

frovedis::ccs_matrix<T,I,O>

NAME

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

SYNOPSIS

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

Constructors

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

Public Member Functions

```
void debug_print ();
```

Public Data Members

```
frovedis::node_local<ccs_matrix_local<T,I,O>> data;  
size_t num_row;  
size_t num_col;
```

DESCRIPTION

A CCS (Compressed Column Storage) matrix is one of the popular sparse matrices. 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:

then its CCS representation would be:

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 $i+1$ th element and i th element in column offset indicates number of non-zero elements present in i th column.

```
template <class T, class I=size_t, class O=size_t>
struct ccs_matrix {
    frovedis::node_local<ccs_matrix_local<T,I,O>> data;    // local matrix information
    size_t num_row;    // number of rows in the global sparse matrix
    size_t num_col;    // number of columns in the global sparse matrix
};
```

master	worker0	worker1
-----	-----	-----
ccs_matrix<int,size_t,size_t>	-> ccs_matrix_local<int,	-> ccs_matrix_local<int,
	size_t,size_t>	size_t,size_t>
*data: node_local<	val: vector<int>	val: vector<int>
ccs_matrix	{1,1,2,2,4,3}	{1,1,2,2,4,3}
_local<int,	idx: vector<size_t>	idx: vector<size_t>
size_t,size_t>>	{0,1,0,1,0,1}	{0,1,0,1,0,1}
	off: vector<size_t>	off: vector<size_t>
	{0,1,1,1,2,4,4,4,6}	{0,1,1,1,2,4,4,4,6}
num_row: size_t (4)	local_num_row: size_t (2)	local_num_row: size_t (2)
num_col: size_t (8)	local_num_col: size_t (8)	local_num_col: size_t (8)

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

Constructor Documentation

ccs_matrix ()

This is the default constructor which creates an empty distributed ccs matrix without any memory allocation at worker nodes.

ccs_matrix (crs_matrix<T,I,0>& m)

This is the implicit conversion constructor to construct a distributed ccs matrix from the input distributed crs matrix of the same types.

Public Member Function Documentation

void debug_print ()

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

Public Data Member Documentation

data

An instance of `node_local<ccs_matrix_local<T,I,0>>` type to contain the reference information related to local matrices at worker nodes.

num_row

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

num_col

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

SEE ALSO

`ccs_matrix_local`, `crs_matrix`