Package: amatrix.metal (via r-universe)

July 10, 2026

Type Package

Title Metal Sparse Backend for 'amatrix' on macOS

Version 0.1.0

Description Experimental sparse-first GPU backend for the 'amatrix' package using Apple 'Metal'. Provides a direct Objective-C++ bridge for sparse matrix times dense matrix products on macOS; on other platforms (or without the 'Metal' frameworks) it builds a mock bridge and reports itself unavailable, so installation is safe anywhere.

License MIT + file LICENSE

URL https://github.com/bbuchsbaum/amatrix

BugReports https://github.com/bbuchsbaum/amatrix/issues

Encoding UTF-8

Depends R (>= 4.3)

Imports amatrix, methods

Suggests testthat (>= 3.0.0)

OS_type unix

SystemRequirements Apple Metal and Foundation frameworks (macOS only); Xcode Command Line Tools

NeedsCompilation yes

Config/testthat/edition 3

Repository https://bbuchsbaum.r-universe.dev

Date/Publication 2026-07-10 06:16:28 UTC

RemoteUrl https://github.com/bbuchsbaum/amatrix

RemoteRef HEAD

RemoteSha 219f07843470aa27a4fa6389e9927f968715015f

RemoteSubdir backends/amatrix.metal

Contents

`amatrix_metal_profile` *Profile Metal backend bridge timing*

Description

Enable, reset, and read lightweight native timing counters for the Metal backend bridge. Counters cover sparse and dense uploads, sparse product command submission and synchronization, deferred pending waits, and host materialization. Profiling is disabled by default to avoid overhead in normal use.

Usage

```
amatrix_metal_profile(reset = FALSE)

amatrix_metal_profile_enable(enabled = TRUE, reset = FALSE)

amatrix_metal_profile_reset()
```

Arguments

`reset`	Logical scalar. For `amatrix_metal_profile()`, reset counters after reading them. For `amatrix_metal_profile_enable()`, reset counters before changing the enabled state.
`enabled`	Logical scalar. Whether native Metal bridge profiling should collect timing counters.

Value

`amatrix_metal_profile()` returns a named numeric vector of timing counters in milliseconds plus call counts. The enable and reset helpers return invisibly.

Examples

```
## Not run:
amatrix_metal_enable_probe(register = TRUE)
amatrix_metal_profile_enable(TRUE, reset = TRUE)

# Run Metal-backed operations here.

amatrix_metal_profile()
amatrix_metal_profile_enable(FALSE)

## End(Not run)
```

Index