

## Re: 2.6.25-rc8: FTP transfer errors

---

*Source:* <http://linux.derkeiler.com/Mailing-Lists/Kernel/2008-04/msg04278.html>

---

- *From:* Willy Tarreau <[w@xxxxxx](mailto:w@xxxxxx)>
  - *Date:* Sat, 12 Apr 2008 10:44:00 +0200
- 

Hi guys,

I've read quite a bunch of this thread, and I think there's some misunderstanding between both parts, as well as inappropriate expectations in both cases.

On Fri, Apr 11, 2008 at 11:07:02AM -0700, David Miller wrote:

We had Mark's bug fixed in 15 minutes once the bisect result was known, even after Ilpo and myself had scanned through the changesets.

This proves the utility of bisect and in fact that trying to intuit the cause by continuing to study changesets and code would have been a complete waste of time.

Yes, Mark, we used to do things that way for every bug in the kernel. And as a result many bugs sat unfixed for weeks if not months. Many of us have left the cave, feel free to join us.

We should be very careful about git-bisect. First, it does not necessarily point to the bug, but to the commit which exhibits the bug, so simply reverting the commit might just hide the bug again. I want to ensure that people do not forget that it does not replace a brain, it enhances your eyes by pointing to a change related to the problem.

While it is a powerful tool, we must accept that it cannot efficiently work in some circumstances, such as :

- the machine cannot be rebooted often. I've been used to work for customers who plan changes once a week, and change absolutely nothing on their production if unplanned. This means one bisect step per week. Often, those people even require that your changes pass through a week of non-regression testing on a pre-production system (which was my case), with no overlapping between changes, so then you can count on one git-bisect iteration every two weeks.
- the problem only happens in peak traffic hours on production, and the loss of service has already gone far beyond the annual quota.

The only case they will accept an upgrade if you engage your full responsibility that it will definitely fix the problem. I've already been in such a situation, you say to the guy in front of you that you're putting your balls on the table, it will work (and sometimes you're only 90% confident). You obviously cannot do this to just check if the current bisect exhibits the problem or not.

– the reporter has very few spare time. I do have friends in this situation. Basically, when your schedule is full of customers visits one month ahead, it's very hard to find several consecutive hours to track the problem down. Sometimes you're happy if you can spend two hours on it in a week. BTW, many developers are also in the same situation. Also most of the time, this must be done at the customer's and some of them do not accept people out of work hours. Then the problem may lay for weeks or months.

– the problem is very reproducible but takes a lot of time before triggering (typically memory leaks).

In these situations, either git-bisect will not be usable, or will take a lot of time to converge (up to several weeks), so will reveal inefficient. So the reporter will either stay with the last known working version, or with the new one accompanied with a workaround.

For this reason, we should not "force" reporters to git-bisect. Just ask them if they can do so, otherwise investigations on their bug will not progress until someone else reports the same one, with some time to bisect it. And there is nothing wrong with that IMHO. If the problem only affects one person and this person has a solution, is that really much of a problem ? Sure it would be better fixed, but nobody suffers from it. On the other hand, being aware that there exists a person somewhere experiencing a specific bug is useful to the developers, because when they think they might have fixed it, they can ping him for validation.

Now, from a developer's point of view, the reporters should not consider that development in free software is a public service and that developers have a strong obligation to find and fix new bugs. Mark said his time is paid for, but most of the people here will tend to take that as a customer-provider insult since their time is also paid for, and while the reporter's work may consist in consulting customers without much schedule freedom, the developer's work consists in delivering new features in a more or less agreed schedule. So everyone's time is valuable.

Of course it's better when developers help, and we must keep in mind that they're the better placed to understand their code (even more when it's recent). But due to the long chain of contributors, the ones in direct contact with the reporter are not often the ones who will be able to debug the code. So they need to know a bit more to find whom to ping first.

Both Mark and Ilpo said something true here. It's that they feel concerned when a bug is reported in an area they have worked on. It is possible that none of the people who have worked on this bug was responsible of it, and in this case it's important to insist on the code author about the fact that he's not only a code author but also has to support his code, and that next time he'll be welcome to check if his code might have caused the reported problem.

But clearly, for scalability reasons, we cannot expect people in the middle of the chain to investigate all bugs. Their experience in the area is much better used at assisting both reporters and code authors at taking the right direction though.

So if I can conclude, both reporter's and developer's time is valuable and may not be spent on chasing every bug down. git-bisect is very good at saving developer's time in exchange of approximately the same amount of time on the reporter's side, which makes the whole process scalable. Sometimes for various reasons the reporter cannot do this (or not efficiently). We should not call him names in this case, just tell him that we cannot go further on this bug without much more information, and that he'll be asked for tests when someone else reports it and debugs it. If the person expected more investigative support, he should have gone with a commercially supported distro.

Now speaking for my case, I know that as a developer, I'm faster than many others to find bugs in \*my\* code, but am of little help when it comes to external contributions to my code. As a user, I will not always be able to git-bisect (or that would be inefficient, see reasons above). But I know that a report is a report, and even if I have a workaround, I feel it as a moral obligation to report the bug, and I want to be able to do it without the fear of being aghressed due to my lack of involvement in the fix. An no Dave, I'm not hypocritic when I say this. I really hate people who say "oh yes I know about this bug, I've already encountered it but did not care to report it". I just want to ensure that people will always report bugs, whatever the level of help they will be able to provide. It's important to know if a problem happens for the first time or is very wide-spread since version X or Y. And for such a case, I agree that bugzilla would at least help not losing those reports.

Willy

--

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@xxxxxxxxxxxxxxxxx  
More majordomo info at <http://vger.kernel.org/majordomo-info.html>  
Please read the FAQ at <http://www.tux.org/lkml/>