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?

The graph shows the number of Google hits for various terms. The x-axis represents different terms such as `autotools`, `AC`, `AM`, `LT`, `gnulib`, `m4`, and `cmake`, while the y-axis represents the number of hits. The terms are divided into two categories: rock(s) (blue bars) and suck(s) (red bars). The data points are as follows:

- `autotools`: rock(s) 486, suck(s) 518
- `AC`: rock(s) 1, suck(s) 1
- `AM`: rock(s) 1, suck(s) 470
- `LT`: rock(s) 6, suck(s) 788
- `gnulib`: rock(s) 6, suck(s) 3
- `m4`: rock(s) 123, suck(s) 202
- `cmake`: rock(s) 3,420, suck(s) 2,000
Users of Autotools: ohloh numbers

<table>
<thead>
<tr>
<th>Tool</th>
<th>ohloh count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Autoconf</td>
<td>623</td>
</tr>
<tr>
<td>Automake</td>
<td>574</td>
</tr>
<tr>
<td>Libtool</td>
<td>322</td>
</tr>
<tr>
<td>Gnulib</td>
<td>26</td>
</tr>
<tr>
<td>M4</td>
<td>57</td>
</tr>
<tr>
<td>CMake</td>
<td>260</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Year</th>
<th>AC tests</th>
<th>AM tests</th>
<th>LT tests</th>
<th>Gnulib tests</th>
</tr>
</thead>
<tbody>
<tr>
<td>2003</td>
<td>94</td>
<td>5</td>
<td>71</td>
<td>31</td>
</tr>
<tr>
<td>2008</td>
<td>117</td>
<td>10</td>
<td>93</td>
<td>41</td>
</tr>
<tr>
<td>2009</td>
<td>120</td>
<td>11</td>
<td>97</td>
<td>44</td>
</tr>
<tr>
<td>2010</td>
<td>121</td>
<td>13</td>
<td>99</td>
<td>47</td>
</tr>
</tbody>
</table>
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
There is still room for improvement . . .

Please report bugs, hacks, submit improvements!

Thank you!