

## Re: Reducing load for LAMP app?

---

*Source:* <http://coding.derkeiler.com/Archive/PHP/comp.lang.php/2008-01/msg00546.html>

---

- *From:* Jerry Stuckle <[jstucklex@xxxxxxxxxxxxxx](mailto:jstucklex@xxxxxxxxxxxxxx)>
  - *Date:* Mon, 07 Jan 2008 21:06:47 -0500
- 

Jeremy wrote:

Jerry Stuckle wrote:

Jeremy wrote:

Gilles Ganault wrote:

Hello,

I'm no LAMP expert, and a friend of mine is running a site which is a bit overloaded. Before upgrading, he'd like to make sure there's no easy way to improve efficiency.

A couple of things:

– MySQL : as much as possible, he keeps query results in RAM, but apparently, each is session-specific, which means that results can't be shared with other users.

Is there something that can be done in that area, ie. keep the maximum amount of MySQL data in RAM, to avoid users (both logged-on and guests) hitting the single MySQL server again and again?

– His hoster says that Apache server is under significant load. At this point, I don't have more details, but generally speaking, what are the well-know ways to optimize PHP apps?

Thank you.

## Re: Reducing load for LAMP app?

memcached: <http://memcached.sf.net> (for caching dynamic data in memory)

apc: <http://us2.php.net/apc> (caches script bytecode to reduce compilation overhead)

These two modules will help enormously. I guarantee it. Using apc is pretty much transparent, but memcached will require modifying your database abstraction layer using the memcached functions (<http://php.net/manual/en/ref.memcache.php>).

Not necessarily. MySQL and the OS both cache data – and generally do a better job at a lower cost than using memcached.

Not necessarily. Using memcached on the application layer can save a lot of network traffic to and from the database server, and can help immensely. Quite immensely. The OS doesn't cache data from a separate database server. I doubt the OP is running a local database server, as he explicitly refers to the "MySQL server".

No, it doesn't. But I'll bet he does run it locally. Virtually everyone I know refers to their MySQL database as "the MySQL Server". In fact, I find it's more often assumed to be on the same server when not specified.

But whether it's local or remote, that's only going to help if you know you're fetching the same data, and that data hasn't changed on the server. With a database, that's often not a safe assumption.

<Snipped re: connection pooling>

**WRONG, WRONG, WRONG!** This **INCREASES** resource usage on the server. With persistent connections, you must have the maximum number of connections *\*ever\** required allocated *\*all of the time\** – even if no one is using your server.

MySQL persistent connections should not be used except in extreme cases – like when you're running 100+ connections per second.

Sorry, but have you ever done any research on this topic? Connection pooling is a well-established mechanism for boosting performance on high-traffic servers. Keeping connections open is not a significant drain on server resources, and can save a lot of overhead in creating and tearing down connections. Also, you don't really know how many connections per second his application is handling. I suggested he try it and gauge the results for himself. If it really is slower, don't do it.

## Re: Reducing load for LAMP app?

Yes, I have. In fact, I have many years of experience with persistent connections, going back to DB2 in the 80's. And several years with MySQL. And my comments stand.

Yes, persistent connections can help with very heavily used sites – in the thousands of hits per second. But below that, the overhead of making a connection – especially on a local machine – is small compared to the need to maintain potentially hundreds of open connections because you might use them at some point in time.

What you're suggesting is he shoot in the dark and hope he hits something. And if he hits the wrong thing, he could easily hurt system performance more than he helps it.

The **FIRST** thing to do when having performance problems is to identify the cause of the performance problem.

That is certainly true.

You're making suggestions without having any idea where the hold up is.

And what would you suggest? Should I do a code review for him? Rewrite his queries? Or should I suggest a few common tweaks that might help from a configuration standpoint? Or maybe we should all just stick to flame wars instead of generating any useful discussion?

Just what I did suggest. Identify the source of the problem first. And I gave him some suggestions. Looks like he followed them up and found some problems. And from what he found, the problem is not related to database – but general resource shortage. So your suggestion of persistent connections would take more resources and aggravate that problem instead of helping it.

I have over 20 years of experience in tuning systems. I never make changes until I've identified the cause(s) of the problem.

In at least two cases, your suggestions could actually **HURT** performance. And the third case may or may not help. If, for instance, the majority of the resource usage is spent in long MySQL queries,

People are always saying "X is the bottleneck, so improving Y won't help." This is just not true. It won't help as much as improving X, but every cycle saved on redundantly compiling PHP code is a cycle that can be used doing useful computation. **NOT** running a bytecode cache is pretty pointless. Why not do it?

Oh yes, it is very true. If X is the bottleneck, you aren't going to make any significant improvement by

Re: Reducing load for LAMP app?

changing Y. In fact, if you fix X, you may not even need to change Y.

Of course, there is always the possibility that Y is a secondary problem. If so, that will show up after fixing X.

Jeremy

--

=====

Remove the "x" from my email address

Jerry Stuckle

JDS Computer Training Corp.

jstucklex@xxxxxxxxxxxxxx

=====

.