White Paper

Scapa
Technologies

Scapa® Test and Performance Platform

- Stress Testing
- Load Testing
- Capacity Test
- Soak Testing
- Scalability Testing
- Performance Testing
- Benchmarking
- Reliability Testing
- Performance Tuning
- Performance Optimization
- Bottleneck Identification
- Maximizing User Densities
- Migration Testing
- Performance Comparison
- Server Consolidation Testing
- Diagnostic Testing
- Right Sizing Systems
- Service Availability

Scapa Expedite Methodology

Testing for Improved System Performance, Capacity, Scalability, Reliability and Continuity

CONTENTS

Methodology Overview

Scapa Expedite

Scapa Expedite is a proven methodology for system performance testing, designed to resolve issues of capacity, scalability, continuity and reliability. Evolved over a number of years, by Scapa Technologies' consultants and partner organizations whose experience of performance testing is unrivalled, it minimizes the risk that the capacity of a system is insufficient and will adversely affect the performance, scalability, capacity and continuity of your mission-critical IT systems and their delivery methods. These systems should be assured in order to avoid a significant impact on their return on investment (ROI) as well as end-user morale, service delivery and ultimately, the bottom line.

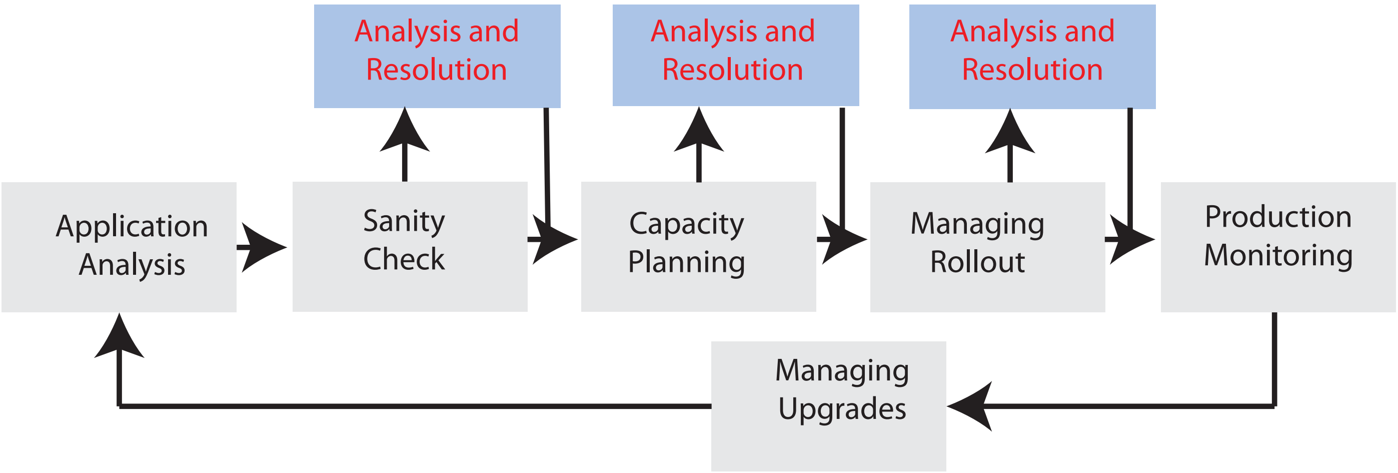
Performance testing, however, is often viewed as a complex and expensive process, in many cases completely divorced from the rest of IT implementation and management. The Scapa Expedite Methodology has been developed to re-engage performance testing with the rest of the IT function, and to offer significantly higher benefits at lower cost.

The key is to do testing sensibly. By this we mean,

- There is no point in generating a lot of information that cannot be acted upon, or information that is too late to impact on decision making processes.

- The cost of the testing process needs to be in proportion with the value to the business. As a rule of thumb, between 5% and 10% of the total cost of a project should be allocated to performance testing.

The Scapa Expedite methodology is based on a sequence of standard test activities, each of which outlines specific objectives and can be applied singly or in combination at various points in a project where most benefit can be derived. The methodology is designed to help you understand your system's capabilities and, where problems or inadequacies are uncovered, to advise on how to improve performance in a timely and cost-effective manner.



The diagram above shows the key steps of the methodology. These are described in more detail later.

A key element of the methodology is to supplement realistic tests which mimic business usage patterns on near-production systems with diagnostic tests on test systems. The earlier the tests are performed and problems are found, the easier they are to resolve, from both a technical and a contractual perspective.

Scapa Expedite - Key Steps

Application Analysis

The first objective of the methodology is to gain an understanding of the main characteristics of the systems, as most performance problems arise from the complex relationship between applications, their usage patterns and the infrastructure on which they are deployed.

The following information is requested, and from this basis a project plan will be produced:

- Technology – Architecture, Server Platform(s), Database, Middleware, Protocols, Front End, Network
- User Base – Number, Location, Type
- Application – Analysis of main system transactions to be modeled
- Implementation Plan – Phases, dates, testing windows, availability of development and test hardware, planned availability of production hardware
- Data – Existing, Migrated, Fabricated

Sanity Check

The aim of this activity is to undertake some preliminary tests to check that the system is relatively sound whilst there is still time for it to be fixed. Tests are performed on test or development hardware, which may be significantly lower in specification than the anticipated production environment. The tasks in this activity tend to focus on gross features of individual applications, and as a rule of thumb, the following five very simple transactions (tested in the following order) will show up most of the fundamental flaws in an application architecture:

- **Login:** A transaction which logs a user into the system, authenticates them via password etc., and then logs them out.
- **Read:** A simple transaction that retrieves a document or a customer's details, or anything that involves accessing persistent data in the system.
- **Update:** A transaction that changes a document and stores it, or updates a customer's data or something of that nature.
- **Search:** Most applications have a facility to search for something in the database, e.g. a product with specific criteria or a document with a specific subject.
- **Batch:** Many applications have a batch process that updates the database. We run this to find out how long it takes. If it is anticipated that batch and user processes will run simultaneously, we test the batch process and at least the update transaction at the same time to check for contention.

These preliminary tests will determine whether the performance of the test environment is functioning at a level that inspires confidence. For example, if we expect to service 1000 users on 10 application servers in production, and we can log 100 users on through one test server, confidence levels are reasonably high. If, however, only 10 can log on, there is a problem. It is worth stating that it is also useful to build tests around transaction rates, (i.e. updates per second etc.) and not just user count.

At this point it is also worth looking at interference amongst transactions in individual applications and, in the case of systems where multiple applications are hosted on the same physical hardware, amongst the applications that are likely to be hosted on the/each machine.

If the sanity check shows that there are performance issues with the application then we would undertake some problem analysis and resolution (outlined in a later section). Otherwise, with confidence levels high, we can proceed to capacity planning.

Capacity Planning

The objective of this phase is to determine if the proposed hardware configuration can support the planned number of users and to advise on the maximum capacity of the system. At this stage it is usually still possible to work with relatively simple transactions (login, read, update, search), rather than with complex business transactions mimicking real user behavior. In our experience, the system is unlikely to be in its final state, so the process is necessarily slightly approximate.

In this stage, we aim to understand the transaction levels that the server(s) will be required to support and the way that the transaction rate scales as more servers (or more CPUs) are added. We start the process with individual transactions, then a mix of transactions in an application and finally, if appropriate, multiple applications. Particular attention is paid to update transactions as these are least likely to scale.

A key output of this phase is a test system calibration metric, i.e. a formula for predicting the capacity of the production environment based upon the capacity of the test environment. If the production environment is available, this can be calculated directly from performance test results on the two environments. If not, an extrapolation can be performed based on scaling up within the limits of the test environment.

The test system calibration metric is usually quite stable during a project and relatively independent of the transaction actually being performed but if, for example, the search transaction scales better than the update transaction, different formulas can be used for each transaction type.

If the Capacity Planning exercise shows that there are performance issues with the application then the next activity is Problem Analysis and Resolution. Otherwise we proceed to Managing Rollout.

Managing Rollout

The objective of this activity is to establish that the application will perform in the “real world”. We do this by using a mix of real business transactions representing real system usage and which are performed at a rate which exceeds the transaction rate anticipated for the system. The tasks in this activity include:

- Phased rollout tests - testing before each phase of a multi-site or multi-application rollout.
- Local/remote network tests - to check the behavior of the system in remote sites in comparison to the behavior on the LAN.
- Multi-site comparative network tests - to check that different remote sites experience similar performance.
- Failover tests - testing how the system behaves when a server is switched off or a disk is pulled in RAID etc., and also testing disaster recovery systems.

With the test calibration metric derived during the capacity planning stage, it is possible to do a lot of this work on the test environment. It is also usually possible to identify a number of business transactions for which the anticipated transaction rate is unlikely to cause any problems on the test environment. These can be removed from the transaction mix to simplify the testing process, if required.

Before initial rollout it is important to test a range of transactions on the production environment, concentrating on the ones thought likely to be problematic. This also offers an opportunity to confirm and/or make any adjustments to the test calibration metric derived earlier. Our methodology counsels running one or two simple diagnostic transactions at this point and correlating the performance level to that of the mix of business transactions. This is because in the later phases of phased rollouts, the opportunities for access to the production system for testing are usually reduced.

Again, if at any stage during the managed rollout, performance issues with the application are discovered, then the next activity is problem analysis and resolution. Otherwise we proceed to an optional production monitoring stage.

Production Monitoring

As stated before, this is an optional activity within the methodology. Monitoring the system in a live environment will provide an insight into its performance characteristics and can be very helpful where the systems’ performance is guaranteed against a Service Level Agreement (SLA). It can also provide an early warning alarm when performance parameters are breached. In our experience, the best transactions to use for early-warning purposes are, typically, the simple transactions developed for the sanity check because their performance characteristics are best understood. However, other, more complex business transactions may be specified as part of the SLA and these can also be monitored.

Problem Analysis and Resolution

The objective of this activity is to pinpoint the location of the performance bottleneck, to advise on resolving the problem and to retest until satisfactory performance is achieved. Due to the complexity of most application environments there are often multiple routes to resolving a problem. For example, adding or reconfiguring hardware, reconfiguring software (typically at the database or system level), applying system performance enhancement tools, changing the application itself or, if necessary, changing the usage patterns of the system.

We would advise that diagnostic testing typically works best with simple transaction sets on test systems under intense monitoring from third-party tools (e.g. database tools or Perfmon). In early stages it is common to use the simple (login/read/write/search) transactions developed for the sanity check. Later, we would use real business transactions and go through a sequence of simplification/expansion cycles where transactions are pared down until the problem is diagnosed and a fix is tested. If the fix solves the problem, we would then use further, more complex transactions to locate any additional issues, and the cycle iterated until all problems are resolved.

During this process, systems performance metrics are monitored, such as CPU, memory and network statistics and others. Using these tools, it is often possible to determine which element of the architecture is limiting the performance of the system. In the case of multi-tier systems, the problem element within a specific tier can be pinpointed.

It is the case that some systems are simply CPU-bound, or memory-bound, or bandwidth-bound. In many cases this can be fixed by adding hardware, or in extreme cases by application fixes.

We find that the majority of the more elusive system problems are caused by serialization through locks (either database-level locks or application-level locks implemented through database tables), or through inadequate thread pool sizes in middleware, or serialization of database drivers.

Managing Upgrades

Once the system is up and running, we advise close management of hardware and software upgrades. The approach to perfomance testing system upgrades includes most of the activities listed above, with the added benefit of comparing performance figures with a previously recorded benchmark, and an existing test calibration metric.

Consulting Services

Scapa Expedite Methodology

The Scapa Expedite Methodology has been developed and is used by our highly skilled consultants and partner organizations working worldwide, either on-site or remotely to deliver a variety of performance testing activities across a wide-range of applications and environments, from off the shelf to custom applications.

Scapa consultants and partners specialize in performance testing of BMC Software® Remedy-based solutions, such as the ITSM® suite of applications, systems based on the Mendix™ application platform, web applications and desktop virtualization deployments (e.g. Citrix® XenApp ™/XenDesktop®, VMware® View™ and Microsoft® Remote Desktop Services® and Terminal Server®).

Our services include:

- End-to-end Testing (load, stress, capacity, performance, scalability, reliability, soak)
- Scalability Profiling
- Bottleneck Identification
- Benchmarking
- Capacity Planning
- Performance Comparison
- Performance Tuning
- Performance Optimization
- Maximizing User Densities
- Migration Testing
- Server Consolidation Testing
- Service Availability
- Diagnostic Testing
- Desktop Virtualization
- Right Sizing Systems
- Proof-of-concept

The following are examples of the types of projects our consultants and partners regularly complete for customers:

Health Check-up

Where there are concerns that systems will not accommodate future demands, with their wide-ranging experience in projects spanning many verticals and an even greater variety of IT systems, our consultants and partners can conduct thorough reviews of current and future capacity requirements. By running a variety of tests to quantify the performance, scalability and capacity of the systems, they are able to identify the available headroom, thereby minimizing potential performance risks.

Training Services

For businesses that want to be self-sufficient with respect to performance testing, our training course is adapted and personalized to customers' specific IT systems. Our consultants and partners help customers run tests on their systems with their specific application mix to kick start the required testing activities. This typically happens within five days.

Pre-Flight Assurance Testing

The philosophy behind Scapa Expedite is to promote continuous and on-going testing throughout the application life cycle. However, for many businesses, testing can be a last minute activity. Our expert consultants and partners regularly deliver pre-flight tests in 1 to 3 weeks, identifying problem areas quickly and providing businesses with the peace of mind that performance, stability and capacity-related issues are resolved prior to rollout.

Fire Fight Services

The root cause of system reliability, performance, scalability or capacity issues can be elusive and can cause untold damage to your business. Scapa consultants and partners have substantial on-site project experience and have helped some of the biggest companies in the world to identify underlying system issues when all internal resources have been exhausted.

Conclusion

Performance testing is often viewed as a complex and expensive process and is often separated from IT systems implementation and management. The Scapa Expedite Methodology has been developed to re-engage performance testing with the rest of the IT function, and to offer significantly higher benefits at lower cost. It is flexible enough to be adapted to a customer's requirements and is designed to provide an insight into systems' capabilities and, where problems or inadequacies are uncovered, to advise on how to improve performance in a timely and cost-effective manner.

About Scapa Test and Performance Platform

Scapa Test and Performance Platform (TPP) features real data utilization, quick test creation, and dynamic multi-load manipulation among its capabilities, making it ideal to emphasize the influence of user data and for rapid test creation and real time analysis which are key concepts of an effective testing approach.

For a full list of features and to learn more about the Scapa Test and Performance Platform visit **http://www.scapatech.com**