Technische Universität München
Institut für Informatik
Dr. Ralf-Peter Mundani
Michael Lieb, M. Sc.

SS 2013

Parallel Programming and HPC

Exercise Sheet 1: Performance Measurement

1 Memory Access

a) In this task we analyze the number of time steps necessary to execute the i^{th} operation of the following for-loop:

```
for i = 0 to N do
    a[i] = ( 1 + s ) · b[i] + t · c[i] · d[i] + e[i]
od
```

You can assume that all scalar quantities are kept in registers during the entire program execution. We assume to have a processor with certain limitations. It can read/write data and execute arithmetic operations simultanously. However, it is **not** possible to simultanously load and store data within one execution step. The incrementation of the loop counter can be ignored.

How many (machine) cycles are necessary in order to execute the i^{th} full iteration, if

i. a scalar processor is used which in each cycle is able to **load two words** *or* to **store one word** as well as to execute **one multiplication** *or* **one addition**,

ii. a superscalar processor (MULT & ADD) is used which is able to **load** *or* to **store one word per cycle**,

iii. a superscalar processor (MULT & ADD) is used which is able to **load** *or* to **store two words per cycle**?

b) Interleave two evaluations of the function using the feature set of processor a.iii). How many cycles are necessary?

2 Speedup and Parallel Efficiency

The speedup S for solving a problem on p processing elements compared to one processing element can be calculated with Amdahl's law

$$S = \frac{p}{\sigma \cdot p + (1 - \sigma)}$$

where σ denotes the percentage of serial work to be done. The efficiency E of this parallel approach can be calculated with $E = \frac{S}{p}$.

Consider some parallel program with a percentage of serial work of 4%. Calculate the maximum amount p of processing elements, thus, the parallel efficiency E is at least 90%.