Data Systems

Project Phase 1

Team Systumm Phaad denge:

- Nachiket Patil (2020101018)
- Abhishek Sharma (2020101050)
- Ayan Agrawal (2020101034)

Normal (Dense) Matrices

Page layout

We split the `N x N` matrix into smaller `M x M` submatrices and each submatrix is written to a disk block.

`M = sqrt((BLOCK_SIZE * 1024) / sizeof(int))` which comes out to be `45`.
(`BLOCK_SIZE=8` has been assumed)

Also, we can deduce that `#blocks per row = ceil(N/M)` and `#total blocks = (blocks per row)^2 = ceil(N/M)*ceil(N/M)`.

This technique is used because this makes the transponse much more efficient. Storing in the row major order is very easy but transpose will take a lot of time, which is undesirable.

Operations

ASSUMPTIONS TAKEN :

1. It is assumed that before any of the operation other than `LOAD` is ran on a matrix, the matrix is loaded into the memory from a csv file.
2. The matrix loaded would be a valid square matrix of size `NxN` .

LOAD MATRIX

In this basically, we are reading the csv file only once but writing in each page multiple times.

Following steps are performed to carry out the operation:

1. We read through the first `M` elements of the first row and it is being written to the first page, then next `M` elements are similarly written onto the next page. This process is continued till we reach the end of first row.
2. Then, we move to the next row of the file and start appending back to the first page.
3. After the `M` rows are written in all the `ceil(N/M)` pages, then those pages are skipped and we start writing to this new page, which is the first page of the next set of `N/M` pages.
4. This is continued till we reach the end of the file. In this way, we have written the whole matrix in the form of smaller sub-matrices.

`SYNTAX : LOAD MATRIX <matrix_name>`

PRINT MATRIX

This command is used to print out the particular sub-matrix of a matrix on the terminal. If the matrix contains more columns that `PRINT_COUNT(=20)` , then first `PRINT_COUNT` columns are printed else all are printed. Same will happen with the rows.

Following steps are performed to carry out the operation:

1. We make use of the cursor for the matrix here. We keep it at the start point initially. First row of the first page is written at first.

2. Then, we go to the next page using cursor. Here also we write the first page of the next page and move to the next page similarly.
3. After the whole first row of the matrix is written, we move cursor to the first page again but on the second row.
4. After the `M` rows have been written then the cursor has already reached the end of the page, so we move the cursor to the next block which is `N/M` pages skipping the first page.
5. This is now done until we have reached the end of the last page. This will ensure that the whole matrix is written.

`SYNTAX : PRINT MATRIX <matrix_name>`

TRANSPOSE MATRIX

This command is used to perform transpose operation on the matrix we have stored. The transpose operation is in-place, i.e., no new memory is written.

Following steps are performed to carry out the operation:

1. Suppose we visualize the page setup as a matrix of size `ceil(N/M) x ceil(N/M)`. Then, first observation we can make is that all the pages on the diagonal of the matrix would be internally swapped, i.e, swapping of the corresponding symmetrical elements in the rows of a page.
2. Now for the non-diagonal pages, we just have to swap the pages with the corresponding symmetric page. In this way, when we read through the rows in the pages, we would encounter

`SYNTAX : TRANSPOSE MATRIX <matrix_name>`

EXPORT MATRIX

The command is used to export the matrix to the data folder as a file named <matrix_name.csv>. This command is similar to PRINT command.

Following steps are performed to carry out the operation:

1. The cursor is used to iterate the pages , we write the first row on the first page. And after writing complete row of the matrix, we move the cursor back to first page, but this time the cursor is pointing to second row.
2. After M rows are written, the cursor has reached the end of the page. We move the cursor to the block after skipping `ceil(N/M)` pages.
3. In this way, we can write the complete matrix and save the file in data folder.

`SYNTAX : EXPORT MATRIX <matrix_name>`

COMPUTE MATRIX

This command is similiar to TRANSPOSE command, where we were swapping symmetric index w.r.t to the matrix. We have to compute the matrix

$$A_{new} = A - A^T \tag{1}$$

Following steps are performed to carry out the operation:

1. On every page we have submatrix of size M , so we iterate for every blocks from `i = 0` to `ceil(N/M)` , and for every i, we iterate the blocks with `j = i to ceil(N/M)` . For every block we find its symmetric block page , and do the following operation:

$$\begin{aligned} A(i,j) &= A(i,j) - A(j,i) \\ A(j,i) &= -A(i,j) \end{aligned} \tag{2}$$

2. This new matrix is stored as `<matrix_name>_RESULT` in pages.

`SYNTAX : COMPUTE <matrix_name>`

RENAME MATRIX

This command is used to assign a new name to an existing matrix in the memory.

Following steps are performed to carry out the operation:

1. In this operation, the old entry which is stored as key-value pair of `{matrix_name, matrix_pointer}` in matrixCatalogue is removed and a new entry is made in it with `{new_matrix_name, matrix_pointer}` entry.

2. Also, all the page names are renamed now from `old_matrix_name_page_i` to `new_matrix_name_page_i`.
3. This ensures that now whenever the matrix is called with its new name, the pointer to that matrix is fetched.

`SYNTAX : RENAME MATRIX <matrix_name> <new_matrix_name>`

CHECKSYMMETRY

This command is used to check if the matrix is a symmetric matrix or not, i.e., the lower triangular matrix is a mirror image of the upper triangular matrix keeping the diagonal as the mirror.

Following steps are performed to carry out the operation:

1. The mechanism for this operation is similar to that of the `TRANSPOSE` operation, just that instead of swapping the elements or pages, we are checking if they are equal or not.
2. We simply iterate through the pages and check the equality of the above mentioned elements and pages, if any one of them is not equal then `FALSE` value is returned, else `TRUE`.

`SYNTAX : CHECKSYMMETRY <matrix_name>`