Load Testing Case Study

the need

The IT department of a university was recently given the challenge of determining the cause of sporadic slowdowns to their SCT Banner system. Much of the time the application performed quickly and users did not experience system slowdowns. At other times, however, the Banner system slowed down, user screens timed out, and class registration time went from a few seconds to over a minute. Faculty and students became increasingly frustrated.

In an attempt to solve the problem, the IT staff purchased monitoring and tuning software modules to examine performance. They also brought in consultants to diagnose the system problems. Unfortunately, their tuning efforts were expended on a system that was not experiencing heavy workload. The monitoring tool did capture heavy workload and that workload was analyzed; yet no changes were made during the heavy workload period because the Banner system had to be available.

For the purpose of loading up the system for further analysis, the IT staff set up a registration simulation in a classroom using faculty members. Unfortunately, the simulation did not create enough workload. These performance-tuning efforts failed because:

- A heavy workload could not be created by a handful of users.
- The monitoring software measured system performance during heavy workload; however, no changes could be made to the system, which could be measured by the monitoring software.
- Experimentation and analysis could not be done on the production system during heavy workload because the system had to be available.
- IT staff were "flying blind" in trying to correct these predictable but unidentified performance problems and were only able to register 300 students per day during a subsequent late registration process.

the inplexus solution

Even more frustrated, the university called on Inplexus to find and solve the problem. Our strategy was different. To find and solve the problem, we had to first reproduce the problem. What did Inplexus do differently? We produced and tuned the Banner system without interrupting the production environment.

In order to create an adequate load test, we interviewed university staff to find out what application activities occur during heavy workload. University staff provided Inplexus approximate totals for each of the activities. Questions asked included: How many students were registering for a class at the same time? How many students were performing a class search at any given time? How many faculty members were reviewing their rosters during registration?

Armed with this information, Inplexus deployed software scripts to simulate the workload encountered during busy times. Once the software was configured and the scripts completed, each individual virtual user simulated active users of the Banner system. While creating workload, Inplexus trapped system metrics on the network, application server and database software, the application server machine and database machine. Most importantly, the response time of the virtual user was tracked via the automated load-testing tool.

As the workload increased, bottlenecks became obvious. Adjustments were then made to the bottleneck component(s) and the same test was run again to see if performance improved. Iterations of this workload testing and adjustment continued until hardware limits were reached.

Inplexus' methodical analysis and adjustments to the system were successful in finding and relieving many configuration bottlenecks. The university was then shown which components were problems and which were not contributors to the previous sluggish system performance. The university also has quantified values for how much workload the Banner system can handle during peak usage. Inplexus also made recommendations for hardware changes to improve peak load throughput. Their SCT Banner system now can handle heavier workload during peak usage.

Let us help you get the maximum efficiency from your Banner database. Thank you for considering Inplexus.