NAME

llvm-profdata - Profile data tool

SYNOPSIS

llvm-profdata *command* [*args...*]

DESCRIPTION

The **llvm-profdata** tool is a small utility for working with profile data files.

COMMANDS

- *merge*
- *show*
- *overlap*

MERGE

SYNOPSIS

llvm-profdata merge [*options*] [*filename...*]

DESCRIPTION

llvm-profdata merge takes several profile data files generated by PGO instrumentation and merges them together into a single indexed profile data file.

By default profile data is merged without modification. This means that the relative importance of each input file is proportional to the number of samples or counts it contains. In general, the input from a longer training run will be interpreted as relatively more important than a shorter run. Depending on the nature of the training runs it may be useful to adjust the weight given to each input file by using the **-weighted-input** option.

Profiles passed in via **-weighted-input**, **-input-files**, or via positional arguments are processed once for each time they are seen.

OPTIONS

--help

Print a summary of command line options.

--output=<output>, -o

Specify the output file name. *Output* cannot be **-** as the resulting indexed profile data can't be written to standard output.

--weighted-input=<weight,filename>
Specify an input file name along with a weight. The profile counts of the supplied **filename** will be scaled (multiplied) by the supplied **weight**, where **weight** is a decimal integer >= 1. Input files specified without using this option are assigned a default weight of 1. Examples are shown below.

--input-files=<path>, -f
Specify a file which contains a list of files to merge. The entries in this file are newline-separated. Lines starting with '#' are skipped. Entries may be of the form <filename> or <weight>,<filename>.

--remapping-file=<path>, -r
Specify a file which contains a remapping from symbol names in the input profile to the symbol names that should be used in the output profile. The file should consist of lines of the form **<input-symbol> <output-symbol>**. Blank lines and lines starting with **#** are skipped.

The *llvm-cxxmap* tool can be used to generate the symbol remapping file.

--instr (default)
Specify that the input profile is an instrumentation-based profile.

--sample
Specify that the input profile is a sample-based profile.

The format of the generated file can be generated in one of three ways:

--binary (default)

Emit the profile using a binary encoding. For instrumentation-based profile the output format is the indexed binary format.

--extbinary

Emit the profile using an extensible binary encoding. This option can only be used with sample-based profile. The extensible binary encoding can be more compact with compression enabled and can be loaded faster than the default binary encoding.

--text

Emit the profile in text mode. This option can also be used with both sample-based and instrumentation-based profile. When this option is used the profile will be dumped in the text format that is parsable by the profile reader.

--gcc

Emit the profile using GCC's gcov format (Not yet supported).

--sparse[=true|false]
Do not emit function records with 0 execution count. Can only be used in conjunction with -instr. Defaults to false, since it can inhibit compiler optimization during PGO.

--num-threads=<N>, -j
Use N threads to perform profile merging. When N=0, llvm-profdata auto-detects an appropriate number of threads to use. This is the default.

--failure-mode=[any|all]
Set the failure mode. There are two options: 'any' causes the merge command to fail if any profiles are invalid, and 'all' causes the merge command to fail only if all profiles are invalid. If 'all' is set, information from any invalid profiles is excluded from the final merged product. The default failure mode is 'any'.

--prof-sym-list=<path>
Specify a file which contains a list of symbols to generate profile symbol list in the profile. This option can only be used with sample-based profile in extbinary format. The entries in this file are newline-separated.

--compress-all-sections=[true|false]
Compress all sections when writing the profile. This option can only be used with sample-based profile in extbinary format.

--use-md5=[true|false]
Use MD5 to represent string in name table when writing the profile. This option can only be used with sample-based profile in extbinary format.

--gen-partial-profile=[true|false]
Mark the profile to be a partial profile which only provides partial profile coverage for the optimized target. This option can only be used with sample-based profile in extbinary format.

--supplement-instr-with-sample=<file>

Supplement an instrumentation profile with sample profile. The sample profile is the input of the flag. Output will be in instrumentation format (only works with -instr).

--zero-counter-threshold=<float>
For the function which is cold in instr profile but hot in sample profile, if the ratio of the number of zero counters divided by the total number of counters is above the threshold, the profile of the function will be regarded as being harmful for performance and will be dropped.

--instr-prof-cold-threshold=<int>
User specified cold threshold for instr profile which will override the cold threshold got from profile summary.

--suppl-min-size-threshold=<int>
If the size of a function is smaller than the threshold, assume it can be inlined by PGO early inliner and it will not be adjusted based on sample profile.

--debug-info=<path>
Specify the executable or **.dSYM** that contains debug info for the raw profile. When **-debug-info-correlate** was used for instrumentation, use this option to correlate the raw profile.

EXAMPLES

Basic Usage

Merge three profiles:

```
llvm-profdata merge foo.profdata bar.profdata baz.profdata -output merged.profdata
```

Weighted Input

The input file **foo.profdata** is especially important, multiply its counts by 10:

```
llvm-profdata merge --weighted-input=10,foo.profdata bar.profdata baz.profdata --output merged.profdata
```

Exactly equivalent to the previous invocation (explicit form; useful for programmatic invocation):

```
llvm-profdata merge --weighted-input=10,foo.profdata --weighted-input=1,bar.profdata --weighted-input=1,baz.pr
```

SHOW

SYNOPSIS

llvm-profdata show [*options*] [*filename*]

DESCRIPTION

llvm-profdata show takes a profile data file and displays the information about the profile counters for this file and for any of the specified function(s).

If *filename* is omitted or is **-**, then **llvm-profdata show** reads its input from standard input.

OPTIONS

--all-functions
Print details for every function.

--binary-ids
Print embedded binary ids in a profile.

--counts
Print the counter values for the displayed functions.

--show-format=<text|json|yaml>
Emit output in the selected format if supported by the provided profile type.

--function=<string>
Print details for a function if the function's name contains the given string.

--help
Print a summary of command line options.

--output=<output>, -o
Specify the output file name. If *output* is **-** or it isn't specified, then the output is sent to standard output.

--instr (default)
Specify that the input profile is an instrumentation-based profile.

--text
Instruct the profile dumper to show profile counts in the text format of the instrumentation-based profile data representation. By default, the profile information is dumped in a more human readable form (also in text) with annotations.

--topn=<n>
Instruct the profile dumper to show the top **n** functions with the hottest basic blocks in the

summary section. By default, the topn functions are not dumped.

--sample
Specify that the input profile is a sample-based profile.

--memop-sizes
Show the profiled sizes of the memory intrinsic calls for shown functions.

--value-cutoff=<n>
Show only those functions whose max count values are greater or equal to **n**. By default, the value-cutoff is set to 0.

--list-below-cutoff
Only output names of functions whose max count value are below the cutoff value.

--profile-version
Print profile version.

--showcs
Only show context sensitive profile counts. The default is to filter all context sensitive profile counts.

--show-prof-sym-list=[true|false]
Show profile symbol list if it exists in the profile. This option is only meaningful for sample-based profile in extbinary format.

--show-sec-info-only=[true|false]
Show basic information about each section in the profile. This option is only meaningful for sample-based profile in extbinary format.

--debug-info=<path>
Specify the executable or **.dSYM** that contains debug info for the raw profile. When **-debug-info-correlate** was used for instrumentation, use this option to show the correlated functions from the raw profile.

--covered
Show only the functions that have been executed, i.e., functions with non-zero counts.

OVERLAP

SYNOPSIS

llvm-profdata overlap [*options*] [*base profile file*] [*test profile file*]

DESCRIPTION

llvm-profdata overlap takes two profile data files and displays the *overlap* of counter distribution between the whole files and between any of the specified functions.

In this command, *overlap* is defined as follows: Suppose *base profile file* has the following counts: {c1_1, c1_2, ..., c1_n, c1_u_1, c2_u_2, ..., c2_u_s}, and *test profile file* has {c2_1, c2_2, ..., c2_n, c2_v_1, c2_v_2, ..., c2_v_t}. Here c{1|2}_i (i = 1 .. n) are matched counters and c1_u_i (i = 1 .. s) and c2_v_i (i = 1 .. v) are unmatched counters (or counters only existing in) *base profile file* and *test profile file*, respectively. Let sum_1 = c1_1 + c1_2 + ... + c1_n + c1_u_1 + c2_u_2 + ... + c2_u_s, and sum_2 = c2_1 + c2_2 + ... + c2_n + c2_v_1 + c2_v_2 + ... + c2_v_t. *overlap* = min(c1_1/sum_1, c2_1/sum_2) + min(c1_2/sum_1, c2_2/sum_2) + ... + min(c1_n/sum_1, c2_n/sum_2).

The result overlap distribution is a percentage number, ranging from 0.0% to 100.0%, where 0.0% means there is no overlap and 100.0% means a perfect overlap.

Here is an example, if *base profile file* has counts of {400, 600}, and *test profile file* has matched counts of {60000, 40000}. The *overlap* is 80%.

OPTIONS

--function=<string>
Print details for a function if the function's name contains the given string.

--help
Print a summary of command line options.

--output=<output>, -o
Specify the output file name. If *output* is **-** or it isn't specified, then the output is sent to standard output.

--value-cutoff=<n>
Show only those functions whose max count values are greater or equal to **n**. By default, the value-cutoff is set to max of unsigned long long.

--cs Only show overlap for the context sensitive profile counts. The default is to show non-context sensitive profile counts.

EXIT STATUS

llvm-profdata returns 1 if the command is omitted or is invalid, if it cannot read input files, or if there is a mismatch between their data.

AUTHOR

Maintained by the LLVM Team (https://llvm.org/).

COPYRIGHT