|  |
| --- |
| Guidelines – Load Test (Performance - NFR) |
| V-1.0 |

Table of Contents

[1 Objective 3](#_Toc10459032)

[2 Introduction 3](#_Toc10459033)

[2.1 Target Audience 3](#_Toc10459034)

[2.2 Reference 3](#_Toc10459035)

[3 General Recommended Items to Improving Drupal site performance 4](#_Toc10459036)

[4 Managing Page Wise Performance 5](#_Toc10459037)

[4.1 Home Page 5](#_Toc10459038)

[4.2 Media Pages 5](#_Toc10459039)

[4.3 Data Pages 5](#_Toc10459040)

[5 Common Problems during Performance Test 6](#_Toc10459041)

# Objective

Load testing is a kind of Performance Testing, which determines a system's performance under real-life load conditions. This testing helps determine how the application behaves when multiple users access it simultaneously

This document provides guidelines to increase the performance of Drupal website

# Introduction

## Target Audience

Development Team will be the target audience for executing load test.

## Reference

|  |  |  |
| --- | --- | --- |
| **Sr. No** | **URL** | **Description** |
| 1 | https://www.drupal.org/docs/7/managing-site-performance-and-scalability/optimizing-drupal-to-load-faster-server-mysql |  |
|  |  |  |
|  |  |  |

# General Recommended Items to Improving Drupal site performance

Follow the below general action items to increase overall performance of the website

1. Keep the Drupal core, contribute module and themes updated to latest version.
2. Enable Page, Block and Views Caching
3. Enable Aggregate javascript and css files - Front End Improvements (minify js and css for page speed)
4. Include java script files in footer region
5. Include/split JS and CSS files specific to page or feature
6. Disable unnecessary modules. Every module adds to the amount of code that needs to be available for a page load, increasing the number of lookups. Wherever possible use a generic module in place of multiple module that does specific functionalities.
7. Disable DB logging and Use syslog in production environment
8. Reduce 404 Errors - <http://www.brokenlinkcheck.com/>
9. Fast 404 Responses - <https://drupal.org/project/fast_404>
10. Set Minimum Cache Life Time Value to a higher number and use cache clearing modules to clear the caches for specific pages
11. PHP Caching – Enable PHP caching by using .htaccess file
12. Use following contribute module to enable more caching options
    1. Memcache
    2. Redis
    3. File Cache
13. Use search engines Apache Solr or Elastic Search to get quick response
14. Enable Opcache (PHP OPcode caching) in PHP.ini file (Suggesting for Static pages only)

[opcache] zend\_extension=php\_opcache.dll

;Determines if Zend OPCache in enabled

opcache.enable=1

opcache.memory\_consumption=128

opcache.interned\_strings\_buffer=8

opcache.max\_accelerated\_files=4000

opcache.revalidate\_freq=60

opcache.fast\_shutdown=1

1. Avoid using PHP Filter module
2. Use Reverse proxy cache like Varnish
3. Recommended max\_execution\_time will be 30 seconds for any frontend service request.

# Managing Page Wise Performance

Below points will guide you to build the pages, with good performance.

## Home Page

* 1. Use Static node content as home page.
  2. Set Home page URL in Drupal core configuration, this core configuration will help to identify and load the homepage in fast response
     + Navigate to Configuration > System >Basic site settings (admin/config/system/site-information). Under Front page, replace /node with the name of the page you would like to make the home page. To use the home page that was previously created, provide its path /home.
  3. For dynamic content, build most functionality from back end and use less custom code to achieve. For look & feel, you can override Twig template files
  4. Avoid loading php filtered data, large size images/videos in home page
  5. Avoid loading external services, use ajax call if required
  6. Avoid loading external libraries and js files

## Media Pages

* 1. Handling at server level Compress Image - <https://www.drupal.org/project/imageapi_optimize>
  2. Image Optimization – Maintain the compression of ration of an image files. Higher % values will give image quality but bigger file size
  3. Use external video cloud (Eg – Brightcove)
  4. Use CDN (Content Delivery Network) concept like cloudflare to render media files
  5. Avoid <table> html tag

## Data Pages

* 1. Use Views to render the data
  2. Avoid using complex query and multiple iterations
  3. Use pagination
  4. Avoid multiple API calls in single page and don’t use API calls in loop

# Common Problems during Performance Test

Below problems are reported during LT execution, by using Blazemeter tool. Resolutions are provided below.

1. **Average load times < 500 ms**
   * Ensure that *Retrieve all embedded resources* is enabled and set to *6 at a time*.
2. **Average load times near or greater than 5 seconds**
   * Verify that varnish, drupal cache, and memcache is working on the site
   * Ensure that an *HTTP Cache Manager* is added
   * Ensure that your test is requesting the site over https and not defaulting to http
   * Try blocking third party embedded resources using a regex like ^((?!brightcove|googleapis|doubleclick|addthis|janrain).)\*$
   * Disable unnecessary features like graphs or result trees
   * Ensure basic caching implemented.