

Testing needs to get into the real world

Paul Down, a senior consultant at Embarcadero Technologies, says goals-based testing eliminates cost and time burdens common to traditional performance testing methodologies

Almost every application suffers from performance problems at some time. It is difficult to find a well-tuned application and even harder to find one that stays that way in the face of growing user loads, network traffic, and data volumes. Neglecting these performance problems can lead to not only poor end-user experiences, but also application outages and that means real business costs.

The goal of a performance test is, therefore, to determine how a new or existing database, system or application might respond under real world stress and how well it can scale. It is also critical that testers discover whether resources are used effectively. Finally, before any new system can be deployed, organisations must be confident that their existing infrastructure can cope with the new workload, either by harnessing the resources they have or acquiring new ones, such as additional CPUs, and so on.

Today, given the complexity of the challenge, performance testing is often an incredibly painstaking, expensive and time-consuming burden.

Testing times

Every engineer understands the frustrations of traditional approaches to performance-based testing. The tester reaches the end of a day's test cycle only to discover that many hours ago, the test failed because a critical condition was not included in the original test criteria. While this will be annoying for individual engineers, it is the greater business cost in terms of time and effort that is a more pressing concern.

The problem is that typical performance-based testing routines only really provide engineers with basic load-testing functionality; they provide a hypothetical evaluation of systems under test. Provided systems can handle a given workload the test will continue, regardless of any other conditions that might in reality justify an early test termination. It would be much better if performance tests reflected the complexity of real-world production activities by systematically evaluating the performance of applications under stress to identify potential bottlenecks before these systems went into production.

However, the over-simplistic nature of current approaches to performance-based testing means that an engineer might have to run the same test tens of times to achieve such a result and as any senior test manager will tell you, this is a very inefficient method of testing.

Today, the pressure on testers to deliver enhanced business performance has brought the inadequacies of traditional load-based, performance testing to the fore. More than ever before, organisations are completely dependent upon an ever-growing enterprise IT infrastructure to perform basic operational functions. Any system outages or performance drops have a dramatic and costly knock-on effect across the enterprise. Now, because of the time-consuming and expensive nature of current methodologies, engineers are calling for a testing revolution.

Stakeholder pressure for change

The business user of a proposed application or system and the technical team responsible for its ultimate deployment both face difficult challenges today.

Business sponsors must specify performance parameters from the end user's perspective. For example, considerations such as ensuring fast transaction response times, as well as the number of users the system can support must be incorporated into test procedures.

Once business sponsors have defined their requirements, the technical team can begin to calculate the impact of new development projects on overall system performance. System performance is the foundation for a positive end-user experience and refers to acceptable levels of resource utilisation when under stress. Since performance testing is the process of applying load to the system to determine how it reacts, these parameters must be included in the performance testing process. Issues such as CPU utilisation become critical when safe thresholds are exceeded, because if the CPU cannot cope with its workload the end-user experience will be poor.

Ultimately, the business sponsor and the technical team must find a way to achieve

positive end-user experience without taxing the infrastructure. This places a heavy burden on performance test engineers. They are tasked with determining if the application infrastructure and the performance test infrastructure are performing well, which is challenging even under normal circumstances. It can become almost impossible when there are many performance thresholds, including many different transaction response times and system performance statistics. Goals-based performance testing enables test engineers to overcome these difficult obstacles.

The goals-based testing difference

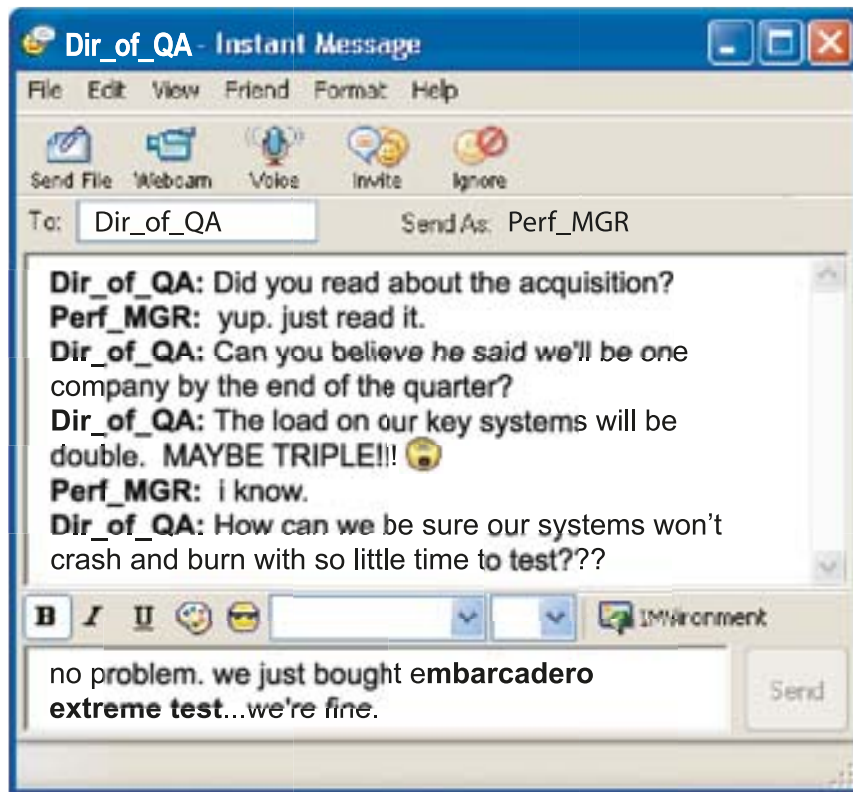
Whereas traditional performance testing can evaluate test systems for simple criteria such as load, the ideal would be that test routines should reflect the challenges of the production environment. In other words, tests should be able to evaluate systems for multiple criteria, including users, activities and data. Traditional testing methods do not offer this multi-dimensional test functionality.

The goals-based testing approach to performance testing delivers the multi-layered capabilities that test engineers are calling for today. This approach does not just operate on the principal of a simple load-based threshold; it allows testers to specify a number of critical conditions, which if attained, will facilitate a subsequent action, such as the premature conclusion of a test, which saves time.

The key difference between established performance testing and goals-based performance testing approaches is that the goals-based approach enables users to ask critical performance questions before the test begins. This pro-active test methodology enables testers to set thresholds and parameters during test configuration to ensure the test results meet all the defined requirements. This dramatically accelerates the testing process because tests can be aborted should thresholds be exceeded.

A second major benefit is that goals-based performance testing gets much closer to simulating real production environment conditions because it supports complex, interlinked test

With Embarcadero Extreme Test, the impossible becomes *possible*.



Extreme Test allows QA teams to use a goals-based approach to create realistic user scenarios that produce more accurate results in a fraction of the time. For more information or to evaluate Embarcadero Extreme Test, please visit www.embarcadero.com/proftester.asp or contact **Chris Birks** on **+44 1628 684443** (chris.birks@embarcadero.co.uk)



Goals-based testing in action – a scenario

Test criteria

- 1,000 concurrent users
- 8-second response time 90% of the time (90th percentile)
- Less than 85% CPU utilisation
- Monitor complete application infrastructure (database, Web server, application server, etc.)

Three phase test

Phase 1: Ramp Up until we exceed eight-second response time or 85% CPU utilisation. If response times reach eight seconds with fewer than 1,000 users, exit the test. If CPU utilisation exceeds 85%, exit the test. If the test exceeds eight seconds with more than 1,000 concurrent users, exit the phase.

Phase 2: Test the system at a constant load. If CPU utilisation exceeds 85%, exit the test. When the phase runs for 15 minutes, exit the phase.

Phase 3: Ramp down until there are no more users.

Results

The engineer ran the test and it exited at Phase 1 because CPU utilisation exceeded 85%. This was determined quickly – the engineer did not have to wait until the test was finished. Moreover, the performance engineer configured database and application server monitoring as one of the test parameters, which helped immediately identify bottlenecks in the database layer and corrected it. The next time, the test ran to completion.

criteria. Unlike load-based testing, users can set a range of thresholds and realistic user scenarios that provide the criteria for success (or failure), before the test begins.

Testers can identify key test goals then set criteria accordingly — defining acceptable thresholds for the various parts of the system, such as response times, CPU utilisation and so on. This approach removes the typical try/repeat cycle that is so common with most load-testing processes. It saves test professionals significant amounts of time they would otherwise have spent waiting for a standard test to finish, only to discover they have an invalid result.

More valuable results

Generally, the goal of performance testing is to determine how many concurrent users your system can handle while continuing to deliver a positive user experience. To meet this goal, the test must answer some important questions:

- 1 What are acceptable user performance levels?
- 2 How many concurrent users are required?
- 3 How fast should the site respond?
- 4 What are acceptable levels of system utilisation? (CPU, memory, network, disk and so on).

Traditional performance testing makes it difficult to answer these questions with any depth of detail. To reach the required degree of testing ‘granularity’, organisations must specify and program these goals into the performance test itself. With traditional tests, engineers

The goals-based test indicated the page delivery time for 1,000 concurrent users, and it indicated the number of concurrent users required to deliver pages inside eight seconds.

By taking a goals-based approach to this scenario, the engineer could accurately answer the question “How many concurrent users can the system support with acceptable performance?” The goals-based approach also minimised the time required to adjust and reconfigure time-based performance tests to answer the same question.

For example, in a time-based performance test, a test schedule is configured. In a goals-based performance test, a load model is configured and thresholds are defined so that instead of applying a time-based transition from increasing users to a constant load, the ramp-up continues until a threshold is exceeded. The result is that either the thresholds are exceeded, the phase ends and the next phase starts, or the whole test terminates because performance is unacceptable and there is no reason to continue the test.

The benefits

Overall, goals-based testing as in the scenario above reduces the total time required to conduct a system evaluation. Moreover, test engineers no longer have to wait around for tests to abort. Most importantly of all, tests can be carried out in a way that reflects how systems are used in the production environment – a first for any kind of performance testing.

‘guess’ the number of users the system can support. The test is then repeated until the four questions (above) are answered. Unfortunately, it can often take much iteration before traditional performance tests achieve their goals.

In stark contrast, goals-based testing enables the technical team to determine the exact requirements for the system before testing begins, and provides more sophisticated answers to the four critical questions above. Effectively, this means testers can ask questions such as: ‘How many users can my application infrastructure support with X second transaction response times without risk of system overload?’ and ‘How can I feel confident that the results returned are accurate?’

Goals-based tests include a series of phases and exit conditions. Within a phase, if the test exceeds any of the designated thresholds, the pre-specified exit actions are initiated. In addition, users can set thresholds that will simply stop the test if exceeded. The user can configure the test with many different thresholds.

For more information about how goals-based testing works in reality, see *Goals-based testing in action – a scenario* above.

Time to get in the real world

Put simply, goals-based testing is a more powerful approach to performance testing because it focuses on modelling the real world. Practically, every system will suffer some kind of performance deterioration or outage at some time.

Yet, today’s end user has zero tolerance for such outages and slow performance. Explosive

growth in technology means increasing system stress, and this leads to potential performance problems across all system components, including the database, web servers, application servers and so on. Neglecting performance problems is costly and laden with risk. Poor performance leads to downtime, outages, and ultimately, a poor end user experience. To avoid this, companies must adopt a broader performance assurance programme in which the performance and stability of critical applications can be constantly evaluated and optimised after they have reached production.

Goals-based testing provides a powerful framework for these broader performance testing and evaluation processes. It enables test engineers to create more sophisticated test environments that reflect the key challenges of today’s highly complex and disparate database, application and system heavy architectures. So much time and money is wasted because of the high volume of repeated tests that result from the inadequacies of established performance-testing methods. With a goals-based approach to testing, organisations can test smarter and more efficiently. They can be pro-active and make allowances for many possible test failures even before testing has begun. Technicians and test engineers can also use the approach to play a fundamental role in overall capacity planning. Finally, with goals-based testing, you can bring your activities into the real world. **PT**