

Re: [PHP] A no brainer...

Re: [PHP] A no brainer...

Source: <http://coding.derkeiler.com/Archive/PHP/php.general/2006-10/msg00908.html>

- *From:* edlazor@xxxxxxxxxx (Ed Lazor)
 - *Date:* Mon, 16 Oct 2006 16:40:34 -0700
-

On Oct 16, 2006, at 6:20 PM, Roman Neuhauser wrote:

Modern filesystems cope well with large directories (plus it's quite trivial to derive a directory hierarchy from the filenames).
Looking at the numbers produced by timing various operations in a directory with exactly 100,000 files on sw RAID 1 (2 SATA disks) in my desktop i'd say this concern is completely baseless.

I knew that you could get PHP to use a directory structure for the session data files, but hearing that you can have 100k files in a single directory and not run into performance issues or problems is news to me. Which OS are you running?

It still uses files, but hopefully you don't hit them very often, especially when you're dealing with the same table records.

A RDBMS is basically required to hit the disk with the data on commit. One of the defining features of a RDBMS, Durability, says that once you commit, the data is there no matter what. The host OS may crash right after the commit has been acked, the data must stay.

You can turn on query caching in MySQL, but this will give you *nothing* for purposes of session storage.

Unless session storage is used to save time in retrieving data, right? I'm seeing your point on the writing, but what about reading?

I think it would be kind of fun to run some actual tests.

Also, having raw data is always faster than having to process it before you can use it.

Re: [PHP] A no brainer...

Re: [PHP] A no brainer...

I don't know what that means.

If you pull a record from the db, you can access the data. Or you can query the db, get the serialized data, de-serialize it, and now access the data.

Bytes in files on disk are as raw
as it gets, you get one roundtrip process -> kernel -> process;
compare the communication protocol MySQL (or just any other DB) uses
where data is marshalled by the client, and unmarshalled by the
server, overhead of the database process(es) taking part in the
write...

So no, it makes no sense for a database to be faster than
filesystem.

I tested this previously and found the database to be faster. The references I gave supported this and listed additional benefits. Things change tho, especially with technology. It seems like we should be able to test this pretty easily. I actually think it would be fun to do as well. Do you have a box we can test this on? Meanwhile, I'll check one of my boxes to see if I can use it. If anything, it'll be interesting to see if two systems report the same.

-Ed

.