

Exercise Sheet 2

Discussion starts from Wednesday, April 24, 2019

2.1 Theory

A sequential program \mathcal{P} can be divided into 5 parts A to E, which must be executed in this order due to their dependencies. Table 1 lists the amount of run-time each part contributes to the run-time of the program. Parts A, C and E cannot be parallelized. Part B can be transformed in max. 4 sub-parts that can be executed in parallel. For the parallel execution of part D, no restrictions exist.

- How many cores are required to achieve a speed-up ≥ 4 ? Which law do you need to follow to answer this question?
- Assume the problem solved in part D to have the double size as in a). However, the overall run-time of the program should not change. Apply Gustavson's law to calculate the resulting speed-up with respect to the overall run-time of the parallelized program from a).

Part	A	B	C	D	E
percentage of run-time	2%	20%	5%	70%	3%

Tabelle 1: Percentage of run-time each part contributes to the overall run-time of the program.

2.2 OpenMP – First Steps

Consider the following C program:

```
int main()
{
#pragma omp parallel
{
printf("Hallo Welt\n");
}
return 0;
}
```

- Compile and execute the program (use “-fopenmp”). How many times “Hallo Welt” is printed out?
- Modify the code so that the program prints out the ID of the current thread as well as the total number of threads. The output should be as follows:

```
Hallo Welt: I am thread number x of y threads
```

Here, x is the ID of the thread and y the total number of threads. Is it possible to control the number of tasks in the program?

2.3 Loop Parallelism

In the repository, the sequential program 0202.c is located.

- a. Add the code required to parallelize the for-loop with OpenMP. Measure the run-time of the sequential and the parallel program.
- b. Parallelize the loop using only "#pragma omp parallel" (not "#pragma omp parallel for" or "#pragma omp for").

General Information: Further information can be found under <https://ls12-www.cs.tu-dortmund.de/daes/lehre/lehrveranstaltungen/sommersemester-2019/rechnerarchitektur-ss19.html>. Submitting solutions to the exercise sheets is not required.