

Package ‘proftools’

April 19, 2009

Title Profile Output Processing Tools for R

Version 0.0-2

Author Luke Tierney <luke@stat.uiowa.edu>

Description Tools for examining Rprof profile output.

Maintainer Luke Tierney <luke@stat.uiowa.edu>

Suggests graph,Rgraphviz

License GPL

Repository CRAN

Date/Publication 2007-10-09 19:58:11

R topics documented:

flatProfile	1
plotProfileCallGraph	3
printProfileCallGraph	4
profileCallGraph2Dot	6
readProfileData	8

Index	10
--------------	-----------

flatProfile	<i>Flat Profile for Rprof Profile Data</i>
-------------	--

Description

Computes a flat profile reflecting time spent in functions themselves (self) and functions plus callees (total).

Usage

```
flatProfile(pd, byTotal = TRUE)
```

Arguments

pd	profile data as returned by readProfileData.
byTotal	logical; sort by total time if true, self time if not.

Details

If `byTotal` is true then the result is analogous to the `by.total` component of the result returned by `summaryRprof`. Otherwise, the result is analogous to the `by.self` component returned by `summaryRprof` but with an additional cumulative self times column. The result returned when `byTotal` is not true is analogous to the flat profile produced by `gprof`.

Value

A matrix with one row per function recorded in the profile data.

Author(s)

Luke Tierney

References

User manual for `gprof`, the GNU profiler.

See Also

[Rprof](#), [summaryRprof](#), [readProfileData](#), [plotProfileCallGraph](#), [printProfileCallGraph](#), [profileCallGraph2Dot](#)

Examples

```
## Not run:
## Rprof() is not available on all platforms
Rprof(tmp <- tempfile())
example(glm)
Rprof()
flatProfile(readProfileData(tmp))
flatProfile(readProfileData(tmp), FALSE)
unlink(tmp)

## End(Not run)
```

plotProfileCallGraph

Plot Call Graph for Rprof Profile Data

Description

Uses the graph and Rgraphviz packages to plot a call graph for profile data produced by Rprof.

Usage

```
plotProfileCallGraph(pd, layout = "dot", score = c("total", "self"),
  transfer = function(x) x, colorMap = NULL,
  mergeCycles = FALSE, edgesColored = TRUE,
  rankDir = "LR", ...)
```

Arguments

pd	profile data as returned by readProfileData.
layout	The layout method to use: One of "dot", "neato", and "twopi".
score	character string specifying whether to use total time or self time for coloring nodes/edges; no color used if missing.
transfer	function; maps score values in unit interval to unit interval
colorMap	character vector of color specifications as produced by rainbow; transfer of score is mapped to color
mergeCycles	logical; whether to merge each cycle of recursion into a single node
edgesColored	logical; whether to color edges
rankDir	The direction that the plot is laid out in, one of either "TB" for Top-to-Bottom or "LR" for Left-to-Right. The default value is "LR". This argument is only useful for dot layouts.
...	additional arguments for the graphNEL plot method.

Details

Color is used to encode the fraction of total or self time spent in each function or call. The scores used correspond to the values in the printed representation produced by printProfileCallGraph. For now, see the gprof manual for further details. The color encoding for a score s and a color map m is $m[\text{ceiling}(\text{length}(m) * \text{transfer}(s))]$

Value

Used for side effect.

Note

Because of lazy evaluation, nested calls like $f(g(x))$ appear in the profile graph as f or one of its callees calling g .

Author(s)

Luke Tierney

References

User manual for gprof, the GNU profiler.

Graphviz: <http://www.research.att.com/sw/tools/graphviz/>

See Also

[Rprof](#), [summaryRprof](#), [readProfileData](#), [flatProfile](#), [printProfileCallGraph](#), [profileCallGraph2Dot](#)

Examples

```
## Not run:
## Rprof() is not available on all platforms
Rprof(tmp <- tempfile())
example(glm)
Rprof()
plotProfileCallGraph(readProfileData(tmp))
plotProfileCallGraph(readProfileData(tmp), score = "total")
unlink(tmp)

## End(Not run)
```

```
printProfileCallGraph
      Print Call Graph for Rprof Profile Data
```

Description

Prints a representation of the call graph for profile data produced by `Rprof`. Output can be directed to a connection or a file.

Usage

```
printProfileCallGraph(pd, file = stdout(), percent = TRUE)
```

Arguments

<code>pd</code>	profile data as returned by <code>readProfileData</code> .
<code>file</code>	a connection or the name of the file where the profile graph will be written.
<code>percent</code>	logical; if true use percent of total time; otherwise use time in seconds

Details

`printProfileCallGraph` produces a printed representation of the call graph for profile data produced by `Rprof`. The representation is analogous to the call graph produced by `gprof` with a few minor changes. eventually more complete documentation of the format will be provided here; for now, reading the `gprof` manual section on the call graph should help understanding this output. The output is similar enough to `gprof` output for the `cgprof` script to be able to produce a visual representation of the call graph via `Graphviz`.

Value

Used for side effect.

Note

Because of lazy evaluation, nested calls like `f(g(x))` appear in the profile graph as `f` or one of its callees calling `g`.

Author(s)

Luke Tierney

References

User manual for `gprof`, the GNU profiler.

`cgprof`: <http://mvertes.free.fr/>

`Graphviz`: <http://www.research.att.com/sw/tools/graphviz/>

See Also

[Rprof](#), [summaryRprof](#), [flatProfile](#), [readProfileData](#), [plotProfileCallGraph](#), [profileCallGraph2Dot](#)

Examples

```
## Not run:
## Rprof() is not available on all platforms
Rprof(tmp <- tempfile())
example(glm)
Rprof()
printProfileCallGraph(readProfileData(tmp))
unlink(tmp)

## If you have graphviz and cgprof installed on a UNIX-like system
## then in R do:

Rprof(tmp <- tempfile())
example(glm)
Rprof()
printProfileCallGraph(readProfileData(tmp), "foo.graph")
```

```

## and then in a shell do (to use the interactive dotty):

cgprof -TX foo.graph

## or (to create a postscript version and view with gv):

cgprof -Tps foo.graph > foo.ps
gv foo.ps
## End(Not run)

```

```
profileCallGraph2Dot
```

Write Call Graph for Rprof Profile Data to Graphviz Dot File

Description

Prints a Graphviz .dot file representation of the call graph for profile data produced by Rprof.

Usage

```

profileCallGraph2Dot(pd, score = c("total", "self"),
                    transfer = function(x) x, colorMap = NULL,
                    filename = "Rprof.dot", landscape = FALSE,
                    mergeCycles = FALSE, edgesColored = TRUE,
                    rankdir = "LR", center = FALSE, size)

```

Arguments

pd	profile data as returned by readProfileData.
score	character string specifying whether to use total time or self time for coloring nodes/edges; no color used if missing.
transfer	function; maps score values in unit interval to unit interval
colorMap	character vector of color specifications as produced by rainbow; transfer of score is mapped to color
filename	name of .dot file
landscape	logical; whether to add the rotate=90 option to the .dot file
mergeCycles	logical; whether to merge each cycle of recursion into a single node
edgesColored	logical; whether to color edges
rankdir	character; value to use for the rankdir= option to specify the direction that the plot is laid out using the dot layout; must be either "TB" for Top-to-Bottom or "LR" for Left-to-Right. The default value is "LR".
center	logical; whether to add the center=1 option to the .dot file.
size	character; string to add as size= option in the .dot file.

Details

Writes the call graph as a Graphviz `.dot` file. Color is used to encode the fraction total or self time spent in each function or call. The scores used correspond to the values in the printed representation produced by `printProfileCallGraph`. For now, see the `gprof` manual for further details. The color encoding for a score `s` and a color map `m` is `ceiling(length(m) * transfer(s))`

Value

Used for side effect.

Note

Because of lazy evaluation, nested calls like `f(g(x))` appear in the profile graph as `f` or one of its callees calling `g`.

Author(s)

Luke Tierney

References

User manual for `gprof`, the GNU profiler.

Graphviz: <http://www.research.att.com/sw/tools/graphviz/>

See Also

[Rprof](#), [summaryRprof](#), [readProfileData](#), [flatProfile](#), [plotProfileCallGraph](#), [printProfileCallGraph](#)

Examples

```
## Not run:
## Rprof() is not available on all platforms
Rprof(tmp <- tempfile())
example(glm)
Rprof()
profileCallGraph2Dot(readProfileData(tmp), filename = tmp)
file.show(tmp)
unlink(tmp)

## If you have graphviz installed on a UNIX-like system
## then in R do:

Rprof(tmp <- tempfile())
example(glm)
Rprof()
profileCallGraph2Dot(readProfileData(tmp), score = "total",
                     filename = "foo.dot")
```

```
## and then in a shell do (to use the interactive dotty):  
  
dotty foo.dot  
  
## or (to create a postscript version and view with gv)  
  
dot -Tps foo.dot > foo.ps  
gv foo.ps  
## End(Not run)
```

readProfileData *Read Rprof Profile Data*

Description

Reads in Rprof profile data for further processing.

Usage

```
readProfileData(filename = "Rprof.out")
```

Arguments

filename Name of a file produced by Rprof()

Details

readProfileData reads the data in the file produced by Rprof into a data structure for processing by other functions. The details of the structure are subject to change.

Value

R representation of Rprof data,

Author(s)

Luke Tierney

See Also

[Rprof](#), [summaryRprof](#), [flatProfile](#), [plotProfileCallGraph](#), [printProfileCallGraph](#), [profileCallGraph2Dot](#)

Examples

```
## Not run:  
  ## Rprof() is not available on all platforms  
  Rprof(tmp <- tempfile())  
  example(glm)  
  Rprof()  
  flatProfile(readProfileData(tmp))  
  flatProfile(readProfileData(tmp), FALSE)  
  unlink(tmp)  
  
## End(Not run)
```

Index

*Topic **programming**

- flatProfile, 1
- plotProfileCallGraph, 2
- printProfileCallGraph, 4
- profileCallGraph2Dot, 5
- readProfileData, 7

*Topic **utilities**

- flatProfile, 1
- plotProfileCallGraph, 2
- printProfileCallGraph, 4
- profileCallGraph2Dot, 5
- readProfileData, 7

flatProfile, 1, 3, 5, 7, 8

plotProfileCallGraph, 2, 2, 5, 7, 8
printProfileCallGraph, 2, 3, 4, 7, 8
profileCallGraph2Dot, 2, 3, 5, 5, 8

readProfileData, 2, 3, 5, 7, 7
Rprof, 2, 3, 5, 7, 8

summaryRprof, 2, 3, 5, 7, 8