



How to contribute to LLVM, Clang, etc

February 2nd, 2014

How to contribute to LLVM
Sylvestre Ledru



License

All codes are published under the « University of Illinois/NCSA Open Source License »
BSD-style

Some of them are also published under MIT
licence (dual-license)

No copyright assignment



Some stats - LLVM

In a Nutshell, The LLVM Compiler Infrastructure...

- ... has had **103,599 commits** made by **379 contributors** representing **1,277,803 lines of code**
- ... is **mostly written in C++** with an **average number of source code comments**
- ... has a **well established, mature codebase** maintained by a **very large development team** with **increasing Y-O-Y commits**
- ... took an estimated **362 years of effort** (COCOMO model) starting with its **first commit in June, 2001** ending with its **most recent commit about 4 hours ago**

30 Day Summary

Jan 1 2014 — Jan 31 2014

1108 Commits

99 Contributors

including 5 new contributors

12 Month Summary

Jan 31 2013 — Jan 31 2014

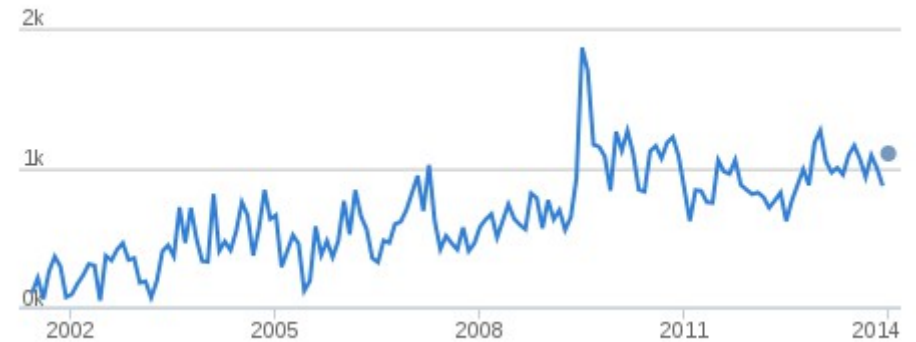
12410 Commits

Up **+1837 (17%)** from previous 12 months

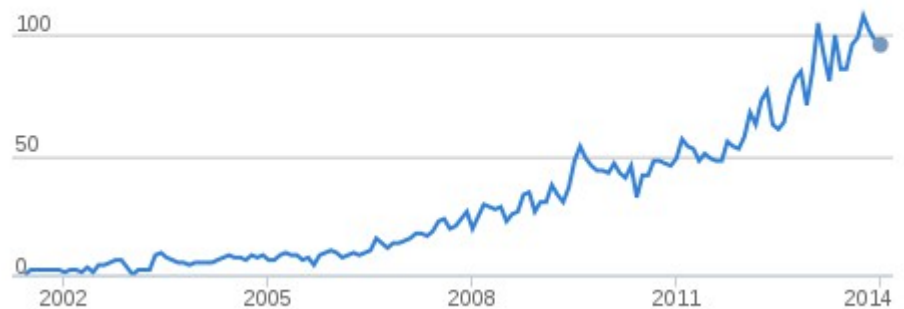
236 Contributors

Up **+57 (31%)** from previous 12 months

Commits per Month



Contributors per Month



February 2nd, 2014

How to contribute to LLVM
Sylvestre Ledru

Source : ohloh.net



Some stats - Clang

In a Nutshell, LLVM/Clang C family frontend...

... has had **49,680 commits** made by **273 contributors** representing **827,471 lines of code**

... is mostly written in C++ with an average number of source code comments

... has a well established, mature codebase maintained by a very large development team with stable Y-O-Y commits

... took an estimated **230 years of effort** (COCOMO model) starting with its **first commit in July, 2007** ending with its **most recent commit about 13 hours ago**

Commits per Month



Contributors per Month



30 Day Summary

Jan 1 2014 — Jan 31 2014

603 Commits

70 Contributors

including 3 new contributors

12 Month Summary

Jan 31 2013 — Jan 31 2014

7494 Commits

Down -390 (4%) from previous 12 months

176 Contributors

Up +34 (23%) from previous 12 months

February 2nd, 2014

How to contribute to LLVM
Sylvestre Ledru

Source : ohloh.net



Community

Friendly

Mainly professional (Apple, Google, ARM,
Linaro, Intel, etc)

With also individual and academic

Usually fast to answer to comments/questions



Mailing lists

Plenty of them (and high traffic!)

LLVM

- LLVM-dev
- LLVM-commits

Clang

- cfe-users
- cfe-dev
- Cfe-commits

Other mailing lists for other LLVM projects



Repositories

Using subversion
git mirrors available

Permissions are granted on all LLVM projects
(clang, compiler-rt, lldb, polly, etc)
and even the websites



Write a patch ?

Make the patch against the current trunks (!)

Made with `svn diff` (`git diff` or `diff -u`)

Must have tests or explain why testing is not possible

Must pass the whole test suite

When relevant, update the release notes



Review process

Documented on

<http://llvm.org/docs/DeveloperPolicy.html>

Minor patches (typo, trivial bug fix, etc) can be committed directly without review

For non-regular contributors, {llvm,cfe,lldb}-commits can be used

No private review



Contributing a patch

- Send to llvm-commits, clang-commits
- Wait for reviewer |\
- | ^ Review loop
- Address comments |/
- Wait for 'LGFM' (Looks go for me)
- Commit patch
 - if you have commit rights -> commit yourself
 - if not -> ask for the patch to be committed



Some advices

- Make the patch as small as possible
- One patch \Leftrightarrow One feature
- Extract unrelated (trivial) fixes into separate patches
- CC possible reviewers



Some advices (bis)

- Pinging patches :
 - Sometimes patches slip through
 - Ping them after 5-7 days (or holidays)
 - Include the latest patch, rebased to 'trunk' in the ping mail
- Make yourself known



Review patches

- Do not need to be a code owner to review patches.
- Reviewers, even if they can not 'LGTM' patches, are highly welcome.
- If you review patches, people are more likely to review your patches.
- Especially review patches in areas you contributed to before.



Large patches / projects

- Discuss the design on the mailing list, before starting the development
- Features are developed, reviewed and committed incrementally
- Make most patches 'trivial'
- Do by no means `_develop_` a large patch in the dark. During the review, you will basically write the feature a second time



Arcanist + phabricator

An other review system !

Phabricator :
A code review platform. Hosted on:
<http://llvm-reviews.chandlerc.com/>



Arcanist + phabricator

The screenshot shows the Phabricator interface for a commit. At the top, there's a navigation bar with the Phabricator logo, a search bar, and utility icons. Below that, the commit title "Improve the detection of the path" is displayed, along with its status "Closed" and visibility "Public".

Metadata for the commit includes:

- Author:** [sylvestre.ledru](#)
- Reviewers:** [rafael](#) (with a green checkmark)
- CCs:** [cfe-commits](#)
- Lint:** [★](#) No Linters Available
- Unit:** [★](#) No Unit Test Coverage
- Commits:** [rL196791: Improve the detection of the path](#)
- Branch:** [/llvm/trunk](#)
- Arcanist Project:** [llvm](#)
- Apply Patch:** [arc patch D2361](#)

On the right side, there is a list of actions: Edit Revision, Automatically Subscribed, Edit Dependencies, Download Raw Diff, Flag For Later, and Award Token. A keyboard shortcuts help icon is also present.

SUMMARY

When clang is used under GNU/Linux in a chroot without /proc mount, it falls back on the BSD method. However, since the buf variable is used twice and fails with sprintf to produce the correct path.

When called as relative (ie ./clang), it was failing with:

```
"" -cc1 [...] -x c++ x.cc
error: unable to execute command: Executable "" doesn't exist!
```

I also took the opportunity to simply the code (the first arg of test_dir was useless).

February 2nd, 2014

How to contribute to LLVM
Sylvestre Ledru



Arcanist + phabricator

Diff 5986

lib/Support/Unix/Path.inc

View Options ▾

Show First 20 Lines • Show All 178 Lines • ▾ Show 20 Lines

```
179 namespace llvm {
180 namespace sys {
181 namespace fs {
182 #if defined(__FreeBSD__) || defined (__NetBSD__) || defined(__Bitrig__) || \
183     defined(__OpenBSD__) || defined(__minix) || defined(__FreeBSD_kernel__) || \
184     defined(__linux__) || defined(__CYGWIN__) || defined(__DragonFly__)
185 static int
186 test_dir(char buf[PATH_MAX], char ret[PATH_MAX],
187          const char *dir, const char *bin)
188 {
189
190     struct stat sb;
191
192     snprintf(buf, PATH_MAX, "%s/%s", dir, bin);
193     if (realpath(buf, ret) == NULL)
194         return (1);
195     if (stat(buf, &sb) != 0)
196         return (1);
197     return (0);
198 }
199
200 static char *
201 getproppath(char ret[PATH_MAX], const char *bin)
202 {
203     char *rv;
204     *rv = *buf[PATH_MAX];
```

```
179 namespace llvm {
180 namespace sys {
181 namespace fs {
182 #if defined(__FreeBSD__) || defined (__NetBSD__) || defined(__Bitrig__) || \
183     defined(__OpenBSD__) || defined(__minix) || defined(__FreeBSD_kernel__) || \
184     defined(__linux__) || defined(__CYGWIN__) || defined(__DragonFly__)
185 static int
186 test_dir(char ret[PATH_MAX], const char *dir, const char *bin)
187 {
188
189
190     struct stat sb;
191     char fullpath[PATH_MAX];
192
193     snprintf(fullpath, PATH_MAX, "%s/%s", dir, bin);
194     if (realpath(fullpath, ret) == NULL)
195         return (1);
196     if (stat(fullpath, &sb) != 0)
197         return (1);
198     return (0);
199 }
200
201 static char *
202 getproppath(char ret[PATH_MAX], const char *bin)
203 {
204     char *rv;
205     *rv = *buf[PATH_MAX];
```

rafael

Line 189 Previous · Next · Reply

extra white space?

February 2nd, 2014

How to contribute to LLVM
Sylvestre Ledru



Arcanist + phabricator

Arcanist :
CLI tools build to interact with phabricator
Works on top of git/mercurial



Get +w permissions

After a few accepted patches
Super easy + fast

Username +
'fullname <email>' +
password hash
To Chris Lattner

Example : Request sent at 15:15, account opened at 22:30



Coding standards

<http://llvm.org/docs/CodingStandards.html>

Defines headers, methods naming, C++ usage
(example : no RTTI/exception), etc



Open Projects

LLVM :

<http://llvm.org/OpenProjects.html>

Clang :

<http://clang.llvm.org/OpenProjects.html>

After that, we have a *few other* projects :
Polly (see later today), lldb, lld, compiler-rt, libc++,
etc



Bug Summary

Bug Type	Quantity	Display?
All Bugs	296	<input checked="" type="checkbox"/>
API		
Argument with 'nonnull' attribute passed null	2	<input checked="" type="checkbox"/>
Dead store		
Dead assignment	100	<input checked="" type="checkbox"/>
Dead increment	11	<input checked="" type="checkbox"/>
Dead initialization	64	<input checked="" type="checkbox"/>
Logic error		
Called C++ object pointer is null	58	<input checked="" type="checkbox"/>
Called C++ object pointer is uninitialized	1	<input checked="" type="checkbox"/>
Called function pointer is null (null dereference)	3	<input checked="" type="checkbox"/>
Dereference of null pointer	18	<input checked="" type="checkbox"/>
Division by zero	3	<input checked="" type="checkbox"/>
Garbage return value	4	<input checked="" type="checkbox"/>
Result of operation is garbage or undefined	8	<input checked="" type="checkbox"/>

- Fix some bugs :
<http://llvm.org/bugs/>
- Fix some (easy) issues found by the static analyzer :
<http://buildd-clang.debian.net/scan-build/>



Questions ?