



Gelato

UNSW

Performance and
Scalability on Itanium

www.gelato.unsw.edu.au

peterc@gelato.unsw.edu.au

© Gelato@UNSW

1

BUGS!

Getting them Stomped

Peter Chubb

Gelato Project

National ICT Australia

and

The University of New South Wales

October 2005

peterc@gelato.unsw.edu.au

© Gelato@UNSW

2

Bugs!

- All software has them
- Your time is wasted on them

It's a truism to say that all software has bugs. In the IA64 world, we seem to hit more bugs per user than some others, perhaps because we're often the first users to try stuff on the Itanium. However, now that that other 64-bit system is becoming popular, some of the more egregious 64-bit bugs are being fixed before we see them. For some reason, Sparc64 and PPC64 didn't find all the problems for us...

Open Source Software

A wonderful concept

Eric Raymond: *Given enough eyeballs all bugs are shallow*

When the open-source software development model *works* then people all over the world collaborate on fixing *your* problems.

But the reverse is true too: with rights come responsibilities. When *you* find a problem, then you should communicate!

One of the reasons that Eric Raymond points out is that in the open source world, the user and the developer can have a common view of what the problem is — because the user can see the source, and, quite often, the developer can share the test case. In the more IP-restrictive closed-source world, that doesn't always happen.

Principles

- An unreported bug is a bug someone else is going to trip on.
- **Report All Bugs**
- and push patches upstream

Next time, it could be you benefitting

Time after time, I've come along to Gelato meetings and find people discussing problems that either have already been fixed upstream, or that noone knows about except the people discussing them.

Almost all open-source software has an 'owner'— someone who cares that it be the best bit of code in the world! Owners want to know about problems, whether they're documentation, incompatibility with other software or hardware, or just plain bugs.

Reporting Bugs: To Whom?

- Who's your upstream contact?
 - Distribution-makers patch their software.
 - So talk to the distributor *first*
- **Debian** <http://bugs.debian.org>
- **RedHat** <https://bugzilla.redhat.com/bugzilla/>
- **SGI** <https://support.sgi.com/caseview>
- **Suse** <https://bugzilla.novell.com>

If you got your software from a place other than the CVS repository on sourceforge (or wherever) then it's likely to have been modified. Therefore, it's important to report problems to the 'vendor' first. For example, bugs in Debian packages should be reported via the Debian bug tracking system, which tracks problems and makes sure that they don't fall out of sight. Often the Debian maintainer is not the same person as the software maintainer; sometimes the problem you're reporting isn't actually in the upstream software, just in the pre-packaged version.

Actually, Debian makes it easy: there's a `reportbug` package; you just have to say,

```
$ reportbug packagename
```

and most of the work in finding the versions, filling in the template etc., is done. Other distributions may have similar mechanisms, I don't know.

Again, for Debian, it's really easy to do

```
$ apt-get source packagename
```

to get the same source that was used to build the binaries that are failing. As a sidenote, it's this kind of developer friendliness that makes Debian so nice.

Generating a Bug Report

- Need a **SIMPLE** recipe for reproducing the problem.
- Need to include relevant information
 - library versions
 - kernel versions
 - hardware architecture
 - data sets,
 - etc

The first step in reporting a bug is being able to reproduce it yourself. If *you* can't reproduce it then *noone* can fix it (after all, the problem may be operator error, or a random hardware failure).

The simpler the process for reproducing the problem, the better, for all concerned.

But what if the maintainer does nothing?

There comes a time though, particularly if the software you're working with is a bit unusual, or the maintainer seems unresponsive, when you need to take matters into your own hands.

What you do next depends on the software. But the general approach is always the same.

- Try the upstream software
- If the problem is there
 - Report to the original maintainer
 - And/or ... Try to fix/work around it yourself
 - Vendor may have a mechanism for this, e.g., NMU for Debian

If you have a fix, even if it's a bad one, tell people. Post it to the relevant mailing lists. Put it into the bugzilla or other tracking systems. A bad fix will likely generate discussion and, often, a correct and simple fix.

Alan Cox: The quickest way to get a bug fixed is an obviously incorrect patch. People can't resist showing you the right way to do something.

Sending an obviously incorrect patch to a mailing list is one way to get responses. Of course if that incorrect patch gets included in the upstream source...

Specific Software

- Kernel (see below)
- Major projects, e.g., glibc, gcc, X (usually have systems)
- Minor projects (Talk to author)

Kernel Bugs

- Find the subsystem maintainer

- MAINTAINERS file

- CC report to appropriate mailing list.

IA64-specific `linux-ia64-owner@vger.kernel.org`

Networking `netdev@oss.sgi.com`

General kernel `linux-kernel@vger.kernel.org`

- Include a patch if possible.

The people on LKML are sometimes very friendly, and sometimes very unfriendly. Grow a thick skin (like a wombat's). Suggest stupid ideas — or sensible ones if you like. (but beware of plausible but *wrong* solutions). IA64-specific things can be sent to the linux-ia64 list or the gelato-technical list — both pretty friendly places.

(I should warn you, some of the other open source communities are *much* more abrasive than the linux kernel mailing list crowd).

Google is forever

Try not to get into flamewars. What you say on a mailing list now can still be picked up months and years later.

Other open-source software

- Large projects generally have a Bugzilla, e.g.,

gcc, glibc <http://sources.redhat.com/bugzilla/>

X

https://bugs.freedesktop.org/enter_bug.cgi?product=x

- (You may need to register to report a bug)
- Smaller projects: send an email to the maintainer.

In Conclusion

- Be part of the open source community
- Share your problems
- Share your fixes
- Others then share with you!

In conclusion then, open source is about being part of the community. There are parts that you'll be *very* interested in, and will take part in the quotidian discussion; other areas that you'll check out only occasionally, as something pops up. But the best way to get problems fixed is to be part of the community, to share problems and results.

BUG stomping

Oct 2005



Gelato

UNSW

Performance and
Scalability on Itanium

www.gelato.unsw.edu.au