frovedis::sliced_colmajor_matrix_local<T>

NAME

`frovedis::sliced_colmajor_matrix_local<T>` - a data structure containing the slicing information of a two-dimensional `frovedis::colmajor_matrix_local<T>`

SYNOPSIS

```
#include <frovedis/matrix/sliced_matrix.hpp>
```

Constructors

sliced_colmajor_matrix_local ()
sliced_colmajor_matrix_local (const `colmajor_matrix_local<T>`& m)
sliced_colmajor_matrix_local (const `std::vector<T>`& v)

Public Member Functions

bool is_valid () const
void debug_print () const

Public Data Members

T* data
size_t ldm
size_t sliced_num_row
size_t sliced_num_col

DESCRIPTION

In order to perform matrix operations on sub-matrices instead of entire physical matrix, frovedis provides some sliced data structures. `sliced_colmajor_matrix_local<T>` is one of them. It is actually not a real matrix, rather it only contains some slicing information of a physical `colmajor_matrix_local<T>`. Thus any changes performed on the sliced matrix, would actually make changes on the physical matrix from which slice was made.

Like `colmajor_matrix_local<T>`, a `sliced_colmajor_matrix_local<T>` is also a template based structure with type **"T"**. This has the below structure:

```
template <class T>
struct sliced_colmajor_matrix_local {
  T* data;               // pointer pointing to the begining of the slice
  size_t ldm;            // leading dimension of the physical matrix
  size_t sliced_num_row; // number of rows in the sliced matrix
  size_t sliced_num_col; // number of columns in the sliced matrix
};
```

E.g., if a physical `colmajor_matrix_local<T>` M has dimensions 4x4 and slice is needed from 2nd row and 2nd column ([1,1]) till 3rd row and 3rd column ([2,2]), then "data" will hold the address of M[1][1] (data -> &M[1][1]),
"ldm" would be 4 (leading dimension of the matrix M, i.e., number of rows),
From 2nd row till 3rd row, number of rows to be sliced is 2, thus "sliced_num_row" would be 2.
From 2nd column till 3rd column, number of columns to be sliced is 2, thus "sliced_num_col" would be 2.

Such matrices are very useful in operations of external libraries like blas, lapack etc.

Constructor Documentation

sliced_colmajor_matrix_local ()

This is the default constructor which creates an empty sliced matrix with num_row = num_col = 0 and "data" points to NULL. In general of no use, unless it is needed to manipulate the slice information explicitly.

sliced_colmajor_matrix_local (const `colmajor_matrix_local<T>`& m)

This is a special constructor for implicit conversion. This constructor treats an entire physical matrix as a sliced matrix. Thus the created `sliced_colmajor_matrix_local<T>` would have the same dimensions as with the input `colmajor_matrix_local<T>` and "data" pointing to the base address of the input `colmajor_matrix_local<T>`.

sliced_colmajor_matrix_local (const `std::vector<T>`& v)

This is a special constructor for implicit conversion. This constructor treats an entire physical vector as a sliced matrix. Thus the created `sliced_colmajor_matrix_local<T>` would have "sliced_num_row" equals to the length of the input `std::vector<T>`, "sliced_num_col" equals to 1 and "data" pointing to the base address of the input vector.

Public Member Function Documentation

bool is_valid () const

This function returns true, if the caller object is a valid sliced matrix, else it returns false.
Kindly note that an empty sliced matrix is also an invalid sliced matrix, since no valid operation can be performed on its data pointing to NULL.

void debug_print () const

It prints the contents of the sliced part of the original (physical) `colmajor_matrix_local<T>` on the user standard output terminal.

Public Data Member Documentation

data

A pointer of type "T" pointing to the starting location of a physical `colmajor_matrix_local<T>` from which slice has been made.

ldm

A size_t attribute to contain the leading dimension of the physical matrix from which slice has been made (number of rows in the physical matrix).

sliced_num_row

A size_t attribute to contain the number of rows in the sliced matrix.

sliced_num_col

A size_t attribute to contain the number of columns in the sliced matrix.

Public Global Function Documentation

make_sliced_colmajor_matrix_local (mat, start_ridx, start_cidx, num_row, num_col)

Parameters
mat: An object of either `colmajor_matrix_local<T>` or `sliced_colmajor_matrix_local<T>` type.
start_ridx: A size_t attribute to indicate the start row index for the slice.
start_cidx: A size_t attribute to indicate the start column index for the slice.
num_row: A size_t attribute to indicate the number of rows to be sliced from the starting row index.
num_col: A size_t attribute to indicate the number of columns to be sliced from the starting column index.

Purpose
This function accepts a valid `colmajor_matrix_local<T>` or `sliced_colmajor_matrix_local<T>` with some slicing information like row and column index from which slicing is to be started, and the size of the output sliced matrix, i.e., number of rows and columns to be sliced from the starting location. On receiving the valid inputs, it outputs a `sliced_colmajor_matrix_local<T>` object containing the slicing information, else it throws an exception.

Example:
If a physical `colmajor_matrix_local<T>` "mat" has the dimensions 4x4 and slicing is required from its 2nd row and 2nd column ([1,1]) till 4th row and 4th column ([3,3]), then this function should be called like:

```
auto smat = make_sliced_colmajor_matrix_local(mat,1,1,3,3);
```

Index of the 2nd row is 1, thus start_row_index = 1.
Index of the 2nd column is 1, thus start_col_index = 1.
From 2nd row till 4th row, number of rows to be sliced is 3, thus num_row = 3.
From 2nd column till 4th column, number of columns to be sliced is 3, thus num_col = 3.

```
Input (mat):          Output (smat):
------------          --------------
1 2 3 4               6 7 8
5 6 7 8       =>      7 6 5
8 7 6 5               3 2 1
4 3 2 1
```

Now if we need to slice further this sliced matrix, "smat" from its 2nd row and 2nd column ([1,1]) till 3rd row and 3rd column ([2,2]), then we would call this function like below:

```
auto ssmat = make_sliced_colmajor_matrix_local(smat,1,1,2,2);
```

Index of the 2nd row of smat is 1, thus start_row_index = 1.
Index of the 2nd column of smat is 1, thus start_col_index = 1.
From 2nd row till 3rd row of smat, number of rows to be sliced is 2, thus num_row = 2.
From 2nd column till 3rd column of smat, number of columns to be sliced is 2, thus num_col = 2.

Kindly note that 2nd row of "smat" is actually the 3rd row of the physical matrix "mat", but this function takes care of it internally. Thus you just need to take care of the index of the input sliced matrix, not the actual physical matrix.

```
Input (smat):         Output (ssmat):
-------------         ---------------
6 7 8                 6 5
7 6 5         =>      2 1
3 2 1
```

Return Value
On success, it returns an object of the type `sliced_colmajor_matrix_local<T>`. Otherwise it throws an exception.

SEE ALSO

colmajor_matrix, sliced_colmajor_vector_local