



Introduction to mod_perl
(part I): *Why mod_perl?*

By Stas Bekman

If you like this article, please make a donation to the [Perl development grant fund](#).

Table of Contents

<u>What is mod_perl?</u>	1
<u>Why is mod_perl so popular?</u>	2
<u>How fast and stable is mod_perl?</u>	3
<u>How many sites run mod_perl?</u>	4
<u>The Road Ahead</u>	5
<u>Where Do I Get It?</u>	6
<u>Books and Documentation</u>	7
<u>Who wrote mod_perl?</u>	8
<u>Getting Involved</u>	9
<u>References</u>	10

What is mod_perl?

In this article I'll give an initial introduction to mod_perl, make you want to give it a try and present a few examples of the well known sites that are powered by mod_perl enabled Apache.

mod_perl is at the heart of the Apache/Perl integration project, which brings together the full power of the Perl programming language and the Apache Web server.

From the outset Apache was designed so that you can extend it by the addition of "modules". Modules can do anything you need to do, such as rewrite HTTP requests, restrict access to certain pages, perform database lookups and so on. Modules are normally written in C, which can be hard work. mod_perl is a module which lets you do all of these things, and more, using Perl -- which makes the development much quicker than C. Apache is the most popular Web server on the Internet and mod_perl is one of the most popular modules for extending it.

Why is mod_perl so popular?

If you love Perl and your favorite web-server is Apache you will love mod_perl at first sight. Once you try it in action you will never look back, you will find mod_perl has everything you need. But even if you do find that there is something missing, just speak up. Before you can count to three, someone will have made it for you. Which of course will make you want to give something in return. Eventually you will contribute something on your own, and that will save time for a huge mod_perl community so that they can create even more things for others to use.

You get the picture — mod_perl empowers its users, who in turn empower mod_perl, which in turn empowers its users, who in turn... It's as simple as the nuclear reaction you learned about at school, or will learn at some point :)

With mod_perl it is possible to write Apache modules entirely in Perl, this lets you easily do things that are more difficult or impossible in regular CGI programs, such as running sub-requests, writing your authentication and logging handlers and much more.

The primary advantages of mod_perl are power and speed. You have full access to the inner workings of the Web server and you can intervene at any stage of HTTP request processing. This allows for customized processing of the various phases, for example URI to filename translation, authorization, response generation and logging.

There are big savings in startup and compilation times. Having the Perl interpreter embedded in the server saves the very considerable overhead of starting an external interpreter for any HTTP request which needs to run Perl code. At least as important is code caching: the modules and scripts are loaded and compiled only once, when the server is first started. Then for the rest of the server's life the scripts are served from the cache, so the server only has to run the pre-compiled code. In many cases this is as fast as running compiled C programs.

There is very little run-time overhead. In particular, under mod_perl there is no need to start a separate process per request, as is often done with other web-server extensions. The most wide-spread such extension mechanism, the Common Gateway Interface (CGI), is replaced entirely with Perl code that handles the response generation phase of request processing. Bundled with mod_perl are two general purpose modules for this purpose: Apache::Registry, which can transparently run existing unmodified Perl CGI scripts and Apache::PerlRun, which does a similar job but allows you to run scripts which are to some extent "dirtier".

mod_perl allows you to configure your Apache server and handlers in Perl (using the PerlSetVar directive and the <PerlE> sections). This makes the administration of servers with many virtual hosts and complex configuration a piece of cake. Hey, you can even define your own configuration directives!

How fast and stable is mod_perl?

Many people ask "How much of a performance improvement does mod_perl give?". Well, it all depends on what you are doing with mod_perl — and possibly whom you ask. Developers report speed boosts from 200% to 2000%. The best way to measure is to try it and see for yourself! (see <http://perl.apache.org/tidbits.html> and <http://perl.apache.org/stories/> for the facts).

Every second of every day, thousands of Web sites all over the world are using mod_perl to serve hundreds of thousands of Web pages. Apache and mod_perl are some of the best-tested programs ever written. Of course they are continually being developed and improved, but you do not have to work on the "bleeding edge" of development — you can use the stable products for your sites and let others do the testing of the new versions for you.

I want to show you just a few of the many very busy and popular sites that are driven by mod_perl. A thousand words can't substitute the experience. Visit the sites and feel the difference. They will persuade you that mod_perl rules!

ValueClick — <http://www.valueclick.com/> serves more than 70M requests per day from about 20 machines. Every response is dynamic, with all sorts of calculation, storing, logging, counting, you name it. All of their "application" programming is done in Perl.

Singles Heaven — <http://singlesheaven.com> is a Match Maker site with 35,000+ members and growing. The site is driven by mod_perl, DBI, Apache::DBI (which provides a persistence to DB connections) and MySQL. The speed is enormous, chatting with mod_perl is a pleasure. Every page is generated by about 10 SQL queries, for it does many dynamic checks on every page — like checking for new emails, users who are watched by various watchdogs and many more. You don't feel these queries are actually happening, the speed is as fast as the "Hello World" script.

Internet Movie Database (Ltd) — <http://www.movedatabase.com/> – serves around 2 million page views per day. All database lookups are handled inside Apache via mod_perl. Each request also goes through several mod_perl handlers and the output is then reformatted on the fly with mod_perl SSI to embed advertising banners and give different views of the site depending on the hostname used.

CMPnet — <http://www.cmpnet.com>, a technology information network, serves about 600k page views per day.

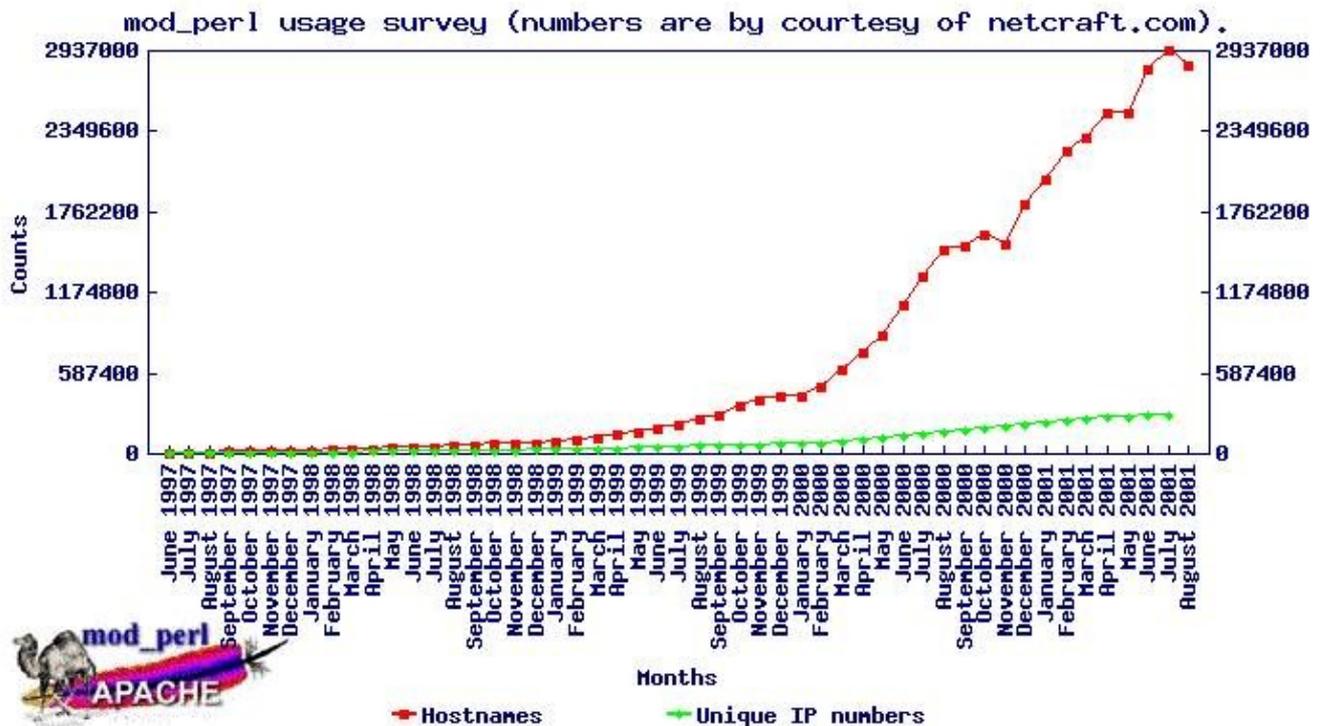
CitySearch.com — <http://www.citysearch.com/> is providing online city guides for more than 100 cities worldwide, citysearch.com helps people find and plan what they want to do and then lets them take action, offering local transactions such as buying event tickets and making hotel and restaurant reservations online. Its traffic exceeds 100,000,000 page views a month.

How many sites run mod_perl?

According to Netcraft (<http://netcraft.com>), as of August, 2001 – 18 million hosts are running the free Apache webserver, which is about 60% of all checked in survey hosts!

Here is the [graph](#) of "Server Share in Internet Web Sites".

What about mod_perl? <http://perl.apache.org/netcraft/reports> that sites running mod_perl account for 2,823,060 host names and 283,180 unique IP addresses. This is actually an underestimate, since when hosts are scanned for running webserver only well known ports are checked (80, 81, 8080 and a few others). If a server runs on unusual port it does not enter the count unless the owner has manually added it to the Netcraft database. Here is a graph of the growth in mod_perl usage:



For the latest numbers see <http://perl.apache.org/netcraft/>.

The Road Ahead

You probably are all excited about the release of Apache 2.0, the next generation of the best web-server. The major new features of this new generation of the webserver are threaded processes, which should make the server more scalable, and of course the very welcomed filtering layer.

You probably are not less excited about the recent release of Perl 5.6, whose main new feature is (almost) stable support for threads, something that existed in the previous Perl version but which was quite shaky.

What has all this to do with mod_perl you ask? mod_perl 2.0 is being developed at this very moment and will benefit enormously from the new Apache and Perl features. The most important improvement will be a reduced process size -- a parsed Perl opcodes tree will be almost completely shared between threads of the same process.

Do you believe in coincidences? Both Perl 5.6 and Apache 2.0 were released in the same week in March, 2000. Looks very suspicious to me. If you get the obvious conspiracy uncovered please let me know.

Of course there are lots of bumps ahead of us. It will take time before all our applications will be able to benefit from the threading features. The main reason lies in fact that most of the Perl modules available from CPAN aren't thread safe. But you shouldn't despair. You can turn off threads for Perl code which is not thread safe or which uses modules which aren't thread safe.

Where Do I Get It?

mod_perl's home is <http://perl.apache.org>. From the site you will be able to download the latest mod_perl software and various documentation; find commercial products and free third party modules; read the success stories; and learn more about mod_perl.

It's quite important to get yourself subscribed to the mod_perl list. If you want know what happens with mod_perl, if you want to know what new features are being developed, if you want to influence and contribute or if you simply want to get help, then you don't want to skip this mailing list. To subscribe to the list send an empty email to modperl-subscribe@apache.org.

Books and Documentation

Lincoln Stein and Doug MacEachern wrote Writing Apache Modules with Perl and C . It can be purchased online from O'Reilly and Amazon.com.

You will find a vast list of mod_perl documentation on the mod_perl home page: <http://perl.apache.org/>.

Who wrote mod_perl?

Well, Doug MacEachern is the person to blame :). He is the guy who gave mod_perl to the mod_perl community. He is the Linus of the mod_perl project.

But as you know in a big community there are always people who love to help, and there is a bunch of developers from all around the world who patch mod_perl, develop entire Perl modules for it, debug the server and advocate it. I'm afraid the list of contributing developers is too long to include here. But you are welcome to join the mod_perl mailing list and see all these folks in action. I promise you, you won't regret the show, since you are going to learn much more than just about mod_perl. See for yourself.

Getting Involved

If you are using mod_perl or planning to use it, it's a good idea to subscribe to the users mod_perl mailing list. You should send an empty email to modperl-subscribe@apache.org in order to do that.

If you are interested in helping out with development of mod_perl 2.0, you are welcome to join us. There are many features that are still need implementing and a lot of testing has to be done. So there is a lot of work if you are knowledgeable developer or even if you just a newbie. And the more help we get the sooner we bring mod_perl 2.0 into production shape. You can subscribe to the developers mailing list by sending email to dev-subscribe@perl.apache.org.

If you are familiar with mod_perl, you probably know about the big fat mod_perl guide that I maintain with help of many people (<http://perl.apache.org/guide/>). However mod_perl 2.0 has quite a few things changed so the new documentation project has been started. You are welcome to check the updates on the <http://perl.apache.org/> site and subscribe to the documentation mailing list to stay up to date by sending email to docs-dev-subscribe@perl.apache.org.

References

The Apache site's URL: <http://www.apache.org/>

The mod_perl site's URL: <http://perl.apache.org/>

CPAN is the Comprehensive Perl Archive Network. The Master site's URL is <http://cpan.org/>. CPAN is mirrored at more than one hundred sites around the world. (<http://cpan.org/SITES.html>)