REGIONALES HOCHSCHUL-
RECHENZENTRUM KAISERSLAUTERN

TECHNISCHE UNIVERSITÄT
KAISERSLAUTERN

RHRK Information

High Performance Computing with the Cluster „Elwetritsch“

Focus: Basics - Parallel Jobs

Instructor: Dr. Josef Schüle

Amdahl's law

In any computing process will be a sequential part that cannot be parallelized.

Let's look at an example:

- perform a large calculation
- write the output into a large file for further visualization/post processing

Obviously this example E can be divided into two subprocesses:

- E_1 : large calculation
- E_2 : write the output

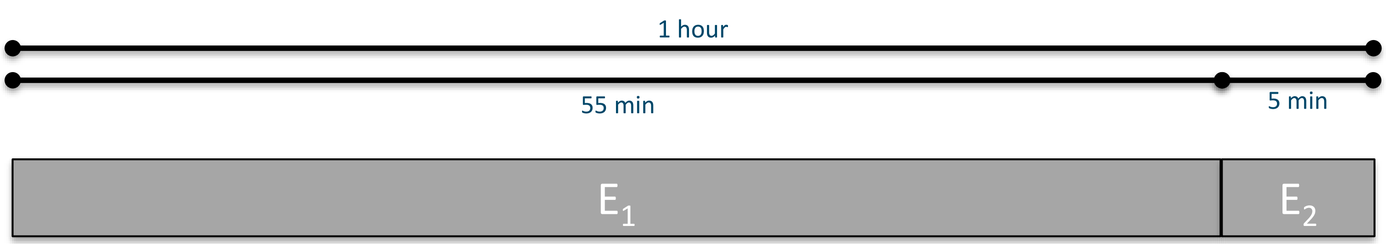
The time to run E may be represented by $T_1(E)$, there the index 1 indicates sequential execution, that is, running on 1 processor core.

Amdahl's law

Obviously, we observe on a single processor core:

- E_2 cannot start before E_1 has finished
- $T_1(E) = T_1(E_1) + T_1(E_2)$
- E_2 is sequential in nature and cannot run faster in parallel

Let's assume that *E* overall takes 1 hour to run sequentially. Out of these E_1 takes 55 minutes and E_2 5 minutes to run.



Amdahl's law

If we use more than 1 processor core, this will only affect E_1, thus using p cores changes the execution time to:

$$T_p(E) = T_p(E_1) + T_1(E_2)$$

In an ideal world, $T_p(E_1)$ is not only smaller then $T_p(E_1)$, but
$T_2(E_1) = ½ T_1(E_1)$
$T_4(E_1) = ¼ T_1(E_1)$
or in general $T_p(E_1) =1/p\ T_1(E_1)$.

We may define a speedup of *E:*

$$S_p(E)=\frac{T_1(E)}{T_p(E)}$$

and a ratio of the parallelizable portion $T_1(E_1)$ to the total runtime $T_1(E)$:

$$f = \frac{T_1(E_1)}{T_1(E)}$$

Amdahl's law

For our example, f = 55/60 = 0.92.

If we for example plan to run *E* on 5 processor cores, we expect:

- the time for E_1 will drop from 55 to 11 minutes
- the overall time will drop from 60 to 11+5=16 minutes
- a speedup $S_5(E)$ of 60/16= **3.75**

We recognize, that our speedup is smaller than 5, thus not optimal.

Amdahl's law

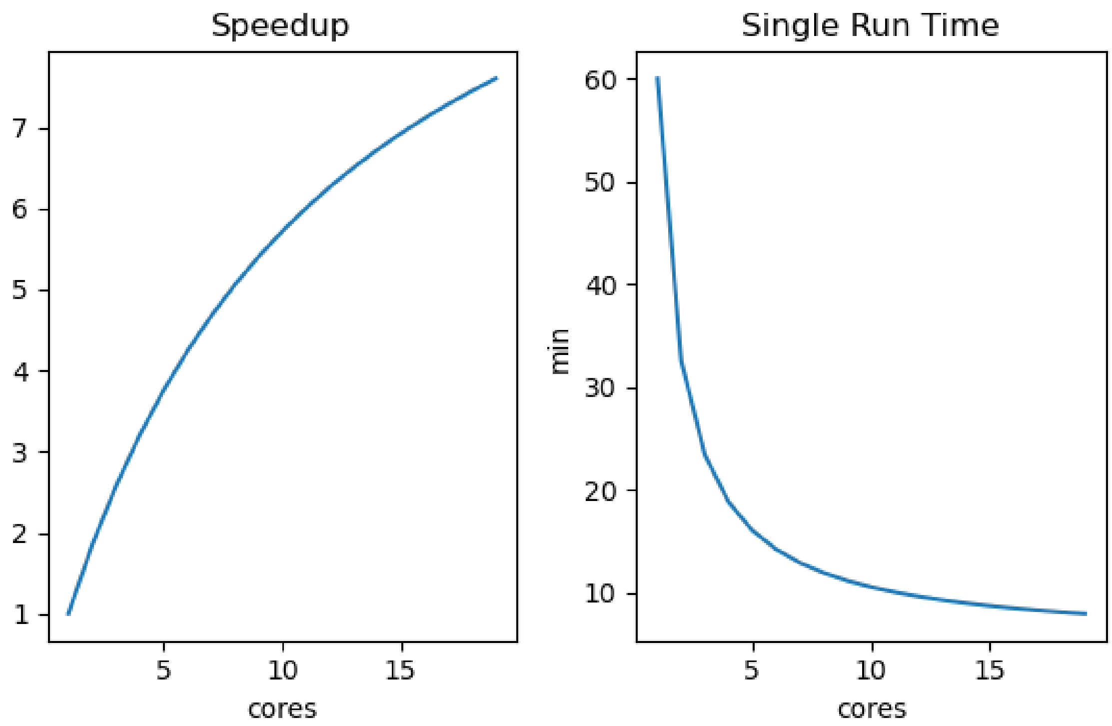
Let's assume, that we not just want to solve *E* once, but we are working with a lot of parameters and have to solve it 100 times.

Now we consider a computer with 20 processor cores.

What are we interested in? There are 2 different aspects when optimizing the way to a solution:

- is it of great importance to have each solution of E fast (time critical prediction)
- request to finish all 100 runs

Amdahl's law



According to our example E, we may predict a speedup of almost 8 and a single run time somewhat below 10 minutes if we use all 20 cores.

Sounds great.

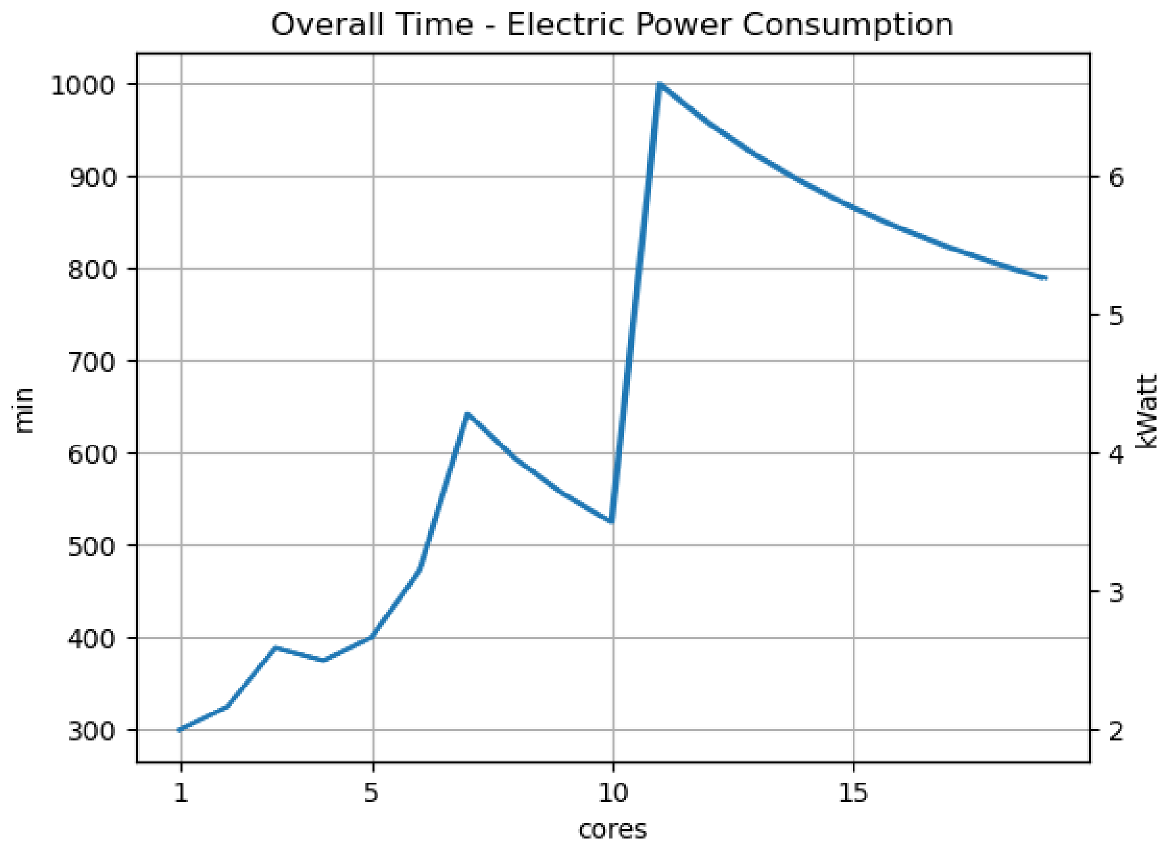
As f=0.92, the highest speedup with an infinite number of cores will be 12.5.

Amdahl's law

With focus on finishing each single calculations, we can gain a lot with parallelization.

Now let's have a look on the time to finish our project, that is, to finish all runs and let's assume (all are speaking about climate - why not we) our server with its 20 cores needs just 400 Watt electric power.

Amdahl's law



We can run 20 different examples on our 20 cores (each 60 min) and thus finish in 300 min or 5 hours in total at a cost of 2 kW.

Or use more cores, wait longer and use more power.

Clear choice to me.

Amdahl's law

What can we learn from this small example:

There is an optimal value of cores to be used

- This optimal value depends on your program
- This optimal value depends on your preferences
- This optimal value depends on whether you run a program once or many times.

- **High Performance Computing on Elwetritsch**
- **Parallel Jobs - Basics**

Vielen Dank
Thank You