R PROFILING AND OPTIMIZATION

GÜNTHER SAWITZKI

Contents

1. Example data

Eingabe
```
nORE <- 10000
xORE <- runif(nORE)
errORE <- rnorm(nORE)
yORE <- 2+ 3 * xORE + errORE
```

Eingabe
```
profinterval <- 0.001
simruns <- 100
Rprof(filename="Rprofsr01.out", interval = profinterval)
for (i in 1:simruns) xxx<- summary(lm(yORE~xORE))
Rprof(NULL)
```

2. A better grip on profile information

The basic information provided by all profilers in R is a protocol of sampled stacks. For each recorded event, the protcol records one line with a text string showing the sampled stack (in reverse order: most recent first). The stack lines may be preceded by header lines with event specific information. The protocol may be interspersed with control information, such as information about the timing interval used. Examples of the protocol format used by the commen profilersin in **??** on page **??**.

We know that the structural information, static information as well as dynamic information, can be represented with the help of a graph. For a static analysis, the graph representation may be thr first choice. For a dynamic analysis, the stackinformaion is our first information. A stack is a connected path in the program

Date: Typeset: June 10, 2013.
Key words and phrases. R programming, profiling, optimization, R program language.
Private Version

.

graph. If we start with nodes and edges, we loose information which is readily available in record of stacks.

As we know that weare working with stacks, we know that they have their peculiarities. Stacks tend to grow and shrink. Subsequent events will have extensions and shrinkages of stacks (if the recording is on a fine scale), or stack sharing common stumbs (if the recording is on a coarser scale).

The graph is a second instance that is (re)constructed from the stack recording.

Here is the way we represent the profilie information:

The profile log file is sanitized:

- Control lines are extracted and recorded in a separate list.
- Head parts, if present, ere extracted and recorded in a matrix that is kept line-aligned with the remainder
- Line content is standardized, for example by removin stray quotation marks ets

After this, the sanitized lines are encoded as a vector of stacks, and references to this.

If necessary, thes steps are done by chunks to reduce memory load.

From the vector of stacks, a vector of nodes (or rather node names) is derived.

The stacks are now encoded by refrences to the nodes table. For convenience, we keep the (sanitized) textual representation of the stacks.

So far, texts are in reverse order. For each stack, we record the trailing leaf, and then we reverse order. The top of stack is now on first position.

Several statistics can be accumulated easily as a side effect.

Conceptually, the data structure consist of three tables (the implementtion may differ, and is subject to change).

The pofiles table is the representation of the input file. Control lines are are collected in a special table. With the control lines removed, the rest is a table, one row per input line. The body of the line, the stack, is encoded as a reference to a stacks table (obligatory) and header information (optional).

The stacks table contains the collected stacks, each stack enoded as a list of references to the node table. This is obligatory. This list is kept in reverse order (root at position 1). A source line representing the stack information may be kept (optional).

The nodes table keeps the names af the nodes.

To illustrate our data structure, we use *Rprofsr01.out* as provided in section **??** on page **??**

This is a temporary hack to get a most recent private recent version of *library(sprof)*.

Eingabe
```
source('~/projects/rforge/sintro/pkg/sprof/R/readProf.R', chdir = TRUE)
source('~/projects/rforge/sintro/pkg/sprof/R/rrle.R', chdir = TRUE)
source('~/projects/rforge/sintro/pkg/sprof/R/sampleRprof.R', chdir = TRUE)
source('~/projects/rforge/sintro/pkg/sprof/R/summary.sprof.R', chdir = TRUE)
source('~/projects/rforge/sintro/pkg/sprof/R/print.sprof.R', chdir = TRUE)
source('~/projects/rforge/sintro/pkg/sprof/R/plot.sprof.R', chdir = TRUE)
# file.edit('~/projects/rforge/sintro/pkg/sprof/R/summary_prof.R')
```

Eingabe
```
rpo <- readProf("Rprofsr01.out")
str_prof(rpo)
```

Ausgabe
```
First line: sample.interval=1000
493 Sampling intervals  at 1000 micros
61 nodes in 54 stacks
41 Terminals
1 Roots:
Sweave
    54

 rpo  Structure: List of 10
 $ firstline : chr "sample.interval=1000"
 $ ctllines  : chr "sample.interval=1000"
 $ ctllinenr : num 1
 $ nodes     : chr [1:61] "!" ".deparseOpts" ".getXlevels" "[" ...
 $ stacksrenc:List of 54
 $ stacks    :'data.frame':        54 obs. of  6 variables:
 $ data      : int [1:493] 1 2 3 1 1 2 3 1 1 2 ...
 $ mem       : NULL
 $ malloc    : NULL
 $ timesRLE  :List of 2
  ..- attr(*, "class")= chr "rle"
 - attr(*, "class")= chr [1:2] "sprof" "list"

 stacks Structure: 'data.frame':        54 obs. of  6 variables:
 $ shortname     : Factor w/ 54 levels "S<Ac","S<etttdweesl",..: 30 21 37 45 36 ...
 $ refcount      : int  109 36 82 6 62 ...
 $ stacklength   : int  14 23 15 18 15 ...
 $ stackheadnodes: int  49 49 49 49 49 ...
 $ stackleafnodes: int  29 12 37 15 13 ...
 $ stacks        : Factor w/ 54 levels "! [.data.frame [ na.omit.data.frame na.omit model.frame.default
```

2.1. **Summary.**

Eingabe
```
 summary_nodes(rpo)
```

Ausgabe
```
                      shortname  root  leaf
!                             ! FALSE  TRUE
.deparseOpts               .dpO FALSE  TRUE
.getXlevels                .gtX FALSE  TRUE
[                             [ FALSE  TRUE
[.data.frame               [.d. FALSE  TRUE
[[                           [[ FALSE FALSE
[[.data.frame              [[.. FALSE  TRUE
%in%                       %in% FALSE  TRUE
<Anonymous>                <An> FALSE FALSE
$                             $ FALSE  TRUE
anyDuplicated              anyD FALSE FALSE
anyDuplicated.default      anD. FALSE  TRUE
as.character               as.c FALSE  TRUE
as.list                    as.l FALSE  TRUE
as.list.data.frame         a... FALSE  TRUE
cat                         cat FALSE  TRUE
chol2inv                   chl2 FALSE  TRUE
```

```
coef                        coef FALSE FALSE
coef.default                cf.d FALSE  TRUE
colnames                    clnm FALSE  TRUE
deparse                     dprs FALSE  TRUE
doTryCatch                  dTrC FALSE FALSE
eval                        eval FALSE FALSE
evalFunc                    evlF FALSE FALSE
FUN                          FUN FALSE  TRUE
lapply                      lppl FALSE  TRUE
list                        list FALSE  TRUE
lm                            lm FALSE  TRUE
lm.fit                      lm.f FALSE  TRUE
match                       mtch FALSE  TRUE
mean                        mean FALSE FALSE
mean.default                mn.d FALSE  TRUE
mode                        mode FALSE  TRUE
model.frame                mdl.f FALSE  TRUE
model.frame.default        mdl.f. FALSE  TRUE
model.matrix               mdl.m FALSE FALSE
model.matrix.default       mdl.m. FALSE  TRUE
model.response             mdl.r FALSE  TRUE
na.omit                     n.mt FALSE  TRUE
na.omit.data.frame          n... FALSE  TRUE
NCOL                        NCOL FALSE FALSE
paste                       past FALSE  TRUE
rep.int                     rp.n FALSE  TRUE
sapply                      sppl FALSE FALSE
simplify2array              smp2 FALSE FALSE
structure                   strc FALSE  TRUE
summary                     smmr FALSE FALSE
summary.lm                  smm. FALSE  TRUE
Sweave                      Swev  TRUE FALSE
sys.call                    sys. FALSE  TRUE
terms                       trms FALSE FALSE
terms.formula               trm. FALSE  TRUE
try                          try FALSE FALSE
tryCatch                    tryC FALSE FALSE
tryCatchList                trCL FALSE FALSE
tryCatchOne                 trCO FALSE FALSE
unique                      uniq FALSE  TRUE
unique.default              unq. FALSE  TRUE
unlist                      unls FALSE  TRUE
vapply                      vppl FALSE  TRUE
withVisible                 wthV FALSE FALSE
```

Eingabe

```
 summary_stacks(rpo)
```

Ausgabe

```
   len refcount root leafs
1   14      109   49    29
2   23       36   49    12
3   15       82   49    37
4   18        6   49    15
```

```
5   15       62   49    13
6   15       29   49    46
7   15        5   49    10
8   13       22   49    48
9   21       25   49     5
10  20        2   49    15
11  16        1   49    30
12  18        8   49    39
13  23        2   49     7
14  21        1   49    26
15  13       20   49    28
16  17        1   49    15
17  23        2   49     8
18  19       20   49    40
19  20        1   49    50
20  22        2   49     1
21  20        2   49    33
22  20        1   49     2
23  14        1   49    38
24  20        8   49    27
25  18       11   49    46
26  15        2   49    43
27  17        1   49    59
28  14        1   49     3
29  15        3   49    32
30  19        1   49    14
31  18        1   49     8
32  16        1   49    26
33  16        1   49    34
34  23        1   49     2
35  15        1   49    19
36  14        1   49     4
37  17        3   49    35
38  24        1   49    50
39  17        1   49    57
40  21        1   49    58
41  15        1   49    20
42  21        1   49    30
43  14        1   49    10
44  17        1   49     5
45  14        1   49    17
46  19        1   49    25
47  18        1   49     2
48  19        1   49    52
49  22        1   49     8
50  16        1   49    60
51  21        1   49    42
52  22        1   49    21
53  20        1   49    59
54   3        1   49    16
```

Eingabe

```
summary_profiles(rpo)
```

Ausgabe

```
$id
[1] "Profile Summary Mon Jun 10 00:18:02 2013"

$len
[1] 493

$uniquestacks
[1] 54

$runs
[1] 366
```

The miscsummary*summary()* rfun]summary@`summary`|textit mtehod for *sprof* objects concatenates thes three functions.

2.2. **Print.**

Eingabe

```
print_nodes(rpo)
```

Ausgabe

```
                         shortname  root  leaf
!                                ! FALSE  TRUE
.deparseOpts                  .dpO FALSE  TRUE
.getXlevels                   .gtX FALSE  TRUE
[                                [ FALSE  TRUE
[.data.frame                  [.d. FALSE  TRUE
[[                              [[ FALSE FALSE
[[.data.frame                 [[.. FALSE  TRUE
%in%                          %in% FALSE  TRUE
<Anonymous>                   <An> FALSE FALSE
$                                $ FALSE  TRUE
anyDuplicated                 anyD FALSE FALSE
anyDuplicated.default         anD. FALSE  TRUE
as.character                  as.c FALSE  TRUE
as.list                       as.l FALSE  TRUE
as.list.data.frame            a... FALSE  TRUE
cat                            cat FALSE  TRUE
chol2inv                      chl2 FALSE  TRUE
coef                          coef FALSE FALSE
coef.default                  cf.d FALSE  TRUE
colnames                      clnm FALSE  TRUE
deparse                       dprs FALSE  TRUE
doTryCatch                    dTrC FALSE FALSE
eval                          eval FALSE FALSE
evalFunc                      evlF FALSE FALSE
FUN                            FUN FALSE  TRUE
lapply                        lppl FALSE  TRUE
list                          list FALSE  TRUE
lm                              lm FALSE  TRUE
lm.fit                        lm.f FALSE  TRUE
match                         mtch FALSE  TRUE
mean                          mean FALSE FALSE
mean.default                  mn.d FALSE  TRUE
mode                          mode FALSE  TRUE
model.frame                  mdl.f FALSE  TRUE
model.frame.default      mdl.f. FALSE  TRUE
model.matrix              mdl.m FALSE FALSE
model.matrix.default     mdl.m. FALSE  TRUE
model.response            mdl.r FALSE  TRUE
na.omit                    n.mt FALSE  TRUE
na.omit.data.frame         n... FALSE  TRUE
NCOL                       NCOL FALSE FALSE
paste                      past FALSE  TRUE
rep.int                    rp.n FALSE  TRUE
sapply                     sppl FALSE FALSE
simplify2array             smp2 FALSE FALSE
structure                  strc FALSE  TRUE
summary                    smmr FALSE FALSE
summary.lm                 smm. FALSE  TRUE
Sweave                     Swev  TRUE FALSE
sys.call                   sys. FALSE  TRUE
terms                      trms FALSE FALSE
terms.formula              trm. FALSE  TRUE
try                         try FALSE FALSE
tryCatch                   tryC FALSE FALSE
tryCatchList               trCL FALSE FALSE
tryCatchOne                trCO FALSE FALSE
unique                     uniq FALSE  TRUE
unique.default             unq. FALSE  TRUE
unlist                     unls FALSE  TRUE
vapply                     vppl FALSE  TRUE
withVisible                wthV FALSE FALSE
```

Eingabe

```
print_stacks(rpo)
```

Ausgabe

```
   len refcount root leafs
1   14      109   49    29
2   23       36   49    12
3   15       82   49    37
4   18        6   49    15
5   15       62   49    13
6   15       29   49    46
7   15        5   49    10
8   13       22   49    48
9   21       25   49     5
10  20        2   49    15
11  16        1   49    30
12  18        8   49    39
13  23        2   49     7
14  21        1   49    26
15  13       20   49    28
16  17        1   49    15
17  23        2   49     8
18  19       20   49    40
19  20        1   49    50
20  22        2   49     1
21  20        2   49    33
```

```
22  20        1   49     2
23  14        1   49    38
24  20        8   49    27
25  18       11   49    46
26  15        2   49    43
27  17        1   49    59
28  14        1   49     3
29  15        3   49    32
30  19        1   49    14
31  18        1   49     8
32  16        1   49    26
33  16        1   49    34
34  23        1   49     2
35  15        1   49    19
36  14        1   49     4
37  17        3   49    35
38  24        1   49    50
39  17        1   49    57
40  21        1   49    58
41  15        1   49    20
42  21        1   49    30
43  14        1   49    10
44  17        1   49     5
45  14        1   49    17
46  19        1   49    25
47  18        1   49     2
48  19        1   49    52
49  22        1   49     8
50  16        1   49    60
51  21        1   49    42
52  22        1   49    21
53  20        1   49    59
54   3        1   49    16
```

Eingabe

```
 print_profiles(rpo)
```

Ausgabe

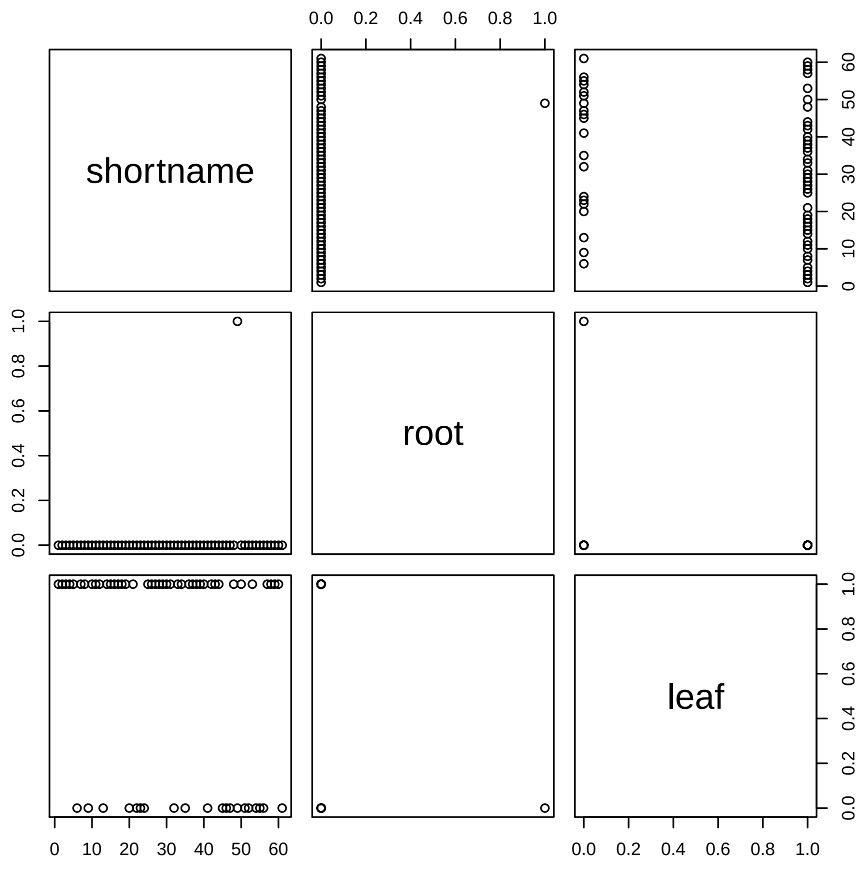
```
$id
[1] "Profile Summary Mon Jun 10 00:18:02 2013"

$len
[1] 493

$uniquestacks
[1] 54

$runs
[1] 366
```

The miscprint*`print()`*rfun]print@`print`|textit method for *`sprof`* objects concatenates these three functions.

2.3. **Plot.**

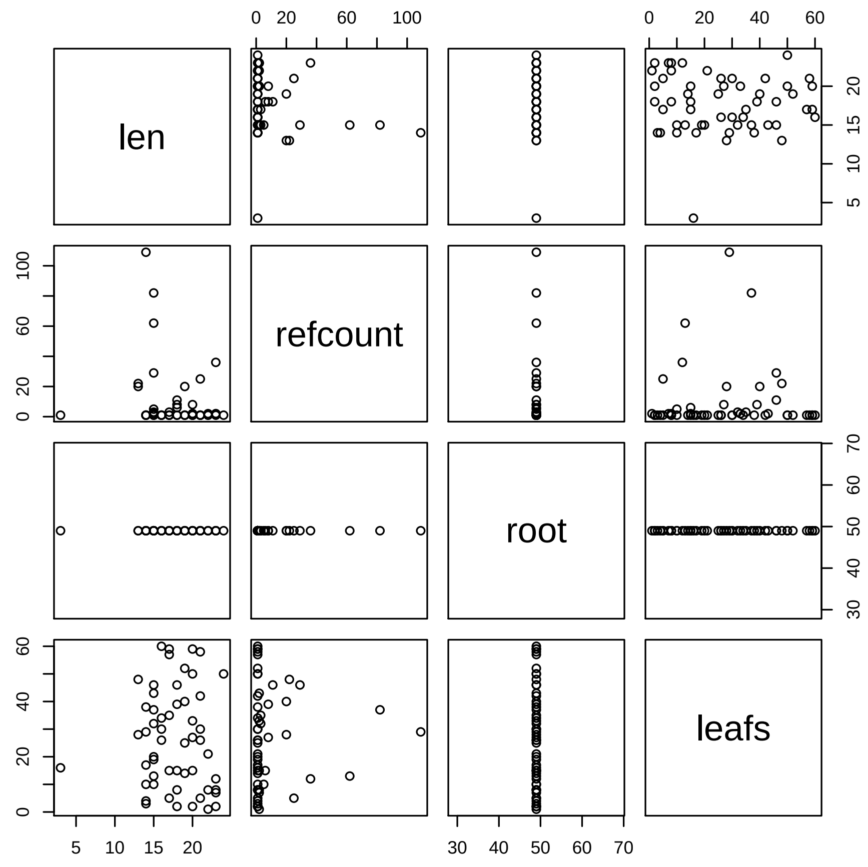
Eingabe

```
plot_nodes(rpo)
```



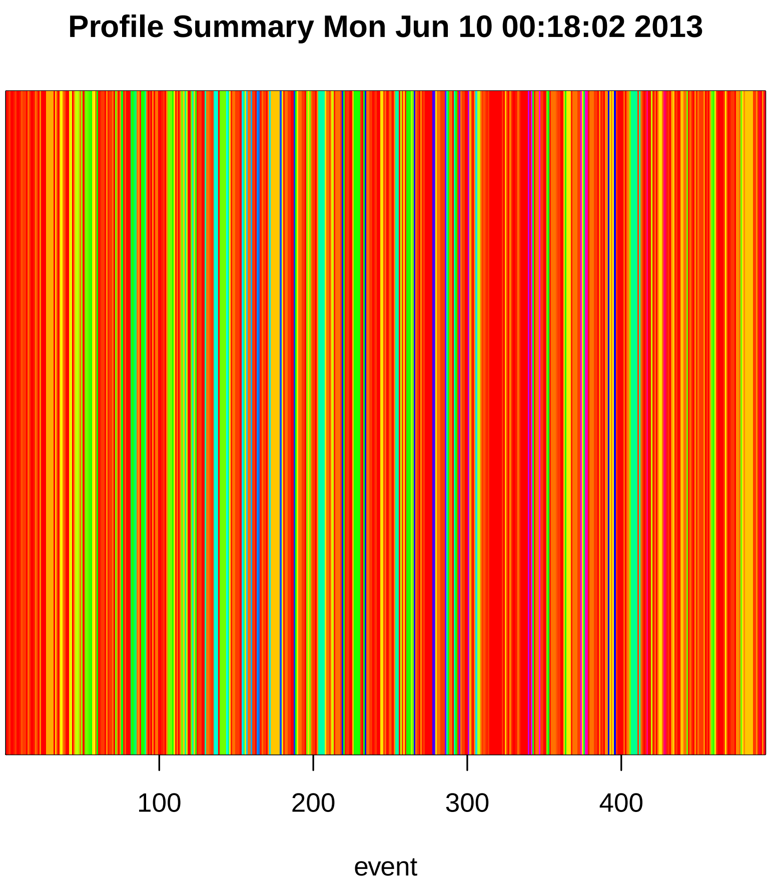
Eingabe

```
plot_stacks(rpo)
```



Eingabe

```
plot_profiles(rpo)
```



The miscplot*`plot()`*rfun]plot@`plot`|textit method for *`sprof`* objects concatenates these three functions.

```
$Source: /u/math/j40/cvsroot/lectures/src/insider/profile/Rnw/profile.Rnw,v $
$Revision: 1.1 $
$Date: 2013/05/20 20:24:04 $
$name:  $
$Author: j40 $
```

Günther Sawitzki
StatLab Heidelberg
Im Neuenheimer Feld 294
D 69120 Heidelberg
E-mail address: `gs@statlab.uni-heidelberg.de`
URL: `http://www.statlab.uni-heidelberg.de/users/gs/`