

Package: peakRAM (via r-universe)

August 26, 2024

Title Monitor the Total and Peak RAM Used by an Expression or Function

Version 1.0.3

URL <http://github.com/tpq/peakRAM>

BugReports <http://github.com/tpq/peakRAM/issues>

Description When working with big data sets, RAM conservation is critically important. However, it is not always enough to just monitor the size of the objects created. So-called ``copy-on-modify" behavior, characteristic of R, means that some expressions or functions may require an unexpectedly large amount of RAM overhead. For example, replacing a single value in a matrix duplicates that matrix in the back-end, making this task require twice as much RAM as that used by the matrix itself. This package makes it easy to monitor the total and peak RAM used so that developers can quickly identify and eliminate RAM hungry code.

License GPL-2

LazyData true

VignetteBuilder knitr

RoxygenNote 7.1.1

Depends R (>= 3.2.2)

Suggests knitr, rmarkdown

Repository <https://tpq.r-universe.dev>

RemoteUrl <https://github.com/tpq/peakram>

RemoteRef HEAD

RemoteSha c990f39099edbbfb526c34dfb8df5aeba25a7658

Contents

peakRAM	2
Index	3

`peakRAM`*Calculate Peak RAM Used*

Description

This function monitors the total and peak RAM used by any number of R expressions or functions.

Usage

```
peakRAM(...)
```

Arguments

... R expressions or function calls. Anonymous functions (e.g., `function() 1:1e7`) also accepted.

Details

When working with big datasets, RAM conservation is critically important. However, it is not always enough to just monitor the size of the objects created. So-called "copy-on-modify" behavior, characteristic of R, means that some expressions or functions may require an unexpectedly large amount of RAM overhead. For example, replacing a single value in a matrix (e.g., with `'[<-'`) duplicates that matrix in the backend, making this task require twice as much RAM as that used by the matrix itself. The `peakRAM` package makes it easy to monitor the total and peak RAM used so that developers can quickly identify and eliminate RAM hungry code.

Value

A data.frame tallying total and peak RAM use.

Examples

```
peakRAM(function() 1:1e7,  
         1:1e7,  
         1:1e7 + 1:1e7,  
         1:1e7 * 2)
```

Index

peakRAM, [2](#)