ccache Compiler cache

CONCEPTS

ccache hashes the preprocessed source, compiler and flags. On a hit it returns the stored object file instead of compiling. Run a build twice and compare 'ccache -s' to see it working.

ccache gcc ... hash inputs
↓ lookup
Cache hit: copy object
↓ miss
Real compiler store result

Each compile is a lookup keyed by source, flags and compiler

Config precedence (highest first)

Source	Example
Command line	ccache max_size=1G gcc ...
Environment	CCACHE_MAXSIZE=1G
Directory config	ccache.conf up the tree
Cache config	$CCACHE_DIR/ccache.conf
System config	/etc/ccache.conf

RUNNING IT

ccache *<compiler> <args>*
Prefix a compiler command

ln -s $(which ccache) ~/bin/gcc
Masquerade as the compiler

make CC='ccache gcc'
Pass the prefix to make

CC="ccache gcc" ./configure
Use it with autotools

CMAKE_C_COMPILER_LAUNCHER
Set to ccache via -D (also CXX)

ccache -V
Print the version

ccache --help
List all options

ccache *<KEY>=<value>* **gcc ...**
Set one config option for this call

STATISTICS

ccache -s
Show summary of hits and misses

ccache -s -v
Show more detailed statistics

STATISTICS (cont.)

ccache -z
Zero the statistics counters

ccache --print-stats
Machine-readable counters

CACHE MANAGEMENT

ccache -M *<size>*
Set maximum cache size (default GiB)

ccache -M 0
Remove the size limit

ccache -F *<num>*
Set maximum number of files

ccache -c
Clean up to the configured limits

ccache -C
Clear all cached files, keep config

ccache --evict-older-than *<age>***d**
Drop entries unused for N days

ccache -d *<dir>* **-s**
Operate on another cache directory

CONFIGURATION

ccache -p
Show effective config and its origin

ccache -k *<key>*
Print one config value

ccache -o *<key>*=*<value>*
Write a config option to the file

CCACHE_DIR=*<dir>*
Cache location

CCACHE_MAXSIZE=*<size>*
Cache size limit (default 5GiB)

CCACHE_DISABLE=1
Bypass ccache entirely

HIT RATE AND CORRECTNESS

CCACHE_BASEDIR=*<dir>*
Rewrite absolute paths under dir

CCACHE_SLOPPINESS=*<list>*
Relax checks to get more hits

HIT RATE AND CORRECTNESS (cont.)

CCACHE_SLOPPINESS=time_macros
Ignore __DATE__ and __TIME__

CCACHE_SLOPPINESS=pch_defines
Allow precompiled headers

CCACHE_COMPILERCHECK=content
Hash compiler binary contents

CCACHE_COMPILERCHECK=*<command>*
Hash the output of a command

CCACHE_NODIRECT=1
Disable direct mode

CCACHE_DEPEND=1
Use depend mode (compiler writes deps)

CCACHE_RECACHE=1
Ignore existing results and overwrite

DEBUGGING AND SHARING

CCACHE_LOGFILE=*<file>*
Log every decision to a file

RECIPES

Trial: measure the speedup

```
ccache -z
make clean && make -j8
make clean && make -j8
ccache -s   # second build should
be mostly hits
```

Transparent via symlinks

```
mkdir -p ~/ccbin
for c in gcc g++ cc c++; do ln -sf
$(which ccache) ~/ccbin/$c; done
export PATH=$HOME/ccbin:$PATH
gcc --version   # now routed
through ccache
```

CMake project with ccache

```
L=CMAKE_C_COMPILER_LAUNCHER=ccache
X=CMAKE_CXX_COMPILER_LAUNCHER=
ccache
cmake -B build -G Ninja -D$L -D$X
cmake --build build && ccache -s
```

Why is it missing?

```
export CCACHE_LOGFILE=/tmp/
ccache.log
make clean && make
grep -iE 'miss|result' /tmp/
ccache.log | head
ccache -p | grep -E 'sloppiness|
base_dir'
```