frovedis::crs_matrix_local<T,I,O>

NAME

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

SYNOPSIS

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

Constructors

crs_matrix_local ();
crs_matrix_local (size_t nrow, size_t ncol);
crs_matrix_local (const `crs_matrix_local<T,I,O>`& m);
crs_matrix_local (`crs_matrix_local<T,I,O>`&& m);

Overloaded Operators

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

Public Member Functions

void set_local_num (size_t ncol);
void savebinary (const std::string& dir);
void debug_print ();
void debug_pretty_print ();
`crs_matrix_local<T,I,O>` transpose () const;
`sparse_vector<T,I>` get_row(size_t r);

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 CRS (Compressed Row Storage) matrix is one of the most 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 matrix (in row-major order).
idx: a vector containing the column indices for each non-zero elements.
off: a vector containing the row-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 CRS representation would be:

```
val: {1, 2, 4, 1, 2, 3, 1, 2, 4, 1, 2, 3}
idx: {0, 4, 7, 3, 4, 7, 0, 4, 7, 3, 4, 7}
off: {0, 3, 6, 9, 12}
```

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

`crs_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 crs_matrix_local {
  std::vector<T> val;     // to contain non-zero elements of type "T"
  std::vector<I> idx;     // to contain column indices of type "I" (default: size_t)
  std::vector<O> off;     // to contain row-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
};
```

Constructor Documentation

crs_matrix_local ()

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

crs_matrix_local (size_t nrow, size_t ncol)

This is the parameterized constructor which creates an empty crs matrix of the given dimension without any memory allocation for the matrix elements.

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

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

crs_matrix_local (crs_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.

Overloaded Operator Documentation

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

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

crs_matrix_local<T,I,O>& operator= (crs_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

sparse_vector<T,I> get_row(size_t r)

It returns the requested row of the target sparse matrix in the form of sparse_vector<T,I> which contains a vector of type "T" for the non-zero elements in the requested row and a vector of type "I" for their corresponding column indices. If r > local_num_row, then it will throw an exception.

void set_local_num (size_t ncol)

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 columns. Number of rows value is set as off.size()-1.

void debug_print ()

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

void debug_pretty_print ()

Unlike debug_print(), it prints the compressed row storage as a view of a two dimensional dense storage on the user terminal. It is mainly useful for debugging purpose.

`crs_matrix_local<T,I,O>` transpose ()

It returns the transposed crs_matrix_local of the source matrix object.

void savebinary (const std::string& dir)

It writes the elements of a crs matrix to the specified directory as little-endian binary data.

The output directory will contain four files, named "nums", "val", "idx" and "off". "nums" is a text file containing the number of rows and number of columns information in first two lines of the file. And rest three files contain the binary data related to compressed row storage.

Public Data Member Documentation

val

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

idx

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

off

An instance of `std::vector<O>` type to contain the row 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

`crs_matrix_local<T,I,O>` make_crs_matrix_local_load(filename)

Parameters
filename: A string object containing the name of the text file having the data to be loaded.

Purpose
This function loads the text data from the specified file and creates a `crs_matrix_local<T,I,O>` object filling the data loaded.

The input file for the sparse data should be in the below format:

```
1:2 3:2
2:5
1:3 3:4 6:3
3:2 4:5
```

Where each sparse row is represented as "column_index:value" (column_index starts at 0). Note that there can be empty rows in the given file indicating no non-zero elements in that row. The desired type triplet of the matrix `<T,I,O>` needs to be explicitly specified when loading the matrix data from reading a file.

Default types for "I" and "O" is "size_t". But "T" type must be mandatorily specified. While loading the matrix data, it will consider number of columns as the maximum value of the column index read.

For example, considering "./data" is a text file having the sparse data to be loaded, then

```
auto m1 = make_crs_matrix_local_load<int>("./data");
auto m2 = make_crs_matrix_local_load<float>("./data");
```

"m1" will be a `crs_matrix_local<int,size_t,size_t>`, whereas
"m2" will be a `crs_matrix_local<float,size_t,size_t>`.

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

`crs_matrix_local<T,I,O>` make_crs_matrix_local_load(filename, num_col)

Parameters
filename: A string object containing the name of the text file having the data to be loaded.
num_col: A size_t attribute specifying the number of columns in the sparse matrix to be loaded.

Purpose
This function serves the same purpose as explained in above data loading function. But since it also accepts the number of columns information, it sets the loaded matrix column number with the given value (without computing the maximum column index as in previous case). Thus it expects, user will pass a valid column number for the loaded sparse matrix.

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

`crs_matrix_local<T,I,O>` make_crs_matrix_local_loadbinary(dirname)

Parameters
dirname: A string object containing the name of the directory having the data to be loaded. It expects four files to be presented inside the specified directory, as follows:

- "nums" (containing number of rows and number of columns separated with new-line),
- "val" (containing binary data for non-zero elements),
- "idx" (containing binary column indices) and
- "off" (containing binary offset values)

Purpose
This function loads the little-endian binary data from the specified directory and creates a `crs_matrix_local<T,I,O>` object filling the data loaded. The desired value type, "T" (e.g., int, float, double etc.) must be specified explicitly when loading the matrix data. If not specified, the other two types "I" and "O" would be size_t as default types.

For example, considering "./bin" is a directory having the binary data to be loaded,

```
auto m1 = make_crs_matrix_local_loadbinary<int>("./bin");
auto m2 = make_crs_matrix_local_loadbinary<float>("./bin");
```

"m1" will be a `crs_matrix_local<int,size_t,size_t>`, whereas
"m2" will be a `crs_matrix_local<float,size_t,size_t>`.

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

`crs_matrix_local<T,I,O>` make_crs_matrix_local_loadcoo(file,zero_origin)

Parameters
file: A string object containing the name of the file having the COO data to be loaded.
zero_origin: A boolean attribute to indicate whether to consider 0-based indices while loading the COO data from file.

Purpose
This function loads the text data from the specified file and creates a `crs_matrix_local<T,I,O>` object filling the data loaded.

The input file for the sparse data should be in the below COO format:

```
1 1 2.0
1 3 2.0
2 2 5.0
3 1 3.0
3 3 4.0
3 6 3.0
4 3 2.0
4 4 5.0
```

Where each row in the given file represents a triplet like `<row-index col-index value>`. The indices are 1-based by default. This file can be loaded as 0-based index, if "zero_origin" parameter is passed as "true" while loading the file. The desired triplet type of the matrix `<T,I,O>` needs to be explicitly specified when loading the matrix data from reading a file.

Default types for "I" and "O" is "size_t". But "T" type must be mandatorily specified. While loading the matrix data, it will consider number of columns as the maximum value of the column index read.

For example, considering "./data" is a text file having the COO data to be loaded, then

```
auto m1 = make_crs_matrix_local_loadcoo<int>("./data");
auto m2 = make_crs_matrix_local_loadcoo<float>("./data");
```

"m1" will be a `crs_matrix_local<int,size_t,size_t>`, whereas
"m2" will be a `crs_matrix_local<float,size_t,size_t>`.

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

std::ostream& operator<<(str, mat)

Parameters
str: A std::ostream& object representing the output stream buffer.
mat: An object of the type `crs_matrix_local<T,I,O>` containing the matrix to be handled.

Purpose
This function writes the contents of the sparse matrix in "index:value" format in the given output stream. Thus a crs matrix can simply be printed on the user terminal as "std::cout << mat", where "mat" is the input matrix.

Return Value
On success, it returns a reference to the output stream.

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

Parameters
m: A const& object of the type `crs_matrix_local<T,I,O>`.
v: A const& object of the type `std::vector<T>`.

Purpose
This function performs matrix-vector multiplication between a sparse crs 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 crs 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: A const& object of the type `crs_matrix_local<T,I,O>`.
m2: A const& object of the type `rowmajor_matrix_local<T>`.

Purpose
It performs matrix-matrix multiplication in between a sparse crs 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

rowmajor_matrix_local, crs_matrix