

16) Find out (typical) capacities, transfer rates and latencies for Cache memory, Main memory, harddisks.

Indicate your source (and if possible what particular hardware it applies to).

17) In this problem we want to investigate questions of cache size and cache miss: Consider a program which creates two random  $n \times n$  matrices (for example using the function `rand`) and then multiplies them. (You will find a framework on the course web page.)

The multiplication uses three iterated loops of the form:

```

for(i = 0; i < n; j++) {
    for(j = 0; j < n; i++) {
        for(k = 0; k < n; i++) {
c[i][j]=c[i][j]+a[i][k]*b[j][k];
        }
    }
}

```

but there are 6 different possibilities (ijk,ikj,kij,kji,jik,jki) to arrange the loops.

a) How do these 6 possibilities perform. (Run the program.) Explain!

b) Take the fastest version of the program and run it for values of  $n$  between 10 and 1000. (To get reasonable runtimes, you might have to add a “repeat” loop for small values of  $n$  or to run the outermost loop (provided it is  $i$  or  $j$ ) only for a fraction of  $n$ , and then rescale the timings.)

Plot the logarithm of the runtime over the matrix size. (You will most likely want to use values of  $n$  that are not equally distributed, but denser at “interesting” locations.)

Describe the resulting curve, using the fact that the cache size on `stokes` is 16kB.

c) A method for improving the performance is what is called “blocking”. We divide the matrices in blocks  $A_{i,j}$  (so if  $A$  is a  $1000 \times 1000$  matrix, and we have  $10 \times 10$  blocks, there are 10000 blocks).

We now calculate the matrix product using the formula

$$C_{i,j} = \sum_k A_{i,k} \cdot B_{k,j}$$

where the multiplication is matrix multiplication (of the blocks).

Why does this improve the performance?

d) Given a cache of size 16kB, what would be a reasonable block size?

e) Implement a blocked matrix multiplication and compare the performance for  $1000 \times 1000$  matrices to the routine in b).