frovedis::ccs_matrix_local<T,I,O>

NAME

`frovedis::ccs_matrix_local<T,I,O>` - A two-dimensional non-distributed sparse matrix with compressed column storage.

SYNOPSIS

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

Constructors

ccs_matrix_local ();
ccs_matrix_local (const `ccs_matrix_local<T,I,O>`& m);
ccs_matrix_local (`ccs_matrix_local<T,I,O>`&& m);
ccs_matrix_local (const `crs_matrix_local<T,I,O>`& m);

Overloaded Operators

`ccs_matrix_local<T,I,O>`& operator= (const `ccs_matrix_local<T,I,O>`& m);
`ccs_matrix_local<T,I,O>`& operator= (`ccs_matrix_local<T,I,O>`&& m);

Public Member Functions

void set_local_num (size_t nrow);
void debug_print ();
`ccs_matrix_local<T,I,O>` transpose ();
`crs_matrix_local<T,I,O>` to_crs();

Public Data Members

`std::vector<T>` val;
`std::vector<I>` idx;
`std::vector<O>` off;
size_t local_num_row;
size_t local_num_col;

DESCRIPTION

A CCS (Compressed Column Storage) matrix is one of the popular sparse matrices with compressed column. It has three major components while storing the non-zero elements, as explained below along with the number of rows and the number of columns in the sparse matrix.

```
val: a vector containing the non-zero elements of the compressed columns
(in column-major order) of the matrix.
idx: a vector containing the row indices for each non-zero elements in "val".
off: a vector containing the column offsets.
```

For example, if we consider the below sparse matrix:

```
1 0 0 0 2 0 0 4
0 0 0 1 2 0 0 3
1 0 0 0 2 0 0 4
0 0 0 1 2 0 0 3
```

then its CCS representation would be:

```
val: {1, 1, 1, 1, 2, 2, 2, 2, 4, 3, 4, 3}
idx: {0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2, 3}
off: {0, 2, 2, 2, 4, 8, 8, 8, 12}
```

column offset starts with 0 and it has n+1 number of elements, where n is the number of columns in the sparse matrix. The difference between i+1th element and ith element in column offset indicates number of non-zero elements present in ith column.

`ccs_matrix_local<T,I,O>` is a two-dimensional template based non-distributed sparse data storage supported by frovedis. The structure of this class is as follows:

```
template <class T, class I=size_t, class O=size_t>
struct ccs_matrix_local {
  std::vector<T> val;     // to contain non-zero elements of type "T"
  std::vector<I> idx;     // to contain row indices of type "I" (default: size_t)
  std::vector<O> off;     // to contain column offsets of type "O" (default: size_t)
  size_t local_num_row;   // number of rows in the sparse matrix
  size_t local_num_col;   // number of columns in the sparse matrix
};
```

This matrix can be loaded from a local crs matrix and also the matrix can be converted back to the local crs matrix. Thus loading/saving interfaces are not provided for local ccs matrix.

Constructor Documentation

ccs_matrix_local ()

This is the default constructor which creates an empty ccs matrix with local_num_row = local_num_col = 0.

ccs_matrix_local (const ccs_matrix_local<T,I,O>& m)

This is the copy constructor which creates a new ccs matrix by deep-copying the contents of the input ccs matrix.

ccs_matrix_local (ccs_matrix_local<T,I,O>&& m)

This is the move constructor. Instead of copying the input matrix, it moves the contents of the input rvalue matrix to the newly constructed matrix. Thus it is faster and recommended to use when input matrix will no longer be used in a user program.

ccs_matrix_local (const crs_matrix_local<T,I,O>& m)

This is the implicit conversion constructor which creates a new ccs matrix by converting the input crs matrix of the same types.

Overloaded Operator Documentation

ccs_matrix_local<T,I,O>& operator= (const ccs_matrix_local<T,I,O>& m)

It deep-copies the input ccs matrix into the left-hand side matrix of the assignment operator "=".

ccs_matrix_local<T,I,O>& operator= (ccs_matrix_local<T,I,O>&& m)

Instead of copying, it moves the contents of the input rvalue crs matrix into the left-hand side matrix of the assignment operator "=". Thus it is faster and recommended to use when input matrix will no longer be used in a user program.

Public Member Function Documentation

void set_local_num (size_t nrow)

It sets the matrix information related to number of rows and number of columns as specified by the user. It assumes the user will provide the valid information related to the number of rows. Number of columns value is set as off.size()-1.

void debug_print ()

It prints the information related to the compressed column storage (val, idx, off, number of rows and number of columns) on the user terminal. It is mainly useful for debugging purpose.

ccs_matrix_local<T,I,O> transpose ()

It returns the transposed ccs_matrix_local of the source matrix object.

crs_matrix_local<T,I,O> to_crs ()

It creates an equivalent crs matrix of the target ccs matrix of the same "val", "idx" and "off" types. Target ccs matrix would remain unchanged.

Public Data Member Documentation

val

An instance of `std::vector<T>` type to contain the non-zero elements of the compressed columns of the sparse matrix.

idx

An instance of `std::vector<I>` type to contain the row indices of the non-zero elements of the compressed columns of the sparse matrix.

off

An instance of `std::vector<O>` type to contain the column offsets.

local_num_row

A size_t attribute to contain the number of rows in the 2D matrix view.

local_num_col

A size_t attribute to contain the number of columns in the 2D matrix view.

Public Global Function Documentation

`ccs_matrix_local<T,I,O>` crs2ccs(m)

Parameters
m: An object of the type `crs_matrix_local<T,I,O>`

Purpose
This function can be used to get a `ccs_matrix_local<T,I>` from a `crs_matrix_local<T,I,O>`. Input matrix would remain unchanged.

Return Value
On success, it returns the created matrix of the type `ccs_matrix_local<T,I>`. Otherwise, it throws an exception.

`crs_matrix_local<T,I,O>` ccs2crs(m)

Parameters
m: An object of the type `ccs_matrix_local<T,I,O>`

Purpose
This function can be used to get a `crs_matrix_local<T,I,O>` from a `ccs_matrix_local<T,I,O>`. Input matrix would remain unchanged.

Return Value
On success, it returns the created matrix of the type `crs_matrix_local<T,I,O>`. Otherwise, it throws an exception.

`std::vector<T>` operator*(m,v)

Parameters
m: An object of the type `ccs_matrix_local<T,I,O>`.
v: An object of the type `std::vector<T>`.

Purpose
This function performs matrix-vector multiplication between a sparse ccs matrix object with a std::vector of same value (T) type. It expects the size of the input vector should be greater than or equal to the number of columns in the input ccs matrix.

Return Value
On success, it returns the resultant vector of the type `std::vector<T>`. Otherwise, it throws an exception.

`rowmajor_matrix_local<T>` operator*(m1,m2)

Parameters
m1: An object of the type `ccs_matrix_local<T,I,O>`.
m2: An object of the type `rowmajor_matrix_local<T>`.

Purpose
It performs matrix-matrix multiplication in between a sparse ccs matrix and a dense rowmajor matrix of the same value (T) type.

Return Value
On success, it returns the resultant rowmajor matrix of the type `rowmajor_matrix_local<T>`. Otherwise, it throws an exception.

SEE ALSO

crs_matrix_local, rowmajor_matrix_local, ccs_matrix