without waiting for the end of the project: database indices were optimized and the need for database locks was eliminated.

Customer benefits

Performance Lab engineers' detailed analysis of the load test results brought to light considerable flaws in the system's initial design, such as: unoptimized indices, excessive data on the client side, a large number of unnecessary database locks, blocking of business processes, and a five-fold slowdown of critical interfaces.

The customer avoided the substantial financial losses associated with the system being unready for deployment. Based on our results and performance optimization recommendations, a developer made major changes to the system.