

Package ‘sparsepp’

September 22, 2018

Type Package

Title 'Rcpp' Interface to 'sparsepp'

Version 1.22

Date 2018-09-19

Description Provides interface to 'sparsepp' - fast, memory efficient hash map.

It is derived from Google's excellent 'sparsehash' implementation.

We believe 'sparsepp' provides an unparalleled combination of performance and memory usage, and will outperform your compiler's `unordered_map` on both counts.

Only Google's `dense_hash_map` is consistently faster, at the cost of much greater memory usage (especially when the final size of the map is not known in advance).

License BSD_3_clause + file LICENSE

Encoding UTF-8

URL <https://github.com/greg7mdp/sparsepp>,

<https://github.com/dselivanov/r-sparsepp>

BugReports <https://github.com/dselivanov/r-sparsepp/issues>

Suggests Rcpp

RoxygenNote 6.0.1

NeedsCompilation no

Author Gregory Popovitch [aut, cph],

Google Inc [aut, cph],

Dmitriy Selivanov [cre]

Maintainer Dmitriy Selivanov <selivanov.dmitriy@gmail.com>

Repository CRAN

Date/Publication 2018-09-22 16:30:03 UTC

R topics documented:

sparsepp-package 2

Index 4

sparsepp-package *sparsepp*

Description

sparsepp provides bindings to the [sparsepp](#) - fast, memory efficient hash map for C++. sparsepp is an open source C++ library derived from Google's excellent sparsehash implementation, but considerably outperform it - <https://github.com/greg7mdp/sparsepp/blob/master/bench.md>. It aims to achieve the following objectives:

- A drop-in alternative for `unordered_map` and `unordered_set`.
- Extremely low memory usage (typically about one byte overhead per entry).
- Very efficient, typically faster than your compiler's `unordered_map/set` or Boost's.
- C++11 support (if supported by compiler).
- Single header implementation - just copy `sparsepp.h` to your project and include it.
- Tested on Windows (vs2010-2015, g++), linux (g++, clang++) and MacOS (clang++).

Author(s)

Maintainer: Dmitriy Selivanov <selivanov.dmitriy@gmail.com>

Authors:

- Gregory Popovitch <greg7mdp@gmail.com> [copyright holder]
- Google Inc [copyright holder]

See Also

Useful links:

- <https://github.com/greg7mdp/sparsepp>
- <https://github.com/dselivanov/r-sparsepp>
- Report bugs at <https://github.com/dselivanov/r-sparsepp/issues>

Examples

```
## Not run:
library(Rcpp)
code = "
// [[Rcpp::plugins(cpp11)]]
#include <Rcpp.h>
using namespace std;
using namespace Rcpp;
// drop-in replacement for unordered_map
// #include <unordered_map>
#include <sparsepp/spp.h>
// [[Rcpp::depends(sparsepp)]]
```

```
using spp::sparse_hash_map;
// @export
// [[Rcpp::export]]
IntegerVector word_count(CharacterVector v) {
  //unordered_map<string, int> smap;
  sparse_hash_map<string, int> smap;
  for(auto x: v) {
    smap[as<string>(x)] ++;
  }
  IntegerVector res(smap.size());
  int i = 0;
  for(auto s:smap) {
    res[i]=s.second;
    i++;
  }
  return(res);
}"
f = tempfile(), fileext = ".cpp")
writeLines(code, f)
sourceCpp(f)
unlink(f)
word_count(sample(letters, 100, T))

## End(Not run)
```

Index

sparsepp (sparsepp-package), [2](#)
sparsepp-package, [2](#)