

Recent Developments in GNU Autotools

Ralf Wildenhues

GHM July 2010
The Hague
Netherlands

Autotools



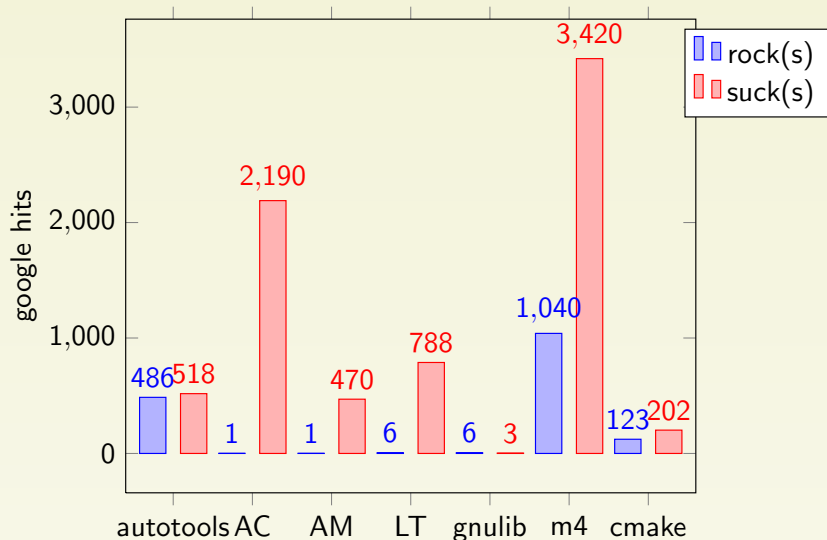
Automacke



What is Autotools?

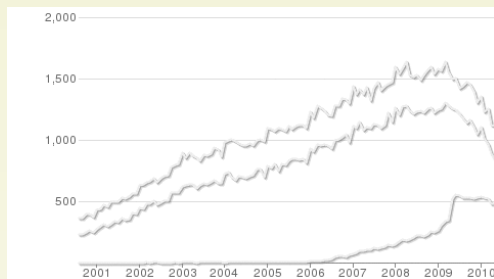
- Autoconf: create portable `configure` and `testsuite` scripts
- Automake: create portable `Makefiles`
- Libtool: create and use shared libraries portably
- Gnulib: pick-and-choose portable `libc` plus lots of goodies

Do we have an image problem?



Users of Autotools: ohloh numbers

Autoconf	623
Automake	574
Libtool	322
Gnulib	26
M4	57
CMake	260



monthly projects using Autoconf, Automake
CMake languages

(open version of ohloh anyone?)

Common perception of Autotools

- portable, conforming to GNU Coding Standards,
- works around incredibly stupid bugs in fairly unusual systems,
- slow tools, generate slow, unreadable code,
- old-fashioned,
- evokes cursing,
- regressions/backward incompatibilities in tools.

Problems

Users suffer from complexity of tools:

- M4 + shell/Posix tools + Autoconf macros + make + “magical” Automake API.
- steep initial learning curve.

Autoconf:

- ok (not good) coverage of perl code (> 80% statement, > 70% condition coverage),
- ok (not good) coverage of macro code,
- API abstraction has gotten better in recent years, but still pretty leaky.

Problems

Automake:

- good coverage of perl API code (90% statement, 80% condition coverage),
- fairly good coverage of macro code and makefile snippets,
- little unit test coverage,
- extremely leaky API abstraction,
- lack of extensibility.

Libtool:

- hard to test: on any given system, 80% of libtool.m4 and 60% of ltmain.sh is dead code,
- parts of ltmain and libltdl need a rewrite.

Did Autotools get Better?

KLoC of tests and rest over the last years:

Year	AC	tests	AM	tests	LT	tests	Gnulib	tests
2003	94	5	71	31	44	6	108	2
2008	117	10	93	41	64	15	488	109
2009	120	11	97	44	65	16	569	172
2010	121	13	99	47	68	19	591	179

Improvements: Fancy

- Autotest: color, hard failures
- `--recheck`
- lazy test execution

- Automake: color for TESTS
- Linux kernel build style `silent-rules`

- no progress bar

Improvements: Optimization

- Autoconf: better scaling behavior in `m4sh` and `m4`
- and in `config.status` (with `awk`)
- shell functions: smaller and faster (20%) configure scripts

- Automake: `parallel-tests` driver
- threaded automake execution
- reliable, efficient multi-file install, `install-sh -C`
- parallel make dist compression

- Libtool: up to 60% faster `libtool --mode=compile`
- some algorithmic bottlenecks in `--mode=link` fixed

Optimizations: Future

- Autoconf: running (some, e.g., `*ONCE`) configure tests in parallel
- general `AC_PARALLEL` statements?
- Automake: further improvements to nonrecursive Makefiles
- Libtool: binary `libtool`?

Improvements: Miscellany

- Autoconf: GPLv3+ plus exception
- better support for Erlang, C99, Objective C(++), Vala
- more precise API definition
- lots of m4sh and m4sugar macros
- multiline substitutions

- Autoconf+Automake: PWD may contain white space

Improvements: Miscellany (2)

- Automake: `AM_MAINTAINER_MODE([enable])`
- `notrans_` manpages
- xz and lzip compression
- `make dist` security issue fixed

Future

- better Fortran support
- w32 and cross compilation
- MSVC

- Automake: better extensibility? `AM_MAKEFILE_INCLUDE`
- Objective C++
- QT
- Cuda?
- better integration for `gcc --coverage`

Improvements: Libtool

- Libtool: the Great Renaming: AC_* -> LT_*
- systems: Amiga OS, BeOS/Haiku, FreeMiNT, kFreeBSD, kNetBSD, RDOS, SFU Interix, compilers on Darwin, WinCE, OpenSolaris, BG/P,
- works with PIE, distcc/ccache
- recursive or nonrecursive embedded libltdl

Libtool Future

- LTO support
- better cross compile and execution support
- per-deplib static/shared
- indirect dependencies

Summary

There is still room for improvement . . .

Please report bugs, hacks, submit improvements!

Thank you!