

TECHNICAL UNIVERSITY OF DENMARK

HIGH-PERFORMANCE COMPUTING

COURSE 02614

Assignment 1

Authors:

Oskar HINT, s161559

Mikkel JENSEN, s123184

Philip RASMUSSEN, s103124

January 5, 2017



Contents

1	Introduction	1
2	Assignment	1
2.1	Nat	1
3	Conclusion	1

1 Introduction

2 Assignment

2.1 Nat

The Sun Studio compiler is used, and therefore the driver required is `matmult_c.studio`.

First, the goal is to write a native function, which performs a matrix-matrix multiplication. The shape has to be suitable for the operation, but it is arbitrary within the limits. We chose the native way, where the matrix is described using double pointers, i.e. `A[i][j]`. The choice of function prototypes and driver therefore follows.

The general function prototype used is displayed below:

Listing 1: Function Prototype

```
void matmult_NNN(int m, int n, int k, double ** A, double ** B, double ** C)
```

The function prototypes are all declared in the header file `ass1_lib.h`. The functions themselves are described in the `ass1_lib.cpp` file, where the `ass1_lib.h` library is included.

The native function takes the integer arguments `m`, `n` and `k`, as well as the double arguments `**A`, `**B` and `**C`. The double arguments are the actual matrices, which in the case of matrix `A` and `B` are generated within the supplied driver based on the submitted shape parameters. In the case of `C`, only the memory is allocated. To perform the matrix-matrix multiplication a for loop for each of the 3 shape parameters is required. Three loop variables `i`, `j` and `l` are introduced for the matrix leading dimensions `m`, `n` and `k`. The `C` matrix has the leading dimensions of `m` and `n`, `C[i][j]`. The function is displayed below:

Listing 2: Function Prototype

```
void matmult_nat(int m, int n, int k, double ** A, double ** B, double ** C){
    for(int i = 0; i < m; i++){
        for(int j = 0; j < n; j++){
            C[i][j] = 0;
            for(int l = 0; l < k; l++){
                C[i][j] += A[i][l]*B[l][j];
            }
        }
    }
}
```

Notice the `+=` sign, which is caused by the fact, that when the function loops over `l`, the multiplications for the different values of `l` are added to the

3 Conclusion