DAT470 Assignment 1

Albin Jansfelt, Amanda Allgurén, Lukas Martinsson

March 2022

1 Problem 1

1.1 How many cores does the machine have?

command used:

```
cat /proc/cpuinfo — grep cores
nproc –all
```

There are 12 cores on the machine, and 48 processors.

1.2 What type of CPUs and what is their clock frequency?

command used: cat /proc/cpuinfo

```
model         : 85
model name    : Intel(R) Xeon(R) Gold 6126 CPU @ 2.60GHz
stepping      : 4
microcode     : 0x2006b06
cpu MHz       : 1000.006
```

The CPUs are of the model *Intel(R) Xeon(R) Gold 6126 CPU @ 2.60GHz* and have a clock frequency of about 1000 mHz.

1.3 How much disk is their in total and in use in the file systems /data and /datainbackup?

command used: df

```
Filesystem                         1K-blocks     Used         Available     Use    Mounted on
/dev/mapper/vg_data-lv_datainbackup  25767708672   4244436848   21523271824   17%    /datainbackup
/dev/mapper/vg_data-lv_data          77551114240   4583432784   72967681456   6%     /data
```

Disk space used and available for the two folders can be read from the table above.

1.4 How much memory does your login shell use?

command used: ps -U $< user >$ –no-headers -o rss — awk ' sum+=$1 END print int(sum/1024) "MB"}'

Plugging in each of our CID in $< user >$, we get 0 MB, 0 MB, and 237 MB.

2 Problem 2

2.1 Q1 - Proportion of serial computations

The function *compute_pi* is run sequentially, but when arriving at $s = p.map(sample\_pi, [n] * args.workers)$, the function *sample_pi* is mapped for all elements and are run in parallel.

To measure the proportion of serial time the Python module *Time* was used. We defined two variables *total_time* and *parallel_time* which measures the total time and the time it takes for the parallel part respectively. The proportion of serial time was then calculated as $serial\_prop = \frac{total\_time - parallel\_time}{total\_time}$.

Thus, the proportion of serial time was 0.906. Connected to Amdahl's law we get the portportion of parallel time $f = 1 - 0.906 = 0.094$.

2.2 Q2 - Speedup graph

See Figure 1 for the theoretical and estimated speedup when increasing the number of cores.

2.3 Q3 - Random Seeding Suggestion

The initial seed number is specified as an argument in the terminal. The default is $seed = 1$. To have each worker use a different seed, we create a list specifying the seed for every worker. The first worker will have $seed = seed + 0$, the second will have $seed = seed + 1$ and so on. This will then be sent to the *sample_pi* function which will generate pseudo-random x and y values.

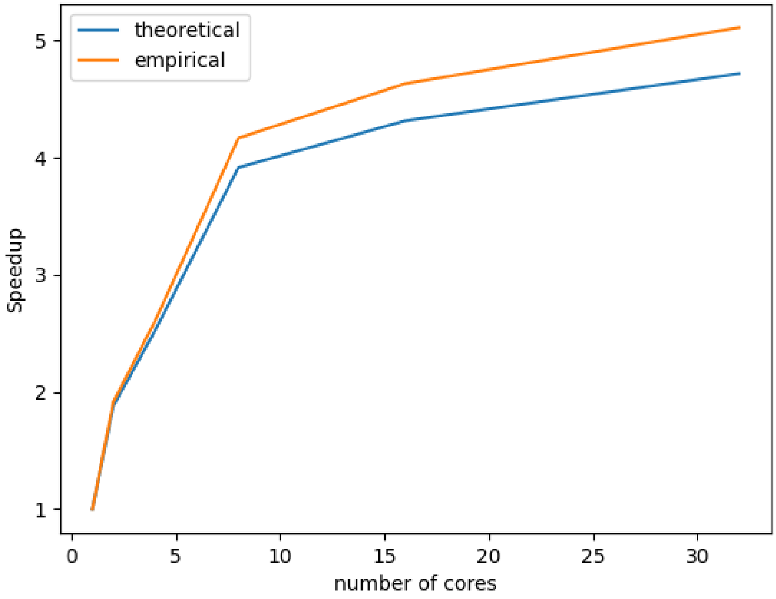

Figure 1: Theoretical and estimated speedup when increasing the number of cores