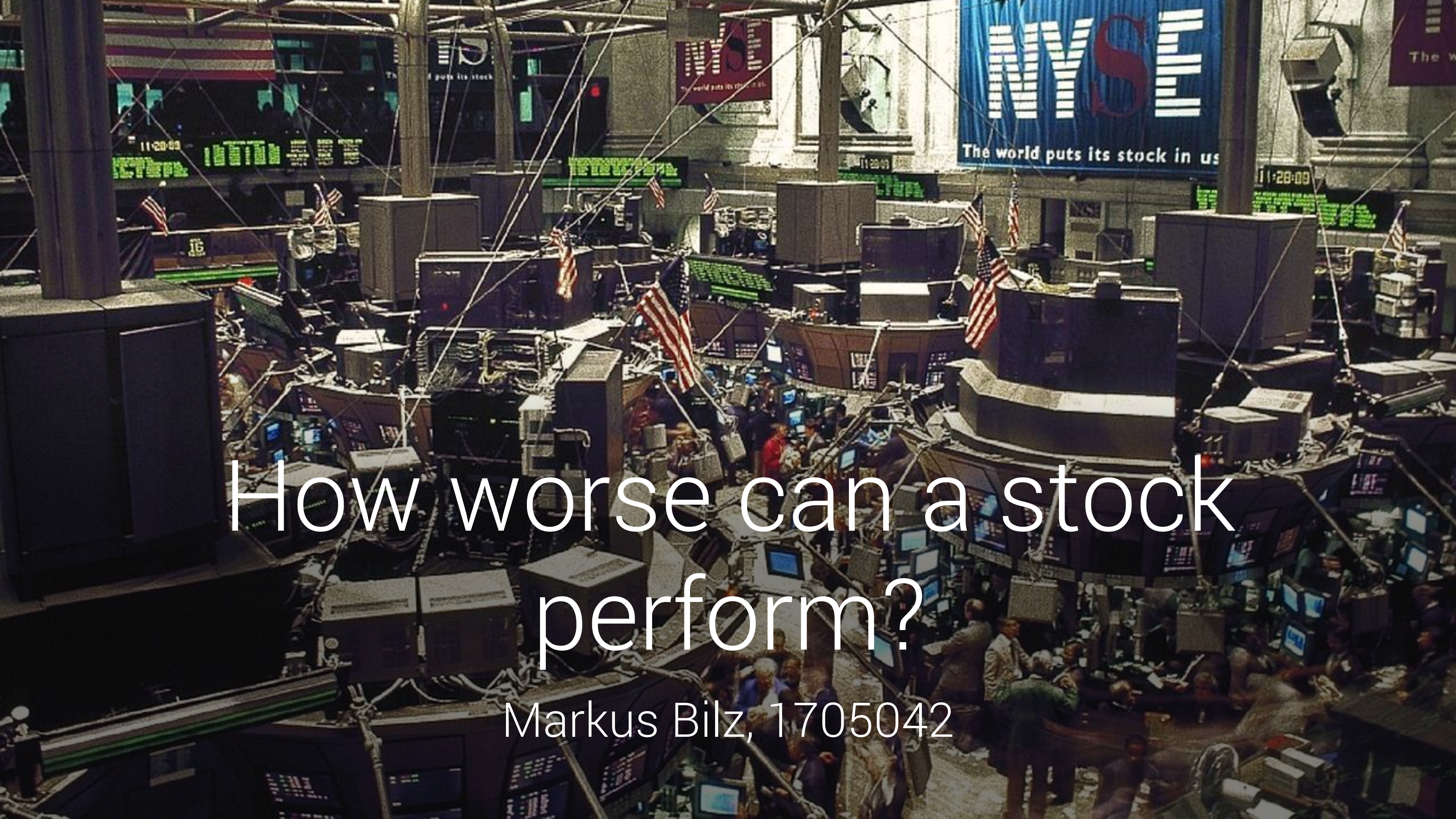
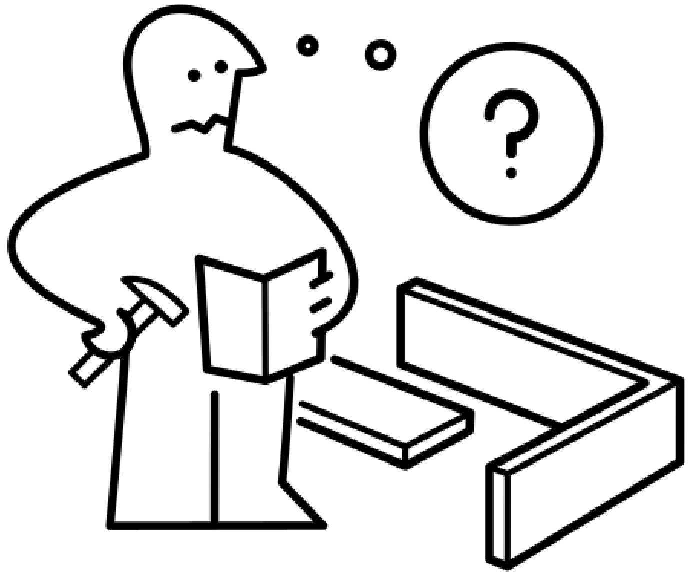
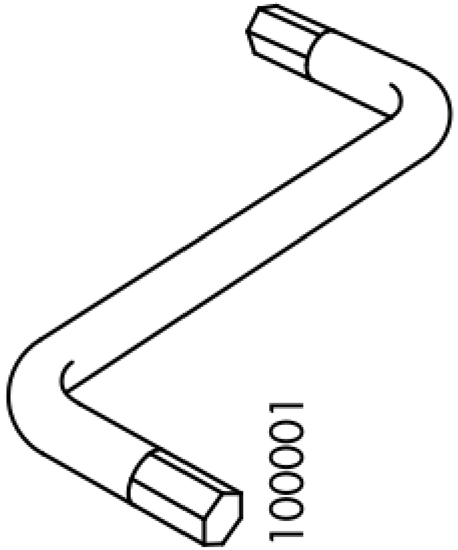


How worse can a stock perform?

Markus Bilz, 1705042



Greatest, likely loss?



Monte Carlo Value at Risk Simulation

Estimating value at risk

```
repeat n times -> first kernel
    repeat t times
        generate normal distributed number
        update interim price
    save end price to path array
extract the nth rank -> second kernel
scale value at risk to holding period
print results
```

1st kernel

(generating random prices)

specification

trivial problem / no interaction between threads

All threads run the same code / no thread divergence...

{demo}

2nd kernel

(extracting the minimal price)

{demo}

specification

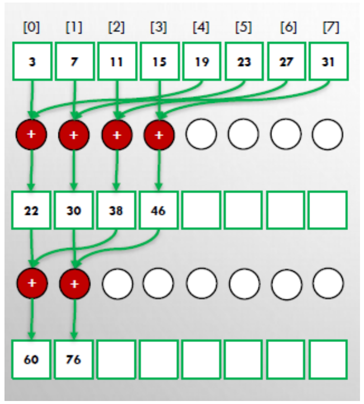
non trivial problem / dependence between threads

potential bank conflicts / idle threads

solved through efficient reduction

...

<insert picture of threads>


[0]
[1]
[2]
[3]
[4]
[5]
[6]
[7]
3
7
11
15
19
23
27
31
+
22
30
38
46
60
76

alternative 2nd kernel (extracting the nth smallest price)

performance evaluation

gpu specification

gpu specification

NVIDIA GeForce 940MX

3 Mb dedicated memory

384 cores

misc

using wakeup call to prevent JIT and lazy initialization

using std::chrono::steady_clock

using tile_size with multiple of two

<insert breakdown
for all kernels>

<insert comparsion
with cpu>

<insert tile_size

vs.

computation time>

<problem size

vs.

computation time>

https://github.com/KarelZe/MC-VAR-Sim