Lab Report

Submitted by: - Parth Parashar

Summary

This was a helpful lab assignment for me as I got to learn a lot about the use of domains.

For the first question, It was fairly easy to judge how the programs were working and what the functionalities of the domains were. I got to learn about the behaviour of domains by making changes to the programs and the mapping of domains.

The outputs and the changed code for those outputs is given in this lab report below.

For the second question and third questions, since they were based on domains as well (including domain maps), I got a deeper understanding of the topic.

For the fourth question which was on matrix multiplication, for converting it to PGAS version was a little challenging. The use of domain maps in the third question gave me an idea on how to convert it to PGAS version. I used domain maps to map domains to the different variables and got the expected output.

After introducing the verification code to the matrix multiplication problem, I was able to correctly verify the output. The domain maps are working properly.

As for the Gauss-Seidel problem, I was stuck on the problem for quite some time. I was not able to figure out how to converge the values which were being updated.

The first thing I did was, I used a temporary variable to store the values in the for loop and just like Jacobi, I used the delta and epsilon values to calculate the requisite values (as well as loop ending condition).

The expected value still was not correct, but it is not being stuck in the while loop which was the case when I had not included the delta and epsilon conditions and was using a while(true) statement.

The second thing which I tried was removed the temporary variable and then ran it again without the reduction condition and used only the same variable changes. The answer still did not converge. The code and outputs are attached in the zip file as well as in this lab document as well.

For the File-I/O problem, I made the changes to the declaration of N as well as integer values and got the correct output after a couple of tries. It took me quite some time to get to the correct output and it was a good introduction to the File IO operations.

For the optional part, I tried to create a worker routine to run the program but was unable to successfully integrate the functionality as required and the output is not as expected.

The code for the same is attached in the zip file.

Question-1: -

Text

Description automatically generated

When we change the domain to D = {1..2}

Graphical user interface, text, application

Description automatically generated

Then the output changes to

Text

Description automatically generated

Question-2: -

Text

Description automatically generated

When changing the domain, we can see that

Text

Description automatically generated

The output changes to: -

Text

Description automatically generated

Question-3: -

Text

Description automatically generated

Now, When I change the mapping,

Text

Description automatically generated

then we can see the following output.

Text

Description automatically generated

When we change the domain to 1..16, then we can see the output changed to: -

Text

Description automatically generated

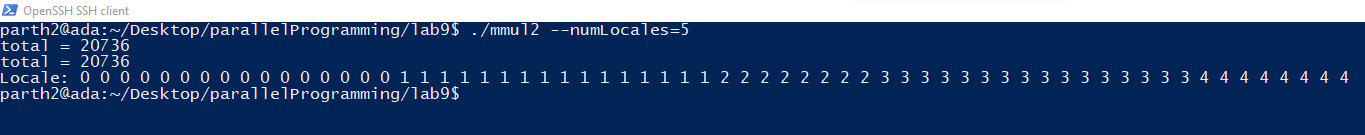
Question-4: -

The code for mmul2.chpl is: -

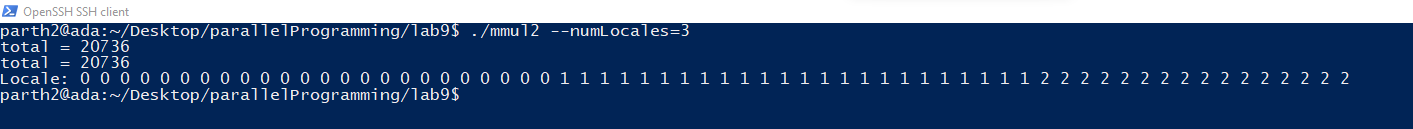
Text

Description automatically generated

The output of this code is given below: -



Changing the number of locales, we can see that the output being generated is correct



Question-5: - When the jacobi.chpl file is run, then

Graphical user interface, text

Description automatically generated

Text

Description automatically generated

Now, for gauss-seidel, we need one array and we will use one buffer and then for tracking convergence, we can use a temporary variable to store the values of a.

The code for the same is given below: -

Graphical user interface, text

Description automatically generated

The output of this program is given below: -

A picture containing text

Description automatically generated

The output was still not converging properly as when I ran it with different numLocales, it showed different outputs.

Then the next iteration of the code which I tried was: -

Graphical user interface, text, application

Description automatically generated

The output for this also did not converge properly as given below: -

Text

Description automatically generated

Question-6: - When FileIO.chpl is run, then we see the following output

Text

Description automatically generated

The program that produced this output is: -

Graphical user interface, text, application

Description automatically generated

The changes needed to be made for the required for N=32 are:-

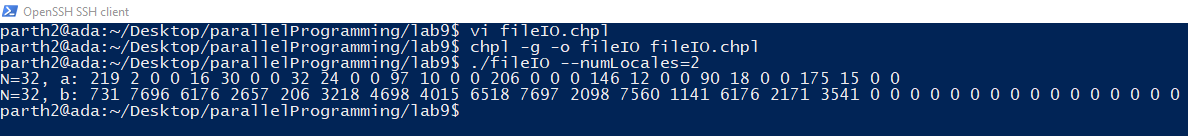
Change the int type as well as the calculation of N by removing the divisor by half.

The code becomes as: -

Graphical user interface, text

Description automatically generated

The output becomes as: -

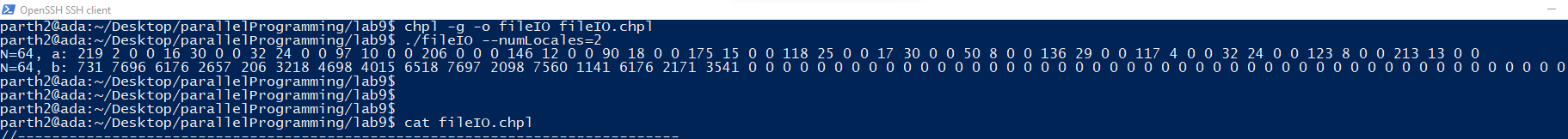


Now for achieving the required 64 integers, we will modify the calculation of N as follows: -

Graphical user interface, text

Description automatically generated

The output for this code is given below: -

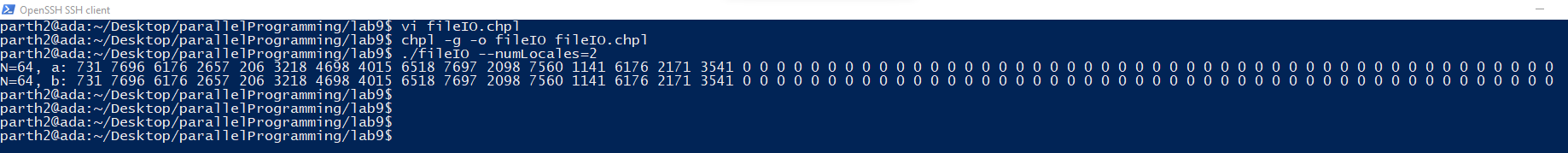


If we want the type of integers to be normal 32 bit integers, then the code can be modified as below: -

Graphical user interface, text, application

Description automatically generated

The output for this code is given below: -



Question-7: - fileIO2.chpl

The code for the same is given below: -

Graphical user interface, text

Description automatically generated