

Re: [ANNOUNCE] Ramback: faster than a speeding bullet

Re: [ANNOUNCE] Ramback: faster than a speeding bullet

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2008-03/msg06176.html>

- *From:* Willy Tarreau <w@xxxxxx>
 - *Date:* Sun, 16 Mar 2008 00:22:21 +0100
-

On Sat, Mar 15, 2008 at 02:33:09PM -0800, Daniel Phillips wrote:

On Saturday 15 March 2008 14:54, Willy Tarreau wrote:

I think it could get major adoption with ordered writes.

It already has ordered write when it is in flush mode.

but IIRC it works in write-through then.

OK, I hear you. There will be an ordered write mode that uses barriers to decide the ordering. It will greatly reduce the speed at which ramback can flush dirty data because of the need to wait synchronously on every barrier, of which there are many. And thus will widen out the window during which UPS power must remain available if power goes out, in order to get all acknowledged transactions on to stable media. The advantage is, the stable media always has a point-in-time version of the filesystem.

Don't expect this mode in the immediate future though, there are bugs to fix in the current driver, which already implements the required performance and stability requirements for a broad range of users.

ah, good!

That is why I keep recommending that a ramback setup be replicated or mirrored, which people in this thread keep glossing over. When replicated or mirrored, you still get the microsecond-level transaction times, and you get the safety too.

Re: [ANNOUNCE] Ramback: faster than a speeding bullet

I agree, but in this case, you should present it this way. You have been insisting too much on the average PC's reliability, the fact that no kernel ever crashed for you, etc... So you are demonstrating that your product is good provided that everything goes perfectly. All people who have experienced software or hardware problems in the past (ie mostly everyone here) will not trust your code because it relies on pre-requisites they know they do not have.

That would have been a miscommunication then. I see arguments coming in that suggest embedded solutions, EMC for examp