Recommendations for Web application performance benchmarks

Asked 16 years, 3 months ago Modified 10 years, 5 months ago

Viewed 12k times

▲

7

▼

I'm about to start testing an intranet web application. Specifically, I've to determine the application's performance.

Please could someone suggest formal/informal standards for how I can judge the application's performance.

performance testing web-applications benchmarking

Share

Improve this question

Follow

edited Jul 14, 2014 at 8:39

Sid M

4,354 ● 4 ● 32 ● 51

asked Sep 3, 2008 at 14:55

JonnyGold

871 ● 6 ● 15 ● 20

4 Answers

Sorted by: Highest score (default)

▲

Use some tool for stress and load testing. If you're using Java take a look at JMeter. It provides different methods

8

✓

to test you application performance. You should focus on:

- **Response time**: How fast your application is running for normal requests. Test some read/write use case
- **Load test**: How your application behaves in high traffic times. The tool will submit several requests (you can configure that properly) during a period of time.
- **Stress test**: Do your application can operate during a long period of time? This test will push your application to the limits

Start with this, if you're interested, there are other kinds of tests.

Share Improve this answer Follow

answered Sep 3, 2008 at 15:04

Marcio Aguiar
14.5k ● 6 ● 40 ● 42

4

"Specifically, I have to determine the application's performance...."

This comes full circle to the issue of requirements, the captured expectations of your user community for what is considered reasonable and effective. Requirements have a number of components

1. General Response time, " Under a load of .... The Site shall have a general response time of less than x, y% of the time..."

2. Specific Response times, " Under a load of .... Credit Card processing shall take less than z seconds, a% of the time..."
3. System Capacity items, " Under a load of .... CPU|Network|RAM|DISK shall not exceed n% of capacity.... "
4. The load profile, which is the mix of the number of users and transactions which will take place under which the specific, objective, measures are collected to determine system performance.

You will notice the the response times and other measures are no absolutes. Taking a page from six sigma manufacturing principals, the cost to move from 1 exception in a million to 1 exception in a billion is extraordinary and the cost to move to zero exceptions is usually a cost not bearable by the average organization. What is considered acceptable response time for a unique application for your organization will likely be entirely different from a highly commoditized offering which is a public internet facing application. For highly competitive solutions response time expectations on the internet are trending towards the 2-3 second range where user abandonment picks up severely. This has dropped over the past decade from 8 seconds, to 4 seconds and now into the 2-3 second range. Some applications, like Facebook, shoot for almost imperceptible response times in the sub one second range for competitive reasons. If you are looking for a hard standard, they just don't exist.

Something that will help your understanding is to read through a couple of industry benchmarks for style, form, function.

- TPC-C Database Benchmark Document
- SpecWeb2009 Benchmark Design Document

Setting up a solid set of performance tests which represents your needs is a non-trivial matter. You may want to bring in a specialist to handle this phase of your QA efforts.

On your tool selection, make sure you get one that can

- Exercise your interface
- Report against your requirements
- You or your team has the skills to use
- You can get training on and will attend with management's blessing

Misfire on any of the four elements above and you as well have purchased the most expensive tool on the market and hired the most expensive firm to deploy it.

Good luck!

Share Improve this answer

Follow

answered Sep 29, 2011 at 14:44

James Pulley

5,682 ● 1 ● 15 ● 14

3

To test the front-end then YSlow is great for getting statistics for how long your pages take to load from a user perspective. It breaks down into stats for each specfic HTTP request, the time it took, etc. Get it at http://developer.yahoo.com/yslow/

Firebug, of course, also is essential. You can profile your JS explicitly or in real time by hitting the profile button. Making optimisations where necessary and seeing how long all your functions take to run. This changed the way I measure the performance of my JS code. http://getfirebug.com/js.html

Share Improve this answer Follow

answered Sep 3, 2008 at 15:00

David McLaughlin
5,178 ● 4 ● 35 ● 36

3

Really the big thing I would think is response time, but other indicators I would look at are processor and memory usage vs. the number of concurrent users/processes. I would also check to see that everything is performing as expected under normal and then peak load. You might encounter scenarios where higher load causes application errors due to various requests stepping on each other.

If you really want to get detailed information you'll want to run different types of load/stress tests. You'll probably want to look at a step load test (a gradual increase of users on system over time) and a spike test (a significant number of users all accessing at the same time where almost no one was accessing it before). I would also run tests against the server right after it's been rebooted to see how that affects the system.

You'll also probably want to look at a concept called HEAT (Hostile Environment Application Testing). Really this shows what happens when some part of the system goes offline. Does the system degrade successfully? This should be a key standard.

My one really big piece of suggestion is to establish what the system is supposed to do before doing the testing. The main reason is accountability. Get people to admit that the system is supposed to do something and then test to see if it holds true. This is key because because people will immediately see the results and that will be the base benchmark for what is acceptable.

Share Improve this answer Follow

answered Sep 3, 2008 at 15:28

kemiller2002

115k ● 28 ● 199 ● 253