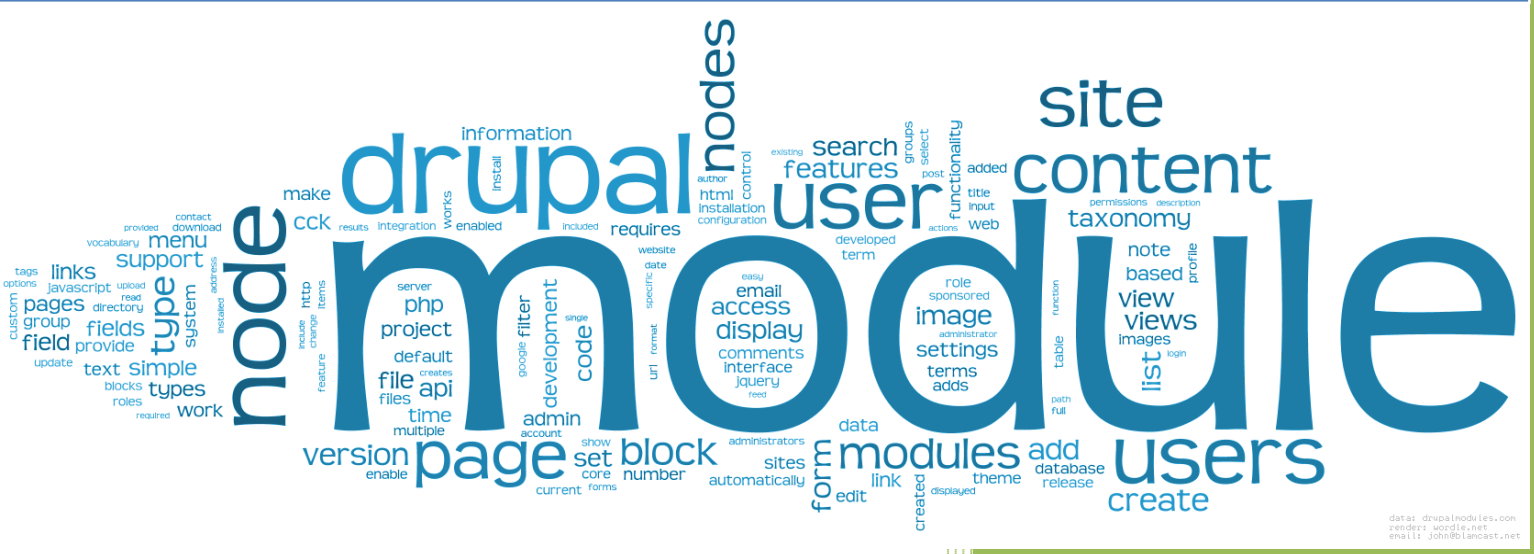
Performance Optimization and tuning in DRUPAL based sites

- A brief run-through Whitepaper



BHARGAVI LAKSHMI KRISHNA MOORTHY RAO

HCL TECHNOLOGIES

DEC 12, 2011

Contents

Introduction

Performance optimization in Drupal based sites- this paper provides a brief run-through on the analysis made on the factor of performance optimization of DRUPAL based sites. DRUPAL, basically as content management system also extends its wings as "Framework solutions" for the web based applications.

Overview &Terminologies

Performance

Software Performance could be many things like Short Response Time, High throughput, Low utilization of Computing resources , stability etc. under a workload.

Scalability

Scability in the perspective of the application's capacity as a Vertical unit and Horizontal Unit. A typical view of Horizontal Unit will be load balancing of the application by deployment of many servers working parallel and Vertical Unit being individual work of Request/Response per server.

Load Balancing

Load Balancing is the deployment of more than one server to process the request from the client machines. This method of deployment of multiple server across enriches with fault tolerance, high availability and ultimately the performance.

Performance Optimization / Tuning

Performance Optimization / Tuning is the methodology of improvising the system or rather the application performance from what is at present.

Multiple Servers

Deployment of the application across multiple servers in the case of Load Balancing uses various algorithm for request handling. One of them is Round Robin. This concept of Multiple Servers denotes usage of one backend server i.e. Database server and multiple web servers.

WEB Site Statistics

Web Site Statics is the data count on number of client request made to the server. There are several metrics derived to analyze the website traffic. Few of them are

- Hits (Request for every page, graphic, video, css, js file etc.)
- Page views (Generating pages like a node, a taxonomy list)
- Visits / Unique visits (A visit is defined as a series of page requests from a uniquely identified client with a time of no more than 30 minutes between each page request. Generally advertisers care about this factor)

Web Analytics Reporting Tool

This web tool processes the website server's log files into more usable and useful reports. Some of the free tools are:

Awstats - Free perl script to analyze your Apache logs, Measures humans as well as crawlers, Measures hits, page views and produces HTML reports.

Google Analytics: Measures humans only (javascript), Focus on visits, new visitors, Map overly, pay-per clicks, advertisements, email marketing etc.

Performance Optimization in a Drupal Site

At the foremost, a site that is built in DRUPAL which needs optimization should be worked at various levels – Optimization at Web server level, optimization at Database level, optimization at PHP level and then finally optimization at DRUPAL level.

Server Level Optimization at Apache

Apache is the most popular, widely used, well supported and feature rich web server with much sophisticated features. Apache supports a variety of features, many implemented as compiled modules which extend the core functionality. Virtual hosting allows one Apache installation to serve many different websites.

Apache Modules

The request sent to the Apache is handled by Apache Modules. Apache has few core modules that handles the basic function of serving static web pages or rather static web content.

Few Apache modules are mod_php, mod_headers, mod_filters, mod_perl, mod_proxy, mod_deflate, mod_rewrite. On each request, Apache Core identifies the required Apache modules to get it loaded for processing the request. Thus required and identified modules are loaded and invoked one after the other such the output of one module will be the input of its successor module in this process flow.

For example, mod_php the module that converts PHP source into HTML output, can pass its output through the output filter stage to mod_deflate, the module that compresses the output sent to the client using gzip compression. Each provides an output filter stage that is chained together in a well defined sequential order.

Apache performance tuning

Performance analysis and monitoring are the major key factors to be observed in hosting a web site. There may arise a time when the performance of the hosted web server is far below its required level. In such scenario the performance needs to be boosted by sharing the application across multiple web servers of similar configuration or tweaking the variables of Apache configuration.

One such typical example will be handling the memory consumption and thereby the memory overflow.

MaxClient is the directive that defines the maximum number of simultaneous or concurrent request that apache could handle at a time. Hence the MinSpareServer / MaxSpareServer (i.e. the minimum/maximum number of child process that Apache can keep it live respectively) are preferred to half the value of Maxclient such that there could be no child thrashing.

Incase of Shared Hosting, such configuration variables are defined individually for each application using .htaccess. Configurations defined in .htaccess override the basic configuration defined in Apache settings. This is implemented by enabling AllowOverride directive in Apache configuration.

Enabling AllowOverride will instruct Apache to look for .htaccess in the parent directory and as well as the sub-directories. Nevertheless, in case AllowOverride is enabled but there is no .htaccess will cause the process to check for .htaccess file redundantly in all the directories and its sub-directories resulting in overhead.

Apache Alternatives

An approach to alternative to Apache or rather Supporting servers could be implemented to handle crisis effectively. Such Supporting servers could be Lighttpd (Lighty) or Nginx – a new corner but more stable than lighty as there are no leaks.

Lighttpd – asynchronous server, as the name says is very light web server and results is enormous speed, high throughout, secured and flexible to all environments. Lighttpd runs as a single process with a single thread and non-blocking I/O. Lighttpd is event-driven and much more faster in rendering the static pages when compared to Apache.

Nginx is an asynchronous web server, unlike Apache, which is a process-based web server. Nginx works as one master process but delegates its work unto worker processes. Nginx will spawn very few or no threads to support concurrent requests, unlike the Apache web server, which will require a new thread for each concurrent request thus resulting in overhead and high lookups. Due to this, one of the most stellar features Nginx provides is its low use of RAM under heavy traffic loads.

The performance difference between lighttpd and nginx is really negligible and Nginx might outperform lighttpd on some workloads and vice versa with other workloads.

Database Level Optimization at MYSQL

Most of the PHP or DRUPAL applications goes in hand with popular RDBMS – MySQL. MySQL has well defined layers at various levels as the front end layer will handle the communication, which then routes to Query Parsing and Optimization, then the data storage layers and the data access layer. The architecture of MySQL is defined in such a way that MySQL can handle multiple storage and data access with its “Storage Engines”. The popular storage engines are MyISAM and InnoDB.

MyISAM

Faster for reads,Less Overhead, less space consumption are the major pros of this engine. However the cons of its usage are its durability and non-transactional as it cannot rollback failed transactions or failed queries . MyISAM takes much time for table repair operations and the read / write operations made by table-level locking thus slowing down the application.

InnoDB

InnoDB are Transactional, has row-level locking and results in better concurrency. Inspite of its high advantages, InnoDB are slower in some cases eg. SELECT COUNT(*) and consumes much disk spaces.

On looking into the pros and cons of both the primary storage engines, one could tune the performance of the database by making the right choice of the database storage engine as the application demands.

Right choice of the storage engine

Before designing the database schema one should analyse and priorities the major operations that are to be carried out in their designed application. Depending on these 2 factors, one could make the right choice of the storage engine for their application.

Some typical situations would be

1. Information oriented applications that has more 'data fetching' operations rather than ' data feeing' operations should go for MyISAM. Because of low 'data feed' operations into the tables, the performance level issues due to the row-level locking will not exist.
2. Banking applications which are transactional oriented make InnonDB as their right choice. As banking process involves integrity of complex data which comprises 'joins' of many tables demands 24x7 uptime with 0% downtime. For such situations, InnoDB is the better option to address data crashes, data backup and logging.

Mysql Tuning

Among various methods to tune the database, commonly used are modifying the normalization-denormaization of tables, caching and redefining the indexing.

Slow Query Log

Slow Query Log has to be enabled in my.cnf file. In Slow Query lists queries taking more than N seconds. Redefining these queries or defining these queries using procedures will help us to make up the speed to some extent. Slow Query Log is very useful to identify bottlenecks.

- Too much of denormalization results in enormous lookups of tables and thereby overhead and slows down the application.
- Indexes are more costly to update and hence indexes should be used with high selectivity.

- Caching methodologies like Query Cache, Table Cache will speed up the process of fetching the data. Enabling the Slow query log will help us to find the query that takes up much execution time. It is prudent to cache such queries for better throughput.

In Drupal perspective

Drupal provides the patch for routing and executing the queries in Master / Slave architecture of the database http://drupal.org/node/147160 . This allows the administrator to define the list of slave servers and the master database. Using such architecture, one can use the slave servers for data fetching operations and master server of data feeding operation. In other terms, SELECYT query can be executed in slave while INSERT / UPDATE / DELETE will be operated in Master server.

PHP Optimization

Load Balancer

Load balancer like HAProxy, Squid are highly sophisticated and Apache running with mod_proxy allows load balancing appliances. High performance systems uses multiple layers of load balancing. Load Balancer has special features like HTTP compression that which reduces amount of data to be transferred for HTTP objects by utilizing gzip compression available in all modern web browsers, TCP buffering, HTTP Caching and other security based features like firewall, intrusion prevention mechanism, client authentication etc.

Op-code Caches

When a PHP program is actually executed, an intermediate language known as operation code (hence forth known as opcode is generated during the parsing process. Hundreds of these optimizations may take place in order to generate the most efficient code possible. Instead of parsing and generating opcode each time the same program is executed, opcode caching keeps the results in memory or disk based cache and only re-parse the file if the original has changed.

Alternative PHP Cache, EAccelerator, Memcache, XCache, Turck MMCache, Zend optimizer+ are the various list of accelerators that optimizes the PHP intermediate code and caches data and compiled code from the PHP byte code compiler.

From the above results, one can come to the following conclusions:

- Dramatic Speed up of applications, specially complex ones like DRUPAL
- Significant decrease in CPU Utilization
- Considerable decrease in memory Utilization
- Biggest impact on a busy and high traffic site
- EAccelerator uses the least memory and provides the most speed

- All op-code caches provide a noticeable improvement for Drupal over a default PHP installation.
- The speed gain is about 3X.
- eAccelerator is marginally better than the XCache or APC both in terms of speed and memory utilization.

A typical example of implementation of accelerators in drupal sites is observed here http://2bits.com/articles/benchmarking-drupal-with-php-op-code-caches-apc-eaccelerator-and-xcache-compared.html

Optimization at DRUPAL

Optimization at Drupal requires analysis on resource consumption - Profiling memory usage, measure DB query times, and duplicate queries for modules or pages with devel.module.

The number of modules enabled on the site may affect performance but should be measured using the techniques outlined above. If the module has slow query times look at tuning the tables with schema changes and using the MySQL optimization tools, such as ANALYSE and EXPLAIN. Other areas for analysis are theme resource consumption if the theme uses a lot of PHP calls.

If the cron jobs are set too frequently there can be a consistent drain on server performance. Drupal has a cache that is very effective for pages served to anonymous users

Modules like watchdog, syslog, sessions, cache and devel are executed on every access of the page. By this way these modules are overloaded with huge data and resulting in less response time.

Media Files

Large Audio and Video ties up resources for long time, specially to slow connections or unstable ones when users try to download again and again.

In such scenario the optimal way to serve the request is like handling the call from the separate box.

E.g.: http://www.domain.com for PHP

http://media.domain.com for media files

Video module in drupal supports this but one has to manually FTP the videos. And other alternative is Content delivery network (CDN) eg.Akamai. This helps us in saving storage space and results in high delivery speed.

Do's on the Drupal Site

Some good practices like disabling unused modules, run cron regularly but not too frequently, enabling throttle will address the tuning factor to some extent.

Drupal Caching

Enable drupal caching / page caching for Anonymous users only. Aggressive caching may have some implications but gives better performance. Certain parts of cache are always turned on and cannot be turned off like Filter, menu and variables.

Filter cache can be turned off in a busy site as it is a pluggable cache using $conf variable in settings.php as 'cache_include' => './includes/cachefile.inc'

Block Caching

This is a contrib module in DRupal 5.x and now its the core in D6. This eliminates the overhead of generating blocks for each page view. Block caching happens separately from page caching and so it will work for logged-in users whether or not page caching is enabled for the site. Cached blocks make just one call to the database and can therefore reduce the overhead required to render blocks that require complex queries and/or calculations such as some Views or Taxonomy-related blocks.

Advanced Caching

For authenticated users – block_cache, comment_cache, node_cache, path_cache, search_cache, taxonomy_cache will be enabled for rendering the page without much lookups into the database.

Slow Modules

There are few slow modules that when enabled consumes much of the resources like

1. Statistics module - Adds extra queries which are even slower on InnoDB
2. gsitemap (XML sitemap): - –high lookups and overhead and statistics say that gsitemap can't handle more than 50,000 nodes per request.

Later to that, Drupal has revised the base structure to address the performance issues as mentioned in http://drupal.org/node/163216

Drupal Tools

Devel module

Devel can help to find out why your site is slow, discover how things are themed, learn what functions are being called, generate random content, clear your cache, switch users on the fly, view every SQL query, how long it took to run etc.

This module can print a summary of all database queries for each page request at the bottom of each page. The summary includes how many times each query was executed on a page (shouldn't run same query multiple times), and query execution time.

This module provides performance statistics logging for a site, such as page generation times, and memory usage, for each page load. This module is useful for developers and site administrators to identify pages that are slow to generate or use excessive memory.

Devel Features include:

- Settings to enable detailed logging or summary logging. The module defaults to no logging at all.
- Detailed logging causes one database row to be written for each page load of the site. The data includes page generation time in milliseconds, and the number of bytes allocated to PHP, time stamp, ...etc.
- Summary logging logs the average and maximum page generation time, average and maximum memory usage, last access time, and number of accesses for each path.
- Summary can be logged to memcache, if configured, so as to not cause extra load on the database.
- This works when APC cannot be used (e.g. certain FastCGI configurations, or when you have many web servers on different boxes. This mode is recommended for live sites.
- Summary can be logged to APC, if installed, and the APC data cache is shared, so as to not cause extra load on the database. This mode is recommended for live sites.
- A settings option is available when using summary mode with APC, to exclude pages with less than a certain number of accesses. Useful for large sites.
- Support for normal page cache.

Measure and Monitor

The site owner cannot wait till the site is sluggish or the users complains or the till the visitors lose interest in the slowdown sites. We have different tools for different tasks. What can go wrong – CPU usage is too high, memory over utilization, too much disk I/O, too much network traffic.

Its the prime responsibility of the site owner to identify who is using the CPU, find out which type like user, system, wait I/O and where from such bottlenecks arises.

- If it is an Apache process, the opcode cache will help (APC, eAccelerator).
- If it is MySQL, then tune the indexes (OPTIMIZE TABLE), split the server to two boxes (web server and db server). Tune the query cache. Optimize MySQL once a week, or once a day.

- Turn off PHP error logging to /var/log/*/error.log
- Consider disabling watchdog module.

Munin and Cacti

Munin and Cacti are the system monitoring instrument that produces easy to understand graphs. It can observe Processor, system memory, Apache, MySQL, network and many other components. It maintains a history of what is going on in the system by day, week, month and year.

vnstat

Vnstat gives network interface statistics in a human readable way. It can give daily or monthly information for each ethernet card, and provide transmit and receive information for each.

htop

htop is similar to top, but displays individual CPU use in a multi-process system and it is widely used. Other such tools are Top,netstat, apachetop, mtop, mytop.

YSlow

Yahoo released YSlow, a Firefox extension that integrates with the popular Firebug tool. YSlow analyzes the front-end performance of the website and tells why it might be slow. For each component of a page (images, scripts, stylesheets) it checks its size, whether it was gzipped, the *Expires*-header, the *ETag*-header, etc. YSlow takes all this information into account and computes a performance grade for the page.

Stress Testing

Stress Testing answers how much requests per second the site can handle, the performance and bottlenecks the site may encounter before you deploy or after you deploy and so on. The challenge is finding a realistic workload and simulating it.

Apache benchmark - ab/ab2 (Apache benchmark)

http://httpd.apache.org/docs/2.0/programs/ab.html

ab is the most commonly used benchmarking tool in the community. It shows have many requests per second the site is capable of serving.

Concurrency can be set to 1 to get end to end speed results or increased to get a more realistic load for the site. In order to test the speed of a single page, turn off page caching and run ab with concurrency of one to get a baseline.

ab -n 1000 -c 1 http://drupal6/node/1

To check scalability turn on the page cache and ramp up concurrent connections (10 to 50) to see how much the server can handle.

“Keep alives” are turned (-k) on as this leads to a more realistic result for a typical web browser. At higher concurrency levels making new connections can be a bottleneck. Also, set compression headers (-H) as most clients will support this feature.

ab -n 1000 -c 10 -k -H 'Accept-Encoding: gzip,deflate' http://drupal6/node/1

Other Server performance testing tool are Siege http://www.joedog.org/JoeDog/Siege

and Jmeter written in Java http://jmeter.apache.org/

Best Practices

Minimize HTTP Requests

Combined files are a way to reduce the number of HTTP requests by combining all scripts into a single script, and similarly combining all CSS into a single stylesheet. Combining files is more challenging when the scripts and stylesheets vary from page to page.

CSS Sprites are the preferred method for reducing the number of image requests. Combine your background images into a single image and use the CSS `background-image` and `background-position` properties to display the desired image segment.

Image maps combine multiple images into a single image.

Use a Content Delivery Network

A content delivery network (CDN) is a collection of web servers distributed across multiple locations to deliver content more efficiently to users. The server selected for delivering content to a specific user is typically based on a measure of network proximity. For example, the server with the fewest network hops or the server with the quickest response time is chosen.

Some large Internet companies own their own CDN, but it's cost-effective to use a CDN service provider, such as Akamai Technologies, EdgeCast, or level3.

Other best practices observed and suggested are placing Stylesheets at the top of the page, placing Scripts at the Bottom of the page, avoiding Redirects, removing Duplicate scripts, making Ajax cacheable, pre-loading Image Components, handling 404, optimizing Images and CSS Sprites, avoiding images scaling in HTML, making favicon.ico Small and Cacheable and avoiding Empty Image SRC.

Resources and Links

General

http://2bits.com/
http://www.lullabot.com/
http://developer.yahoo.com/performance/rules.html

Apache

http://httpd.apache.org/docs/2.0/misc/perftuning.html

MySQL

http://www.mysqlperformanceblog.com/
http://dev.civicactions.net/moin/CodeSprint/SanFransiscoMarch2007/PerformanceAndScalabilitySeminar

Drupal

http://drupal.org

http://buytaert.net/drupal-performance