PPROG, Winter 2015/16

Exercise 1

Task 1.1 (Memory hierarchy)**.** Assume that you have a L1 cache, L2 cache and main memory. The hit rates and access times for each are:

- **L1**: 50% hit rate, 2 cycles for access
- **L2**: 70% hit rate, 15 cycles for access
- **Mem**: 100% hit rate, 200 cycles for access

a) What fraction of accesses are serviced by the L2 cache? From main memory?

b) What is the miss rate and miss time for L2 cache?

c) What is the miss rate and miss time for L1 cache? (**Hint:** Depends on previous answer.)

d) What is the improvement in L1 miss time, if the access time of the main memory is reduced by 10%?

e) Now we remove the L2 cache to add more L1 cache. As a result, the new L1 hit rate is 75% ($Mem_{accesstime} = 200$). What is the change in L1 miss time? Does this architectural change make sense?

f) What is the effective access time?

Task 1.2 (Memory centric runtime analysis)**.** In this exercise we shall investigate the runtime behaviour of the computation $\mathrm{B} = \mathrm{B} + \mathrm{A}^T$ for $\mathrm{A}, \mathrm{B} \in \mathbb{R}^{n \times n}$. Assume that these matrices are so large that they do not fit in the cache. Start by considering the following algorithm for computing the sum $\mathrm{B} = \mathrm{B} + \mathrm{A}^T$:

```
for (i=0; i<n; i++)
    for(j=0; j<n; j++)
        B(i,j) = B(i,j) + A(j,i);
```

a) Name at least two main differences between CPU Cache and main memory. Explain the term cache line.

b) State where in the above algorithm the memory is accessed and whether accesses are consecutive, given that the matrices A and B are stored row-wise.

c) Conduct a runtime analysis of the above algorithm, expressing the runtime in terms of the following quantities:

n**,** dimension of the matrix,

T_A**,** time for the executing an arithmetic operation,

T_M**,** time for an access to the main memory,

T_C**,** time for an access to cache, and

l**,** length of a cache line.

Hint: Assume that variable values are written back to the cache.

d) What is the runtime when the matrix A is stored column wise instead of row wise?

e) Assume the following relations hold:

$$\begin{aligned} T_M &= 80\,T_A \\ T_C &= 8\,T_A \\ l &= 8. \end{aligned}$$

How many times faster is the above algorithm when you store the matrix A column wise instead of row wise?

Task 1.3 (Amdahl's Law)**.** Consider an arbitrary parallel code.

a) Which speedup can be achieved with 16 processors if $\beta = 1 - \alpha = 90\%$ of the code can be perfectly parallelized?

b) How large must the parallelizable share $\beta = 1 - \alpha$ be at least in order to achieve a speedup of 10 with 16 Processors?

Task 1.4 (Speedup and Efficiency)**.** The maximum norm of a vector $\mathbf{x} \in \mathbb{R}^n$ is defined by:

$$\|\mathbf{x}\|_\infty := \max_{i=1,\ldots,n} |x_i|.$$

1. Give the definitions of speedup and efficiency.
2. How many steps in terms of n are required determine the maximum norm of a vector of length n? *Hint*: Count only the arithmetic operations and count the operation for the maximum of two items the same as that for an absolute value.
3. How many steps in terms of n are required if an efficient, parallel algorithm is used with $p = n/2$ processors? *Hint*: Assume that p is a power of 2.
4. Give the speedup and efficiency in terms of p.
5. What are the speedup and efficiency when $p = n/\sqrt{n} = \sqrt{n}$ processors are used?

Discussion of this exercise on 27^{th} October 2015.