NAME

`mrprof` – (Method R Profiler) profile the duration of an Oracle application program execution

SYNOPSIS

`mrprof` [`--advisory`] [`--cdefs=`*file*] [`--create`] [`--elide`] [`--encoding=`*string*] [`--error-emit=`*number*] [`--etc=`*dir*] [`--fix-tim-values`] [`--force-match-plans`] [`--force-match-statement-texts`] [`--help` | `-?`] [`--html`] [`--indent`] [`--license`] [`--load`] [`--man`] [`--ofile=`*string*] [`--open`] [`--param=`*string*] [`--params-file=`*file*] [`--patch-negative-counts`] [`--progress=`*file*] [`--recursive` | `-R`] [`--script=`*string*] [`--script-comment=`*string*] [`--script-extension=`*string*] [`--shareable`] [`--sparam=`*string*] [`--sql-text-width=`*number*] [`--stash=`*number*] [`--trace=`*string*] [`--trcsess`] [`--trcunit=`*number*] [`--version`] [`--workers=`*number*] [`--xsd=`*file*] *file* ...

DESCRIPTION

`mrprof` (Method R Profiler) is a fast, high-precision response time profiler for Oracle Database application programs. It reads Oracle Database event 10046 extended SQL trace data to produce an HTML5 report that application developers, database administrators, and performance analysts use to learn how their program durations are being consumed. Profile reports are useful for code reviews, performance tests, concept proofs, hardware and software evaluations, upgrades, troubleshooting, ... anytime it is important to understand where an Oracle application program's time is being spent.

`mrprof` accounts for every microsecond of time within its input to provide a complete explanation of where a program's time has gone. Other tools to which you may be accustomed, like Oracle's `tkprof`, do not. New `mrprof` users are often surprised by the ramifications of this sensitivity. When using `mrprof`, it is crucial to remember:

> ***The duration your trace file explains must match the duration of the end-user experience you want to diagnose.***

Method R Workbench includes a utility called `mrcrop`, which enables you to perfom high-precision cropping operations upon trace files containing data that you wish to exclude from a diagnosis.

`mrprof` determines its action based upon the type of the file that is passed to it.

`.trc` If you specify a `.trc` file, `mrprof` will produce an XML file and, by default, an HTML file.

`.trc.gz` or `.trc.bz2` If you specify a `.trc.gz` or `.trc.bz2` file, `mrprof` will create a `.trc` file and then process it in the normal way described above.

`mrprof` uses the same case-tolerance rules as its underlying file system. For example, on a Linux system, `x.trc` and `X.trc` are different files, whereas on a Microsoft Windows system, the two file names are synonymous.

OPTIONS

`--advisory` Print advisory text in the HTML profile report. The default is `--advisory`. Use `--noadvisory` to suppress the advisory text.

`--cdefs=`*file* Read call definitions from *file*. The default is

`$HOME/.method-r/workbench/9.2.1.2/cdefs.xml`

`--create` When loading results into the Workbench database, create the database tables and upgrade datatbase tables to the current release. Use this option when starting the Workbench application is not the first post-installation action you perform or if you remove its database. The default is `--nocreate`.

`--elide` Use the negated form of this option to override the current transform parameter `PROFILE-DISPLAYED-THRESHOLD` setting with a value of 0, which means all the details of every subroutine call and every SQL and PL/SQL statement will be in the profile report. Use `--noelide` to ensure all details appear in the profile report. The default is `--elide`, which means that details that are probably irrelevant will not appear in the profile report.

`--encoding=`*string* Declare the encoding for the input trace file. The default is `UTF-8`. Choose one of the following values:

```
UTF-8         UTF-16
ISO-8859-1    ISO-8859-2   ISO-8859-3
ISO-8859-4    ISO-8859-5   ISO-8859-6
ISO-8859-7    ISO-8859-8   ISO-8859-9
ISO-2022-JP   Shift_JIS    EUC-JP
```

`--error-emit=`*number* Emit details for only *number* instances of a given error type into the HTML report. The default is 5. Use 0 to show all errors in detail, but beware: using 0 can cause `mrprof` to consume a lot of memory and cause browser performance problems when viewing the HTML report.

`--etc=`*dir* Use *dir* for external references made to `/etc` from the output HTML. The default value is `/etc` in the directory where you installed Method R Workbench.

`--fix-tim-values` Attempt to fix out-of-order `tim` values (Oracle Database software bugs). The default is `--nofix-tim-values`.

--force-match-plans Report a SQL statement's execution plans as one execution plan when they have the same matching signature but different plan hash values. The first one encountered will be reported and it will contain the aggregation of the statistics for all plans having that matching signature. The default value is `--force-match-plans`. Use `--noforce-match-plans` to disable.

--force-match-statement-texts Report SQL statements as one statement when they have the same matching signature but different sqlid values. The first one encountered will be reported with the aggregation of the statistics for all statements having that matching signature. Use `--noforce-match-statement-texts` to disable. The default is `--force-match-statement-texts`.

--help, -? Print usage information and exit.

--html Create the HTML report, in a `.html` file with the same base name as the input file. The default is `--html`. Use `--nohtml` if you want not to create HTML output.

--indent Indent HTML output. The default is `--noindent`. Indenting HTML is useful primarily for HTML debugging purposes. Note specifically that using `--indent` will cause `mrprof` to generate much larger HTML files and prohibit `mrprof`'s SQL formatter from doing its job properly.

--license Print license key information and exit.

--load Load results into the Workbench application database, making them visible the next time you start the applicaiton. The default is `--noload`.

--man Print the manual page and exit.

--ofile=*string* Use *string* as a format string for the output file base name. Where you specify the string `%s`, `mrprof` will insert the base name of the input file. If you do not specify `%s` in *string*, then `mrprof` will append the base name to the end of the *string* you specify. The default is `--ofile=%s`.

This flexible mechanism allows you to choose output filename prefixes or suffixes. A prefix can include a path name. For example, if your command is `mrprof prod_ora_1492.trc`, the relationship between `--ofile` value and output file name is shown in the following table:

`--ofile` value	Output file name
`%s-spec`	`prod_ora_1492-spec.html`
`1-%s-spec`	`1-prod_ora_1492-spec.html`
`abc-`	`abc-prod_ora_1492.html`
`../`	`../prod_ora_1492.html`
`../output/%s`	`../output/prod_ora_1492.html`
`d:\tmp\mrprof\`	`d:\tmp\mrprof\prod_ora_1492.html`

--open Open each output HTML file after creating it. The default is **--open** if you provide only one input file name; the default is **--noopen** if you provide multiple input file names.

--param=*string* Use string to override a **prof.xml** parameter setting. The string must have the form *name=value*; for example, **--param=PROFILE-DISPLAYED-THRESHOLD=0.01**. You may specify multiple name-value pairs by using separate **--param** option specifications on a single command line.

--params-file=*file* Read transformation parameters from *file*. The default value is **$HOME/.method-r/workbench/9.2.1.2/prof.xml**.

--patch-negative-counts Manipulate parent call **p**, **cr**, and **cu** values that are less than or equal to the sum of the call's children's corresponding **p**, **cr**, and **cu** values (because of Orace Database software imperfections). The default is **--patch-negative-counts**. Use **--nopatch-negative-counts** to not manipulate values.

--progress=*file* Update *file* with a percentage representing how much input trace data has been converted into XML form.

--recursive or -R If a given **file** is a directory, then process it recursively looking for .trc files. The default is **--norecursive**.

--script=*string* Use either Microsoft Windows or Unix style scripting. Valid *string* values are **mswin** and **unix** (values are not case sensitive). The effects of **--script** can be overridden with the individual **--script-*** options. The default value is **MSWin**. This option sets the individual **--script-*** options as follows.

string	--script-extension	--script-comment
mswin	.bat	rem
unix	(empty string)	#

--script-comment=*string* Use *string* as the comment designator for any scripts this program will generate (for example, by using the **--trcsess** option). The default value is **rem**.

--script-extension=*string* Use *string* as the filename extension for any scripts this program will generate (for example, by using the **--trcsess** option). The default value is **.bat**.

--shareable Create HTML output with embedded graphics and CSS that will render properly on someone else's system, even if that person doesn't have the **/etc** directory contents that you do. The default value is **--shareable**. Use **--noshareable** to render HTML with links instead of embedded graphics and CSS. Using **--noshareable** will make HTML files that are

smaller but less convenient.

--sparam=*string* Send the option specified in *string* to the Saxon XSLT processor. You may use this option multiple times on a single command line. Use `--sparam=-?` to see information about which parameters are allowed.

--sql-text-width=*number* Format SQL statement text so that each line contains `number` or fewer bytes. The default is `--sql-text-width=180`.

--stash=*number* Distinguish `WAIT` lines for a given cursor between no more than this `number` of intervening dbcalls so that if some of this cursor's `WAIT` lines belong to some missing dbcall, the `WAIT` lines will be properly attributed. The default is `--stash=100`.

--trace=*string* Write diagnostic information about the input file processing to *f*`.trace`, where *f* is the base name of the input file. The type of information written is controlled by zero or more of the following keys (characters) in *string*. For example, `--trace=ut` will show input lines annotated with call start time values and unaccounted-for details. The default value is the empty string, which causes no *f*`.trace` file to be written.

Key	Writes...
`*`	all tracing information
`1`	basic diagnostic information
`c`	character set conversion setup info
`d`	check tim drift
`h`	shared statement identification details
`l`	input lines
`s`	synthetic dbcall details
`t`	input lines with t0 values appended
`u`	unaccounted-for time details
`U`	unaccounted-for time summary
`w`	`WAIT` line details

--trcsess Generate a script that will run Oracle `trcsess` commands to split a trace file containing information about two or more Oracle sessions into a set of files containing information about exactly one Oracle session per trace file. The script file base name is *f*`-trcsess`, where *f* is the base name of the input file. The script file extension is determined by the setting of `--script-extension`. The default value is `--trcsess`. To suppress the `trcsess` script generation, use `--notrcsess`.

--trcunit=*number* Regard time and duration values in the raw trace file as being expressed in units of *number* seconds. The default value is `.000001`. Oracle Database versions since 9.0 express all trace file time and duration values in microseconds, for which you should use `--trcunit=.000001`, the default. However, on some platforms prior to version 11.2, the Oracle

kernel reports time and durations in 1,024-nanosecond units. On these platforms, you should use `--trcunit=.000001024`.

`--version` Print the version number and exit.

`--workers=`***number*** Use *number* concurrent workers to process the input files. The default value is `--workers=1`.

`--xsd=`***file*** Emit *file* as an XML schema definition reference into the output XML file if *file* is a non-empty string. The default value is the empty string.

TRANSFORM PARAMETERS

`mrprof` gives you access to several transform parameters that allow you to govern the appearance of your profile reports. You can specify these values either on the command line with the `--param` option, or you can record the values you want permanently in the `prof.xml` file specified by the `--params-file` option. The syntax of this file is:

```
<params>
    <param name="LOCALE-CODE">D</param>
    <!--
    XML comment syntax lets you disable settings without deleting them.
    <param name="CONTRIBUTIONS-MINIMUM-LIST-LENGTH">0</param>
    -->
</params>
```

If a parameter is not listed in the `prof.xml` file, `mrprof` will use its default value. `mrprof` recognizes the following parameters:

`ADVISORY` Print advisory paragraphs that narrate the data in the profile report. The default value is `true`. To suppress the advisory paragraphs, use `false`.

`CONTRIBUTIONS-MINIMUM-LIST-LENGTH` `mrprof` will print contributions in profiles that have at least this many rows in them, provided that this many contributing elements exist. To see all contributors for a given profile, set the value to `0`. However, be aware that printing all contributions may seriously degrade transform runtime and memory performance. The default value is `4`.

`CREATE-SHAREABLE-HTML` Create shareable HTML files by including images and styles directly within the HTML itself. The default value is `true`.

`DISTINCT-TEXT-THRESHOLD` If the number of similar but distinct texts exceeds this value for a given statement, then `mrprof` highlights the distinct text count for the statement. The default value is `1`.

`EXECS-PER-PARSE-THRESHOLD` If the application executes fewer than this quantity times the parse call count, then `mrprof` highlights the parse call count.

The default value is 2.

LIBRARY-CACHE-MISSES-PER-STATEMENT-THRESHOLD If the number of Oracle database library cache misses is greater than this value for a given statement, then mrprof highlights the library cache miss count. The default value is 1.

LIOS-PER-ROW-THRESHOLD If the number of LIOs is greater than this value times the number of rows processed, then mrprof highlights the LIO count. The default value is 10.

LOCALE-CODE mrprof uses this value to determine its output number format. The domain of values and an example of each follows. The default value is A:

LOCALE-CODE	Format
A	123,456,789.000000
B	123 456 789.000000
C	123.456.789,000000
D	123 456 789,000000
E	123,456,789 · 000000
F	123'456'789,000000
G	12,34,56,789.000000
H	1,2345,6789.000000

PIOS-PER-LIO-THRESHOLD If the number of PIOs is greater than this value times the number of LIOs, then mrprof highlights the PIO count. The default value is .1.

PROFILE-DISPLAYED-THRESHOLD mrprof will print details for any cursor or statement whose magnitude of contribution to total experience duration is greater than or equal to this value times the magnitude of the total experience duration. That is, if a contribution's elapsed time is e, the value of PROFILE-DISPLAYED-THRESHOLD is t, and the total experience duration is R, then mrprof will print details for the contribution only if $|e| \geq t|R|$. It elides details for cursors and statements whose contributions fall short of this threshold. To see all contributors for a given profile, set the value to -1. However, be aware that rendering all contributions may seriously degrade transform runtime and memory performance. The default value is .05.

PROFILE-RELEVANCE-THRESHOLD mrprof will highlight rows for elements whose magnitude of contribution to total experience duration is greater than or equal to this value times the magnitude of the total experience duration. That is, if a contribution's elapsed time is e, the value of PROFILE-RELEVANCE-THRESHOLD is t, and the total experience duration is R, then mrprof will highlight the contribution only if $|e| \geq t|R|$. The

default value is `.20`.

`ROWS-PER-EXEC-THRESHOLD` If there are no Oracle fetch calls for a statement and the application manipulates fewer than this many rows per Oracle `EXEC` call, then `mrprof` highlights the `EXEC` call count. The default value is `2`.

`ROWS-PER-FETCH-THRESHOLD` If the application retrieves fewer than this many rows per Oracle `FETCH` call, then `mrprof` highlights the `FETCH` call count. The default value is `2`.

`SHOW-CONFIDENTIALITY-LABEL` If this value is `true`, then `mrprof` will render a confidentiality label at the top of the report. The default value is `false`.

`SHOW-CUMULATIVE-DURATIONS` If this value is `true`, then `mrprof` will render columns showing cumulative duration information. Using `true` makes it easier to understand the relevance of a given row of data. Using `false` conserves horizontal space in the report, making it easier to print. The default value is `false`.

`SHOW-LITERAL-HIGHLIGHTING` If this value is `true`, then `mrprof` will highlight literal values in SQL statements. The default value is `true`.

`SHOW-PROFILER-STATEMENT-ID` If this value is `true`, then `mrprof` will render its internal statement ID for each statement printed in the report. The default value is `false`.

`SHOW-ZERO-VALUE-STRING` If this value is `true`, then `mrprof` will print an unintrusive middot character into the report instead of actual zero values. The default value is `true`, which removes significant visual clutter from the printed data.

`SQL-TEXT-PREFIX-LENGTH` `mrprof` will print up to this many characters of statement text in each Profile by Cursor row. Decreasing the size of this number conserves more horizontal space in the report. The default value is `48`.

CALL SUBTYPES AND TOLERANCES

`mrprof` uses highlighting to attract your attention to items in the report that you shouldn't overlook. For example if the average disk read duration for an experience is too high, `mrprof` will highlight it for you. But how does `mrprof` know how to define "too high"? That's where subtypes and tolerances come in.

`mrprof` highlighting rules are driven by configuration parameters that you can adjust. For example, on an old system you might want to highlight disk read calls that take longer than 10 ms (0.010 seconds). On a newer system, you might want to highlight disk read calls that take longer than 2 ms.

But in reality, things are even more complicated than that. There's no such thing as a single correct tolerance for disk read calls of all sizes. For example,

imagine that your tolerance for a 2-block read is 4 ms. If a 2-block read takes 5 ms, you want to see a highlight on that tolerance violation. But if a 32-block read takes 5 ms, you probably don't want alarm bells going off. `mrprof` allows you to specify tolerances using a powerful sub-typing mechanism that can both highlight exceptional calls and allow you to group by call attributes.

`mrprof` subtyped definitions are recorded in the `cdefs.xml` file specified by your `--cdefs` option. Here is an example of a `mrprof` call subtype definition:

```
<call name="db file scattered read">
    <subtype
        qualifier="$p3 &lt;= 16"
        qualifier-label="blocks &le; 16"
        tolerance="0.004"
    />
    <subtype
        qualifier="$p3 &lt;= 32"
        qualifier-label="16 &lt; blocks &le; 32"
        tolerance="0.008"
    />
    <subtype
        qualifier-label="blocks &gt; 32"
        tolerance="0.016"
    />
</call>
```

With the example definition shown above, whenever `mrprof` finds a call named `db file scattered read`, it will evaluate the qualifier expressions (in document order) until it finds a match.

Qualifier expressions use Perl expression syntax, but to use the Perl `<`, `>`, and `&` operator symbols, you must refer to them in XML form (as you can see in the `cdefs.xml` example above):

Symbol	XML form
`<`	`<`
`>`	`>`
`&`	`&`

You can refer to the following variables:

`$e` The elapsed duration of the subroutine call.

`$p1, $p2, $p3` The value of `PARAMETER1`, `PARAMETER2`, or `PARAMETER3`, respectively, for the given call. See your Oracle Database `V$EVENT_NAME` documentation for details.

Thus, `$p3 <= 16` in the example above means "the `PARAMETER3` value is less

than or equal to 16". For `db file scattered read` calls, `PARAMETER3` is the number of blocks retrieved by an O/S read call, so the subtype definitions above will sub-type `db file scattered read` calls based on how many blocks they return:

- If the call returns 16 or fewer blocks, then `mrprof` will name the call `db file scattered read [blocks <= 16]` and use a duration tolerance of 0.004 seconds.
- Otherwise, if the call returns 32 or fewer blocks, then `mrprof` will name the call `db file scattered read [16 < blocks <= 32]` and use a duration tolerance of 0.008 seconds.
- Otherwise, `mrprof` will name the call `db file scattered read [blocks > 32]` and use a duration tolerance of 0.016 seconds.

In the call subtype defintion syntax, each qualifier-label attribute is optional. If you specify a qualifier label, `mrprof` will append that label in [square brackets] to the end of the call name.

Likewise, each tolerance attribute value is optional. If you specify a tolerance, `mrprof` will use it to determine whether to highlight the sub-typed call's mean call duration. If the mean duration of a sub-typed call exceeds the stated tolerance, then `mrprof` will highlight the mean duration value.

Note: `mrprof` does not honor the use of qualifier or qualifier-label attributes for synthetic calls like `CPU: PARSE dbcalls` and `unaccounted-for within dbcalls`. Use these attributes only with the Oracle timed event call names described in `V$EVENT_NAME`.

SYNTHETIC CALLS

Imagine tracing a program that runs for an hour, from 8:00 to 9:00, creating exactly an hour's worth of trace data. But imagine that someone disabled tracing at 8:05 and then re-enabled it at 8:55. Your trace file will still span the 8:00–9:00 interval, but it will contain execution details for only 10 minutes (8:00–8:05 and 8:55–9:00). For the other 50 minutes (8:05–8:55), the trace file would not contain all the details about what the program did.

If your team needed you to explain how the 8:00–9:00 program spent its hour, you couldn't do it without explaining that 50 minutes (83%) of that hour were inexplicable. `mrprof` has the same problem. It will see the 50-minute gap and explain it the best it can, basically by naming the gap and telling you everything it knows about it. But with 83% of your time unexplained, you're probably going to have to collect another trace file for the program.

Even if your trace file exactly explains the user experience you want to analyze, imperfections in the Oracle Database kernel's instrumentation will cause gaps of unexplained time in a trace file, too. Regardless of where the gaps come

from, `mrprof` uses contextual information from the trace file to give you as much information about all of its gaps as it can. It *synthesizes* calls to explain what it found. These synthetic calls appear in "Profile by Subroutine" and "Profile by Database Call" sections of the profile report. Each synthetic call whose duration is relevant to your analysis is described in the profile advisory text printed in the HTML report.

TROUBLESHOOTING

How do I make my skew pictograms look like the ones at method-r.com?

Skew pictograms are supposed to resemble a city skyline. If yours look like a row of empty boxes or diamonds with question marks in them, then you are missing a font. Install either Profiler Pictogram (distibuted in `fonts/profilerpictogram.tff`), or Arial Unicode MS.

ENVIRONMENT

MRPROF_JVM_OPTS

You can pass options to the Java virtual machine by setting `MRPROF_JVM_OPTS`. For example, you can change `mrprof`'s maximum Java heap size to 4GB by using `MRPROF_JVM_OPTS=-Xmx4G`. Options set with this environment variable will behave as if they were entered on the command line to the left of the other options that you enter explicitly. You can see all the options you can use with the `java -X` command.

JAVA_HOME and PATH

Put the directory of the Java executable that you want to use at the head of your `PATH`, or set the `JAVA_HOME` environment variable to the parent of the `bin` directory that contains the Java executable that you want to use. `JAVA_HOME` takes precedence over `PATH`.

EXIT STATUS

Exit status is 0 for success, 1 for failure.

BUGS AND DEFICIENCIES

`mrprof` works only with extended SQL trace files generated by Oracle Database versions 11.1 and newer.

AUTHOR

Jeff Holt, Cary Millsap

SUPPORT

`mrprof` 9.2.1.2

For support, visit https://method-r.com/support.

COPYRIGHT AND LICENSE