ALTREP

some new stuff in R

mainly due to: L. Tierney, G. Becker and T. Kalibera

ALTREP

- ALTREP
- ALTernative object REPresentations of vectors
  - some help for really long vectors
  - some help for deferring operations
- Basic strategies:
  - provide a **wrapper** for vectors that can hold attributes, new special symbols
  - defer string computations
  - provide tools for simple sequences (eg 1:n)

ALTREP

- The summary functions **mean, min, max, sum**, and **prod** have been updated for INTSXP and REALSXP vectors, but not yet for CPLXSXP or LGLSXP vectors.
- Basic subsetting operations for INTSXP and REALSXP have also been modified, so calls to **head** and **sample**, among others, do not force allocation.
- Many more functions could be modified in this way (eg do_for)

Wrappers

```
wrapper <- function(x, srt = 0, nna = 0) .Internal(wrap_meta(x, srt, nna))
```

```
> x <- wrapper(c(1, 2, 3))
> y <- x
> attr(y, "foo") <- "stuff"
```

result in duplicating and modifying the wrapper object value of `y`, but the payload is not duplicated and is shared by the values of `x` and `y`:

```
> .Internal(inspect(x))
@3299280 14 REALSXP g0c0 [NAM(2)]  wrapper [srt=0,no_na=0]
  @3288b98 14 REALSXP g0c3 [NAM(2)] (len=3, tl=0) 1,2,3
> .Internal(inspect(y))
@3298c60 14 REALSXP g0c0 [NAM(1),ATT]  wrapper [srt=0,no_na=0]
  @3288b98 14 REALSXP g0c3 [NAM(2)] (len=3, tl=0) 1,2,3
ATTRIB:
  @333c6e8 02 LISTSXP g0c0 []
    TAG: @2189008 01 SYMSXP g0c0 [MARK] "foo"
    @3289948 16 STRSXP g0c1 [NAM(2)] (len=1, tl=0)
      @32899f0 09 CHARSXP g0c1 [gp=0x60] [ASCII] [cached] "stuff"
```

Wrappers

- wrappers also provide a location to put some descriptions of the data in the payload
- a description of whether the data are sorted, and if so whether increasing or decreasing
- a description of whether the data contain any NA values

Deferred Coercion

- converting an integer or real to a string is an expensive operation and there are many cases where the result is not used or partially used
  - eg row labels in regression matrices
- Initially the resulting object contains only a reference to the original numeric object
  - along with the scipen option setting in effect
- If elements are requested individually these are converted on request and saved .
- If the data pointer is requested, then the full vector is converted and the reference to the original data is dropped.

Deferred Coercion

```
> x <- 1:1000
> y <- as.character(x)
> .Internal(inspect(y))
@2802830 16 STRSXP g0c0 [NAM(1)]   <deferred string conversion>
  @25114c0 13 INTSXP g0c0 [NAM(2)]  1 : 1000 (compact)
> head(x)
[1] 1 2 3 4 5 6
> y[1] <- "a"
> .Internal(inspect(y))
@2802830 16 STRSXP g0c0 [NAM(1)]   <expanded string conversion>
  @331a690 16 STRSXP g0c7 [] (len=1000, tl=0)
    @1d2d388 09 CHARSXP g0c1 [MARK,gp=0x61] [ASCII] [cached] "a"
    @2696ac8 09 CHARSXP g0c1 [MARK,gp=0x60] [ASCII] [cached] "2"
    @1d16038 09 CHARSXP g0c1 [MARK,gp=0x60] [ASCII] [cached] "3"
    @26a36e8 09 CHARSXP g0c1 [MARK,gp=0x60] [ASCII] [cached] "4"
    @2a57148 09 CHARSXP g0c1 [gp=0x60] [ASCII] [cached] "5"
    ...
```

R 3.4/3.5

```
> n <- 10000000
> x <- rnorm(n)
> y <- rnorm(n)
> system.time(lm(y ~ x))
   user  system elapsed
 17.927   0.982  18.911
> system.time(lm(y ~ x))
   user  system elapsed
  9.225   0.703   9.929
```

R 3.6

```
> n <- 10000000
> x <- rnorm(n)
> y <- rnorm(n)
> system.time(lm(y ~ x))
   user  system elapsed
  1.989   0.601   2.590
> system.time(lm(y ~ x))
   user  system elapsed
  1.886   0.610   2.496
```

Compact Integer Vectors

- Vectors n1:n2 , seq_along(n) and seq_len(m): represented compactly in terms of their start and end values.
- In R 3.4.4 (and older):
- ## > x <- 1:1e10
- ## Error: cannot allocate vector of size 74.5 Gb
- where in R 3.5.0++ and the ALTREP branch it works (very fast):
- x <- 1:1e10

Compact Integer Vectors

- The sequence functions produce long vectors (or compact versions) which will be of type "double", as long as the limits don't exceed the maximal exact integer 2^53
- `> typeof(1:1e20)`
- **`Error in 1:1e+20 : result would be too long a vector`**
- `> typeof(1:1e15)`
- `[1] "double"`

An Example Package

- Luke Tierney has put an example package on github that shows how to use ALTREP
- ALTREP includes sample classes for memory mapped integer and real vectors.
  - The file can be opened for reading and writing or in read-only mode.
  - When used by ALTREP-aware code these will not result in allocating memory for holding all the data.
  - Using non-aware functions may result in attempts to allocate large objects.
  - The class provides an option for signaling an error when the raw data pointer is requested.
- A variant is also available as a small experimental package simplemmap. https://github.com/ALTREP-examples/Rpkg-simplemmap

Other Resources:

- Luke Tierney (Dec. 2017): http://homepage.stat.uiowa.edu/~luke/talks/nzsa-2017.pdf
- Martin Maechler

  ftp://stat.ethz.ch/Teaching/maechler/R/eRum_2018_ProgR-ALTREP.html
- Documentation:

  https://svn.r-project.org/R/branches/ALTREP/ALTREP.htm