HP Customer Perspective, Retail Sector

How to prepare, plan, and execute effective performance testing strategies for large, global SAP implementations

White paper

Table of contents

About TUSC

TUSC is a global provider of enterprise IT services and solutions. TUSC engineers help clients design and implement end-to-end Enterprise Performance Management (EPM) strategies by addressing strategic, financial, and operational management processes across the enterprise. With over 20 years experience, TUSC has provided EPM strategic and tactical support for enterprises of all types and sizes, helping them manage implementation, upgrade, and migration of strategic IT systems. On average, TUSC EPM clients have realized a 200–300% in end-to-end system performance results.

Introduction

The company is growing. Operations are expanding, reaching out to new geographies, conquering new markets, and acquiring new brands. Sales are booming and the newest, trendiest products are virtually flying off the shelves. But what's happening behind the glitz and glamour of the new storefronts and popular brands that now bear the company logo? Are current backend systems prepared to handle the integration of different technologies? Can they support the expanding volume of products, orders, records, and transactions? Can it be said with confidence that a store manager in New York will be able to order inventory from a warehouse in China or the customer in Singapore can pay for her purchase with a local bank card?

Companies rely on enterprise-class business systems such as SAP to automate key business processes—from supply chain, to customer relationship management, manufacturing, and finance. But without a clear strategy for implementing, integrating, and maintaining these systems, the risk of failure may be too great.

In 2005, a large, multi-national retailer hired TUSC to design, develop, and execute a comprehensive performance testing strategy for its enterprise SAP implementation. The "Retail Giant"—another name used to refer to the client in this paper—has recently acquired four new brands, each using different systems for logistics, merchandise planning, order entry, and financial reporting. The multi-phased task of moving these disparate systems to an SAP platform fell to one of the global systems integrators. And, TUSC was given the mandate to verify that the Retail Giant's new foundation was meeting the highest standards of availability and performance. This paper describes the joint efforts of TUSC and the Retail Giant to build, manage, and execute a performance testing strategy, set up a performance Center of Excellence (CoE), create reusable testing assets and increase visibility into application load, and stress testing data to ultimately optimize system performance and availability prior to going live. It will also highlight the best practices that TUSC has developed over the years for building a solid, consistent, and repeatable strategy for performance validation and share the knowledge that can help any organization verify that its mission-critical IT systems can support its business needs.

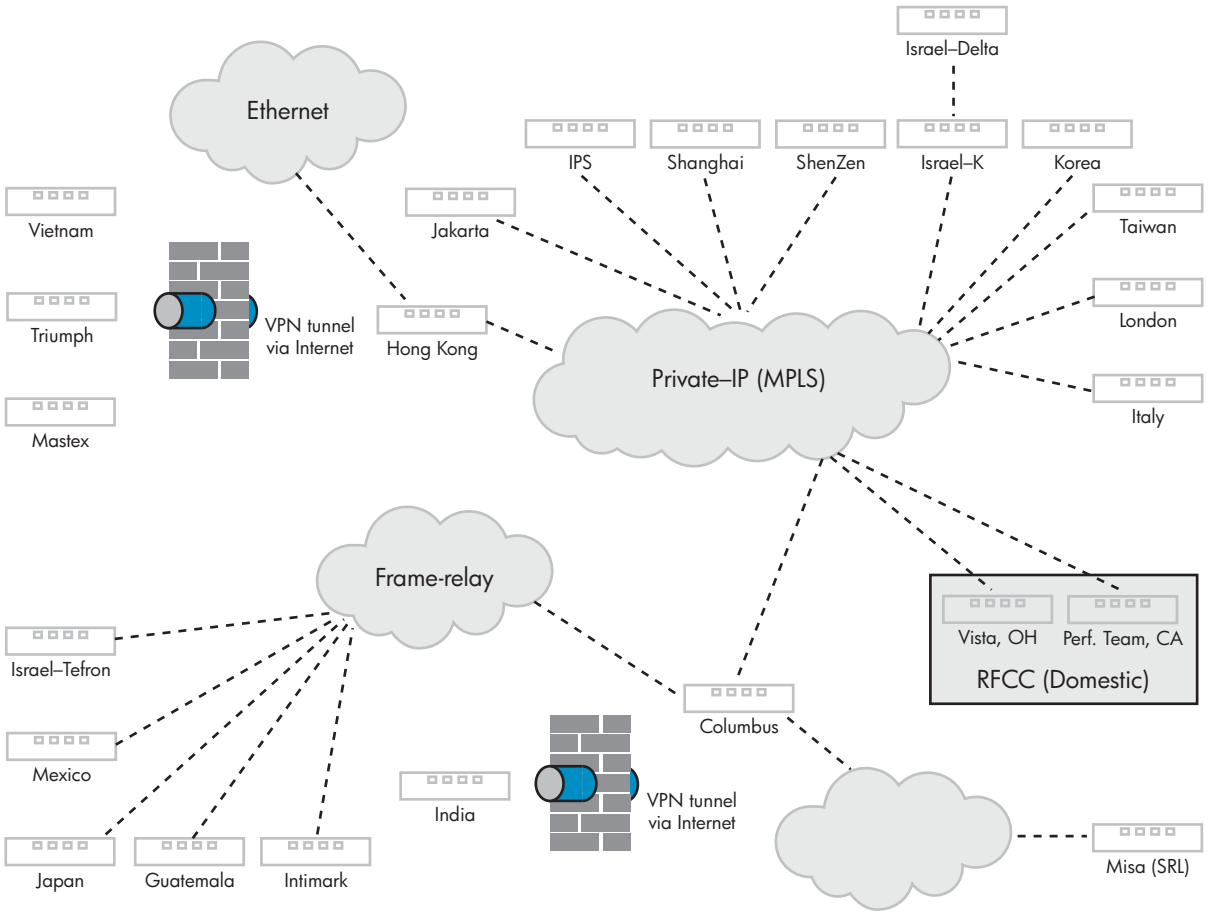
Facilitating corporate support for automated SAP performance testing

As performance engineering consultants, TUSC typically gets engaged with the project when the commitment to performance testing has already been made, budgets allocated, and project goals set. But corporate IT leaders may face a number of challenges convincing the business that its new SAP implementation, a major upgrade or a significant change, require end-to-end automated performance testing prior to going live.

Figure 1. The Retail Giant's SAP infrastructure is a complex system that spans multiple applications and geographies.

SAP Applications
- SAP ERP Financials
- SAP ERP Human Capital Management
- SAP ERP Operations
- SAP NetWeaver Web Portal
- SAP R/3
- SAP CCMS

HP Products
- HP LoadRunner
- HP Performance Center



Today's software applications are incredibly complex, with many potential points of failure in every single business process. SAP business systems consist of many integrated application and infrastructure components, and each SAP implementation is unique—with specialized customizations, modifications, and links to third-party and proprietary software. It can't be assumed that the multi-faceted, enterprise-scale SAP business system will support thousands of users when the go-live switch is turned on. Prior to going live, it is important to get an accurate picture of end-to-end system performance, verify that the application meets service-level agreements (SLAs) and other business requirements, and analyze test results to fine-tune the processes and infrastructure to reduce performance problems.

Don't allow the company's success to backfire. If the system becomes unavailable due to too many concurrent users, or employees can't view the product inventory because the results take too long to load, or the financial system can't handle the load of too many simultaneous transactions, the business will suffer. An effective, well-constructed automated performance testing strategy will help the company make better, more accurate release decisions, prevent system downtime, and avoid availability problems. This is where IT and business should agree: Resources spent on performance management is a direct investment in business success.

Designing an effective performance management strategy

Application performance management is much more than running a couple of stress tests. It requires a building-block approach that involves several phases. The methodology that TUSC is using with its clients can easily be applied to a performance testing group within corporate IT or a performance Center of Excellence (CoE). IT is essentially a service provider

to a business, so its line of business (LOB) becomes the "client" who signs (literally or figuratively) a contract with IT to validate its systems' performance and availability. While specific implementations and approaches may differ, the business can use the same basic framework for setting up an effective performance testing strategy as TUSC has been using for over 20 years.

Phase one: project initiation

During this phase in the performance testing project, TUSC EPM works with the client to assign resources to a project, agree on the project goals and objectives, define roles and responsibilities, and finalize the project scope and timelines.

For the Retail Giant, the effort to migrate all backend systems to SAP was so large, that it too had to be divided into stages. During the first stage, Retail Giant was planning to roll out SAP ERP Financials, followed by SAP ERP HR and Operations in the second stage. During project initiation meetings, the Retail Giant agreed that the first part of TUSC's engagement will involve validating the performance of its SAP financial systems: general ledger, accounts receivable, accounts payable, contract accounting, credit management, electronic invoicing and payments, and collections management.

Knowing the scope of the project helped TUSC consultants allocate resources, finalize budgets, and define the exact deliverables that the Retail Giant could expect at the end of the engagement.

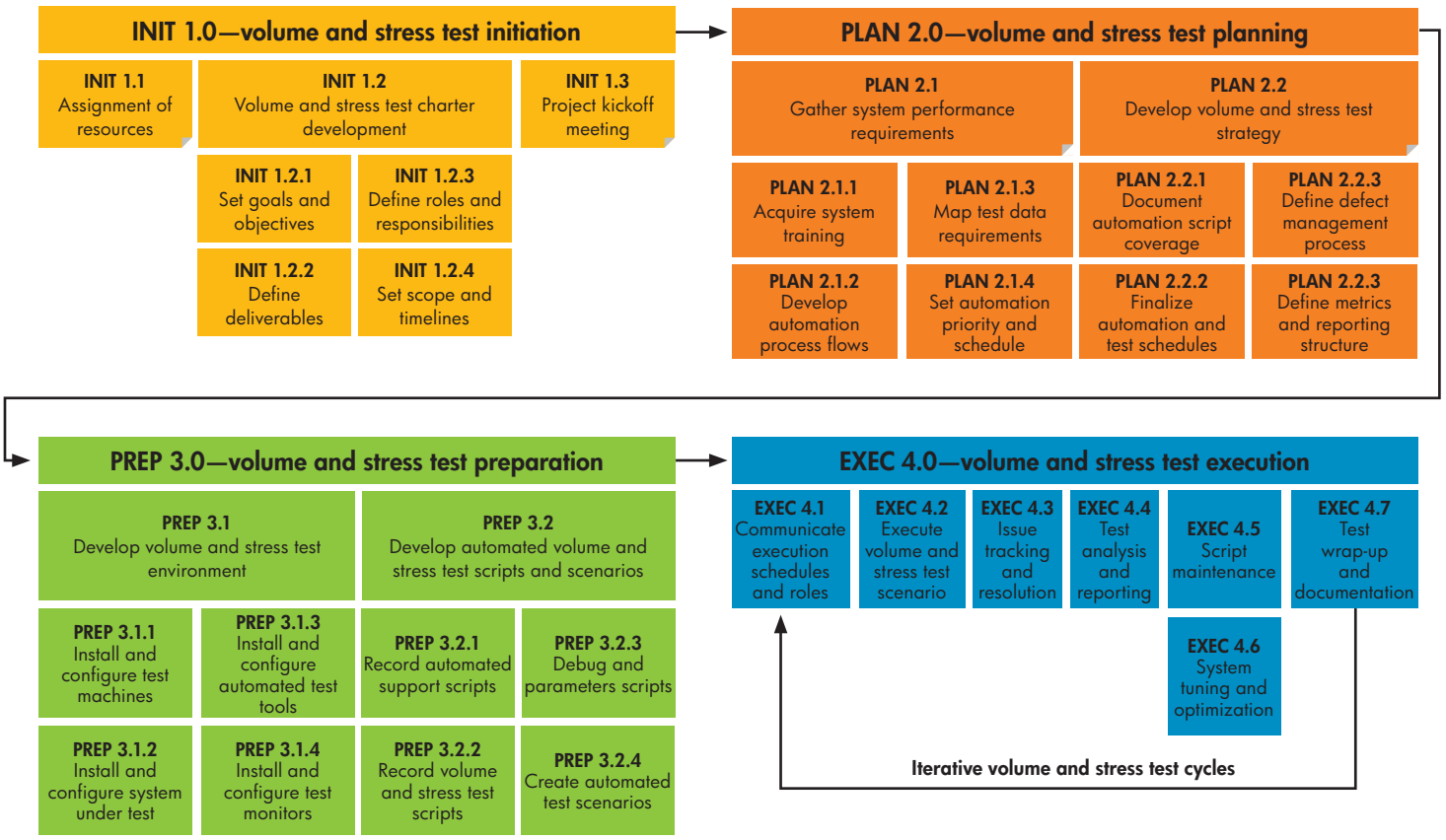
Phase two: volume and stress test planning

The planning phase is the primary time to gather performance requirements. For each performance testing project, there are several types of requirements, and different expert teams inside the client's organization who can help collect, document, and prioritize those requirements.

For example, to collect business requirements for the Retail Giant, TUSC engineers met with the company's business analysts. They received an overview of the Retail Giant's SAP financials system and a demonstration of how the applications are being used in day-to-day operations. Seeing how the data from the cash registers in over 1,500 stores in North America is being transferred into the central accounting system or how each night at every store the system calculated the day's sales numbers helped TUSC engineers understand which transactions are used most frequently and will have the highest performance impact. Based on the data that TUSC collected from the business analysis and the end users, they were able to generate a list of key business processes that reflected most important activities performed by business users on the client's SAP system.

TUSC has extensive experience in implementing SAP applications, and its engineers are intimately familiar with the most critical and frequently used transactions. Still, seeing how these transactions are being used by the Retail Giant in real life and documenting the detailed flows gave TUSC consultants an important foundation for creating a master list of transactions that need to be measured under load and for building automated process flows.

Figure 2. Load and stress testing process overview



This exercise also helped TUSC gather test data. The business analysts were also very helpful in identifying the right types of data for accurately emulating sales orders, payments, or funds transfers.

Another critical type of requirement is technical requirements. At the Retail Giant, TUSC consultants met with IT management, systems administrators, database administrators (DBAs), and the infrastructure teams to create detailed diagrams of the client's SAP infrastructure. Since performance testing is done on the real application infrastructure, TUSC had to identify which components can be used and which ones needed to be excluded from the test.

TUSC and Retail Giant's IT had an ongoing relationship during the performance management project. TUSC ran the tests, reported the results, and suggested possible problem root cause. And it was the job of the corporate IT to perform detailed problem diagnosis, fix the cause, and notify the TUSC team that the transaction or component was ready for retest.

Finally, TUSC had to agree with the Retail Giant's IT and LOB management on the detailed system requirements. System requirements are high-level goals that govern the pass/fail status of the load-testing process. For example, if the client decides that their SAP system must support two thousand concurrent users with a transaction response time of under four seconds, anything less than that would be considered a performance problem.

Retail Giant set the following system requirements:

- The SAP system must support 2,000 users at normal periods and 3,500 at peak periods. The peak time is 10 p.m.– 2 a.m.—when the system collects and processes all sales data from all the stores.
- The system must process several thousand transactions and some different load targets for different subsystems and interfaces include tens of thousands transactions per second.
- All transactions must have response time of three seconds.

Last, but not least, TUSC had to determine how the performance testing is going to be handled—as a project-specific initiative or as a Center of Excellence (CoE). TUSC performance engineers recommended that the Retail Giant build an internal competency center and consolidate all performance management functions, processes, and best practices under the CoE umbrella.

A performance CoE provides a management and automation platform for performance optimization processes, as well as leadership and advocacy to help the entire organization understand the importance of performance validation and optimization. All the expertise, testing assets, toolsets, and best practices developed by TUSC a part of this engagement could be reused by Retail Giant's performance CoE for future projects.

The performance CoE was also going to be the central repository for all defects uncovered during the performance validation process, and the keeper of all project schedules, timelines, and deliverables. With the central point of control of all performance testing activities and the dashboard for performance and project-related key performance indicators (KPIs), the CoE became one of the key factors of success for the Retail Giant's engagement. It is also worth mentioning that at the moment, the Retail Giant's performance CoE consists only of two full-time resources. This is the great advantage of the CoE model—there is no need for a large workforce to run performance tests. Establish a fully operational performance Center of Excellence (CoE) with only a small number of employees, and then adjust staff, resources, and capabilities based on the changing testing requirements.

Selecting the performance management toolset

The TUSC consultants are skilled in working with most of the performance management tools on today's market, and over the years most of TUSC's customers have standardized on HP LoadRunner—the industry-leading, cost-effective enterprise performance validation software. For clients who are just starting the toolset selection process, TUSC recommends HP LoadRunner for its comprehensive approach to performance testing, ability to obtain an accurate picture of end-to-end system performance and the combination of end-user, system-level, and code-level testing and diagnostics tools.

Like many other enterprise customers, the Retail Giant has standardized on HP LoadRunner since 2000. And as they began to plan a large-scale performance validation project for their SAP implementation, the choice of the software solution was obvious. HP LoadRunner can accurately test the end-to-end performance of the SAP system by emulating thousands of users, measuring end-user response times, monitoring application components under load, and providing robust analysis and reports to help resolve any performance issues prior to going live.

Using minimal hardware resources, HP LoadRunner stresses an application from end-to-end, applying consistent loads and using the data to identify scalability issues that can affect real users once your SAP system goes into production. HP LoadRunner supports performance testing for a wide range of application environments and protocols. It also provides system-specific monitors, including SAP R/3 and SAP Computer Center Management System (CCMS).

For Retail Giant, TUSC took advantage of the fact that HP LoadRunner has both the Web mode for scripting SAP transactions that go through the SAP Netweaver Web Portal and the SAP R/3 graphical user interface (GUI) protocol for scripting the processes that utilize the SAP R/3 GUI. This provided coverage for all parts of the client's heterogeneous SAP environment and did not leave any parts of it untested.

Given HP LoadRunner's undisputed market leadership, the highest level of partnership between HP and SAP and the fact that SAP uses HP LoadRunner to perform "go-live" checks for its own customers, TUSC engineers felt comfortable recommending HP LoadRunner for the Retail Giant's mission-critical performance testing project.

Benefits of the performance testing CoE:

- **Improves quality** by providing and enforcing consistent processes across all testing projects, helps reduce defects prior to production
- **Drives productivity** by using common test assets, sharing institutional knowledge and best practices across the enterprise
- **Reduces costs** by leveraging consolidated resources and centralized skills
- **Fosters interaction and collaboration** between line-of-business (LOB) customers and performance testing project teams

Phase three: performance test preparation

The test preparation phase can be divided into two areas: developing the test environment and building the automated volume and stress test scripts and scenarios. The information collected in the planning phase can be turned into automated test components that can be leveraged to drive a repeatable, realistic load on the system.

The test environment part of the process involves installing and configuring test machines, preparing the system under test, setting up the test tools, and deciding which machines and servers will be monitored under load.

The first step in the automation test preparation process is to record automated scripts. TUSC uses an HP LoadRunner script recorder to capture the business processes into test scripts, which are often referred to as "Vuser scripts" or "Vusers." The critical part is to record the various SAP business processes from start to finish—this helps determine the amount of time needed to complete a business process.

In order to accurately emulate real users, it is important to configure think times, connection speeds, and error handling. This is why it was so important to talk to different types of users in the planning phase—TUSC engineers gained a good understanding of how a store clerk or a finance manager would handle specific transactions. Someone with more experience with the system may complete their tasks quickly, often bypassing steps, while other users may take longer to complete the same transaction.

The next step is to insert test parameters. User data recorded in the script needs to be replaced with parameters that draw unique, relevant values from the data pool. This way, every simulated user in the system will be entering a unique, realistic order number or account code, making tests much closer to emulating real production environment.

Finally, based on the data that was gathered during the planning phase and the system requirements, load test scenarios are set up. The load test scenarios create groups of Vusers based on their transactions. A varying number of Vusers are assigned to individual business processes to emulate user groups performing multiple transactions. The groups are then assigned to the load-generating machines to prepare for the load-test execution. Running tests from the central point of control provides the ability to organize, manage, and accurately monitor the test execution process.

The load generators also have to be specially set up to realistically simulate the end-user experience. In the case of Retail Giant for example, TUSC had to make sure that the desired Web browser and SAP GUI client versions were installed and configured on each load generator machine. Also, to account for the client's multi-national operating environment, the load generators had to be set up in all major retail and office locations—including several U.S. cities and Hong Kong.

Phase four: test execution

Instead of saying "load test," it should be more about "load tests," because in reality there are many different types of load tests, each giving a different angle on application performance and providing additional information to measure the business risk associated with releasing the application into production. Each type of test must have its own timeline, resource requirements, entrance and exit criteria, and reporting requirements. Consider the table below for the list of six commonly used test types. Note how the development time for scripts and scenarios is usually a

Table 1. Performance test types

Test name	Required resources	Common metrics	Entrance criteria	Exit criteria
Automated script development	Performance test team: 100% Business SME's and Development team: 5–10%	Smoke testing—no formal metrics	• Access to stable build/data. • Test tools/environment configured. • Can script business flows.	• Automated scripts recorded and parameterized (accurate simulation of production)
Performance SMOKE testing	Performance test team: 100% Business SME's and Development team: 5–10% Infrastructure team: 10–20%	Smoke testing—no formal application metrics "Vanilla" Network Bandwidth Utilization stats	• Script development complete • Test scenarios complete	• Automated scripts replay as expected and generate accurate impact on the system under test.
Performance BASELINE testing	Performance test team: 100% Business SME's and Development team: 10–20% Infrastructure team: 20–40%	System performance baseline report: end-to-end business process transaction timings System hardware performance statistics	• Unit/functional test complete • Code frozen/migrated to QA • Test environment loaded/functional	• All performance scripts executed successfully (multiple runs) • Baseline metrics acquired
Performance INCREMENTAL LOAD testing	Performance test team: 100% Business SME's: 5–10% Development team: 25–50% Infrastructure team: 25–50%	System performance load report: business transaction timings under load System hardware performance under load	• Performance baseline tests completed • Production data loaded • Test environment; ready for load	• System meets requirements under various scenarios up to 100% of expected user load levels.
Performance DURATION testing	Performance test team: 100% Business SME's and Development team: 5–10% Infrastructure team: 10–25%	System performance duration report: business transaction timings under load System hardware performance under load	• Performance baseline tests completed • Production data loaded • Test environment; ready for duration	• System performance meets requirements throughout the execution of the duration test.
Performance THRESHOLD testing	Performance test team: 100% Business SME's: 5–10% Development team: 25–50% Infrastructure team: 25–50%	System performance threshold report: system breaking points Hardware capacity metrics	• Incremental load testing completed • Test environment; ready for threshold testing	• System failure achieved • Breaking points and bottlenecks identified
Final system tuning	Performance test team: 100% Business SME's: 5–10% Development team: 25–75% Infrastructure team: 25–75%	Final system performance report: optimal end-to-end business timings Optimal system hardware performance metrics	• Final system optimization completed • Production code frozen	• Final round of baseline, load and threshold testing completed for production code

third—or less—of the execution time requirements. So, don't think of test execution as a one-time run. It is a continuous process that involves many types of tests, many iterations, and re-runs in order to get a more efficient picture of application performance.

1. **Smoke test** validates that all performance test scripts can be replayed correctly and that the resulting impact on the system meets expectations.
2. **Baseline test** is run with only a small load on the system to verify that the system is functioning within reasonable technical parameters. Baseline tests are run at the beginning and end of the performance testing project to measure and document performance improvements.
3. **Incremental load test** validates that the system performance meets requirements as incremental load is generated on the system. (0% up to 100% of expected users).
4. **Performance duration test** verifies that the system performance meets requirements while under load (100%) for extended periods of time (e.g., up to 24 hours).
5. **Performance threshold test** identifies system breaking points by increasing load until the system no longer meets performance requirements.
6. **Final system tuning** is a final round of performance testing focusing on the tuning and optimization of the system.

Any issues that are uncovered in the testing process must be properly logged and tracked. Only through collaboration with different groups in the organization: business analysts, development managers, QA managers, performance engineers, application architects, and infrastructure specialists is it possible to rapidly resolve application or infrastructure problems.

Poor performance is not always a hardware issue. Often times an application error can cause the transaction response time to be unacceptably slow, but the problem is not discovered during functional testing. It may be identified but not fixed because it is not considered an urgent issue.

At the Retail Giant for example, TUSC consultants were continuously seeing slow response times on the product inventory query. After thorough analysis, they were able to discover that while the inventory was displaying correctly, it was loading all the query results, instead of 20 per page. It turned out, this was a known application issue, but functional testers gave it low priority and there was no fix date for it. Understanding the issue's impact on application performance, helped TUSC engineers find a solution and get much better results.

Some of the test results can be seen while the test is running. Monitoring during a performance test gives a complete picture of what's happening with the application infrastructure. If the CPU starts to max out at 100 users, that quickly indicates something is not right, and changes can be made before proceeding with the load test.

More in-depth results are available during the analysis phase. As performance engineering consultants, the TUSC team typically runs the test, performs preliminary analysis, and communicates the results to the client's DBAs, system administrators, or other experts who perform detailed analysis, diagnostics, and tuning.

TUSC consultants helped the Retail Giant coordinate the performance analysis by interpreting load test results as they relate to the business. They provided both the business view, and the cross-component view, because while it is important to understand

which technical components may be slowing down the system, IT also needs to know how these routers, switches, and servers are affecting the business. By correlating the business needs with the most critical business processes and the IT infrastructure that these processes rely on, TUSC was able to make recommendations to the client on where to focus their efforts. Performance testing is an iterative, collaborative process. After the bottlenecks are resolved, tests need to be rerun to verify the fix and document performance improvements.

All throughout the execution phase, TUSC engineers worked on improving and fine-tuning performance scripts and scenarios, and providing ongoing project reporting and documentation. Both business and technical stakeholders of the project had complete visibility into current status, results, and next steps.

Retail Giant—what TUSC learned from the load test

During the planning phase, the Retail Giant identified the SAP business function of creating and changing articles as very high risk to the business. This transaction was conducted through the SAP Web Portal using WebDynpro technology with custom-built code. The major concern for performance was the large amount of worldwide users concurrently on the system along with the large number of data fields being populated. During testing of approximately 150 concurrent users TUSC identified performance issues on certain transactions with response times averaging two minutes. This was completely unacceptable to the business.

Turns out, this was another functional issue, rather than an infrastructure problem. Part of the processing of this screen requires the system to load check table values into memory. In the case of this particular transaction, some 5,000 records have to be loaded. There is a data structure provided by the language that wasn't sufficiently sized by default. As the consultants were entering the check table values, the system was resizing the structure by a factor of 16 until it could fit all the records into it (start with size 16, double the size of a new one, copy the old one into the new one, and then get rid of the old one, repeat until the size is 5,000).

This is a very expensive operation. Multiply this by 20 or more users, the performance degradation grows quickly. After some experimentation, TUSC engineers found a data structure that gave the client tremendous performance gains. Instead of two minutes, this transaction was taking as little as ten seconds.

The creation of purchase orders and sales orders in SAP was also a risk concern for the business. These were currently being created and submitted through several different systems by the different business units. This phase of the projects would again consolidate all of the sales orders and purchase order creation within a single system, SAP. The concern was again the addition of large numbers of users concurrently in the SAP system worldwide and the performance of these transactions to create up to 700 sales orders and purchase orders in one day was critical to the business. During testing there were performance issues because of the large amount of costing conditions that were computed for each line item within an order. Load tests were run again and better performance resulted in more purchase and sales orders being created in less time.

It is important to note that not all performance issues can be easily fixed, and not all transactions can be tuned to meet the business SLAs. This was the case with the CJ20N transaction for creating a project within SAP. The response time for this transaction consistently averages over 20 seconds, which was in violation of the SLA. No matter what the TUSC engineers tried, nothing seemed to change the slow response times. Then, by working with the internal local BASIS and SAP support specialists, they discovered that this particular operation was simply a resource-intensive transaction. It was a fact known to SAP and little could be done to optimize it. Although TUSC was not able to improve the response times, by communicating and collaborating with the right experts both within the client's organization and at SAP, TUSC got the business to agree to reset its expectations—and SLAs—for this particular business process.

Performance testing best practices summary

Whether you are running a performance CoE, validating a large-scale business application implementation, or running load tests to verify performance of the newly upgraded application, these best practices can help improve the outcome of your performance testing project.

- **Implement consistent reporting, checkpoints, KPIs, and scorecards throughout the performance testing project.** It is critical to keep all team members—test leads, development managers, project managers, quality assurance (QA) managers, etc. updated on the progress of the performance testing project. Through consistent reporting and input from various stakeholders, it is easier to keep the project on track and make sure that testing goals align with the business requirements.
- **Plan for multiple types of performance test cycles (smoke tests, load, threshold, duration, failover, etc.).** Performance testing is not a single event. There are multiple types of tests that need to be run and rerun to validate and improve application performance and scalability. Each of these tests provides different types of data and a different view of the business risks associated with the application's release into production.
- **Budget for time to fix and optimize the system after uncovering performance issues.** Work with DBAs, system administrators, project managers, and business analysts to help diagnose and fix application and infrastructure issues that are impairing system performance. It may be necessary to run tests multiple times to get to the bottom of the issue, and then again to validate the fix.
- **Develop isolated performance test environment that matches production as closely as possible.** Performance testing is about realistically emulating the user experience in production. Try to create an environment that closely mimics the production system. This way, when the application goes live, there will be no surprises from unexpected performance problems involving infrastructure components that weren't included in the test environment.
- **Proactively involve resources from the business, development team, and support groups to create open communication and a sense of teamwork with all parties involved.** It is hard to over-emphasize the importance of collaboration and communication not only within the performance testing team, but between all the business and technical stakeholders. Get them involved in the project early, share project goals, progress and results, and consider their input. Without the support of the entire team, the performance testing project cannot be properly aligned with the business goals, and deliver desired business results.

Five golden rules of performance testing

1. Implement consistent reporting, checkpoints, KPIs, and scorecards throughout the performance testing project.
2. Plan for multiple types of performance test cycles (smoke tests, load, threshold, duration, failover, etc.).
3. Budget for time to fix and optimize the system after uncovering performance issues.
4. Develop isolated performance test environment that matches production as closely as possible.
5. Proactively involve resources from business, development team, and support groups. Create open communication, sense of teamwork with all parties involved.

For more information

To find out more about the topics discussed in this paper, visit the following links:

TUSC—for information about Enterprise Performance Management and other TUSC services

HP LoadRunner—for information about this industry-leading performance testing software

Quality Management Solutions for SAP Applications—to explore how HP solutions can help increase testing efficiency, lower costs and deliver strong business results for your SAP applications.

Acknowledgements

With thanks to the contributing author:

David J Kaucher
TUSC—Management Consultant
EPM—Performance Engineering
www.tusc.com

This is an HP Indigo print.

Technology for better business outcomes

To learn more, visit www.hp.com/go/QMforSAPsoftware

© Copyright 2009 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

*DRAFT*4AA2-3934ENW, February 2009