frovedis::ell_matrix<T,I>

NAME

`frovedis::ell_matrix<T,I>` - A two-dimensional row-wise distributed ELL sparse matrix.

SYNOPSIS

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

Constructors

ell_matrix ();
ell_matrix (`crs_matrix<T,I,O>`& m);

Public Member Functions

void debug_print ();
`crs_matrix<T,I,O>` to_crs();

Public Data Members

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

DESCRIPTION

A ELL matrix is one of the most popular sparse matrices with elements stored in column-major order. In this matrix representation, all the non-zero elements of a row are shifted (packed) at left side and all the rows are padded with zeros on the right to give them equal length.

It has two 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 left-shifted (zero-padded) non-zero elements of
the sparse matrix stored in column-major order.
idx: a vector containing the corresponding column indices of the non-zero elements.
```

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 ELL image can be thought of as:

```
values        indices
1 2 4          0 4 7
1 2 3   =>     3 4 7
1 2 4          0 4 7
1 2 3          3 4 7
```

And its column-major memory representation would be:

```
val: {1, 1, 1, 1, 2, 2, 2, 2, 4, 3, 4, 3}
idx: {0, 3, 0, 3, 4, 4, 4, 4, 7, 7, 7, 7}
```

`ell_matrix<T,I>` is a two-dimensional template based distributed sparse data storage supported by frovedis. It contains public member "data" of the type `node_local<ell_matrix_local<T,I>>`. The actual distributed matrices are contained in all the worker nodes locally, thus named as `ell_matrix_local<T,I>` (see manual of ell_matrix_local) and "data" is the reference to these local matrices at worker nodes. It also contains dimension information related to the global matrix i.e., number of rows and number of columns in the original sparse matrix.

The structure of this class is as follows:

```
template <class T, class I=size_t>
struct ell_matrix {
  frovedis::node_local<ell_matrix_local<T,I>> 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
};
```

For example, if the above sparse matrix with 4 rows and 8 columns is distributed row-wise over two worker nodes, then the distribution can be shown as:

```
master                     worker0                          worker1
-----                      -----                            -----
ell_matrix<int,size_t>     -> ell_matrix_local<int,size_t>  -> ell_matrix_local<int,size_t>
   *data: node_local<          val: vector<int>                 val: vector<int>
         ell_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>>                ({0,3,4,4,7,7})                  ({0,3,4,4,7,7})
    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)
```

The `node_local<ell_matrix_local<int,size_t>>` object "data" is simply a (*)handle of the (->)local matrices at worker nodes.

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 ell matrix.

Constructor Documentation

ell_matrix ()

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

ell_matrix (crs_matrix<T,I,O>& m)

This is the implicit conversion constructor to construct a distributed ell matrix from the input distributed crs matrix of the same "val" and "idx" type.

Public Member Function Documentation

crs_matrix<T,I,O> to_crs()

This method can be used to convert the target distributed ell matrix into a distributed crs matrix of the same "val" and "idx" type.

void debug_print ()

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

Public Data Member Documentation

data

An instance of node_local<ell_matrix_local<T,O>> 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

crs_matrix, jds_matrix, ell_matrix_local