

# Load Testing Worksheet

**Source:** Drupal Performance Agency

**# Pages:** 5

**Last Modified:** 05:26 PM Jan 21, 2008

**License:** Creative Commons Attribution-ShareAlike2.0

**Description:** *Tips on load testing your web site.*

*This worksheet was designed by the Drupal Performance Agency. For more information, email [perf@tag1consulting.com](mailto:perf@tag1consulting.com).*

---

## **Load Generating Tools**

*There are many free and open source tools that can be used to generate loads on your website. Here are a few that we recommend.*

### I. ApacheBench (ab)

#### 1. Overview:

“ab is a tool for benchmarking your Apache Hypertext Transfer Protocol (HTTP) server. It is designed to give you an impression of how your current Apache installation performs. This especially shows you how many requests per second your Apache installation is capable of serving.”

#### 2. Details:

As “ab” is very simple to use and is included with the Apache web server, it is a very popular choice for generating load tests. The two most common options to set when running “ab” are “-n” for setting the number of page requests to make, and “-c” for setting the concurrent number of page requests to make at the same time. For example, to load the page at “<http://localhost/>” 1000 times, 10 requests at a time, simply run:

```
$ ab -n1000 -c10 http://localhost/
```

The tool will then generate a full page of useful output, summarizing the results of your test. This will include statistics on how long each page request took, how much data was transferred, how many connections results in errors, and what fraction of the time was spent connecting, processing the request, and waiting for a response.

The above example will generate a load of anonymous users. If you want to generate a load of logged in users, you can use ab's “-C” flag to specify a cookie. You will first need to log into your website with a web browser. Then, find the session ID that was assigned to your

**Modified On:** 05:26 PM Jan 21, 2008

logged in user. The simplest way to find both the cookie name and cookie value is to use your web browser's built-in cookie tool.

3. Links:

- a) *Project's web page:* <http://httpd.apache.org/docs/trunk/programs/ab.html>
- b) Benchmarking Drupal code with ab: <http://drupal.org/node/79237>

## II. Siege

1. Overview:

“Siege is an http regression testing and benchmarking utility. It was designed to let web developers measure the performance of their code under duress, to see how it will stand up to load on the internet. Siege supports basic authentication, cookies, HTTP and HTTPS protocols. It allows the user hit a web server with a configurable number of concurrent simulated users. Those users place the webservice 'under siege.’”

2. Details:

Siege is particularly useful for simulating a load on your website. You can set up lists of pages for siege to visit, and forms for siege to automatically submit. As with ab, siege supports a “-c” flag for setting the number of concurrent sessions to open at the same time. Other common options include “-f” for specifying a list of URLs that siege should access during the load testing, “-i” for Internet mode telling siege to randomly access the listed URLs, “-d” for adding a random delay in seconds between individual page requests, and “-t” for telling siege how long it should run. For example, to simulate ten users randomly loading website pages for 5 minutes with a 0-3 second delay between page loads, run:

```
siege -c 10 -i -t 5m -d 3 -f url_list.txt
```

Your URL list can be manually created, or you can use the specially created Sproxy to harvest real URL information from your live website.

3. Links:

- a) *Project's web page:* <http://www.joedog.org/JoeDog/Siege>
- b) Sproxy project page: <http://www.joedog.org/JoeDog/Sproxy>

## III. Apache JMeter

1. Overview:

“Apache JMeter may be used to test performance both on static and dynamic

resources (files, Servlets, Perl scripts, Java Objects, Data Bases and Queries, FTP Servers and more). It can be used to simulate a heavy load on a server, network or object to test its strength or to analyze overall performance under different load types. You can use it to make a graphical analysis of performance or to test your server/script/object behavior under heavy concurrent load.”

## 2. Details:

JMeter provides a graphical user interface for building load tests. It is more complicated to get set up and running, but also offers numerous powerful options. The links section below includes references to a nice write up about load testing Drupal with JMeter.

## 3. Links:

- a) *Project's web page:* <http://jakarta.apache.org/jmeter/>
- b) Load testing Drupal with JMeter, part 1:  
<http://www.johnandcailin.com/blog/john/load-test-your-drupal-application-scalability-apache-jmeter>
- c) Load testing Drupal with JMeter, part 2:  
<http://www.johnandcailin.com/blog/john/load-test-your-drupal-application-scalability-apache-jmeter%3A-part-two>

## IV. wget

### 1. Overview:

“GNU Wget is a [free software](#) package for retrieving files using HTTP, HTTPS and FTP, the most widely-used Internet protocols. It is a non-interactive command line tool, so it may easily be called from scripts, cron jobs, terminals without X-Windows support, etc.”

### 2. Details:

The wget command line tool has the advantage of being very easy to script. One very quick way to generate a load with wget is to use the “-r” to recursively download an entire website. Wget is designed for downloading, but if you just want it to generate a load you can set the “--spider” option. For a complete list of options, run “wget --help”.

### 3. Links:

- a) *Project's web page:* <http://www.gnu.org/software/wget/>
- 

## ***Load Testing Tips***

### **I. Test in a realistic configuration**

When load testing your website, even if you're only interested in some specific new code that you have written or downloaded, it's wise to simulate your actual configuration as much as possible. This includes being sure that the versions of Apache, MySQL, PHP and Drupal running on your test server(s) is the same as what is on your production server(s). Also try to use as similar a configuration for each as possible, including enabling the same Apache modules, allocating the same size of MySQL buffers, enabling the same PHP options, and enabling the same Drupal modules. It is often true that your code will interact with other code and configuration options in ways that you don't expect. By spending the time now making your development environment match your production environment, you save your time later deploying then rolling back code that breaks in unexpected ways.

### **II. Testing with realistic data**

Ideally, load in a very recent snapshot of your real website's database tables into your development database. If you instead test with a subset of your real data or with automatically generated data, you may end up optimizing the wrong things. This is especially important if testing out code that involves making SQL queries.

### **III. Logged-in users versus anonymous visitors**

When you enable Drupal's page cache, much of the work involved in serving pages is optimized away for anonymous visitors. This is not true for logged-in user's, so it's important to simulate both types of user's. Furthermore, on some websites different Roles can put a very different load on your servers, so be sure to include to test with all of your roles.

### **IV. Posting content**

When new content is posted to your website, all the various caches will usually be flushed. It is important to simulate a real type of load, posting whatever types of content your users will actually be posting in real life. This will include things like creating new users, posting comments, and posting new nodes.

V. Profiling load effects

1. Drupal devel module:

The Drupal devel module is a very powerful tool, useful for profiling your website while under a load. It can help you to isolate where bottlenecks are.

*Project URL:* <http://drupal.org/project/devel>

2. MySQL slow query log:

You can enable MySQL's slow query log to generate a log of any database queries that take longer than a configurable threshold of time.

*Documentation URL:* <http://dev.mysql.com/doc/refman/5.0/en/slow-query-log.html>