

vB4Mance Part 4: The Basics - Understanding vBulletin4 PHP Script and Hosting

- 1. In my earlier tech blog posts I've covered a wide array of topics mostly focused on vBulletin4 databases and how to improve their performance. We talked about vBulletin4 database model changes, converting to InnoDB as well as covered vBulletin benchmark load testing under various technology stacks.

While these discussions were a fun and exciting read for the technically savvy crowd on vBulletin.com they've left a lot of unanswered questions and perplexing concepts for some of us that are just starting out with vBulletin4 and community building in general. How does vBulletin4 work? What hosting server environment will I need to get started? What is a LAMP software stack? How do really big vBulletin sites work? What is the meaning of life?

For my next foray into the realm of technical writing I shall try to decipher, explain and illustrate how vBulletin4 actually functions on the server level. We'll discuss the differences between shared and dedicated hosted environments along with general principles of their infrastructure.

So, without further ado - let us joyfully embark upon this prodigious quest for technical vBulletin4 knowledge. Still with me?

How does the vBulletin4 PHP script work?

vBulletin4 code is written in what's called a "Hypertext Preprocessor" scripting language; we all know it as our beloved "PHP". PHP is an extremely popular coding language due to a variety of reasons not limited to its general flexibility, availability of pre-made functions, overall simplicity and well it's also completely free!

LAMP/WAMP: Because vBulletin is developed exclusively using PHP it requires a LAMP or WAMP web environment to work. This is really simple "LAMP is just an acronym for basic software components needed to successfully execute a PHP script to the end-user, ie. Your site's visitor. These modules are Linux (L), Apache (A), MySQL (M) and PHP (P), and in case you are using Windows "just replace the first letter with W instead of L, creating WAMP.

The basics are as follows - Linux or Windows is the operating system of the server, Apache is a service that gets and sends requests to and from the server to the user, PHP is a module that reads and interprets the code's instructions and MySQL is the storage center for all of the generated data. LAMP technology has been around since the dawn of man "in fact some of the earliest PHP classes have been written on stones by cavemen and only recently uncovered by modern archeologists.

Now that we've briefly introduced the four fundamental server software components "let's do a quick tour of the actual page rendering process and discuss key functionality of each step. Please reference and follow my "LAMP Stack" diagram in an effort to appease me and validate my artistic credibility:

The first step of any web transaction in vBulletin4 is a call from the visitor's browser to the

web-server requesting data. In my diagram, a fine gentleman who also appears to be very confused has entered a forum URL into his browser requesting a vBulletin4 thread page. That particular webpage request call goes out over the internet and is routed to a very specific web-server where the vBulletin4 showthread.php file resides. This happens to be a LAMP server running Apache, MySQL and PHP. Because the visitor's request arrives via a specifically designated channel the web-server is able to accept and forward the request to the Apache web server module. Apache web server checks the request, identifies it as a vBulletin PHP file and starts working on creating a return data package to the visitor. Since Apache does not understand PHP code it forwards the request to a pre-compiled PHP module in order to interpret the PHP script code. PHP module quickly reads the instructions and begins executing the required commands. In this case - to build a vBulletin thread page, PHP sends multiple database queries to MySQL server requesting post id numbers, checking user group permissions and obtaining various information required to display a thread to the user. MySQL server receives the queries and searches the specified database for the data requested by PHP. After finding all of the requested database field values, MySQL returns the data to PHP so that it can finish logical computations as instructed by the vBulletin PHP script. Once PHP successfully executes the instructions contained in the vBulletin PHP file, Apache builds the page requested and sends it back to the visitor. In our diagram, a forum thread page requested by the user appears in their browser. Great Success!

While it looks like lots of different activities had just taken place - the time to complete these computations could have been just fractions of a second, depending greatly on your server's computing capacity. Each of these unique processes requires processing power as well as an allotment of RAM memory. In this example - scalability and website performance relationship appears to follow a simple rule - more server power equals higher performance and improved loading times. The reality of hosting is entirely different however and highly constrained by budgets. Everyone can't just go out and purchase expensive equipment - especially when they are just starting their vBulletin4 community. Due to budget constraints, web server hosting is available at various price points and tailored to specific computing capacity requirements.

This also happens to be just the right spot in the article to pick up our discussion of just some of the popular available hosting options and their characteristics.

vBulletin4 Hosting: Knowing how much server power you need.

Typically, the processing and memory requirements for any vBulletin site can be calculated using the number of total posts and the general activity levels of the users. As an example of the lower hosting scale; a brand new vBulletin4 community with just a few users, articles and threads requires very little server capacity and can save lots of money by using cheap - shared hosting. An example on the far end of the spectrum would be a vBulletin community that has 5 million posts and over 500 users online at any given point in time participating on the site. Major server capacity is required to power this active community and - dedicated hosting is typically used in this scenario.

vBulletin4 Hosting: - Shared and - Dedicated hosting explained.

Shared Hosting (Also See Diagram) - GREAT for starting a new vBulletin4 community!

Just like the name implies, shared hosting is a machine that is disproportionately shared by many different sites. Site X, Y and Z are all using the same software, processor and services. The customer cannot modify or install new software on a shared environment without risking breaking the neighboring sites. Because Internet Service Providers can easily cram tens if not hundreds of sites into just one server - shared hosting is extremely cheap. The drawback to shared hosting is that your site's speed and performance is at the mercy of all of your neighbors. If your community starts growing - you will start running out of server capacity - as well as, if others' sites begin consuming more resources, your site will slow down as well.

It's also worth noting that while still extremely popular, "Shared Hosting" is becoming a concept of the past and is slowly being replaced with comparably-priced elastic cloud-based and VPS hosting services (we will cover these a bit later).

Shared Hosting Pros: Really fast and easy to start a new vBulletin4 community, extremely affordable. Easy server maintenance and administration.

Shared Hosting Cons: Potential performance issues, lack of flexibility to install custom software, limited community growth.

Dedicated "Big-Board" Hosting (Also See Diagram)

A vBulletin community that achieves the coveted "Big-Board" status has thousands if not millions of total posts and very high user activity. At this point, the site will require a server that is dedicated to serving the site exclusively, meaning that the entire server's processing and memory power is available to just one customer. An extremely large vBulletin community site could actually have an entire farm of dedicated web-servers as well as separate database servers, all working together to maximize speed and performance. Another nice thing about being on a dedicated server is that you can modify it " you can install everything and anything you want. Lots of expert technical knowledge is required to operate a dedicated server and that is precisely why dedicated hosting is available in two flavors; managed and unmanaged. "Managed Dedicated Hosting" is more expensive because you are paying professional technicians to administer all aspects of your server's operation; it's like having your own technical staff. In an "Unmanaged Dedicated Hosting" situation " you are solely responsible for operating, configuring and fixing your own server. Typically, dedicated hosting means that you are renting server(s) from an internet service provider and do not actually own the hardware. (see co-location)

Dedicated Hosting Pros: Flexibility of server optimization and customization, maximum performance, unlimited growth.

Dedicated Hosting Cons: Expensive and could require expert technical knowledge to maintain.

Co-Located Servers " a quick word.

Co-Location is identical to "Dedicated Hosting" in every aspect, including the "Managed" or "Unmanaged" option. The main difference is that when you are co-locating a dedicated server, you own the hardware and are paying an Internet Service Provider for space in their server rack as well as management and network bandwidth usage.

In conclusion and looking out into the bright vBulletin4 future!

With basic hosting and PHP scripts explained and out of the way - In the next installment, we are going to cover Virtual Private Server hosting, Elastic Cloud hosting and vBulletin4 performance and other server optimization tips / tricks! We are going to make it fast, real fast - so please stay tuned and become my buddy on the INETSkunkWorks twitter account for vB4 performance updates. (Shameless plug).

<http://help.digipower.vn/questions/59/>